

Detnet Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 6, 2018

S. Bryant  
M. Chen  
Huawei Technologies  
March 05, 2018

Operation of Deterministic Networks over MPLS  
draft-bryant-detnet-mpls-dp-00

Abstract

This document specifies Deterministic Networking data plane operation over an MPLS Packet Switched Networks.

This document is a derivative work from draft-ietf-detnet-dp-sol-01.

Whether this is published as a stand-alone text, or serves as a focal point to refine the MPLS design and the key point are subsequently re-merged with draft-ietf-detnet-dp-sol-01 is a matter for the DETNET WG, as is the editorship of any WG text.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
2.1. Terms used in this document . . . . .	3
2.2. Abbreviations . . . . .	4
3. Requirements language . . . . .	5
4. DetNet Over an MPLS Underlay . . . . .	5
5. DetNet over MPLS Encapsulation Components . . . . .	8
5.1. Basic Data Plane Encapsulation . . . . .	8
5.2. DetNet Control Word . . . . .	9
5.3. Flow identification . . . . .	10
5.4. OAM Indication . . . . .	11
5.5. Flow Aggregation . . . . .	11
5.6. Aggregation at the LSP . . . . .	12
5.7. Aggregating DetNet flows as a new DetNet flow . . . . .	12
6. Simple Aggregation at the DetNet layer . . . . .	13
7. Indication of the DetNet Payload Type. . . . .	13
8. Operation of the PREF Functions . . . . .	14
8.1. Operation of PR . . . . .	14
8.2. Operation of EF . . . . .	15
8.3. Packet reordering considerations . . . . .	15
8.4. Indication of PREF processing . . . . .	16
8.5. Placement of PR and EF in a node . . . . .	16
9. Operation at DetNet Node types . . . . .	17
9.1. Operation at an Ingress Edge (PE) Node . . . . .	17
9.2. Operation at a Relay node (S-PE) . . . . .	17
9.3. Operation at an Egress Edge (PE) Node . . . . .	18
9.4. Operation at a Transit(P) Node . . . . .	18
10. Other DetNet data plane considerations . . . . .	19
10.1. Class of Service . . . . .	19
10.2. Quality of Service . . . . .	19
10.3. Bidirectional traffic . . . . .	20
10.4. Layer 2 addressing and QoS Considerations . . . . .	21
11. Management and control considerations . . . . .	22
11.1. S-Label assignment and distribution . . . . .	22
11.2. Explicit routes . . . . .	22
12. Security considerations . . . . .	22
13. IANA considerations . . . . .	23
13.1. Acknowledgments . . . . .	23
14. References . . . . .	23
14.1. Normative References . . . . .	23
14.2. Informative References . . . . .	23

Authors' Addresses . . . . .	26
------------------------------	----

## 1. Introduction

This document is a derivative work from [draft-ietf-detnet-dp-sol-01].

Editor's Note: We need to point to the exact version that this was derived from, not a generic version of [I-D.ietf-detnet-dp-sol].

Whether this is published as a stand-alone text, or serves as a focal point to refine the MPLS design and the key point are subsequently re-merged with draft-ietf-detnet-dp-sol-01 is a matter for the DETNET WG.

Deterministic Networking (DetNet) is a service that can be offered by a network to DetNet flows. DetNet provides these flows extremely low packet loss rates and assured maximum end-to-end delivery latency. General background and concepts of DetNet can be found in [I-D.ietf-detnet-architecture].

This document specifies the encapsulation and operation of deterministic networking over an MPLS data-plane. The approach is modeled on the operation of Pseudowires (PW) over an MPLS Packet Switched Network (PSN) [RFC3985][RFC4385].

The DetNet transport layer functionality that provides congestion protection for DetNet flows is assumed to be in place in a DetNet node.

This document does not define the associated control plane functions, or Operations, Administration, and Maintenance (OAM). It also does not specify traffic handling capabilities required to deliver congestion protection and latency control for DetNet flows at the DetNet transport layer.

## 2. Terminology

### 2.1. Terms used in this document

Editor's note: This section needs to be reviewed when the body of the text is closer to completion.

This document uses the terminology established in the DetNet architecture [I-D.ietf-detnet-architecture] and the DetNet Data Plane Solution Alternatives [I-D.ietf-detnet-dp-alt].

**T-Label** A label used to identify the LSP used to transport a DetNet flow across an MPLS PSN, e.g., a hop-by-hop label used between label switching routers (LSR).

**S-Label** A DetNet "service" label that is used between DetNet nodes that implement also the DetNet service layer functions. An S-Label is also used to identify a DetNet flow at DetNet service layer.

**Local-ID** A DetNet Edge and Relay node internal construct that uniquely identifies a DetNet flow within a node and never appear on-wire. It may be used to select proper forwarding and/or DetNet specific service function.

**PREF** A Packet Replication and Elimination Function (PREF) does the replication and elimination processing of DetNet flow packets in edge or relay nodes. The replication function is essentially the existing 1+1 protection mechanism. The elimination function reuses and extends the existing duplicate detection mechanism to operate over multiple (separate) DetNet member flows of a DetNet compound flow.

**DetNet Control Word** A control word used for sequencing and identifying duplicate packets at the DetNet service layer.

## 2.2. Abbreviations

Editor's note: This section needs to be reviewed when the body of the text is closer to completion.

The following abbreviations used in this document:

AC Attachment Circuit.

CE Customer Edge equipment.

CoS Class of Service.

CW Control Word.

d-CW DetNet Control Word.

DetNet Deterministic Networking.

DF DetNet Flow.

L2VPN Layer 2 Virtual Private Network.

LSR Label Switching Router.

MPLS Multiprotocol Label Switching.

MPLS-TP Multiprotocol Label Switching - Transport Profile.

MS-PW Multi-Segment Pseudowire (MS-PW).

NSP Native Service Processing.

OAM Operations, Administration, and Maintenance.

PE Provider Edge.

PREF Packet Replication and Elimination Function.

PSN Packet Switched Network.

PW Pseudowire.

QoS Quality of Service.

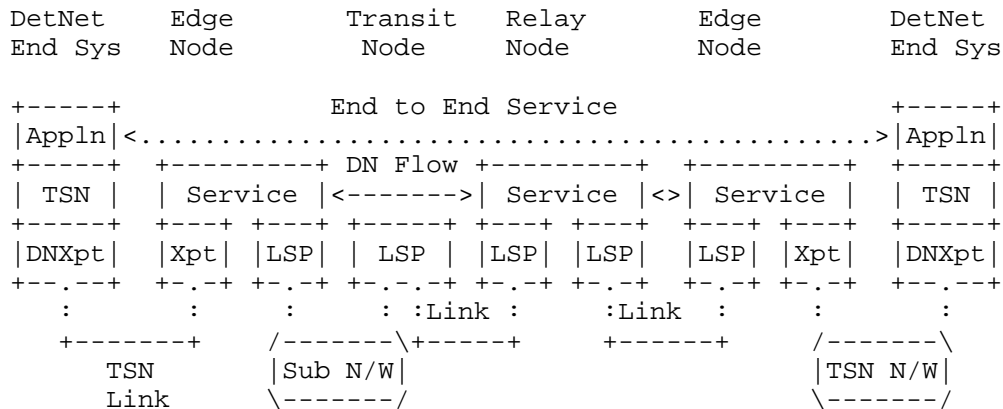
TSN Time-Sensitive Network.

### 3. Requirements language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

### 4. DetNet Over an MPLS Underlay

Figure 1 Shows the basic components of a DetNet enabled MPLS network used to transport TSN traffic using an MPLS transport.



LSP = MPLS Transport  
 DNXpt & Xpt = DetNet Transport

Figure 1: A Simple DetNet Enabled MPLS Network

TSN End Systems send and receive packets over the DetNet. The supported packet types are IP and Ethernet.

Edge Nodes are responsible for the imposition and disposition of the required DetNet encapsulation. These are functionally similar to pseudowire (PW) Terminating Provide Edge (T-PE) equipment.

Relay nodes are strategically placed and used enhance the reliability of delivery by enabling the replication of packets so that multiple copies, possibly over multiple paths. They also reduce the impact of replication by the eliminating surplus copies of DetNet packets. Replication and elimination may also be performed at ingress and egress edge nodes respectively.

Edge nodes, and relay nodes are aware of the needs of particular DetNet flows and take care to process them in accordance with the required performance needs.

Transit nodes are normal MPLS Label Switching Routers (LSRs). They are unaware of the special requirements of DetNet flows, although they may be configured to provide traffic engineering services to them to enhance the prospect of them meeting the required service level agreement (SLA).

The MPLS LSP may be provided by any MPLS method (LSP, RSVP-TE, MPLS-TP, or MPLS Segment Routing (SR)).

Figure 2 illustrates how end to end MPLS-based DetNet service is provided in a more detail. In this case, the end systems are able to send and receive DetNet flows. For example, an end system sends data encapsulated in MPLS. The 'X' in the end systems, edge and relay nodes represents potential DetNet flow packet replication and elimination points. Here the relay nodes may change the underlying transport, for example tunneling MPLS over IP [draft-xu-mpls-sr-over-ip], or simply interconnect network segments.

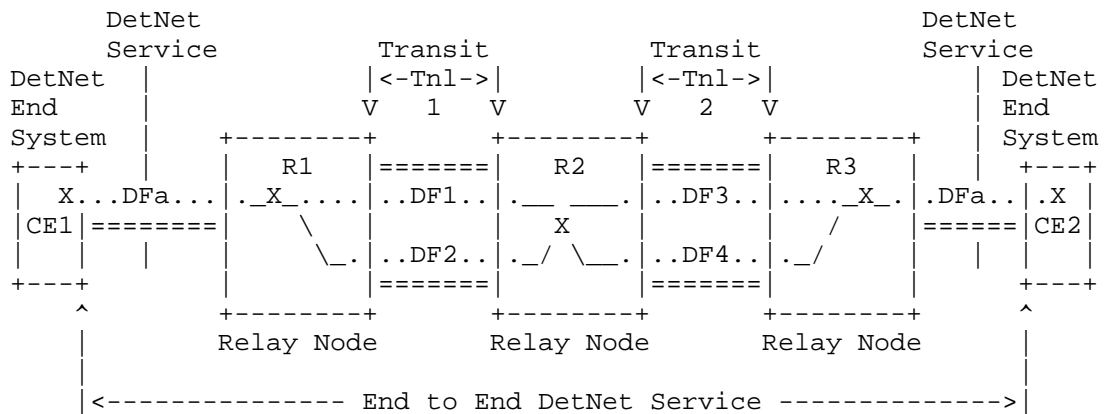


Figure 2: Flows in a DetNet Enabled MPLS Network

An example MPLS DetNet network fragment and packet flow is illustrated in Figure 3.

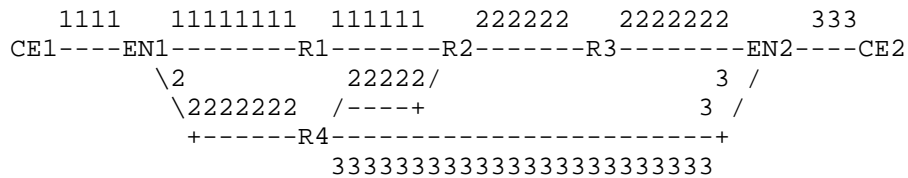


Figure 3: Example Packet flow in DetNet Enabled MPLS Network

Customer Equipment CE1 send a packet into the DetNet enabled MPLS network. Edge Node EN1 makes a copy of the packet and encapsulates each copy as a DetNet packet (packet 1 and packet 2). It sends one copy (1) to Relay Node R1 and the other copy (2) to Relay Node R4. R1 sends packet copy 1 to R2. R4 sends one copy to R2, and a further copy (3) to EN2. R2 receives copy (2) before copy 1, and so eliminated copy (1) sending only (2) to EN2. EN2 receives copy (3) first sending it to CE2 and eliminating copy (2). Note the number

illustrates the replication instance and should not be confused with the sequence number which remains unchanged in all packet copies.

The above is of course illustrative of many network scenarios that can be configured. Between a pair of relay nodes there may be one or more transport nodes that simply forward the DetNet traffic, but these are omitted for clarity.

## 5. DetNet over MPLS Encapsulation Components

To carry DetNet over MPLS the following is required:

1. A method of identifying the MPLS payload type.
2. A method of identifying the DetNet flow group to the processing element.
3. A method of distinguishing DetNet OAM packets from DetNet data packets.
4. A method of carrying the DetNet sequence number.
5. A suitable LSP to deliver the packet to the egress PE.
6. A method of carrying queuing and forwarding indication.

In this design an MPLS service label (the S-Label), similar to a pseudowire (PW) label [RFC3985], is used to identify both the DetNet flow identity and the payload MPLS payload type satisfying (1) and (2) in the list above. OAM traffic discrimination happens through the use of the Associated Channel method described in [RFC4385]. The sequence number is carried in the DetNet Control word which carries the Data/OAM discriminator. The LSP used to transport the DetNet packet may be of any type (MPLS-LDP, MPLS-TE, MPLS-TP[RFC5921], or MPLS-SR[I-D.ietf-spring-segment-routing-mpls]). The LSP (T-Label) label and/or the S-Label may be used to indicate the queue processing as well as the forwarding parameters.

Note that when the network consists only of DetNet enabled nodes with no aggregation, Penultimate Hop Popping (PHP) means that the only label in the label stack is the S-label.

### 5.1. Basic Data Plane Encapsulation

Figure 4 shows a DetNet data plane MPLS encapsulation. This is modeled on the encapsulation of pseudowires over MPLS [RFC3985].

The encapsulation consists of:



- o DetNet control word (d-CW) containing sequencing information for packet replication and duplicate elimination purposes, and the OAM indicator. There MUST a separate sequence number space for each DetNet flow.
- o DetNet Label (S-label) that identifies a DetNet flow to the peer node that is to process it. The S-Label is allocated from the platform label space.
- o Zero or more MPLS transport LSP label(s) (T-label) used to direct the packet along the label switched path (LSP) to the next peer node along the path. When Penultimate Hop Popping is in use there will be no label T-label in the protocol stack on the final hop.
- o The necessary data-link encapsulation is then applied prior to transmission over the physical media.

RFC Editor - if you ever get this text please remove this para (text to make compiler work).

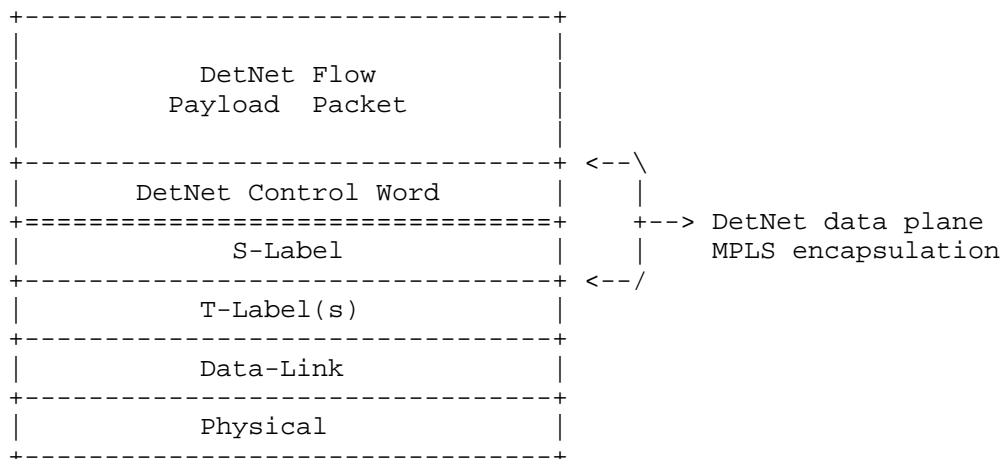


Figure 4: MPLS Encapsulation of DetNet

Flow aggregation may be necessary to achieve the required scaling. The extensions to basic encapsulation needed to support flow aggregation are described in Section 5.5.

## 5.2. DetNet Control Word

A DetNet Control Word (d-CW) conforms to the Generic PW MPLS Control Word (PWMCW) defined in [RFC4385] and is shown in Figure 5. In a DetNet data packet the upper nibble of the d-CW MUST be set to zero

(0). The effective sequence number bit length is between 0 and 28 bits, and configured either by a control plane or manually for each DetNet flow. The sequence number is aligned to the right (least significant bits) and unused bits MUST be set to zero (0). Each DetNet flow MUST have its own sequence number counter. The sequence number is incremented by one for each new packet.

Editor's note: We need to consider whether it is better to allow a multiplicity of sequence number lengths with a length configured for each flow, or a uniform sequence number of 28. On reflection it seems better to go for the simplicity of a standard length. If for any reason a different length becomes desirable, then it is relatively simple to define another type of DetNet d-CW with a different standard sequence number length and there is no ambiguity of operation since the sequence number length is a parameter of the S-Label.

The d-CW MUST always be present in a packet. In cases where the sequence number is not used (e.g., for DetNet-t-flows) the control plane or the manual configuration has to define zero (0) bit length sequence number and the value of the sequence number MUST be set to zero (0).

Editors Note: Do we set length zero or say it is not used? Also need to add text to stop relay nodes and egress edge nodes from processing the s/n otherwise only one packet ever gets through!

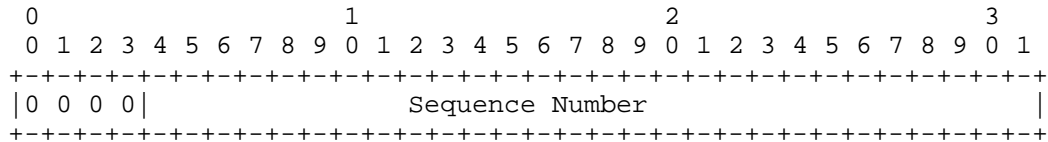


Figure 5: DetNet Control Word

### 5.3. Flow identification

DetNet flow identification at a DetNet service layer is realized by an S-label. It maps a DetNet flow to a specific d-CW in a DetNet node.

For a data flow the S-label used for flow identification MUST be bottom label of the label stack for a DetNet-s- or DetNet-st-flow and MUST precede the d-CW.

Editor's note: We have specified the above for `_data_` to leave the door open for the GAL based OAM method as an alternative to the ACH mechanism currently specified. When using GAL the GAL label would be

after the S-Label. Do we leave this option in or shut the door on it?

An S-label for a single DetNet flow MUST be unique at the peer at the node that is to process it. The S-label is stored in the platform label table to allow for DetNet packet processing independent of the interface on which the packet is received on.

#### 5.4. OAM Indication

OAM follows the procedures set out in [RFC5085] with the restriction that only Virtual Circuit Connectivity Verification (VCCV) type 1 is supported.

Editor's Note - is this restriction acceptable? Do we need to support GAL based OAM indication?

As shown in Figure 3 of [RFC5085] when the first nibble of the d-CW is 0x0 the payload following the d-CW is normal user data. However, when the first nibble of the d-CW is 0x1, the payload that follows the d-DW is an OAM payload with the OAM type indicated by the value in the d-CW Channel Type field.

The reader is referred to [RFC5085] for a more detailed description of the Associated Channel mechanism, and to the DetNet work on OAM for more information DetNet OAM.

#### 5.5. Flow Aggregation

The ability to aggregate individual flows, and their associated resource control, into a larger aggregate is an important technique for improving scaling of control in the data, management and control planes. The DetNet data plane allows for the aggregation of DetNet flows, to improved scaling. There are three methods of introducing flow aggregation:

1. Aggregate at the LSP (Transport)
2. Aggregating DetNet flows as a new DetNet flow
3. Simple Aggregation at the DetNet layer

A further method of using SR to perform aggregation is for further study.

The resource control and management aspects of aggregation (including the queuing/shaping/ policing implications) will be covered in other documents.

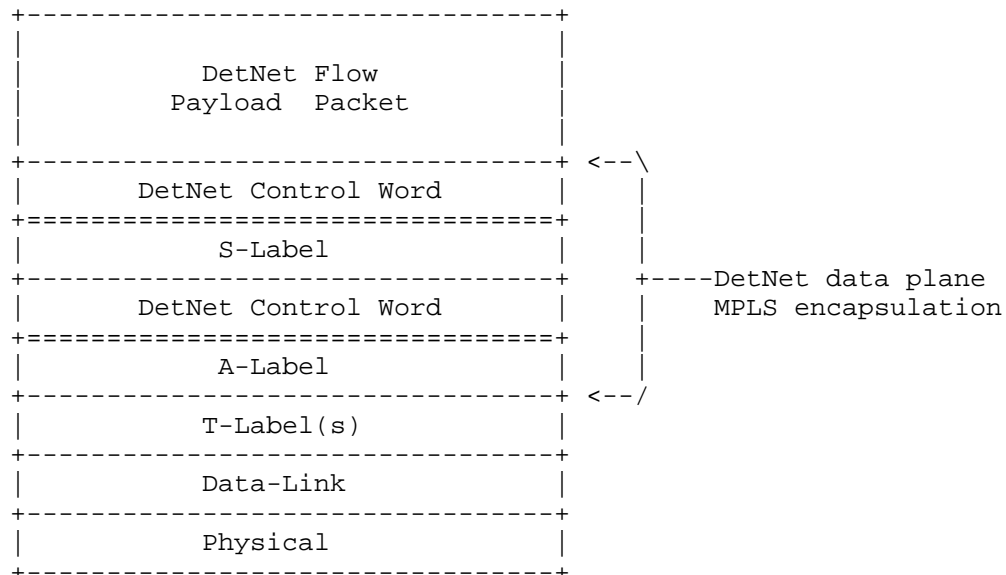
## 5.6. Aggregation at the LSP

DetNet flows transported via MPLS can leverage MPLS-TE's existing support for hierarchical LSPs (H-LSPs), see [RFC4206]. H-LSPs are typically used to aggregate control and resources, they may also be used to provide OAM or protection for the aggregated LSPs. Arbitrary levels of aggregation naturally falls out of the definition for hierarchy and the MPLS label stack [RFC3032]. DetNet nodes which support aggregation (LSP hierarchy) map one or more LSPs (labels) into and from an H-LSP. Both carried LSPs and H-LSPs may or may not use the TC field, i.e., L-LSPs or E-LSPs. Such nodes will need to ensure that traffic from aggregated LSPs are placed (shaped/policed/enqueued) onto the H-LSPs in a fashion that ensures the required DetNet service is preserved.

Additional details of the traffic control capabilities needed at a DetNet-aware node may be covered in the new service descriptions mentioned above or in separate future documents. Management and control plane mechanisms will also need to ensure that the service required on the aggregate flow (H-LSP or DSCP) are provided, which may include the discarding or remarking mentioned in the previous sections.

## 5.7. Aggregating DetNet flows as a new DetNet flow

An aggregate can be built by layering DetNet flows as shown below:

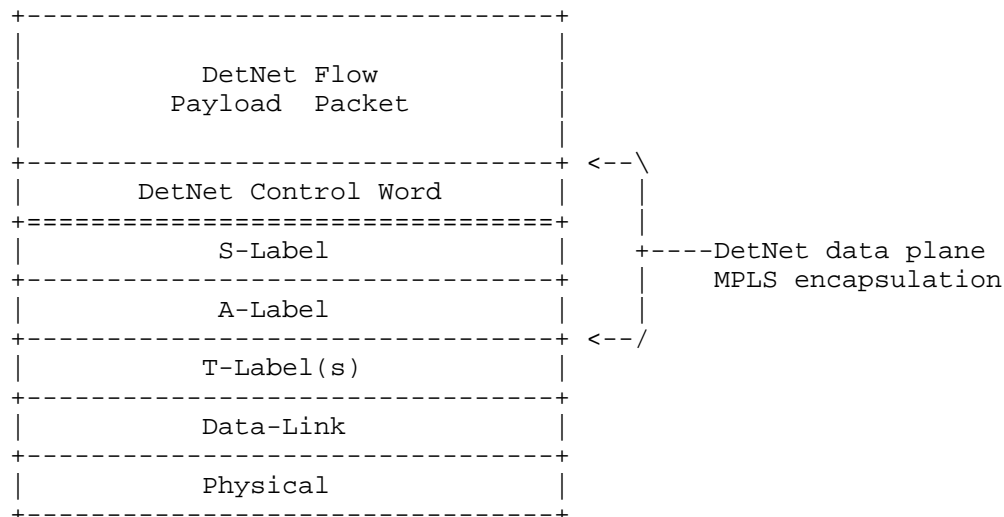


Both the Aggregation (A) label and the S-label have their MPLS S bit set indicating bottom of stack, and the d-CW allows the PREF functions to work.

It is a property of the A-label that what follows is d-CW followed by an S-label. A relay node processing the A-label would not know the underlying payload type. This would only be known to a node that was a peer of the node imposing the S-label. However there is no real need for it to know the payload type during aggregation processing.

## 6. Simple Aggregation at the DetNet layer

Another approach would be not to include a d-CW for the aggregated flow. This would be functionally similar to aggregation at the transport layer using H-LSPs, but would confine knowledge of the aggregation to the DetNet layer. Such an approach shares the disadvantage that PREF operations would not be possible. OAM operation in this mode is for further study.



## 7. Indication of the DetNet Payload Type.

The only node types that needs to know the payload type is the ingress node which has to know how to process the packet it receives on the ingress AC, and egress edge node which has to know how to prepare the packet for transmission on the egress AC.

On ingress a DetNet edge node has to classify the packets into those that are for transmission as Detnet packets and those that are for transmission as "normal" packets at one of more lower priorities.

The packet type is indicated to the egress edge node through the value of the S-label. Thus when the egress edge node looks up the S-label one of the parameters returned is the packet type which in turn tells the egress edge node how to prepare the packet for transmission over the egress AC.

Editor's note: Do we only find one type on the ingress and egress or do we need to run missed mode, i.e. will we have to send some packets across the network as Ethernet packets and some as IP packets?

## 8. Operation of the PREF Functions

The Packet Replication and Elimination Functions (PREF) are designed to enhance the reliability of delivery by network layer packet replication (PR), whilst at the same time minimizing network congestion and the duplicate delivery of packets over an egress AC by eliminating duplicate packets (EF).

PR and EF are independent functions that operate on a DetNet flow at strategic places in the network. The placement of a function is a matter for the network operator.

### 8.1. Operation of PR

A PR function creates two or more copies of a packet, and forwards a copy to each of the designated peers of the replicating node. A PR function may be placed in an ingress edge node, a relay node or an egress edge node.

We consider first a DetNet relay node. The packet is received from the upstream DetNet peer, and if present the T-label is popped. The S-label is looked up in the platform label table and if the forwarding instructions indicate that replication is required, the following happens for each next hop in the DetNet layer:

- o A copy of the payload is made.
- o An identical copy of the d-CW from the original packet is pushed.
- o The S-Label required by the next hop in the DetNet layer is pushed.
- o The DetNet packet copy is forwarded on the LSP to the next hop in the DetNet layer using normal MPLS forwarding.

An ingress node operates in the same way as a relay node, except that it is responsible for initial encapsulation of the packet. The packet is received from the AC, classified and prepared for

forwarding over the DetNet network as described in Section 9.1, except that for each next-hop in the DetNet Layer (i.e. each node that is to receive a copy of the DetNet packet) :

- o A copy of the payload is made.
- o The d-CW is constructed using the next sequence number in the sequence associated with this service.
- o An identical copy of the d-CW is pushed.
- o The S-Label required by the next hop in the DetNet layer to recognize this service is pushed.
- o The DetNet packet copy is forwarded on the LSP to the next hop in the DetNet layer using normal MPLS forwarding.

An egress node operates in the same way as as described in Section 9.1, except that where the S-Label indicates that the the packet is to be forwarded to a multiply attached system in the payload layer a similar copy (modified as needed to conform to any MAC requirements) is forwarded out of each egress AC.

Editor's Note: For normal PW, there will be one to one AC to PW binding relationship, for DetNet, no matter the ingress or egress side, there may be multiple AC corresponding to a single DetNet PW (S-label), should we state explicitly?

#### 8.2. Operation of EF

The EF function eliminates duplicate copies of a packet. The node identifies the service from the packet S-Label. If the S-Label indicates that the packet elimination function is required to operate on this service at this node, it uses the sequence number in the d-CW to determine whether or not the packet is a duplicate that must be eliminated, and precedes accordingly.

The EF may be placed in a Relay node, or an Edge Egress node.

#### 8.3. Packet reordering considerations

The DetNet service layer introduces packet reordering functionality for use in DetNet edge and relay node and end system packet processing. The reordering functionality MAY be enabled in a DetNet node. The reordering functionality relies on a presence of sequence numbers the d-CW.

#### 8.4. Indication of PREF processing

The indication that PR or EF processing is needed at a DetNet Relay node or a DetNet egress edge node is carried as an implicit characteristic of the S-Label. Thus, when a service is established the ingress edge node is configured to use an active rather than a static zero sequence number, and DetNet relays and egress edge nodes are configured to run PR and/or EF functionality on on services identified by specific S-labels.

#### 8.5. Placement of PR and EF in a node

This section is not normative.

The placement of the PR and EF functions within the node is a matter for the node designers, and this specification makes no determination on this matter. However the placement may well have implications for the management and control of a node, and thus the following is worth noting.

In a bladed system a common processing model is to analyze the packet for forwarding close to the ingress interface and to either to fully prepare it for forwarding as part of ingress processing, or to package it with internal metadata for final preparation close to the egress interface. In such systems the natural place to perform replication is as part of the ingress processing since egress processors often do not have the capability (for example processing power) to further process the packet, often do not normally have the required data paths to facilitate replication for transmission to other line cards.

Furthermore, in bladed systems it is to be expected that a packet from a peer in the MPLS layer may arrive on any of the blades. Whilst in principle some constraints could be applied on which interfaces the packets will arrive on, experience shows that such constraints are operationally impractical. To perform elimination state must be shared amongst the forwarders performing elimination. Sharing the state required for elimination between forwarders on different blades without sacrificing performance is technically difficult. Thus, the natural place for elimination in a distributed design is close to the egress interface.

On mono-processor systems these constraints do not apply in quite the same way, although even in this case it is necessary for the equipment designer to consider the implications of particular forwarder design, for example the allocation of Network Processing Unit (NPU) cores to particular interfaces.



A bladed system may place the PR and EF functions on a processing function other than the ingress or egress forwarding processor, whereupon the mono-processor considerations apply.

As noted above the placement of the PR and EF functions is a matter for the system designer. However it is important that nothing in the control, configuration, or OAM system results in undue difficulties for any of the forwarding models.

## 9. Operation at DetNet Node types

This section considers the operations at the various DetNet node types in more detail.

### 9.1. Operation at an Ingress Edge (PE) Node

An ingress Edge Node first classifies received traffic into DetNet flows and non-Detnet traffic. The DetNet flows are sent over the DetNet network using the procedures defined in the document. The classification process and the handling of non-DetNet traffic is out of scope for this document.

The processing of non-DetNet flows is currently outside the scope of this document

The packet is encapsulated as described in Section 5.1. First, if the flow is to be carried as IP the MAC header and checksum are removed. Otherwise the checksum is removed. The d-CW is constructed using the next sequence number associated with the service, and is pushed. The S-label corresponding to this DetNet flow is then pushed. The packet is then forwarded to the required egress edge-node by pushing the required T-Labels and the required data-link headers.

Editor's Note: we need text on how we indicate IPv4 vs IPv6. In MPLS they are carried on different FECs. Presumable we need to have a different S-Label for each IP type. This implies a different s/n for each IP type, but since it seems unlikely that we need to maintain them in a common sequence that looks to be OK.

### 9.2. Operation at a Relay node (S-PE)

At a relay node packet forwarding follows the same process used in a PW S-PE [RFC5659], save for the additional processing required for any required PREF processing.

The packet is received from the incoming LSP and any T-labels which are still present are popped. The S-label is looked up in the

platform label table to determine the forwarding parameters. If PREF processing is required the process described in Section 8 is followed, and if required the locally held sequence number information associated with the S-label is updated to avoid future duplicate forwarding.

For each copy of the packet to be forwarded the S-label is swapped for the S-label required by either the next relay node, or by the egress edge node. For each copy of the packet, the T-label(s) needed to reach the next relay node or the egress edge node is pushed and the packet forwarded towards its next hop in the MPLS layer.

### 9.3. Operation at an Egress Edge (PE) Node

At an egress edge node, the S-Label is looked up in the platform label table and is used to determine the egress packet processing parameters. The sequence number in d-CW and the recorded sequence number history is compared to perform any required ER processing Section 8.2. If the packet is not eliminated, the d-CW is stripped.

If the packet is to be treated as an IP packet, this is processed in the normal way for an IP packet egressing an MPLS tunnel (for example the data-link header is constructed) and the packet forwarded towards its destination.

Editor's note: Do we need to look the IP address up in a forwarding table, if so, which one, or is the next hop a forwarding parameter. Is the MAC address a forwarding parameter, or do we need to run ARP as normal?

If the packet is to be treated as an Ethernet packet it is forwarded unmodified, save for the addition of the CRC as described in [RFC4448].

### 9.4. Operation at a Transit(P) Node

Operation at a transit (P) node is normal MPLS forwarding. The outer label is either swapped or popped as required, and the packet is forwarded along the LSP. If an entropy label is present in the label stack, this may be used by the Equal Cost Multi-Path (ECMP) selection process. No other label is inspected as part of forwarding.

The Traffic Class field and/or incoming LSP transport label may be used to indicate how the packet is to be processed and queued including which forwarding resources are to be applied to the packet ((RFC3270)). Any of the methods of constructing a physical LSP (for RSVP-TE signaling or MPLS-TP style controller based configuration) or a virtual LSP (Segment Routing

[I-D.ietf-spring-segment-routing-mpls]) may be used to indicate not only the next hop, but the priority of processing and any physical resources dedicated to the service [I-D.bryant-rtgwg-enhanced-vpn].

## 10. Other DetNet data plane considerations

### 10.1. Class of Service

Class and quality of service, i.e., CoS and QoS, are terms that are often used interchangeably and confused. In the context of DetNet, CoS is used to refer to mechanisms that provide traffic forwarding treatment based on aggregate group basis and QoS is used to refer to mechanisms that provide traffic forwarding treatment based on a specific DetNet flow basis. Examples of existing network level CoS mechanisms include DiffServ which is enabled by IP header differentiated services code point (DSCP) field [RFC2474] and MPLS label traffic class field [RFC5462], and at Layer-2, by IEEE 802.1p priority code point (PCP).

CoS for DetNet flows carried in PWs and MPLS is provided using the existing MPLS Differentiated Services (DiffServ) architecture [RFC3270]. Both E-LSP and L-LSP MPLS DiffServ modes MAY be used to support DetNet flows. The Traffic Class field (formerly the EXP field) of an MPLS label follows the definition of [RFC5462] and [RFC3270]. The Uniform, Pipe, and Short Pipe DiffServ tunneling and TTL processing models are described in [RFC3270] and [RFC3443] and MAY be used for MPLS LSPs supporting DetNet flows. MPLS ECN MAY also be used as defined in ECN [RFC5129] and updated by [RFC5462].

One additional consideration for DetNet nodes which support CoS services is that they MUST ensure that the CoS service classes do not impact the congestion protection and latency control mechanisms used to provide DetNet QoS. This requirement is similar to requirement for MPLS LSRs to that CoS LSPs do not impact the resources allocated to TE LSPs via [RFC3473].

### 10.2. Quality of Service

Quality of Service (QoS) mechanisms for flow specific traffic treatment typically includes a guarantee/agreement for the service, and allocation of resources to support the service. Example QoS mechanisms include discrete resource allocation, admission control, flow identification and isolation, and sometimes path control, traffic protection, shaping, policing and remarking. Example protocols that support QoS control include Resource ReSerVation Protocol (RSVP) [RFC2205] and RSVP-TE [RFC3209] and [RFC3473]. The existing MPLS mechanisms defined to support CoS [RFC3270] can also be used to reserve resources for specific traffic classes.

In addition to explicit routes Section 11.2, and packet replication and elimination, described in Section 8 above, DetNet provides zero congestion loss and bounded latency and jitter. As described in [I-D.ietf-detnet-architecture], there are different mechanisms that maybe used separately or in combination to deliver a zero congestion loss service. These mechanisms are provided by the either the MPLS or IP layers, and may be combined with the mechanisms defined by the underlying network layer such as 802.1TSN.

A baseline set of QoS capabilities for DetNet flows carried over MPLS can be provided with Traffic Engineering (MPLS-TE) [RFC3209] and [RFC3473]. TE LSPs can also support explicit routes (path pinning). Current service definitions for packet TE LSPs can be found in "Specification of the Controlled Load Quality of Service", [RFC2211], "Specification of Guaranteed Quality of Service", [RFC2212], and "Ethernet Traffic Parameters", [RFC6003]. Additional service definitions are expected in future documents to support the full range of DetNet services. In all cases, the existing label-based marking mechanisms defined for TE-LSPs and even E-LSPs are use to support the identification of flows requiring DetNet QoS.

Packets that are marked with a DetNet Class of Service value, but that have not been the subject of a completed reservation, can disrupt the QoS offered to properly reserved DetNet flows by using resources allocated to the reserved flows. Therefore, the network nodes of a DetNet network MUST:

- o Defend the DetNet QoS by discarding or remarking (to a non-DetNet CoS) packets received that are not the subject of a completed reservation.
- o Not use a DetNet reserved resource, e.g. a queue or shaper reserved for DetNet flows, for any packet that does not carry a DetNet Class of Service marker.

### 10.3. Bidirectional traffic

Some DetNet applications generate bidirectional traffic. Using MPLS definitions [RFC5654] there are associated bidirectional flows, and co-routed bidirectional flows. MPLS defines a point-to-point associated bidirectional LSP as consisting of two unidirectional point-to-point LSPs, one from A to B and the other from B to A, which are regarded as providing a single logical bidirectional transport path. This would be analogous of standard IP routing, or PWs running over two reciprocal unidirectional LSPs. MPLS defines a point-to-point co-routed bidirectional LSP as an associated bidirectional LSP which satisfies the additional constraint that its two unidirectional component LSPs follow the same path (in terms of both nodes and

links) in both directions. An important property of co-routed bidirectional LSPs is that their unidirectional component LSPs share fate. In both types of bidirectional LSPs, resource allocations may differ in each direction. The concepts of associated bidirectional flows and co-routed bidirectional flows can be applied to DetNet flows as well. PWs [RFC3985] are always created as bidirectional constructs.

While the MPLS data planes must support bidirectional DetNet flows, there are no special bidirectional features with respect to the data plane other than need for the two directions take the same paths. Fate sharing and associated vs co-routed bidirectional flows can be managed at the control level. Note, that there is no stated requirement for bidirectional DetNet flows to be supported using the same MPLS Labels in each direction, and indeed to do so would introduce significant implementation issues. Control mechanisms will be need to support such bidirectional flows DetNet over MPLS, but such mechanisms are out of scope of this document. An example control plane solution for MPLS can be found in [RFC7551].

#### 10.4. Layer 2 addressing and QoS Considerations

Editor's note: Add references needed by this section

The Time-Sensitive Networking (TSN) Task Group of the IEEE 802.1 Working Group have defined (and are defining) a number of amendments to IEEE 802.1Q [IEEE8021Q] that provide zero congestion loss and bounded latency in bridged networks. IEEE 802.1CB [IEEE8021CB] defines packet replication and elimination functions that should prove both compatible with and useful to, DetNet networks.

As is the case for DetNet, a Layer 2 network node such as a bridge may need to identify the specific DetNet flow to which a packet belongs in order to provide the TSN/DetNet QoS for that packet. It also will likely need a CoS marking, such as the priority field of an IEEE Std 802.1Q VLAN tag, to give the packet proper service.

Although the flow identification methods described in IEEE 802.1CB [IEEE8021CB] are flexible, and in fact, include IP 5-tuple identification methods, the baseline TSN standards assume that every Ethernet frame belonging to a TSN stream (i.e. DetNet flow) carries a multicast destination MAC address that is unique to that flow within the bridged network over which it is carried. Furthermore, IEEE 802.1CB [IEEE8021CB] describes three methods by which a packet sequence number can be encoded in an Ethernet frame.

Ensuring that the proper Ethernet VLAN tag priority and destination MAC address are used on a DetNet/TSN packet may require further

clarification of the customary L2/L3 transformations carried out by routers and edge label switches. Edge nodes may also have to move sequence number fields among Layer 2, PW, and IPv6 encapsulations.

## 11. Management and control considerations

While management plane and control planes are traditionally considered separately, from the Data Plane perspective there is no practical difference based on the origin of flow provisioning information. This document therefore does not distinguish between information provided by a control plane protocol, e.g., RSVP-TE [RFC3209] and [RFC3473], or by a network management mechanisms, e.g., RestConf [RFC8040] and YANG [RFC7950].

Editor's note: Not sure if RSVP-TE can be used, indeed unsure what routing protocols can be use other than to create point to point MPLS transport paths. Normally we construct a a single path through the network with RSVP-TE, but here we need to construct an explicit mesh at the DetNet layer. The classical routing protocols are not really capable of constructing graphs of the sort needed here either.

### 11.1. S-Label assignment and distribution

A mechanism based on the existing PW label distribution protocol [RFC8077] can be used to exchange labels between DetNet nodes. The protocol may however need extending depending on the preferred format of the DetNet flow identifiers.

A mechanism to create the flow graph through the relay nodes will need to be specified. Initially this is likely to be based on a network controller of some sort rather than a distributed routing protocol.

The details of the control plane protocol solution required for the label distribution and the management of the label number space are out of scope of this document.

### 11.2. Explicit routes

Editor's note describe the applicability of explicit routes as a method of constructing paths

## 12. Security considerations

The security considerations of DetNet in general are discussed in [I-D.ietf-detnet-architecture] and [I-D.sdt-detnet-security]. Other security considerations will be added in a future version of this draft.

### 13. IANA considerations

This document makes no IANA requests.

#### 13.1. Acknowledgments

We acknowledge the authors of draft-ietf-detnet-dp-sol-01.

### 14. References

#### 14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <<https://www.rfc-editor.org/info/rfc4448>>.
- [RFC5085] Nadeau, T., Ed. and C. Pignataro, Ed., "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, DOI 10.17487/RFC5085, December 2007, <<https://www.rfc-editor.org/info/rfc5085>>.

#### 14.2. Informative References

- [I-D.bryant-rtgwg-enhanced-vpn]  
Bryant, S. and J. Dong, "Enhanced Virtual Private Networks (VPN+)", draft-bryant-rtgwg-enhanced-vpn-01 (work in progress), October 2017.
- [I-D.ietf-detnet-architecture]  
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-04 (work in progress), October 2017.
- [I-D.ietf-detnet-dp-alt]  
Korhonen, J., Farkas, J., Mirsky, G., Thubert, P., Zhuangyan, Z., and L. Berger, "DetNet Data Plane Protocol and Solution Alternatives", draft-ietf-detnet-dp-alt-00 (work in progress), October 2016.

- [I-D.ietf-detnet-dp-sol]  
Korhonen, J., Andersson, L., Jiang, Y., Finn, N., Varga, B., Farkas, J., Bernardos, C., Mizrahi, T., and L. Berger, "DetNet Data Plane Encapsulation", draft-ietf-detnet-dp-sol-01 (work in progress), January 2018.
- [I-D.ietf-spring-segment-routing-mpls]  
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-12 (work in progress), February 2018.
- [I-D.sdt-detnet-security]  
Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell, J., Austad, H., Stanton, K., and N. Finn, "Deterministic Networking (DetNet) Security Considerations", draft-sdt-detnet-security-01 (work in progress), July 2017.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.
- [RFC2211] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", RFC 2211, DOI 10.17487/RFC2211, September 1997, <<https://www.rfc-editor.org/info/rfc2211>>.
- [RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, DOI 10.17487/RFC2212, September 1997, <<https://www.rfc-editor.org/info/rfc2212>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.



- [RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <<https://www.rfc-editor.org/info/rfc3270>>.
- [RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <<https://www.rfc-editor.org/info/rfc3443>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <<https://www.rfc-editor.org/info/rfc3985>>.
- [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, DOI 10.17487/RFC4206, October 2005, <<https://www.rfc-editor.org/info/rfc4206>>.
- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<https://www.rfc-editor.org/info/rfc4385>>.
- [RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, DOI 10.17487/RFC5129, January 2008, <<https://www.rfc-editor.org/info/rfc5129>>.
- [RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <<https://www.rfc-editor.org/info/rfc5462>>.
- [RFC5654] Niven-Jenkins, B., Ed., Brungard, D., Ed., Betts, M., Ed., Sprecher, N., and S. Ueno, "Requirements of an MPLS Transport Profile", RFC 5654, DOI 10.17487/RFC5654, September 2009, <<https://www.rfc-editor.org/info/rfc5654>>.

- [RFC5659] Bocci, M. and S. Bryant, "An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge", RFC 5659, DOI 10.17487/RFC5659, October 2009, <<https://www.rfc-editor.org/info/rfc5659>>.
- [RFC5921] Bocci, M., Ed., Bryant, S., Ed., Frost, D., Ed., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, DOI 10.17487/RFC5921, July 2010, <<https://www.rfc-editor.org/info/rfc5921>>.
- [RFC6003] Papadimitriou, D., "Ethernet Traffic Parameters", RFC 6003, DOI 10.17487/RFC6003, October 2010, <<https://www.rfc-editor.org/info/rfc6003>>.
- [RFC7551] Zhang, F., Ed., Jing, R., and R. Gandhi, Ed., "RSVP-TE Extensions for Associated Bidirectional Label Switched Paths (LSPs)", RFC 7551, DOI 10.17487/RFC7551, May 2015, <<https://www.rfc-editor.org/info/rfc7551>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8077] Martini, L., Ed. and G. Heron, Ed., "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", STD 84, RFC 8077, DOI 10.17487/RFC8077, February 2017, <<https://www.rfc-editor.org/info/rfc8077>>.

#### Authors' Addresses

Stewart Bryant  
Huawei Technologies

Email: [stewart.bryant@gmail.com](mailto:stewart.bryant@gmail.com)

Mach Chen  
Huawei Technologies

Email: [mach.chen@huawei.com](mailto:mach.chen@huawei.com)

DetNet  
Internet-Draft  
Intended status: Standards Track  
Expires: September 6, 2018

N. Finn  
Huawei Technologies Co. Ltd  
J-Y. Le Boudec  
EPFL  
B. Varga  
J. Farkas  
Ericsson  
March 5, 2018

DetNet Bounded Latency  
draft-finn-detnet-bounded-latency-00

Abstract

This document presents a parameterized timing model for Deterministic Networking so that existing and future standards can achieve bounded latency and zero congestion loss.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions Used in This Document . . . . .	3
3. Terminology and Definitions . . . . .	4
4. DetNet bounded latency model . . . . .	4
4.1. Flow creation . . . . .	4
4.2. End-to-end model . . . . .	5
4.3. Relay system model . . . . .	5
5. Computing End-to-end Latency Bounds . . . . .	7
5.1. Examples of Computations . . . . .	8
6. Achieving zero congestion loss . . . . .	8
6.1. A General Formula . . . . .	8
7. Queuing model . . . . .	9
7.1. Queuing data model . . . . .	9
7.2. IEEE 802.1 Queuing Model . . . . .	11
7.2.1. Queuing Data Model with Preemption . . . . .	11
7.2.2. Transmission Selection Model . . . . .	12
7.3. Other queuing models, e.g. IntServ . . . . .	14
8. Parameters for the bounded latency model . . . . .	14
8.1. Sender parameters . . . . .	14
8.2. Relay system parameters . . . . .	14
9. References . . . . .	15
9.1. Normative References . . . . .	15
9.2. Informative References . . . . .	15
Authors' Addresses . . . . .	17

## 1. Introduction

The ability for IETF Deterministic Networking (DetNet) or IEEE 802.1 Time-Sensitive Networking (TSN) to provide the DetNet services of bounded latency and zero congestion loss depends upon A) configuring and allocating network resources for the exclusive use of DetNet/TSN flows; B) identifying, in the data plane, the resources to be utilized by any given packet, and C) the detailed behavior of those resources, especially transmission queue selection, so that latency bounds can be reliably assured. Thus, DetNet is an example of an INTSERV Guaranteed Quality of Service [RFC2212]

As explained in [I-D.ietf-detnet-architecture], DetNet flows are characterized by 1) a maximum bandwidth, guaranteed either by the transmitter or by strict input metering; and 2) a requirement for a guaranteed worst-case end-to-end latency. That latency guarantee, in turn, provides the opportunity for the network to supply enough buffer space to guarantee zero congestion loss. To be of use to the

applications identified in [I-D.ietf-detnet-use-cases], it must be possible to calculate, before the transmission of a DetNet flow commences, both the worst-case end-to-end network latency, and the amount of buffer space required at each hop to ensure against congestion loss.

Rather than defining, in great detail, specific mechanisms to be used to control packet transmission at each output port, this document presents a timing model for sources, destinations, and the network nodes that relay packets. The parameters specified in this model:

- o Characterize a DetNet flow in a way that provides externally measureable verification that the sender is conforming to its promised maximum, can be implemented reasonably easily by a sending device, and does not require excessive over-allocation of resources by the network.
- o Enable reasonably accurate computation of worst-case end-to-end latency, in a way that requires as little detailed knowledge as possible of the behavior of the Quality of Service (QoS) algorithms implemented in each device, including queuing, shaping, metering, policing, and transmission selection techniques.

Using the model presented in this document, it should be possible for an implementor, user, or standards development organization to select a particular set of QoS algorithms for each device in a DetNet network, and to select a resource reservation algorithm for that network, so that those elements can work together to provide the DetNet service.

This document does not specify any resource reservation protocol or server. It does not describe all of the requirements for that protocol or server. It does describe a set of requirements for resource reservation algorithms and for QoS algorithms that, if met, will enable them to work together.

## 2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The lowercase forms with an initial capital "Must", "Must Not", "Shall", "Shall Not", "Should", "Should Not", "May", and "Optional" in this document are to be interpreted in the sense defined in [RFC2119], but are used where the normative behavior is defined in documents published by SDOs other than the IETF.

### 3. Terminology and Definitions

This document uses the terms defined in [I-D.ietf-detnet-architecture].

### 4. DetNet bounded latency model

#### 4.1. Flow creation

The bounded latency model assumes the use of the following paradigm for provisioning a particular DetNet flow:

1. Perform any onfiguration required by the relay systems in the network for the classes of service to be offered, including one or more classes of DetNet service. This configuration is general; it is not tied to any particular flow.
2. Characterize the DetNet flow in terms of limitations on the sender Section 8.1 and flow requirements Section 8.2.
3. Establish the path that the DetNet flow will take through the network from the source to the destination(s). This can be a point-to-point or a point-to-multipoint path.
4. Select one of the DetNet classes of service for the DetNet flow.
5. Compute the worst-case end-to-end latency for the DetNet flow. In the process, determine whether sufficient resources are available for that flow to guarantee the required latency and provide zero congestion loss.
6. Assuming that the resources are available, commit those resources to the flow. This may or may not require adjusting the parameters that control the QoS algorithms at each hop along the flow's path.

This paradigm can be static and/or dynamic, and can be implemented using peer-to-peer protocols or with a central server model. In some situations, backtracking and recursing through this list may be necessary.

Issues such as un-provisioning a DetNet flow in favor of another when resources are scarce are not considered. How the path to be taken by a DetNet flow is chosen is not considered in this document.

## 4.2. End-to-end model

[Suggestion: This is the introduction to network calculus. The starting point is a model in which a relay system is a black box.]

### 4.3. Relay system model

[NWF I think that at least some of this will be useful. We won't know until we see what J-Y has to say in Section 4.2. I'm especially interested in whether J-Y thinks that the "output delay" in Figure 1 is useful in determining the number of buffers needed in the next hop. It is possible that we can define the parameters we need without this section.]

In Figure 1 we see a breakdown of the per-hop latency experienced by a packet passing through a relay system, in terms that are suitable for computing both hop-by-hop latency and per-hop buffer requirements.

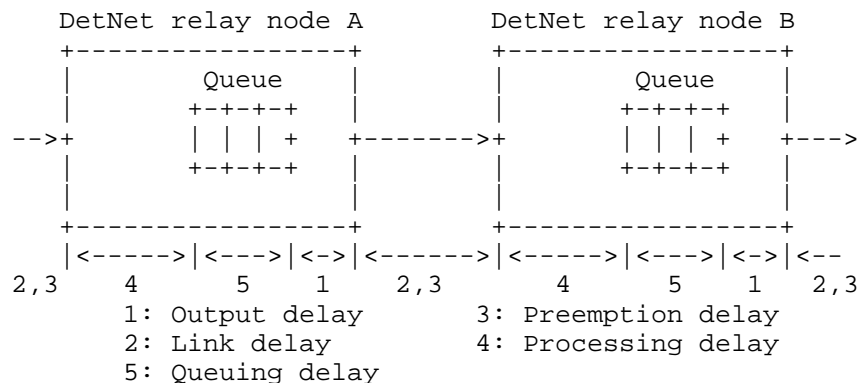


Figure 1: Timing model for DetNet or TSN

In Figure 1, we see two DetNet relay nodes (typically, bridges or routers), with a wired link between them. In this model, the only queues we deal with explicitly are attached to the output port; other queues are modeled as variations in the other delay times. (E.g., an input queue could be modeled as either a variation in the link delay [2] or the processing delay [4].) There are five delays that a packet can experience from hop to hop.

## 1. Output delay

The time taken from the selection of a packet for output from a queue to the transmission of the first bit of the packet on the physical link. If the queue is directly attached to the physical port, output delay can be a constant. But, in many

implementations, the queuing mechanism in a forwarding ASIC is separated from a multi-port MAC/PHY, in a second ASIC, by a multiplexed connection. This causes variations in the output delay that are hard for the forwarding node to predict or control.

2. Link delay

The time taken from the transmission of the first bit of the packet to the reception of the last bit, assuming that the transmission is not suspended by a preemption event. This delay has two components, the first-bit-out to first-bit-in delay and the first-bit-in to last-bit-in delay that varies with packet size. The former is typically measured by the Precision Time Protocol and is constant (see [I-D.ietf-detnet-architecture]). However, a virtual "link" could exhibit a variable link delay.

3. Preemption delay

If the packet is interrupted (e.g. [IEEE8023br] preemption) in order to transmit another packet or packets, an arbitrary delay can result.

4. Processing delay

This delay covers the time from the reception of the last bit of the packet to that packet being eligible, if there were no other packets in the queue, for selection for output. This delay can be variable, and depends on the details of the operation of the forwarding node.

5. Queuing delay

This is the time spent from the insertion of the packet into a queue until the packet is selected for output on the next link. We assume that this time is calculable based on the details of the queuing mechanism.

Not shown in Figure 1 are the other output queues that we presume are also attached to that same output port as the queue shown, and against which this shown queue competes for transmission opportunities.

The initial and final measurement point in this analysis (that is, the definition of a "hop") is the point at which a packet is selected for output. In general, any queue selection method that is suitable for use in a DetNet network includes a detailed specification as to exactly when packets are selected for transmission. Any variations in any of the delay times 1-4 result in a need for additional buffers in the queue. If all delays 1-4 are constant, then any variation in the time at which packets are inserted into a queue depends entirely on the timing of packet selection in the previous node. If the



delays 1-4 are not constant, then additional buffers are required in the queue to absorb these variations. Thus:

- o Variations in output delay (1) require buffers to absorb that variation in the next hop, so the output delay variations of the previous hop (on each input port) must be known in order to calculate the buffer space required on this hop.
- o Variations in processing delay (4) require additional output buffers in the queues of that same Detnet relay node. Depending on the details of the queueing delay (5) calculations, these variations need not be visible outside the DetNet relay node.

## 5. Computing End-to-end Latency Bounds

End-to-end latency bounds can be computed using the delay model in Section 4.3. Here it is important to be aware that for several queuing mechanisms, the worst-case end-to-end delay is less than the sum of the per-hop worst-case delays. An end-to-end latency bound for one detnet flow can be computed as

$$\text{end\_to\_end\_latency\_bound} = \text{non\_queuing\_latency} + \text{queuing\_latency}$$

The two terms in the above formula are computed as follows. First, at the  $h$ -th hop along the path of this detnet flow, obtain an upper bound per-hop  $\text{non\_queuing\_latency}[h]$  on the sum of delays 1,2,3,4 of Figure 1. These upper-bounds are expected to depend on the specific technology of the node at the  $h$ -th hop but not on the T-SPEC of this detnet flow. Then set  $\text{non\_queuing\_latency} =$  the sum of  $\text{per-hop\_non\_queuing\_latency}[h]$  over all hops  $h$ .

Second, compute  $\text{queuing\_latency}$  as an upper bound to the sum of the queuing delays along the path. The value of  $\text{queuing\_latency}$  depends on the T-SPEC of this flow and possibly of other flows in the network, as well as the specifics of the queuing mechanisms deployed along the path of this flow.

For several queuing mechanisms,  $\text{queuing\_latency}$  is less than the sum of upper bounds on the queuing delay (5) at every hop. Section 5.1 gives such practical computation examples.

For other queuing mechanisms the only available value of  $\text{queuing\_latency}$  is the sum of the per-hop queuing delay bounds. In such cases, the computation of per-hop queuing delay bounds must account for the fact that the T-SPEC of a detnet flow is no longer satisfied at the ingress of a hop, since burstiness increases as one flow traverses one detnet node.

### 5.1. Examples of Computations

[[ JYLB: THIS IS WHERE DETAILS OF END-TO-END LATENCY COMPUTATION ARE GIVEN FOR PER-FLOW QUEUING AND FOR TSN WITH ATS]]

## 6. Achieving zero congestion loss

When the input rate to an output queue exceeds the output rate for a sufficient length of time, the queue must overflow. This is congestion loss, and this is what deterministic networking seeks to avoid.

### 6.1. A General Formula

To avoid congestion losses, an upper bound on the backlog present in the queue of Figure 1 must be computed during path computation. This bound depends on the set of flows that use this queue, the details of the specific queuing mechanism and an upper bound on the processing delay (4). The queue must contain the packet in transmission plus all other packets that are waiting to be selected for output.

A conservative backlog bound, that applies to all systems, can be derived as follows.

The backlog bound is counted in data units (bytes, or words of multiple bytes) that are relevant for buffer allocation. For every class we need one buffer space for the packet in transmission, plus space for the packets that are waiting to be selected for output. Excluding transmission and preemption times, the packets are waiting in the queue since reception of the last bit, for a duration equal to the processing delay (4) plus the queuing delay (5).

Let

- o `nb_classes` be the number of classes of traffic that may use this output port
- o `total_in_rate` be the sum of the line rates of all input ports that send traffic of any class to this output port. The value of `total_in_rate` is in data units (e.g. bytes) per second.
- o `nb_input_ports` be the number input ports that send traffic of any class to this output port
- o `max_packet_length` be the maximum packet size for packets of any class that may be sent to this output port. This is counted in data units.

- o `max_delay45` be an upper bound, in seconds, on the sum of the processing delay (4) and the queuing delay (5) for a packet of any class at this output port.

Then a bound on the backlog of traffic of all classes in the queue at this output port is

```
backlog_bound = ( nb_classes + nb_input_ports ) *  
max_packet_length + total_in_rate* max_delay45
```

## 7. Queuing model

[[ JYLB: THIS IS WHERE DETAILS OF END-TO-END LATENCY COMPUTATION ARE GIVEN FOR PER-FLOW QUEUING AND FOR TSN WITH ATS]]

### 7.1. Queuing data model

Sophisticated QoS mechanisms are available in Layer 3 (L3), see, e.g., [RFC7806] for an overview. In general, we assume that "Layer 3" queues, shapers, meters, etc., are instantiated hierarchically above the "Layer 2" queuing mechanisms, among which packets compete for opportunities to be transmitted on a physical (or sometimes, logical) medium. These "Layer 2 queuing mechanisms" are not the province solely of bridges; they are an essential part of any DetNet relay node. As illustrated by numerous implementation examples, the "Layer 3" some of mechanisms described in documents such as [RFC7806] are often integrated, in an implementation, with the "Layer 2" mechanisms also implemented in the same system. An integrated model is needed in order to successfully predict the interactions among the different queuing mechanisms needed in a network carrying both DetNet flows and non-DetNet flows.

Figure 2 shows the (very simple) model for the flow of packets through the queues of an IEEE 802.1Q bridge. Packets are assigned to a class of service. The classes of service are mapped to some number of physical FIFO queues. IEEE 802.1Q allows a maximum of 8 classes of service, but it is more common to implement 2 or 4 queues on most ports.

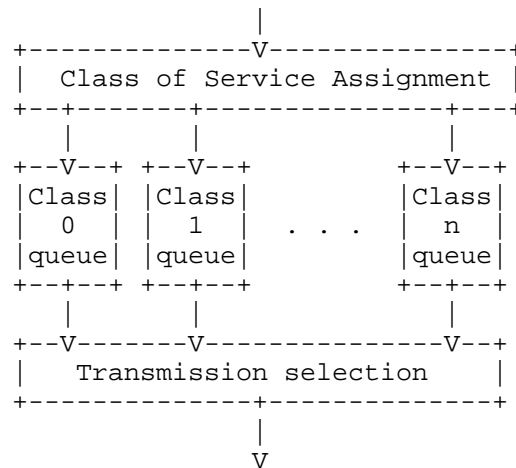


Figure 2: IEEE 802.1Q Queuing Model: Data flow

Some relevant mechanisms are hidden in this figure, and are performed in the "Class n queue" box:

- o Discarding packets because a queue is full.
- o Discarding packets marked "yellow" by a metering function, in preference to discarding "green" packets.

The Class of Service Assignment function can be quite complex, since the introduction of [IEEE802.1Qci]. In addition to the Layer 2 priority expressed in the 802.1Q VLAN tag, a bridge can utilize any of the following information to assign a packet to a particular class of service (queue):

- o Input port.
- o Selector based on a rotating schedule that starts at regular, time-synchronized intervals and has nanosecond precision.
- o MAC addresses, VLAN ID, IP addresses, Layer 4 port numbers, DSCP. (Work items expected to add MPC and other indicators.)
- o The Class of Service Assignment function can contain metering and policing functions.

The "Transmission selection" function decides which queue is to transfer its oldest packet to the output port when a transmission opportunity arises.

## 7.2. IEEE 802.1 Queuing Model

### 7.2.1. Queuing Data Model with Preemption

Figure 2 must be modified if the output port supports preemption ([IEEE8021Qbu] and [IEEE8023br]). This modification is shown in Figure 3.

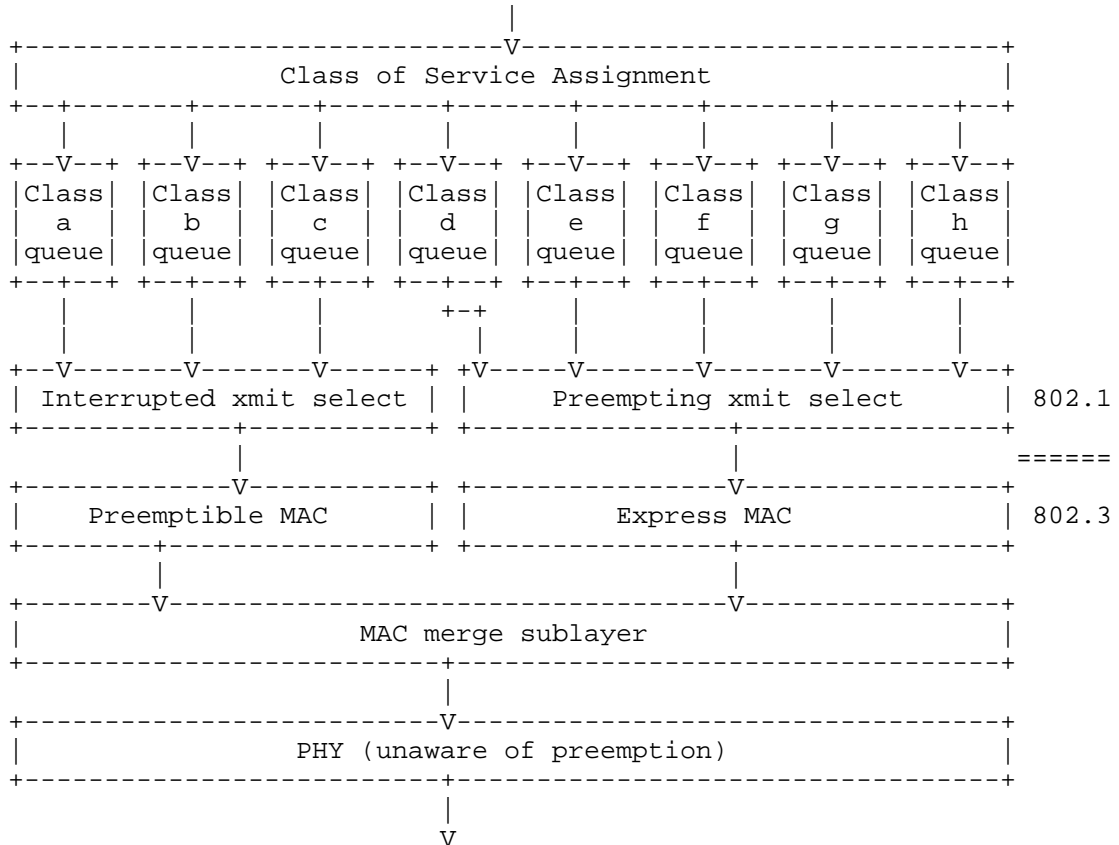


Figure 3: IEEE 802.1Q Queuing Model: Data flow with preemption

From Figure 3, we can see that, in the IEEE 802 model, the preemption feature is modeled as consisting of two MAC/PHY stacks, one for packets that can be interrupted, and one for packets that can interrupt the interruptible packets. The Class of Service (queue) determines which packets are which. In Figure 3, the classes of service are marked "a, b, ..." instead of with numbers, in order to avoid any implication about which numeric Layer 2 priority values correspond to preemptible or preempting queues. Although it shows

three queues going to the preemptible MAC/PHY, any assignment is possible.

#### 7.2.2. Transmission Selection Model

In Figure 4, we expand the "Transmission selection" function of Figure 3.

Figure 4 does NOT show the data path. It shows an example of a configuration of the IEEE 802.1Q transmission selection box shown in Figure 2 and Figure 3. Each queue *m* presents a "Class *m* Ready" signal. These signals go through various logic, filters, and state machines, until a single queue's "not empty" signal is chosen for presentation to the underlying MAC/PHY. When the MAC/PHY is ready to take another output packet, then a packet is selected from the one queue (if any) whose signal manages to pass all the way through the transmission selection function.

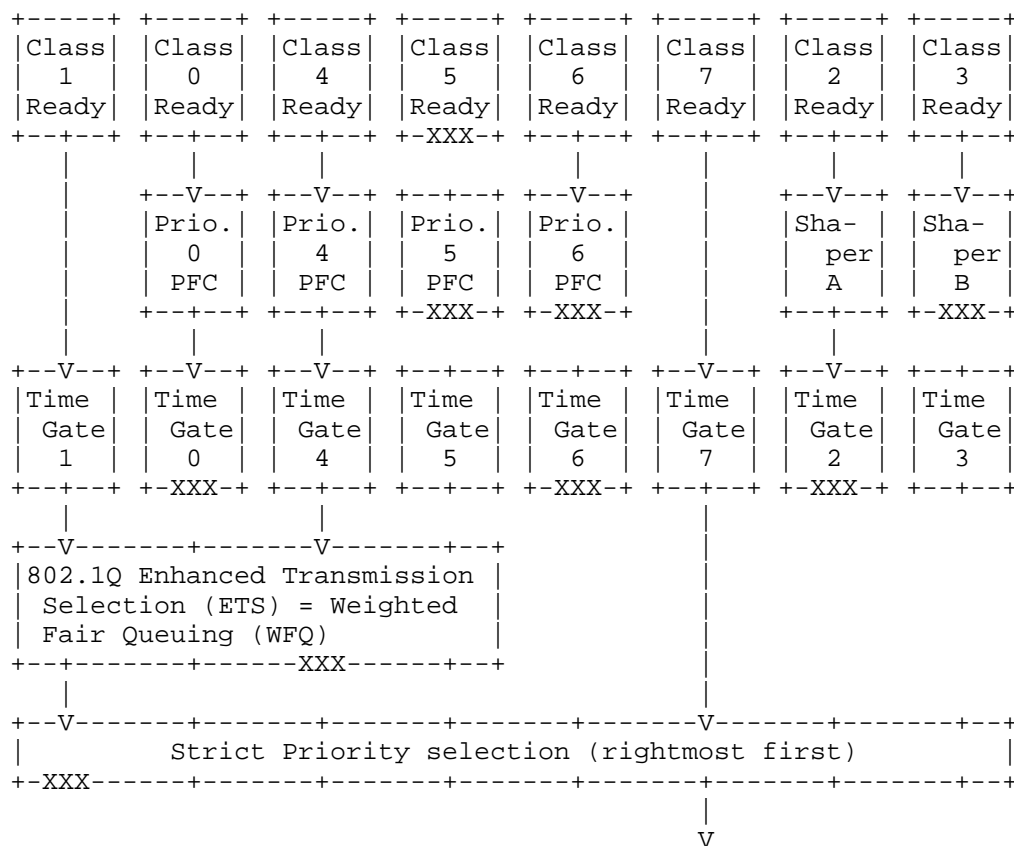


Figure 4: 802.1Q Transmission Selection

The following explanatory notes apply to Figure 4

- o The numbers in the "Class n Ready" boxes are the values of the Layer 2 priority that are assigned to that Class of Service in this example. The rightmost CoS is the most important, the leftmost the least. Classes 2 and 3 are made the most important, because they carry DetNet flows. It is all right to make them more important than the priority 7 queue, which typically carries critical network control protocols such as spanning tree or IS-IS, because the shaper ensures that the highest priority best-effort queue (7) will get reasonable access to the MAC/PHY. Note that Class 5 has no Ready signal, indicating that that queue is empty.
- o Below the Class Ready signals are shown the Priority Flow Control gates (IEEE Std 802.1Qbb-2011 Priority-based Flow Control, now [IEEE8021Q] clause 36) on Classes of Service 1, 0, 4, and 5, and

two 802.1Q shapers, A and B. Perhaps shaper A conforms to the IEEE Std 802.1Qav-2009 (now [IEEE8021Q] clause 34) credit-based shaper, and shaper B conforms to [IEEE8021Qcr] Asynchronous Traffic Shaper. Any given Class of Service can have either a PFC function or a shaper, but not both.

- o Next are the IEEE Std 802.1Qbv time gates ([IEEE8021Qbv]). Each one of the 8 Classes of Service has a time gate. The gates are controlled by a repeating schedule that restarts periodically, and can be programmed to turn any combination of gates on or off with nanosecond precision. (Although the implementation is not necessarily that accurate.)
- o Following the time gates, any number of Classes of Service can be linked to one or more instances of the Enhanced Transmission Selection function. This does weighted fair queuing among the members of its group.
- o A final selection of the one queue to be selected for output is made by strict priority. Note that the priority is determined not by the Layer 2 priority, but by the Class of Service.
- o An "XXX" in the lower margin of a box (e.g. "Prio. 5 PFC" indicates that the box has blocked the "Class n Ready" signal.
- o IEEE 802.1Qch Cyclic Queuing and Forwarding [IEEE802.1Qch] is accomplished using two or three queues (e.g. 2 and 3 in the figure), using sophisticated time-based schedules in the Class of Service Assignment function, and using the IEEE 802.1Qbv time gates [IEEE8021Qbv] to swap between the output buffers.

### 7.3. Other queuing models, e.g. IntServ

[[NWF More sections that discuss specific models]]

## 8. Parameters for the bounded latency model

### 8.1. Sender parameters

### 8.2. Relay system parameters

[[NWF This section talks about the parameters that must be passed hop-by-hop (T-SPEC? F-SPEC?) by a resource reservation protocol.]]



## 9. References

### 9.1. Normative References

- [I-D.ietf-detnet-architecture]  
Finn, N. and P. Thubert, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-00 (work in progress), September 2016.
- [I-D.ietf-detnet-dp-alt]  
Korhonen, J., Farkas, J., Mirsky, G., Thubert, P., Zhuangyan, Z., and L. Berger, "DetNet Data Plane Protocol and Solution Alternatives", draft-ietf-detnet-dp-alt-00 (work in progress), October 2016.
- [I-D.ietf-detnet-use-cases]  
Grossman, E., "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-14 (work in progress), February 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, DOI 10.17487/RFC2212, September 1997, <<https://www.rfc-editor.org/info/rfc2212>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC6658] Bryant, S., Ed., Martini, L., Swallow, G., and A. Malis, "Packet Pseudowire Encapsulation over an MPLS PSN", RFC 6658, DOI 10.17487/RFC6658, July 2012, <<https://www.rfc-editor.org/info/rfc6658>>.
- [RFC7806] Baker, F. and R. Pan, "On Queuing, Marking, and Dropping", RFC 7806, DOI 10.17487/RFC7806, April 2016, <<https://www.rfc-editor.org/info/rfc7806>>.

### 9.2. Informative References

## [IEEE802.1Qch]

IEEE, "IEEE Std 802.1Qch-2017 IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks Amendment 29: Cyclic Queuing and Forwarding (amendment to 802.1Q-2014)", 2017, <<http://www.ieee802.org/1/files/private/ch-drafts/>>.

## [IEEE802.1Qci]

IEEE, "IEEE Std 802.1Qci-2017 IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment 30: Per-Stream Filtering and Policing", 2017, <<http://www.ieee802.org/1/files/private/ci-drafts/>>.

## [IEEE802.1Q]

IEEE 802.1, "IEEE Std 802.1Q-2014: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks", 2014, <<http://standards.ieee.org/getieee802/download/802-1Q-2014.pdf>>.

## [IEEE802.1Qbu]

IEEE, "IEEE Std 802.1Qbu-2016 IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment 26: Frame Preemption", 2016, <<http://standards.ieee.org/getieee802/download/802.1Qbu-2016.zip>>.

## [IEEE802.1Qbv]

IEEE 802.1, "IEEE Std 802.1Qbv-2015: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", 2015, <<http://standards.ieee.org/getieee802/download/802.1Qbv-2015.zip>>.

## [IEEE802.1Qcr]

IEEE 802.1, "IEEE P802.1Qcr: IEEE Draft Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment: Asynchronous Traffic Shaping", 2017, <<http://www.ieee802.org/1/files/private/cr-drafts/>>.

## [IEEE802.1TSN]

IEEE 802.1, "IEEE 802.1 Time-Sensitive Networking (TSN) Task Group", <<http://www.ieee802.org/1/>>.

## [IEEE802.3]

IEEE 802.3, "IEEE Std 802.3-2015: IEEE Standard for Local and metropolitan area networks - Ethernet", 2015, <<http://standards.ieee.org/getieee802/download/802.3-2015.zip>>.

[IEEE8023br]

IEEE 802.3, "IEEE Std 802.3br-2016: IEEE Standard for Local and metropolitan area networks - Ethernet - Amendment 5: Specification and Management Parameters for Interspersing Express Traffic", 2016, <<http://standards.ieee.org/getieee802/download/802.3br-2016.pdf>>.

#### Authors' Addresses

Norman Finn  
Huawei Technologies Co. Ltd  
3101 Rio Way  
Spring Valley, California 91977  
US

Phone: +1 925 980 6430  
Email: [norman.finn@mail01.huawei.com](mailto:norman.finn@mail01.huawei.com)

Jean-Yves Le Boudec  
EPFL  
IC Station 14  
Lausanne EPFL 1015  
Switzerland

Email: [jean-yves.leboudec@epfl.ch](mailto:jean-yves.leboudec@epfl.ch)

Balazs Varga  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary

Email: [balazs.a.varga@ericsson.com](mailto:balazs.a.varga@ericsson.com)

Janos Farkas  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary

Email: [janos.farkas@ericsson.com](mailto:janos.farkas@ericsson.com)

DetNet  
Internet-Draft  
Intended status: Informational  
Expires: December 27, 2019

N. Finn  
Huawei Technologies Co. Ltd  
J-Y. Le Boudec  
E. Mohammadpour  
EPFL  
J. Zhang  
Huawei Technologies Co. Ltd  
B. Varga  
J. Farkas  
Ericsson  
June 25, 2019

DetNet Bounded Latency  
draft-finn-detnet-bounded-latency-04

Abstract

This document presents a timing model for Deterministic Networking (DetNet), so that existing and future standards can achieve the DetNet quality of service features of bounded latency and zero congestion loss. It defines requirements for resource reservation protocols or servers. It calls out queuing mechanisms, defined in other documents, that can provide the DetNet quality of service.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 27, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology and Definitions . . . . .	3
3. DetNet bounded latency model . . . . .	4
3.1. Flow creation . . . . .	4
3.1.1. Static flow latency calculation . . . . .	4
3.1.2. Dynamic flow latency calculation . . . . .	5
3.2. Relay node model . . . . .	6
4. Computing End-to-end Latency Bounds . . . . .	8
4.1. Non-queuing delay bound . . . . .	8
4.2. Queuing delay bound . . . . .	8
4.2.1. Per-flow queuing mechanisms . . . . .	9
4.2.2. Per-class queuing mechanisms . . . . .	9
4.3. Ingress considerations . . . . .	10
4.4. Interspersed non-DetNet transit nodes . . . . .	11
5. Achieving zero congestion loss . . . . .	11
5.1. A General Formula . . . . .	11
6. Queuing techniques . . . . .	12
6.1. Queuing data model . . . . .	12
6.2. Preemption . . . . .	14
6.3. Time-scheduled queuing . . . . .	15
6.4. Credit-Based Shaper with Asynchronous Traffic Shaping . . . . .	16
6.4.1. Flow Admission . . . . .	19
6.5. IntServ . . . . .	20
6.6. Cyclic Queuing and Forwarding . . . . .	22
6.6.1. CQF timing sequence . . . . .	23
6.6.2. CQF latency calculation . . . . .	24
7. References . . . . .	24
7.1. Normative References . . . . .	24
7.2. Informative References . . . . .	25
Authors' Addresses . . . . .	26

## 1. Introduction

The ability for IETF Deterministic Networking (DetNet) or IEEE 802.1 Time-Sensitive Networking (TSN, [IEEE8021TSN]) to provide the DetNet services of bounded latency and zero congestion loss depends upon A)

configuring and allocating network resources for the exclusive use of DetNet/TSN flows; B) identifying, in the data plane, the resources to be utilized by any given packet, and C) the detailed behavior of those resources, especially transmission queue selection, so that latency bounds can be reliably assured. Thus, DetNet is an example of an IntServ Guaranteed Quality of Service [RFC2212]

As explained in [I-D.ietf-detnet-architecture], DetNet flows are characterized by 1) a maximum bandwidth, guaranteed either by the transmitter or by strict input metering; and 2) a requirement for a guaranteed worst-case end-to-end latency. That latency guarantee, in turn, provides the opportunity for the network to supply enough buffer space to guarantee zero congestion loss.

To be of use to the applications identified in [RFC8578], it must be possible to calculate, before the transmission of a DetNet flow commences, both the worst-case end-to-end network latency, and the amount of buffer space required at each hop to ensure against congestion loss.

This document references specific queuing mechanisms, defined in other documents, that can be used to control packet transmission at each output port and achieve the DetNet qualities of service. This document presents a timing model for sources, destinations, and the DetNet transit nodes that relay packets that is applicable to all of those referenced queuing mechanisms.

Using the model presented in this document, it should be possible for an implementor, user, or standards development organization to select a particular set of queuing mechanisms for each device in a DetNet network, and to select a resource reservation algorithm for that network, so that those elements can work together to provide the DetNet service.

This document does not specify any resource reservation protocol or server. It does not describe all of the requirements for that protocol or server. It does describe requirements for such resource reservation methods, and for queuing mechanisms that, if met, will enable them to work together.

## 2. Terminology and Definitions

This document uses the terms defined in [I-D.ietf-detnet-architecture].

### 3. DetNet bounded latency model

#### 3.1. Flow creation

This document assumes that following paradigm is used for provisioning DetNet flows:

1. Perform any configuration required by the DetNet transit nodes in the network for the classes of service to be offered, including one or more classes of DetNet service. This configuration is done beforehand, and not tied to any particular flow.
2. Characterize the new DetNet flow, particularly in terms of required bandwidth.
3. Establish the path that the DetNet flow will take through the network from the source to the destination(s). This can be a point-to-point or a point-to-multipoint path.
4. Select one of the DetNet classes of service for the DetNet flow.
5. Compute the worst-case end-to-end latency for the DetNet flow, using one of the methods, below (Section 3.1.1, Section 3.1.2). In the process, determine whether sufficient resources are available for that flow to guarantee the required latency and to provide zero congestion loss.
6. Assuming that the resources are available, commit those resources to the flow. This may or may not require adjusting the parameters that control the filtering and/or queuing mechanisms at each hop along the flow's path.

This paradigm can be implemented using peer-to-peer protocols or using a central server. In some situations, a lack of resources can require backtracking and recursing through this list.

Issues such as un-provisioning a DetNet flow in favor of another when resources are scarce are not considered, here. Also not addressed is the question of how to choose the path to be taken by a DetNet flow.

##### 3.1.1. Static flow latency calculation

The static problem:

Given a network and a set of DetNet flows, compute an end-to-end latency bound (if computable) for each flow, and compute the resources, particularly buffer space, required in each DetNet transit node to achieve zero congestion loss.

In this calculation, all of the DetNet flows are known before the calculation commences. This problem is of interest to relatively static networks, or static parts of larger networks. It gives the best possible worst-case behavior. The calculations can be extended to provide global optimizations, such as altering the path of one DetNet flow in order to make resources available to another DetNet flow with tighter constraints.

The static flow calculation is not limited only to static networks; the entire calculation for all flows can be repeated each time a new DetNet flow is created or deleted. If some already-established flow would be pushed beyond its latency requirements by the new flow, then the new flow can be refused, or some other suitable action taken.

This calculation may be more difficult to perform than that of the dynamic calculation (Section 3.1.2), because the flows passing through one port on a DetNet transit node affect each others' latency. The effects can even be circular, from Flow A to B to C and back to A. On the other hand, the static calculation can often accommodate queuing methods, such as transmission selection by strict priority, that are unsuitable for the dynamic calculation.

### 3.1.2. Dynamic flow latency calculation

The dynamic problem:

Given a network whose maximum capacity for DetNet flows is bounded by a set of static configuration parameters applied to the DetNet transit nodes, and given just one DetNet flow, compute the worst-case end-to-end latency that can be experienced by that flow, no matter what other DetNet flows (within the network's configured parameters) might be created or deleted in the future. Also, compute the resources, particularly buffer space, required in each DetNet transit node to achieve zero congestion loss.

This calculation is dynamic, in the sense that flows can be added or deleted at any time, with a minimum of computation effort, and without affecting the guarantees already given to other flows.

The choice of queuing methods is critical to the applicability of the dynamic calculation. Some queuing methods (e.g. CQF, Section 6.6) make it easy to configure bounds on the network's capacity, and to make independent calculations for each flow. Other queuing methods (e.g., transmission selection by strict priority), make this calculation impossible, because the worst case for one flow cannot be computed without complete knowledge of all other flows. Other queuing methods (e.g. the credit-based shaper defined in [IEEE8021Q] section 8.6.8.2) can be used for dynamic flow creation, but yield



poorer latency and buffer space guarantees than when that same queuing method is used for static flow creation (Section 3.1.1).

### 3.2. Relay node model

A model for the operation of a DetNet transit node is required, in order to define the latency and buffer calculations. In Figure 1 we see a breakdown of the per-hop latency experienced by a packet passing through a DetNet transit node, in terms that are suitable for computing both hop-by-hop latency and per-hop buffer requirements.

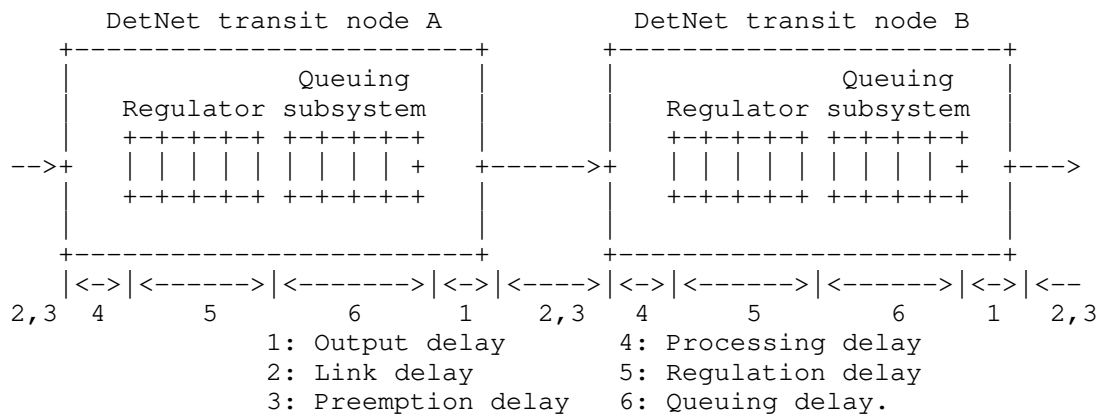


Figure 1: Timing model for DetNet or TSN

In Figure 1, we see two DetNet transit nodes (typically, bridges or routers), with a wired link between them. In this model, the only queues we deal with explicitly are attached to the output port; other queues are modeled as variations in the other delay times. (E.g., an input queue could be modeled as either a variation in the link delay [2] or the processing delay [4].) There are six delays that a packet can experience from hop to hop.

#### 1. Output delay

The time taken from the selection of a packet for output from a queue to the transmission of the first bit of the packet on the physical link. If the queue is directly attached to the physical port, output delay can be a constant. But, in many implementations, the queuing mechanism in a forwarding ASIC is separated from a multi-port MAC/PHY, in a second ASIC, by a multiplexed connection. This causes variations in the output delay that are hard for the forwarding node to predict or control.

#### 2. Link delay

The time taken from the transmission of the first bit of the packet to the reception of the last bit, assuming that the transmission is not suspended by a preemption event. This delay has two components, the first-bit-out to first-bit-in delay and the first-bit-in to last-bit-in delay that varies with packet size. The former is typically measured by the Precision Time Protocol and is constant (see [I-D.ietf-detnet-architecture]). However, a virtual "link" could exhibit a variable link delay.

3. Preemption delay

If the packet is interrupted in order to transmit another packet or packets, (e.g. [IEEE8023] clause 99 frame preemption) an arbitrary delay can result.

4. Processing delay

This delay covers the time from the reception of the last bit of the packet to the time the packet is enqueued in the regulator (Queuing subsystem, if there is no regulation). This delay can be variable, and depends on the details of the operation of the forwarding node.

5. Regulator delay

This is the time spent from the insertion of the last bit of a packet into a regulation queue until the time the packet is declared eligible according to its regulation constraints. We assume that this time can be calculated based on the details of regulation policy. If there is no regulation, this time is zero.

6. Queuing subsystem delay

This is the time spent for a packet from being declared eligible until being selected for output on the next link. We assume that this time is calculable based on the details of the queuing mechanism. If there is no regulation, this time is from the insertion of the packet into a queue until it is selected for output on the next link.

Not shown in Figure 1 are the other output queues that we presume are also attached to that same output port as the queue shown, and against which this shown queue competes for transmission opportunities.

The initial and final measurement point in this analysis (that is, the definition of a "hop") is the point at which a packet is selected for output. In general, any queue selection method that is suitable for use in a DetNet network includes a detailed specification as to exactly when packets are selected for transmission. Any variations in any of the delay times 1-4 result in a need for additional buffers in the queue. If all delays 1-4 are constant, then any variation in

the time at which packets are inserted into a queue depends entirely on the timing of packet selection in the previous node. If the delays 1-4 are not constant, then additional buffers are required in the queue to absorb these variations. Thus:

- o Variations in output delay (1) require buffers to absorb that variation in the next hop, so the output delay variations of the previous hop (on each input port) must be known in order to calculate the buffer space required on this hop.
- o Variations in processing delay (4) require additional output buffers in the queues of that same DetNet transit node. Depending on the details of the queueing subsystem delay (6) calculations, these variations need not be visible outside the DetNet transit node.

#### 4. Computing End-to-end Latency Bounds

##### 4.1. Non-queueing delay bound

End-to-end latency bounds can be computed using the delay model in Section 3.2. Here it is important to be aware that for several queueing mechanisms, the worst-case end-to-end delay is less than the sum of the per-hop worst-case delays. An end-to-end latency bound for one DetNet flow can be computed as

$$\text{end\_to\_end\_latency\_bound} = \text{non\_queueing\_latency} + \text{queueing\_latency}$$

The two terms in the above formula are computed as follows. First, at the h-th hop along the path of this DetNet flow, obtain an upper bound per-hop\_non\_queueing\_latency[h] on the sum of delays 1,2,3,4 of Figure 1. These upper-bounds are expected to depend on the specific technology of the DetNet transit node at the h-th hop but not on the T-SPEC of this DetNet flow. Then set non\_queueing\_latency = the sum of per-hop\_non\_queueing\_latency[h] over all hops h.

##### 4.2. Queueing delay bound

Second, compute queueing\_latency as an upper bound to the sum of the queueing delays along the path. The value of queueing\_latency depends on the T-SPEC of this flow and possibly of other flows in the network, as well as the specifics of the queueing mechanisms deployed along the path of this flow.

For several queueing mechanisms, queueing\_latency is less than the sum of upper bounds on the queueing delays (5,6) at every hop. This occurs with (1) per-flow queueing, and (2) per-class queueing with

regulators, as explained in Section 4.2.1, Section 4.2.2, and Section 6.

For other queuing mechanisms the only available value of `queuing_latency` is the sum of the per-hop queuing delay bounds. In such cases, the computation of per-hop queuing delay bounds must account for the fact that the T-SPEC of a DetNet flow is no longer satisfied at the ingress of a hop, since burstiness increases as one flow traverses one DetNet transit node.

#### 4.2.1. Per-flow queuing mechanisms

With such mechanisms, each flow uses a separate queue inside every node. The service for each queue is abstracted with a guaranteed rate and a delay. For every flow the per-node delay bound as well as end-to-end delay bound can be computed from the traffic specification of this flow at its source and from the values of rates and latencies at all nodes along its path. Details of calculation for IntServ are described in Section 6.5.

#### 4.2.2. Per-class queuing mechanisms

With such mechanisms, the flows that have the same class share the same queue. A practical example is the credit-based shaper defined in section 8.6.8.2 of [IEEE8021Q]. One key issue in this context is how to deal with the burstiness cascade: individual flows that share a resource dedicated to a class may see their burstiness increase, which may in turn cause increased burstiness to other flows downstream of this resource. Computing latency upper bounds for such cases is difficult, and in some conditions impossible [charny2000delay][bennett2002delay]. Also, when bounds are obtained, they depend on the complete configuration, and must be recomputed when one flow is added. (The dynamic calculation, Section 3.1.2.)

A solution to deal with this issue is to reshape the flows at every hop. This can be done with per-flow regulators (e.g. leaky bucket shapers), but this requires per-flow queuing and defeats the purpose of per-class queuing. An alternative is the interleaved regulator, which reshapes individual flows without per-flow queuing ([Specht2016UBS], [IEEE8021Qcr]). With an interleaved regulator, the packet at the head of the queue is regulated based on its (flow) regulation constraints; it is released at the earliest time at which this is possible without violating the constraint. One key feature of per-flow or interleaved regulator is that, it does not increase worst-case latency bounds [le\_boudec\_theory\_2018]. Specifically, when an interleaved regulator is appended to a FIFO subsystem, it does not increase the worst-case delay of the latter.

Figure 2 shows an example of a network with 5 nodes, per-class queuing mechanism and interleaved regulators as in Figure 1. An end-to-end delay bound for flow  $f$ , traversing nodes 1 to 5, is calculated as follows:

$$\text{end\_to\_end\_latency\_bound\_of\_flow\_f} = C_{12} + C_{23} + C_{34} + S_4$$

In the above formula,  $C_{ij}$  is a bound on the aggregate response time of queuing subsystem in node  $i$  and interleaved regulator of node  $j$ , and  $S_4$  is a bound on the response time of the queuing subsystem in node 4 for flow  $f$ . In fact, using the delay definitions in Section 3.2,  $C_{ij}$  is a bound on sum of the delays 1,2,3,6 of node  $i$  and 4,5 of node  $j$ . Similarly,  $S_4$  is a bound on sum of the delays 1,2,3,6 of node 4. A practical example of queuing model and delay calculation is presented Section 6.4.

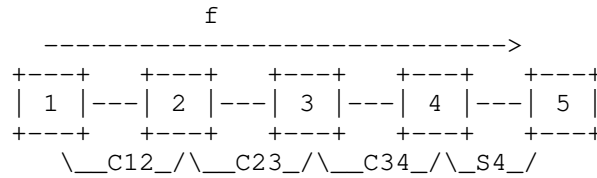


Figure 2: End-to-end latency computation example

REMARK: The end-to-end delay bound calculation provided here gives a much better upper bound in comparison with end-to-end delay bound computation by adding the delay bounds of each node in the path of a flow [TSNwithATS].

#### 4.3. Ingress considerations

A sender can be a DetNet node which uses exactly the same queuing methods as its adjacent DetNet transit node, so that the latency and buffer calculations at the first hop are indistinguishable from those at a later hop within the DetNet domain. On the other hand, the sender may be DetNet unaware, in which case some conditioning of the flow may be necessary at the ingress DetNet transit node.

This ingress conditioning typically consists of a FIFO with an output regulator that is compatible with the queuing employed by the DetNet transit node on its output port(s). For some queuing methods, simply requires added extra buffer space in the queuing subsystem. Ingress conditioning requirements for different queuing methods are mentioned in the sections, below, describing those queuing methods.

#### 4.4. Interspersed non-DetNet transit nodes

It is sometimes desirable to build a network that has both DetNet aware transit nodes and DetNet non-aware transit nodes, and for a DetNet flow to traverse an island of non-DetNet transit nodes, while still allowing the network to offer latency and congestion loss guarantees. This is possible under certain conditions.

In general, when passing through a non-DetNet island, the island causes delay variation in excess of what would be caused by DetNet nodes. That is, the DetNet flow is "lumpier" after traversing the non-DetNet island. DetNet guarantees for latency and buffer requirements can still be calculated and met if and only if the following are true:

1. The latency variation across the non-DetNet island must be bounded and calculable.
2. An ingress conditioning function (Section 4.3) may be required at the re-entry to the DetNet-aware domain. This will, at least, require some extra buffering to accommodate the additional delay variation, and thus further increases the worst-case latency.

The ingress conditioning is exactly the same problem as that of a sender at the edge of the DetNet domain. The requirement for bounds on the latency variation across the non-DetNet island is typically the most difficult to achieve. Without such a bound, it is obvious that DetNet cannot deliver its guarantees, so a non-DetNet island that cannot offer bounded latency variation cannot be used to carry a DetNet flow.

#### 5. Achieving zero congestion loss

When the input rate to an output queue exceeds the output rate for a sufficient length of time, the queue must overflow. This is congestion loss, and this is what deterministic networking seeks to avoid.

##### 5.1. A General Formula

To avoid congestion losses, an upper bound on the backlog present in the regulator and queuing subsystem of Figure 1 must be computed during resource reservation. This bound depends on the set of flows that use these queues, the details of the specific queuing mechanism and an upper bound on the processing delay (4). The queue must contain the packet in transmission plus all other packets that are waiting to be selected for output.

A conservative backlog bound, that applies to all systems, can be derived as follows.

The backlog bound is counted in data units (bytes, or words of multiple bytes) that are relevant for buffer allocation. For every class we need one buffer space for the packet in transmission, plus space for the packets that are waiting to be selected for output. Excluding transmission and preemption times, the packets are waiting in the queue since reception of the last bit, for a duration equal to the processing delay (4) plus the queuing delays (5,6).

Let

- o `nb_classes` be the number of classes of traffic that may use this output port
- o `total_in_rate` be the sum of the line rates of all input ports that send traffic of any class to this output port. The value of `total_in_rate` is in data units (e.g. bytes) per second.
- o `nb_input_ports` be the number input ports that send traffic of any class to this output port
- o `max_packet_length` be the maximum packet size for packets of any class that may be sent to this output port. This is counted in data units.
- o `max_delay45` be an upper bound, in seconds, on the sum of the processing delay (4) and the queuing delays (5,6) for a packet of any class at this output port.

Then a bound on the backlog of traffic of all classes in the queue at this output port is

$$\text{backlog\_bound} = ( \text{nb\_classes} + \text{nb\_input\_ports} ) * \text{max\_packet\_length} + \text{total\_in\_rate} * \text{max\_delay45}$$

## 6. Queuing techniques

### 6.1. Queuing data model

Sophisticated queuing mechanisms are available in Layer 3 (L3, see, e.g., [RFC7806] for an overview). In general, we assume that "Layer 3" queues, shapers, meters, etc., are precisely the "regulators" shown in Figure 1. The "queuing subsystems" in this figure are not the province solely of bridges; they are an essential part of any DetNet transit node. As illustrated by numerous implementation examples, some of the "Layer 3" mechanisms described in documents

such as [RFC7806] are often integrated, in an implementation, with the "Layer 2" mechanisms also implemented in the same node. An integrated model is needed in order to successfully predict the interactions among the different queuing mechanisms needed in a network carrying both DetNet flows and non-DetNet flows.

Figure 3 shows the general model for the flow of packets through the queues of a DetNet transit node. Packets are assigned to a class of service. The classes of service are mapped to some number of regulator queues. Only DetNet/TSN packets pass through regulators. Queues compete for the selection of packets to be passed to queues in the queuing subsystem. Packets again are selected for output from the queuing subsystem.

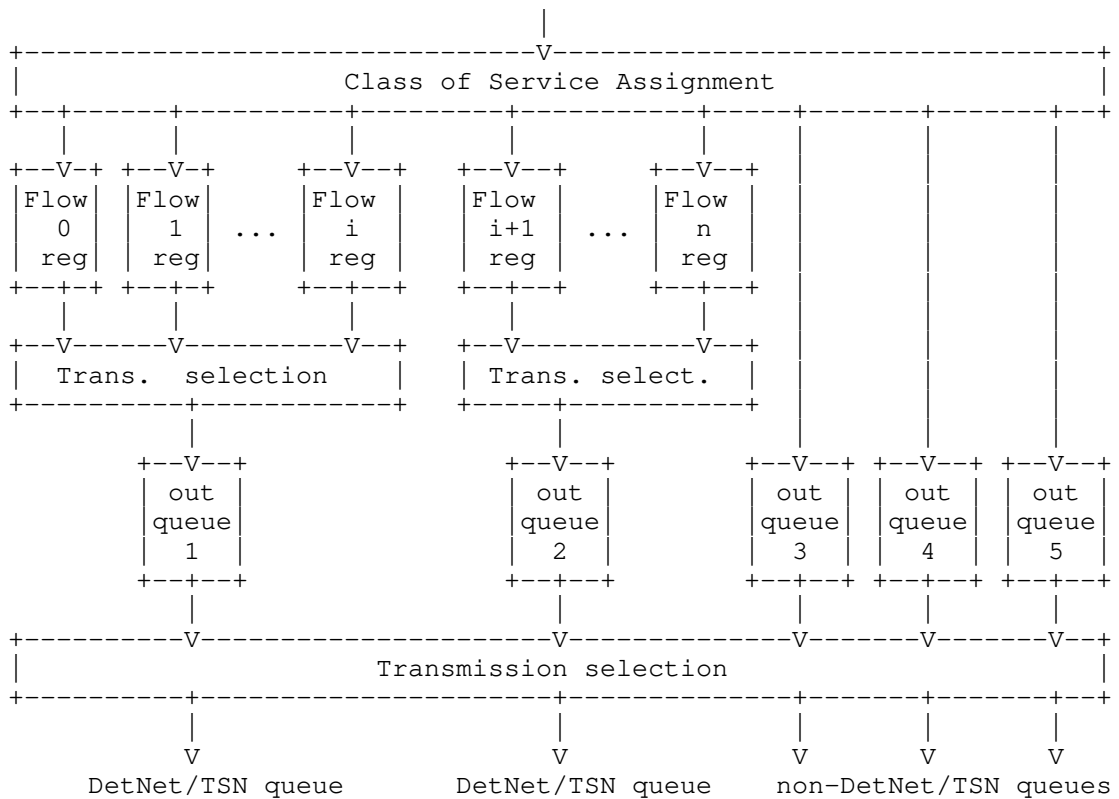


Figure 3: IEEE 802.1Q Queuing Model: Data flow

Some relevant mechanisms are hidden in this figure, and are performed in the queue boxes:

- o Discarding packets because a queue is full.



- o Discarding packets marked "yellow" by a metering function, in preference to discarding "green" packets.

Ideally, neither of these actions are performed on DetNet packets. Full queues for DetNet packets should occur only when a flow is misbehaving, and the DetNet QoS does not include "yellow" service for packets in excess of committed rate.

The Class of Service Assignment function can be quite complex, even in a bridge [IEEE8021Q], since the introduction of per-stream filtering and policing ([IEEE8021Q] clause 8.6.5.1). In addition to the Layer 2 priority expressed in the 802.1Q VLAN tag, a DetNet transit node can utilize any of the following information to assign a packet to a particular class of service (queue):

- o Input port.
- o Selector based on a rotating schedule that starts at regular, time-synchronized intervals and has nanosecond precision.
- o MAC addresses, VLAN ID, IP addresses, Layer 4 port numbers, DSCP. ([I-D.ietf-detnet-ip], [I-D.ietf-detnet-mpls]) (Work items are expected to add MPC and other indicators.)
- o The Class of Service Assignment function can contain metering and policing functions.
- o MPLS and/or pseudowire ([RFC6658]) labels.

The "Transmission selection" function decides which queue is to transfer its oldest packet to the output port when a transmission opportunity arises.

## 6.2. Preemption

In [IEEE8021Q] and [IEEE8023], the transmission of a frame can be interrupted by one or more "express" frames, and then the interrupted frame can continue transmission. This frame preemption is modeled as consisting of two MAC/PHY stacks, one for packets that can be interrupted, and one for packets that can interrupt the interruptible packets. The Class of Service (queue) determines which packets are which. Only one layer of preemption is supported -- a transmitter cannot have more than one interrupted frame in progress. DetNet flows typically pass through the interrupting MAC. Best-effort queues pass through the interruptible MAC, and can thus be preempted.

### 6.3. Time-scheduled queuing

In [IEEE8021Q], the notion of time-scheduling queue gates is described in section 8.6.8.4. Below every output queue (the lower row of queues in Figure 3) is a gate that permits or denies the queue to present data for transmission selection. The gates are controlled by a rotating schedule that can be locked to a clock that is synchronized with other DetNet transit nodes. The DetNet class of service can be supplied by queuing mechanisms based on time, rather than the regulator model in Figure 3. Generally speaking, this time-aware scheduling can be used as a layer 2 time division multiplexing (TDM) technique.

Consider the static configuration of a deterministic network. To provide end-to-end latency guaranteed service, network nodes can support time-based behavior, which is determined by gate control list (GCL). GCL defines the gate operation, in open or closed state, with associated timing for each traffic class queue. A time slice with gate state "open" is called transmission window. The time-based traffic scheduling must be coordinated among the DetNet transit nodes along the path from sender to receiver, to control the transmission of time-sensitive traffic.

Ideally all network devices are time synchronized and static GCL configurations on all devices along the routed path are coordinated to ensure that length of transmission window fits the assigned frames, and no two time windows for DetNet traffic on the same port overlap. (DetNet flows' windows can overlap with best-effort windows, so that unused DetNet bandwidth is available to best-effort traffic.) The processing delay, link delay and output delay in transmitting are considered in GCL computation. Transmission window for a certain flow may require that a time offset on consecutive hops be selected to reduce queueing delay as much as possible. In this case, TSN/DetNet frames transmit at the assigned transmission window at every node through the routed path, with zero congestion loss and bounded end-to-end latency. Then, the worst-case end-to-end latency of the flow can be derived from GCL configuration. For a TSN or DetNet frame, denote the transmission window on last hop closes at `gate_close_time_last_hop`. Assuming talker supports scheduled traffic behavior, it starts the transmission at `gate_open_time_on_talker`. Then worst case end-to-end delay of this flow is bounded by `gate_close_time_last_hop - gate_open_time_on_talker + link_delay_last_hop`.

It should be noted that scheduled traffic service relies on a synchronized network and coordinated GCL configuration. Synthesis of GCL on multiple nodes in network is a scheduling problem considering all TSN/DetNet flows traversing the network, which is a non-

deterministic polynomial-time hard (NP-hard) problem. Also, at this writing, scheduled traffic service supports no more than eight traffic classes, typically using up to seven priority classes and at least one best effort class.

#### 6.4. Credit-Based Shaper with Asynchronous Traffic Shaping

Consider a network with a set of nodes (DetNet transit nodes and hosts) along with a set of flows between hosts. Hosts are sources or destinations of flows. There are four types of flows, namely, control-data traffic (CDT), class A, class B, and best effort (BE) in decreasing order of priority. Flows of classes A and B are together referred to AVB flows. It is assumed a subset of TSN functions as described next.

It is also assumed that contention occurs only at the output port of a TSN node. Each node output port performs per-class scheduling with eight classes: one for CDT, one for class A traffic, one for class B traffic, and five for BE traffic denoted as BE0-BE4 (according to TSN standard). In addition, each node output port also performs per-flow regulation for AVB flows using an interleaved regulator (IR), called Asynchronous Traffic Shaper (ATS) in TSN. Thus, at each output port of a node, there is one interleaved regulator per-input port and per-class. The detailed picture of scheduling and regulation architecture at a node output port is given by Figure 4. The packets received at a node input port for a given class are enqueued in the respective interleaved regulator at the output port. Then, the packets from all the flows, including CDT and BE flows, are enqueued in a class based FIFO system (CBFS) [TSNwithATS].

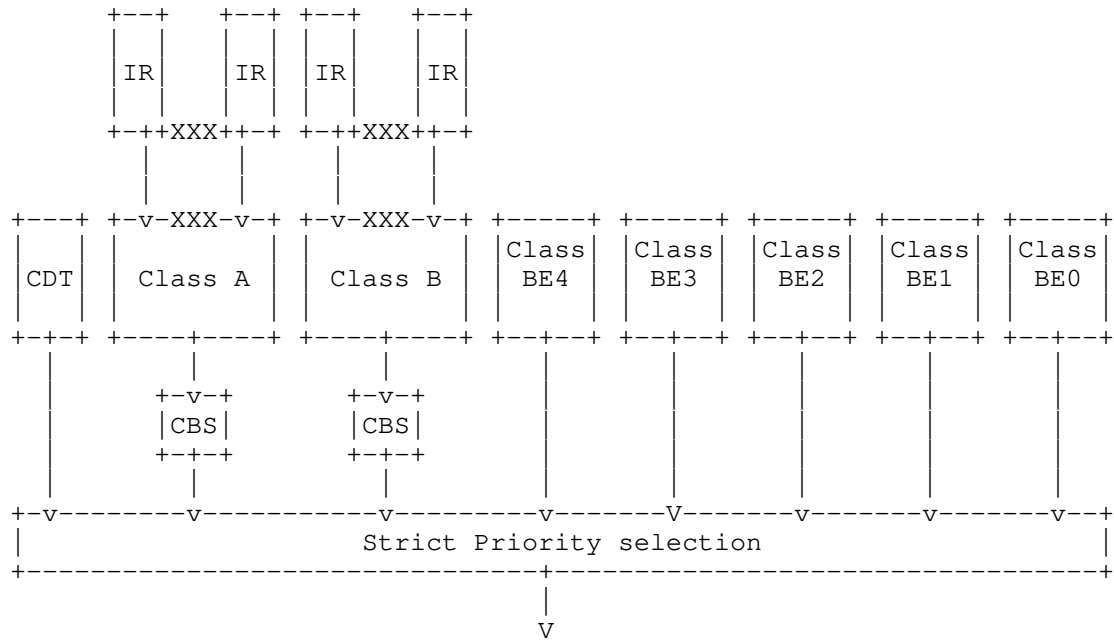


Figure 4: Architecture of a TSN node output port with interleaved regulators (IRs)

The CBFS includes two Credit-Based Shaper (CBS) subsystems, one for each class A and B. The CBS serves a packet from a class according to the available credit for that class. The credit for each class A or B increases based on the idle slope, and decreases based on the send slope, both of which are parameters of the CBS. The CDT and BE0-BE4 flows in the CBFS are served by separate FIFO subsystems. Then, packets from all flows are served by a transmission selection subsystem that serves packets from each class based on its priority. All subsystems are non-preemptive. Guarantees for AVB traffic can be provided only if CDT traffic is bounded; it is assumed that the CDT traffic has leaky bucket arrival curve with two parameters  $r_h$  as rate and  $b_h$  as bucket size, i.e., the amount of bits entering a node within a time interval  $t$  is bounded by  $r_h t + b_h$ .

Additionally, it is assumed that the AVB flows are also regulated at their source according to leaky bucket arrival curve. At the source hosts, the traffic satisfies its regulation constraint, i.e. the delay due to interleaved regulator at hosts is ignored.

At each DetNet transit node implementing an interleaved regulator, packets of multiple flows are processed in one FIFO queue; the packet at the head of the queue is regulated based on its leaky bucket

parameters; it is released at the earliest time at which this is possible without violating the constraint. The regulation parameters for a flow (leaky bucket rate and bucket size) are the same at its source and at all DetNet transit nodes along its path. A delay bound of CBFS for an AVB flow  $f$  of class A or B can be computed if the following condition holds:

sum of leaky bucket rates of all flows of this class at this node  $\leq R$ , where  $R$  is given below for every class.

If the condition holds, the delay bound is:

$$d_f = T + (b_t - L_{\min_f})/R - L_{\min_f}/c$$

where  $L_{\min_f}$  is the minimum packet length of flow  $f$ ;  $c$  is the output link transmission rate;  $b_t$  is the sum of the  $b$  term (bucket size) for all the flows having the same class as flow  $f$  at this node. Parameters  $R$  and  $T$  are calculated as follows for class A and class B, separately:

If  $f$  is of class A:

$$R = I_A (c - r_h) / c$$

$$T = L_{nA} + b_h + r_h L_n / c / (c - r_h)$$

where  $L_{nA}$  is the maximum packet length of class B and BE packets;  $L_n$  is the maximum packet length of classes A, B, and BE.

If  $f$  is of class B:

$$R = I_B (c - r_h) / c$$

$$T = (L_{BE} + L_A + L_{nA} I_A / (c_h - I_A) + b_h + r_h L_n / c) / (c - r_h)$$

where  $L_A$  is the maximum packet length of class A;  $L_{BE}$  is the maximum packet length of class BE.

Then, an end-to-end delay bound is calculated by the formula Section 4.2.2, where for  $C_{ij}$ :

$$C_{ij} = \max(d_{f'})$$

where  $f'$  is any flow that shares the same CBFS class with flow  $f$  at node  $i$  and the same interleaved regulator as flow  $f$  at node  $j$ .

More information of delay analysis in such a DetNet transit node is described in [TSNwithATS].

#### 6.4.1. Flow Admission

The delay calculation requires some information about each node. For each node, it is required to know the idle slope of CBS for each class A and B ( $I_A$  and  $I_B$ ), as well as the transmission rate of the output link ( $c$ ). Besides, it is necessary to have the information on each class, i.e. maximum packet length of classes A, B, and BE. Moreover, the leaky bucket parameters of CDT ( $r_h, b_h$ ) should be known. To admit a flow/flows, their delay requirements should be guaranteed not to be violated. As described in Section 3.1, the two problems static and dynamic are addressed separately. In either of the problems, the rate and delay should be guaranteed. Thus,

The static admission control:

The leaky bucket parameters of all flows are known, therefore, for each flow a delay bound can be calculated. The computed delay bound for every flow should not be more than its delay requirement. Moreover, the sum of the rate of each flow ( $r_f$ ) should not be more than the rate allocated to each class ( $R$ ). If these two conditions hold, the configuration is declared admissible.

The dynamic admission control:

For dynamic admission control, we allocate to every node and class A or B, static value for rate ( $R$ ) and maximum burstiness ( $b_t$ ). In addition, for every node and every class A and B, two counters are maintained:

$R_{acc}$  is equal to the sum of the leaky-bucket rates of all flows of this class already admitted at this node; At all times, we must have:

$$R_{acc} \leq R, \text{ (Eq. 1)}$$

$b_{acc}$  is equal to the sum of the bucket sizes of all flows of this class already admitted at this node; At all times, we must have:

$$b_{acc} \leq b_t. \text{ (Eq. 2)}$$

A new flow is admitted at this node, if Eqs. (1) and (2) continue to be satisfied after adding its leaky bucket rate

and bucket size to  $R_{acc}$  and  $b_{acc}$ . A flow is admitted in the network, if it is admitted at all nodes along its path. When this happens, all variables  $R_{acc}$  and  $b_{acc}$  along its path must be incremented to reflect the addition of the flow. Similarly, when a flow leaves the network, all variables  $R_{acc}$  and  $b_{acc}$  along its path must be decremented to reflect the removal of the flow.

The choice of the static values of  $R$  and  $b_t$  at all nodes and classes must be done in a prior configuration phase;  $R$  controls the bandwidth allocated to this class at this node,  $b_t$  affects the delay bound and the buffer requirement.  $R$  must satisfy the constraints given in Annex L.1 of [IEEE8021Q].

#### 6.5. IntServ

Integrated service (IntServ) is an architecture that specifies the elements to guarantee quality of service (QoS) on networks. To satisfied guaranteed service, a flow must conform to a traffic specification (T-spec), and reservation is made along a path, only if routers are able to guarantee the required bandwidth and buffer.

Consider the traffic model which conforms to token bucket regulator  $(r, b)$ , with

- o Token bucket depth  $(b)$ .
- o Token bucket rate  $(r)$ .

The traffic specification can be described as an arrival curve:

$$\alpha(t) = b + rt$$

This token bucket regulator requires that, during any time window  $t$ , the number of bit for the flow is limited by  $\alpha(t) = b + rt$ .

If resource reservation on a path is applied, IntServ model of a router can be described as a rate-latency service curve  $\beta(t)$ .

$$\beta(t) = \max(0, R(t-T))$$

It describes that bits might have to wait up to  $T$  before being served with a rate greater or equal to  $R$ .

It should be noted that, the guaranteed service rate  $R$  is a share of link's bandwidth. The choice of  $R$  is related to the specification of flows which will transmit on this node. For example, in strict priority policy, considering a flow with priority  $j$ , its share of

bandwidth may be  $R=c-\sum(r_i)$ ,  $i < j$ , where  $c$  is the link bandwidth,  $r_i$  is the token bucket rate for the flows with priority higher than  $j$ . The choice of  $T$  is also related to the specification of all the flows traversing this node. For example, in a generalized processor sharing (GPS) node,  $T = L / R + L_{\max}/c$ , where  $L$  is the maximum packet size for the flow,  $L_{\max}$  is the maximum packet size in the node across all flows. Other choice of  $R$  and  $T$  are also supported, according to the specific scheduling of the node and flows traversing this node.

As mentioned previously in this section, delay bound and backlog bound can be easily obtained by comparing arrival curve and service curve. Backlog bound, or buffer bound, is the maximum vertical derivation between curves  $\alpha(t)$  and  $\beta(t)$ , which is  $v=b+rT$ . Delay bound is the maximum horizontal derivation between curves  $\alpha(t)$  and  $\beta(t)$ , which is  $h = T+b/R$ . Graphical illustration of the IntServ model is shown in Figure 5.

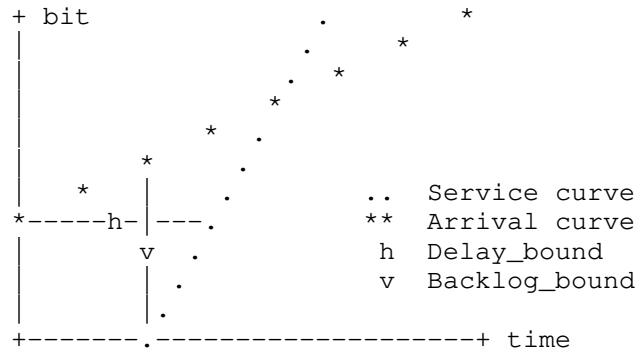


Figure 5: Computation of backlog bound and delay bound. Note that arrival and service curves are not necessary to be linear.

The output bound, or the next-hop arrival curve, is  $\alpha_{\text{out}}(t) = b + rT + rt$ , where burstiness of the flow is increased by  $rT$ , compared with the arrival curve.

We can calculate the end-to-end delay bound for a path including  $N$  nodes, among which the  $i$ -th node offers service curve  $\beta_i(t)$ ,

$$\beta_i(t) = \max(0, R_i(t-T_i)), \quad i=1, \dots, N$$

By concatenating these IntServ nodes, an end-to-end service curve can be computed as

$$\beta_{\text{e2e}}(t) = \max(0, R_{\text{e2e}}(t-T_{\text{e2e}}))$$



where

$$R_{e2e} = \min(R_1, \dots, R_N)$$

$$T_{e2e} = T_1 + \dots + T_N$$

Similarly, delay bound, backlog bound and output bound can be computed by using the original arrival curve  $\alpha(t)$  and concatenated service curve  $\beta_{e2e}(t)$ .

## 6.6. Cyclic Queuing and Forwarding

Annex T of [IEEE8021Q] describes Cyclic Queuing and Forwarding (CQF), which provides bounded latency and zero congestion loss using the time-scheduled gates of [IEEE8021Q] section 8.6.8.4. For a given DetNet class of service, a set of two or three buffers is provided at the output queue layer of Figure 3. A cycle time  $T_c$  is configured for each class  $c$ , and all of the buffer sets in a class swap buffers simultaneously throughout the DetNet domain at that cycle rate, all in phase.

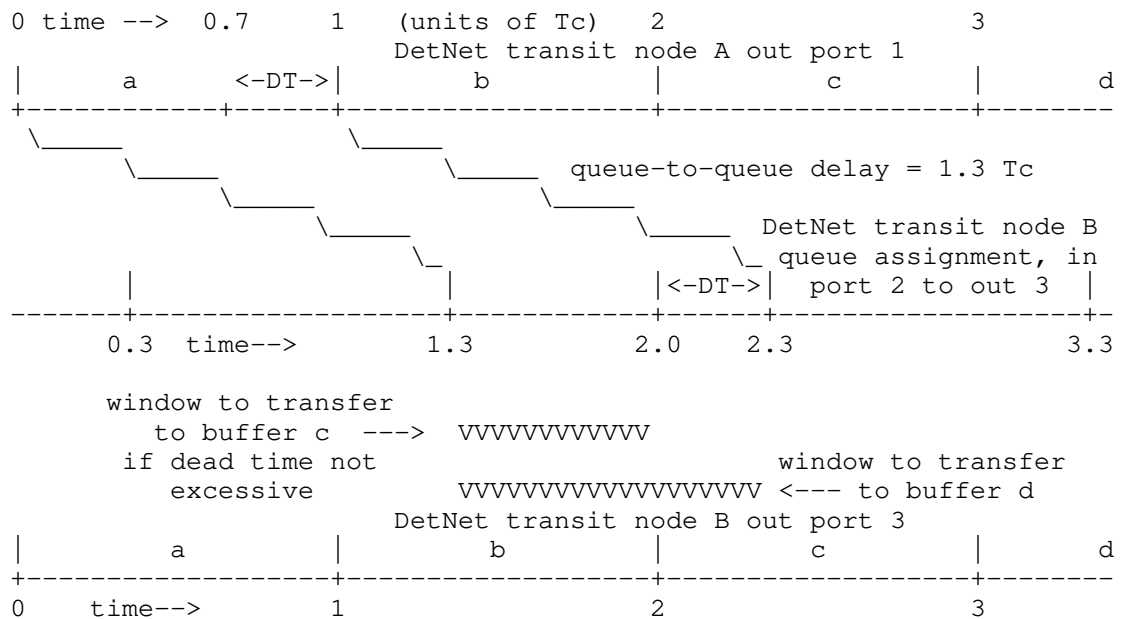


Figure 6: CQF timing diagram

Figure 6 shows two DetNet transit nodes A and B, including three timelines for:

1. The output queues on port 1 in node A.
2. The input gate function ([IEEE8021Q], 8.6.5.1) that assigns packets received on port 1 of transit node B to output queues on port 2 of transit node B.
3. The output queues on port 2 of node B.

In this figure, the output ports on the two nodes are synchronized, and a new buffer starts transmitting at each tick, shown as 0, 1, 2, ... The output times shown for timelines 1 and 3 are the times at which packets are selected for output, which is the start point of the output time (1) of Figure 1. The queue assignments times on timeline 3 take place at the beginning of the queuing delay (6) of Figure 1. Time-based CQF, as described here, does not require any regulator queues. In the shown in the figure, the total time for delays 1 through 6 of Figure 1 is  $1.3T_c$ . Of course, any value is possible.

#### 6.6.1. CQF timing sequence

In general, as shown in Figure 6, the windows for buffer assignment do not align perfectly with the windows for buffer transmission. The input gates (the center timeline in Figure 6) must switch from using one buffer to using another buffer in sync with the (delayed) received data, at times offset by the dead time from the output buffer switching (the bottom timeline in Figure 6).

If the dead time  $DT$  in Figure 6 is not excessive, then it is feasible to subtract the dead time from the cycle time  $T_c$ , and use the remainder as the input window. In the example in Figure 6, packets from node A buffer a can be transferred from the input port to node B's buffer "c" during the window shown by the upper row "VVVV...". Input must cease by time = 2.0, because that is when transit node B starts transmitting the contents of buffer c. In this case, only two output buffers are in use, one filling and one outputting.

If the dead time is too large (e.g., if the delays placed the middle timeline's switching points at  $n+0.9$ , instead of  $n+0.3$ ), three buffers are used by node B. This case is shown by the lower row "VVVV..." in Figure 6. In this case, node B places the data received from node A buffer a into node B buffer d between the times 1.3 and 2.3 in Figure 6. Buffer b starts outputting at time = 2.0, while buffer d is filling. Thus, three buffers are in use, one filling, one waiting, and one emptying.

### 6.6.2. CQF latency calculation

The per-hop latency is trivially determined by the wire delay and the queuing delay. Since the wire delay is either absorbed into the queueing delay (dead time is small and two buffers are used) or padded out to a whole cycle time  $T_c$  (three buffers are used) the per-hop latency is always an integral number of cycle times  $T_c$ , with a latency variation at the output of the final hop of  $T_c$ .

Ingress conditioning (Section 4.3) may be required if the source of a DetNet flow does not, itself, employ CQF.

Note that there are no per-flow parameters in the CQF technique. Therefore, there is no requirement for per-hop configuration when a new DetNet flow is added to a network, except perhaps for ingress checks to see that the transmitter does not exceed the contracted bandwidth.

## 7. References

### 7.1. Normative References

- [I-D.ietf-detnet-architecture]  
Finn, N., Thubert, P., Varga, B., and J. Farkas,  
"Deterministic Networking Architecture", draft-ietf-detnet-architecture-08 (work in progress), September 2018.
- [I-D.ietf-detnet-ip]  
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP", draft-ietf-detnet-ip-00 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls]  
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS", draft-ietf-detnet-mpls-00 (work in progress), May 2019.
- [RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, DOI 10.17487/RFC2212, September 1997, <<https://www.rfc-editor.org/info/rfc2212>>.
- [RFC6658] Bryant, S., Ed., Martini, L., Swallow, G., and A. Malis, "Packet Pseudowire Encapsulation over an MPLS PSN", RFC 6658, DOI 10.17487/RFC6658, July 2012, <<https://www.rfc-editor.org/info/rfc6658>>.

- [RFC7806] Baker, F. and R. Pan, "On Queuing, Marking, and Dropping", RFC 7806, DOI 10.17487/RFC7806, April 2016, <<https://www.rfc-editor.org/info/rfc7806>>.
- [RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <<https://www.rfc-editor.org/info/rfc8578>>.

## 7.2. Informative References

- [bennett2002delay]  
J.C.R. Bennett, K. Benson, A. Charny, W.F. Courtney, and J.-Y. Le Boudec, "Delay Jitter Bounds and Packet Scale Rate Guarantee for Expedited Forwarding", <<https://dl.acm.org/citation.cfm?id=581870>>.
- [charny2000delay]  
A. Charny and J.-Y. Le Boudec, "Delay Bounds in a Network with Aggregate Scheduling", <[https://link.springer.com/chapter/10.1007/3-540-39939-9\\_1](https://link.springer.com/chapter/10.1007/3-540-39939-9_1)>.
- [IEEE8021Q]  
IEEE 802.1, "IEEE Std 802.1Q-2018: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks", 2018, <<http://ieeexplore.ieee.org/document/8403927>>.
- [IEEE8021Qcr]  
IEEE 802.1, "IEEE P802.1Qcr: IEEE Draft Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment: Asynchronous Traffic Shaping", 2017, <<http://www.ieee802.org/1/files/private/cr-drafts/>>.
- [IEEE8021TSN]  
IEEE 802.1, "IEEE 802.1 Time-Sensitive Networking (TSN) Task Group", <<http://www.ieee802.org/1/>>.
- [IEEE8023]  
IEEE 802.3, "IEEE Std 802.3-2018: IEEE Standard for Ethernet", 2018, <<http://ieeexplore.ieee.org/document/8457469>>.
- [le\_boudec\_theory\_2018]  
J.-Y. Le Boudec, "A Theory of Traffic Regulators for Deterministic Networks with Application to Interleaved Regulators", <<http://arxiv.org/abs/1801.08477>>.

## [NetCalBook]

Le Boudec, Jean-Yves, and Patrick Thiran, "Network calculus: a theory of deterministic queuing systems for the internet", 2001, <<https://arxiv.org/abs/1804.10608/>>.

## [Specht2016UBS]

J. Specht and S. Samii, "Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks", <<https://ieeexplore.ieee.org/abstract/document/7557870>>.

## [TSNwithATS]

E. Mohammadpour, E. Stai, M. Mohiuddin, and J.-Y. Le Boudec, "End-to-end Latency and Backlog Bounds in Time-Sensitive Networking with Credit Based Shapers and Asynchronous Traffic Shaping", <<https://arxiv.org/abs/1804.10608/>>.

## Authors' Addresses

Norman Finn  
Huawei Technologies Co. Ltd  
3101 Rio Way  
Spring Valley, California 91977  
US

Phone: +1 925 980 6430  
Email: [nfinn@nfinnconsulting.com](mailto:nfinn@nfinnconsulting.com)

Jean-Yves Le Boudec  
EPFL  
IC Station 14  
Lausanne EPFL 1015  
Switzerland

Email: [jean-yves.leboudec@epfl.ch](mailto:jean-yves.leboudec@epfl.ch)

Ehsan Mohammadpour  
EPFL  
IC Station 14  
Lausanne EPFL 1015  
Switzerland

Email: [ehsan.mohammadpour@epfl.ch](mailto:ehsan.mohammadpour@epfl.ch)

Jiayi Zhang  
Huawei Technologies Co. Ltd  
Q22, No.156 Beiqing Road  
Beijing 100095  
China

Email: zhangjiayi11@huawei.com

Balazs Varga  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary

Email: balazs.a.varga@ericsson.com

Janos Farkas  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary

Email: janos.farkas@ericsson.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 6, 2018

X. Geng  
M. Chen  
Huawei  
Z. Li  
China Mobile  
March 05, 2018

DetNet Configuration YANG Model  
draft-geng-detnet-conf-yang-01

Abstract

This document defines a YANG data Model for Deterministic Networking (DetNet), covering the device / link capabilities and resources. It can be used in network capability advertising, device configuration and status reporting.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminologies . . . . .	4
3. DetNet Configuration Attribute . . . . .	4
3.1. DetNet Topology Attribute . . . . .	4
3.1.1. Node Type . . . . .	5
3.1.2. Replication Capability . . . . .	6
3.1.3. Elimination Capability . . . . .	6
3.1.4. Queuing Management Algorithm . . . . .	6
3.1.5. Resource Reservation Base . . . . .	7
3.1.6. Bandwidth Metric . . . . .	7
3.1.7. Delay Metric . . . . .	8
3.1.8. Synchronization Accuracy . . . . .	9
3.2. DetNet Path Configuration Attribute . . . . .	9
3.2.1. Path Constrains . . . . .	9
3.2.2. Explicit Routing . . . . .	9
3.3. DetNet Flow Configuration Attribute . . . . .	9
3.3.1. Flow Identification . . . . .	9
3.3.2. Traffic Specification . . . . .	10
3.3.3. Encapsulation . . . . .	10
3.3.4. Flow Priority . . . . .	10
3.3.5. Queuing Parameters . . . . .	11
3.3.6. Replication Function . . . . .	11
3.3.7. Elimination Function . . . . .	11
3.3.8. Routing . . . . .	11
3.4. DetNet Status Attribute . . . . .	12
3.4.1. Performance Status . . . . .	12
3.4.2. Replication/Elimination Status . . . . .	13
4. DetNet Configuration YANG Model . . . . .	13
4.1. DetNet Topology YANG Model . . . . .	13
4.2. DetNet Static Configuration YANG Model . . . . .	19
5. DetNet Configuration Model Classification . . . . .	23
5.1. Fully Distributed Configuration Model . . . . .	23
5.2. Fully Centralized Configuration Model . . . . .	23
5.3. Hybrid Configuration Model . . . . .	24
6. IANA Considerations . . . . .	25
7. Security Considerations . . . . .	25
8. Acknowledgements . . . . .	25
9. References . . . . .	25



9.1. Normative References . . . . .	25
9.2. Informative References . . . . .	26
Authors' Addresses . . . . .	28

## 1. Introduction

A lot of use cases in industry and other areas require the network to provide service that can satisfy strict quality requirements, e.g., extremely low packet loss rate, bounded low latency and jitter, together with other best effort flows [I-D.ietf-detnet-use-cases]. Deterministic Networking (DetNet) is able to provide high quality deterministic service in layer 3 in an IP/MPLS network.

[I-D.ietf-detnet-architecture] defines the whole picture of DetNet; [I-D.dt-detnet-dp-sol] defines DetNet flow encapsulation and forwarding process;

As defined in the [I-D.ietf-detnet-flow-information-model] , DetNet information model can be distinguished as:

- o Flow models describe characteristics of data flows. These models describe in detail all relevant aspects of a flow that are needed to support the flow properly by the network between the source and the destination(s).
- o Service models describe characteristics of services being provided for data flows over a network. These models can be treated as a network operator independent information model.
- o Configuration models describe in detail the settings required on network nodes to serve a data flow properly. Service and flow information models are used between the user and the network operator. Configuration information models are used between the management/control plane entity of the network and the network nodes.

They are shown in the Figure 1.

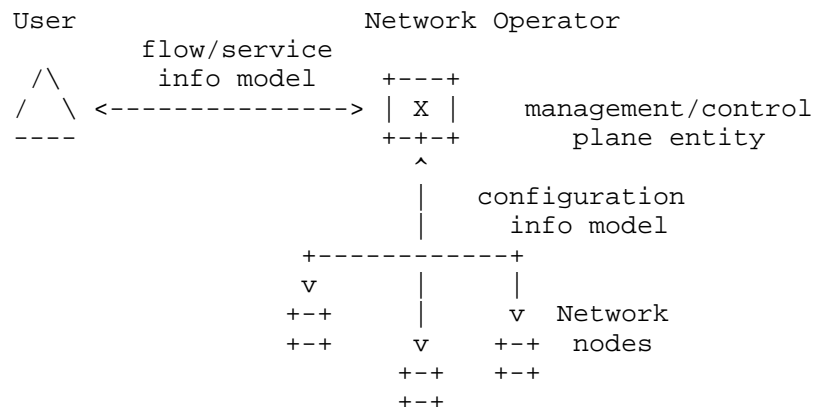


Figure 1. Three Information Models

[I-D.ietf-detnet-flow-information-model] defines the user network interface (UNI), including flow/service information model.

This document defines DetNet configuration information model and YANG data Model, covering the device / link capabilities and resources. It can be used in network capability advertising, device configuration and status reporting. The YANG model is protocol irrelevant, and serves as a base data model that other DetNet specific models can augment.

## 2. Terminologies

This documents uses the terminologies defined in [I-D.ietf-detnet-architecture].

## 3. DetNet Configuration Attribute

This section defines network attributes for DetNet, which are used for capability advertising/collection (section 3.1 DetNet Topology Attribute), flow configuration (section 3.2 DetNet Path Configuration Attribute/ section 3.3 DetNet Device configuration Attribute) and status reporting (section 3.4 DetNet Status Attribute).

### 3.1. DetNet Topology Attribute

DetNet Topology Attribute describes the network topology and capability, which is the basis of path computation and flow transmission.

### 3.1.1.1. Node Type

Figure 2 shows a basic architecture of a DetNet Network. Three types of DetNet nodes are showed in the picture, which play different roles with different functions, as defined in [I-D.ietf-detnet-architecture].

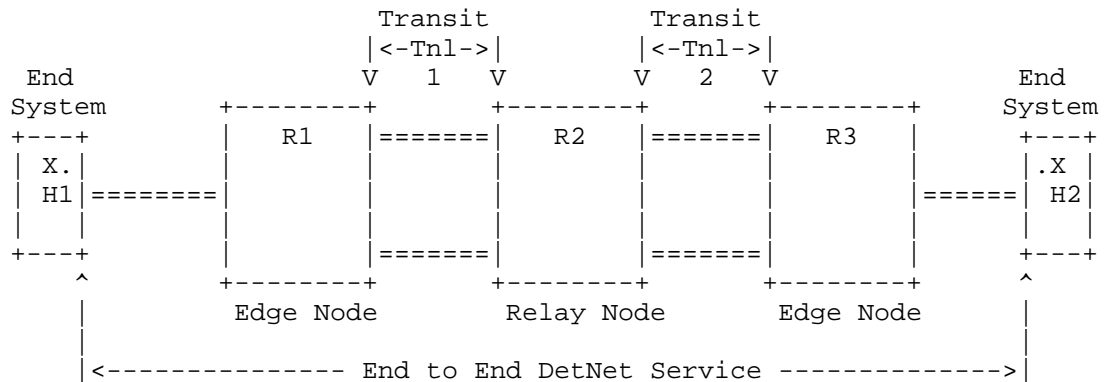


Figure 2. DetNet Architecture

#### Edge node

Edge node is the boundary of a DetNet network, including ingress and egress. The DetNet flow is started at an edge node, then the packet of a DetNet flow is forwarded to the DetNet Network after being encapsulated or recapsulated in the edge node. Once having passed through the network, DetNet flow is ended at another edge node; the packet is decapsulated or recapsulated, and forwarded to the end system or another network. Ingress and Egress may also do replication/elimination, flow aggregation/split and load balance [I-D.thubert-tsvwg-detnet-transport]. Edge node can be proxy of the network and connect to the controller through UNI [I-D.ietf-detnet-flow-information-model].

#### Relay node

Relay node is designed to do replication and elimination in the DetNet network to satisfy the reliability requirement. The packet of a DetNet flow is replicated in one relay node and forwarded to disjoint paths. These paths merge with each other in another relay node, and after the redundant packets being eliminated, only one copy of the flow is forwarded to the next hop. Relay node can identify DetNet flow and guide the packet to the next relay node or edge node, so it can also be the tunnel initial/terminal which is very important to guarantee DetNet explicit route.

### Transit node

The node between relay node/edge node is transit node, just like the p node in MPLS. Packet is transmitted through transit node hop by hop. If the DetNet service requires bounded latency, every node in the network is supposed to do congestion protection, with some queuing management algorithm to guarantee per hop latency, including the transit node.

NodeType attribute specifies which type of DetNet node one device belongs to. It indicates DetNet node capability, which can be used in path computation. These three nodes are explained in capability ascending order above, that is to say normally, the DetNet node capability: Edge node>Relay node> Transit node; the more capable node type can play a less capable node's role, for example, using a Relay node as a transit node. However, this attribute doesn't implicate specific functions of the node, which have their own corresponding attributes stated in the following text.

#### 3.1.2. Replication Capability

ReplicationCapability specifies whether a DetNet node has the capability of packet replication. A DetNet Node with replication capability can: 1) identify the packets that need to be replicated; 2) do packet replication; 3) encapsulate the replicated packets and send them to different next hop.

#### 3.1.3. Elimination Capability

EliminationCapability specifies whether a DetNet node has the capability of packet elimination. DetNet Node with elimination capability can: 1) record the packets that have been received from different port; 2) Filter the redundant packets from the same flow and eliminate the redundant packets; 3) encapsulate the first-received packets and send them to the right next hop.

#### 3.1.4. Queuing Management Algorithm

Queuing Management Algorithm is the most important method of congestion protection, including scheduling, shaping and preemption. IEEE defines some queuing management algorithms to guarantee TSN service quality, most of them can be used in DetNet, for example:

- o Credit-based shaper algorithm [IEEE802.1Q-2014]
- o Frame Preemption[IEEE802.1Qbu]
- o Scheduled Traffic [IEEE802.1Qbv]

- o Per-Stream Filtering and Policing [IEEE802.1Qci]
- o Cyclic Queuing and Forwarding [IEEE802.1Qch]

This attribute specifies which type of Queuing Management Algorithm(s) is(are) used in the output queue for DetNet (except for IEEE802.1Qci, which is normally used in input queue).

Editor's Note: Every queuing management algorithm has its parameters, which are to be defined in the next step work. However, one of the concerns of this part of work is whether it is out of the charter's scope.

#### 3.1.5. Resource Reservation Base

There is a set of parameters that influence reservation operation for the entire device. Those parameters are contained in Reservation Base attribute, including the following parameters:

- o MaxFanInPorts: maximum number of fan-in ports in the device
- o MaxPacketSize: maximum packet size that the node allows to transmit
- o MaxDetNetClasses: maximum number of traffic classes that can be reserved for DetNet

#### 3.1.6. Bandwidth Metric

[I-D.ietf-teas-yang-te-topo] defines the following parameters for bandwidth reservation:

- o Max-link-bandwidth: maximum link bandwidth
- o Max-resv-link-bandwidth: maximum reservable link bandwidth
- o Unreserved-bandwidth(N): unreserved bandwidth for priority N

Considering the features of DetNet, bandwidth reservation parameters for DetNet are defined as follows to augment the te-topology:

- o Maximum DetNet Reservable Bandwidth(N): is represented as a percentage of port transmit rate, that can be used by DetNet of traffic class N and it is also available for other DetNet traffic classes that have lower latency requirements;
- o DetNet Unreserved Bandwidth(N): is represented as a percentage of maximum DetNet Reservable bandwidth that has not been reserved;

For example, there are three classes of DetNet service A, B, and C, with A the lowest latency and C the highest. 'Maximum DetNet Reservable Bandwidth(N)' can be presented as 'MaxBw(N)'; DetNet Unreserved Bandwidth(N) can be presented as 'UnBw(N)'. MaxBw(A) can be used by A; MaxBw(B) can be used by A&B, and MaxBw(C) can be used by A&B&C. So, if MaxBw(A)=10, MaxBw(B)=25, MaxBw(C)=40, and we allocate 15 to A, 30 to B and 10 to C, then UnBw(A)=0, UnBw(B)= 0, UnBw(C)=20.

### 3.1.7. Delay Metric

Delay Metric is used to describe the delay of every hop, which includes the following parameters:

- o Link Delay
- o Maximum Packet Processing Delay
- o Minimum Packet Processing Delay
- o Maximum Output Queuing Delay
- o Minimum Output Queuing Delay

Link Delay specifies the delay along the network media for a packet transmitted from the specified Port of this station to the neighboring Port on a different station.

Operations causing Packet Processing Delay includes: Per-Stream Filtering and Policing([IEEE802.1Qci]), Flow Classification, Looking up in Forwarding Information Base, and etc. It covers the process from the packet being received by the node to the packet being sent to the output queue. It is packet length dependent.

Queuing Delay specifies the delay for a packet in the output queue. It is determined by the Queuing Management Algorithm and Port Transmission Rate.

The delay of every hop is the sum of link delay, packet processing delay and output queuing delay.

Editor's Note: The delay metric is also discussed in IEEE with other considerations, which can be found: <<http://www.ieee802.org/1/files/public/docs2017/cr-finn-timing-model-0617-v00.pdf>> and <<http://www.ieee802.org/1/files/public/docs2017/cr-specht-bridge-timing-0917-v01.pdf>>. More discussions are needed here.

### 3.1.8. Synchronization Accuracy

Most of the DetNet service requires clock synchronization. Synchronization Accuracy is necessary for queuing algorithm configuration and delay prediction. For example, Synchronization Accuracy is an important parameter when calculating the guard band for CQF[IEEE802.1Qch].

Editor's Note: The method used to achieve time synchronization is not specified in this draft.

### 3.2. DetNet Path Configuration Attribute

Path Attribute is used for path configuration in DetNet Edge Node(Ingress).

#### 3.2.1. Path Constrains

DetNet path constrains are mainly based on the application requirement, including maximum latency/number of replication trees, and traffic specification, which can be used to calculate bandwidth requirement[I-D.ietf-detnet-flow-information-model]. There may be other path constrains when the path is established, which can be added in this attribute in the future version.

#### 3.2.2. Explicit Routing

Explicit routing attribute describes an end-to-end path for DetNet flow, by listing nodes along the path in order and specifying their types. The DetNet node type has been specified in section 4.1.1. If service protection is needed, DetNet flow is replicated in relay node, going through different paths, and eliminated in another relay node. It makes the DetNet route a point-to-multipoint-to-point (P-MP-P) path. In [RFC4875], explicit routing of a P-MP LSP is represented by a P-MP tree. Similarly, a P-MP-P tree is needed in DetNet, and the rules of building the tree is to be defined.

### 3.3. DetNet Flow Configuration Attribute

DetNet Configuration Attribute is used for path configuration after the path has been calculated, preparing for the DetNet Flow Transportation.

#### 3.3.1. Flow Identification

Flow Identification is data plane relevant, and it is defined in [I-D.ietf-detnet-flow-information-model].

### 3.3.2. Traffic Specification

Traffic Specification is defined  
in[I-D.ietf-detnet-flow-information-model] .

### 3.3.3. Encapsulation

[I-D.dt-detnet-dp-sol] defines more than one data plane protocols for DetNet service, and DetNet Encapsulation attribute specifies the type of encapsulation used in the node, including:

- o MPLS Pseudo Wire
- o Native IPv6
- o TSN

Notes: In one DetNet domain, the encapsulation should be the same; When a flow goes across different domains, the encapsulation needs to be changed. For example, when an DetNet Edge Node connects two TSN domains, at the entry or exit boundary of the DetNet domain, the encapsulation needs to be changed accordingly. Parameters in the encapsulation also needs to do the mapping. for example, the translation from flow Unique ID defined [IEEE802.1Qcc] to DetNet flow ID defined in [I-D.dt-detnet-dp-sol] should be defined in the configuration of the edge node .

### 3.3.4. Flow Priority

Flow Priority attribute specifies the priority reserved for DetNet flow in PSN header. The transit node can distinguish DetNet flow from non-DetNet flow by DetNet priority. And, if more than one DetNet priority is defined, it can also be used to describe DetNet flows with different quality requirements, e.g. , low latency DetNet flows and high latency DetNet flows.

Notes: In one DetNet domain, the priority reserved for DetNet should be the same. When crossing DetNet domains, the priority should be translated accordingly. For example, the priority transition from TSN domain to DetNet domain is defined in [I-D.varga-detnet-service-model] Annex 2 "Integrating Layer 3 and Layer 2 QoS".

This attribute is also data plane relevant. If there is no priority reserved for DetNet, other attribute should be specified to distinguish DetNet flows. The mapping from flow priority to output queue also makes it necessary to take queuing management



algorithm(section 3.1.4) into consideration when defining the DetNet priority.

#### 3.3.5. Queuing Parameters

Queuing Management Algorithm Type is described in section 3.1.4. Different algorithm use different parameters to manage queue. In a fully-centralized configuration model, the parameters can be distributed by CNC; in a distributed configuration model, the device can configure itself based on the application requirement and flow traffic specification information.

The queuing management configuration parameters and the corresponding YANG model are being defined in IEEE. For example, when stream policing and filtering defined in[IEEE802.1Qci] is deployed in one node, the parameter of Stream filter instance table ([IEEE802.1Qci] 8.6.5.1.1), Stream gate instance table ([IEEE802.1Qci] 8.6.5.1.2), Flow meter instance table ([IEEE802.1Qci] 8.6.5.1.3) should be configured by CNC or other control plane protocol.

#### 3.3.6. Replication Function

This attribute specifies whether the node will do replication to the packet of this flow. Configuration of Replication in relay node is defined in [IEEE802.1CB].

#### 3.3.7. Elimination Function

This attribute specifies whether the node will do elimination to the packet of a flow. For a multicast flow, elimination can be performed on some ports, but not on others in one node. Configuration of Elimination in relay node is defined in [IEEE802.1CB].

#### 3.3.8. Routing

Routing configuration is data plane relevant, but no matter what the encapsulation is, the following attributes should be contained:

- o Flow Identification: in the current data plane design, flow ID, PW label or other relevant information can be used in flow identification. Flow Identification Information may be not needed in Transit Node;
- o Operation: forwarding / replication / elimination / elimination&replication;
- o Next-hop;

- o Encapsulation: the packet should be re-encapsulated after replication or elimination. Usually, encapsulation Information is not needed in the Transit Node;

It is also relevant to the data plane identification. Take MPLS solution defined in [I-D.dt-detnet-dp-sol]

as an example:

Transit Node: Operation at a transit (P) node is normal MPLS forwarding. The outer label is either swapped or popped as required, and the packet is forwarded along the LSP.

Relay Node: S-label is used to identify the flow and indicate whether the packet should be replicated or eliminated or both. In one of the relay nodes in the path, the parameter table can be as follows:

Incoming S-Label	Flow ID	Replication	Elimination	Outcoming S-Label
Label-1	Flow 1	Yes	No	Label-5
Label-2	Flow 2	No	Yes	Label-6
Label-3	Flow 3	Yes	Yes	Label-7

In this table, Label-1/ Label-2/ Label-3 are distributed from the current relay node to the previous relay node in the path; Label-5/ Label-6/ Label-7 are distributed from the next relay node to the current relay node in the path;

### 3.4. DetNet Status Attribute

The DetNet status attributes are provided by the device for each DetNet flow. The Status Attributes describe the status of the flow when it is transmitted in the network.

#### 3.4.1. Performance Status

Performance Status contains:

- o Maximum Link Latency: which is measured by the packet's timestamp
- o Packet Loss: which describes the packet loss of a particular flow in this node

- o Flow Policing and Filtering Status: the illegal behavior of the flow that is recorded by the node

### 3.4.2. Replication/Elimination Status

Detailed discussion of Replication/Elimination status is specified in [IEEE802.1CB].

If the S-label indicates that the packet is supposed to be eliminated, the relay node should read the sequence number of the packet and see whether this packet has been received before. For example, the parameters of one relay node can be:

Flow ID	Sequence Number
Flow 1	1001
Flow 1	1002
Flow 1	1003

If a packet of flow 1 with the sequence number of 1001 is received, it should be dropped in this relay node; If a packet of flow 1 with the sequence number of 1005 is received, it should be forwarded in this relay node, and the parameter talbe will be updated.

## 4. DetNet Configuration YANG Model

This section specifies the network management information that is used for the fully centralized DetNet configuration model. YANG model for other configuration model is to be defined in the future version of the draft.

### 4.1. DetNet Topology YANG Model

```
<CODE BEGINS> file "ietf-detnet-topology@2018-01-15.yang"
module ietf-te-detnet-topology {
  namespace "urn:ietf:params:xml:ns:yang:ietf-detnet-topology";
  prefix "detnet-to";

  import ietf-te-types {
    prefix "te-types";
  }

  import ietf-routing-types {
```

```
    prefix "rt-types";
  }

  import ietf-te-topology {
    prefix "tet";
  }

  import ietf-network {
    prefix "nw";
  }

  import ietf-network-topology {
    prefix "nt";
  }

  organization
    "IETF Deterministic Networking(detnet)Working Group";

  contact
    "WG Web:    <http://tools.ietf.org/wg/detnet/>
    WG List:    <mailto:detnet@ietf.org>

    WG Chair: Lou Berger
               <mailto:lberger@labn.net>

    Editor:    Xuesong Geng
               <mailto:gengxuesong@huawei.com>

    Editor:    Mach Chen
               <mailto:mach.chen@huawei.com>";

  description
    "This YANG module augments the 'ietf-te-topology'
    module with detnet capability data for detnet
    configuration";

  revision "2018-01-15" {
    description "Initial revision";
    reference "RFC XXXX: YANG Data Model for DetNet Topologies";
    //RFC Ed.: replace XXXX with actual RFC number and remove
    // this note
  }

  grouping detnet-link-info-attributes{
    description
      "DetNet capability attributes in a DetNet topology";
    container detnet-performance-metric-attributes{
      description
```

```
        "Link performance information in real time.";
        uses detnet-performance-metric-attributes;
    }
    container detnet-queuing-management-algorithm{
        description
            "Detnet queuing management algorithm used in
            output queue";
        uses detnet-queuing-management-algorithm;
    }
}

grouping detnet-performance-metric-attributes{
    description
        "Link performance information in real time.";
    container maximum-detnet-reservable-bandwidth{
        uses te-types:te-bandwidth;
        description
            "This container specifies the maximum bandwidth
            that is reserved for DetNet on this link.";
    }
    container reserved-detnet-bandwidth{
        uses te-types:te-bandwidth;
        description
            "This container specifies the bandwidth that has
            been reserved for DetNet on this link.";
    }
    container available-detnet-bandwidth{
        uses te-types:te-bandwidth;
        description
            "This container specifies the bandwidth that can
            be used for new DetNet flows on this link.";
    }
    leaf minimum-detnet-device-delay{
        type uint32;
        description
            "Minimum delay in the device for DetNet flows";
    }
    leaf maximum-detnet-device-delay{
        type uint32;
        description
            "Maximum delay in the device for DetNet flows";
    }
}

grouping detnet-queuing-management-algorithm{
    description
        "Detnet queuing management algorithm used in
        output queue";
```

```
leaf queuing-management-algorithm{
  type enumeration{
    enum credit-based-shaping{
      reference
        "IEEE P802.1 Qav";
    }
    enum time-aware-shaping{
      reference
        "IEEE P802.1 Qbv";
    }
    enum cyclic-queuing-and-forwarding{
      reference
        "IEEE P802.1 Qch";
    }
    enum asynchronous-traffic-shaping{
      reference
        "IEEE P802.1 Qcr";
    }
  }
  description
    "Detnet queuing management algorithm type";
}

grouping detnet-node-info-attributes{
  description
    "DetNet capability attributes in a DetNet node";
  container detnet-node-type{
    description
      "Three types of DetNet nodes";
    reference
      "draft-ietf-detnet-architecture-03:
      Deterministic Networking Architecture";
    uses detnet-node-type;
  }
  container detnet-resource-reservation-attributes{
    description
      "Attributes about resource reservation for
      DetNet flows";
    uses detnet-resource-reservation-attributes;
  }
  leaf detnet-elimination-capability{
    type boolean;
    description
      "This node is able to do DetNet packet
      elimination";
  }
}
```

```
    leaf detnet-replication-capability{
      type boolean;
      description
        "This node is able to do DetNet packet
        replication";
    }
  }
}

grouping detnet-node-type{
  description
    "This grouping defines three types of DetNet nodes";
  reference
    "draft-ietf-detnet-architecture-03:Deterministic
    Networking Architecture";
  leaf detnet-node-type{
    type enumeration{
      enum edge-node{
        description
          "An instance of a DetNet relay node that
          includes either a DetNet service layer proxy
          function for DetNet service protection (e.g.
          the addition or removal of packet sequencing
          information) for one or more end systems, or
          starts or terminate congestion protection at
          the DetNet transport layer, analogous to a
          Label Edge Router (LER).";
      }
      enum relay-node{
        description
          "A DetNet node including a service layer
          function that interconnects different DetNet
          transport layer paths to provide service
          protection. A DetNet relay node can be a bridge,
          a router, a firewall, or any other system that
          participates in the DetNet service layer. It
          typically incorporates DetNet transport layer
          functions as well, in which case it is
          collocated with a transit node.";
      }
      enum transit-node{
        description
          "A node operating at the DetNet transport layer,
          that utilizes link layer and/or network layer
          switching across multiple links and/or
          sub-networks to provide paths for DetNet
          service layer functions. Optionally provides
          congestion protection over those paths. An MPLS
          LSR is an example of a DetNet transit node.";
      }
    }
  }
}
```

```

    }
  }
  description
    "The type this node belongs to, which also determines
    the role the node can play in DetNet ";
}
}

grouping detnet-resource-reservation-attributes{
  description
    "This grouping describes reservation operation for
    the entire device";
  leaf MaxFanInPorts{
    type uint32;
    description
      "maximum number of fan-in ports in the device";
  }
  leaf MaxPacketSize{
    type uint32;
    description
      "maximum Packet size the device allows";
  }
  leaf MaxDetNetClasses{
    type uint32;
    description
      "maximum number of traffic classes that can be
      reserved for DetNet";
  }
}

augment "/nw:networks/nw:network/nw:node" {
  when "../nw:network-types/tet:te-topology"
  {
    description
      "";
  }
  description
    "Advertised DetNet link information attributes.";
  uses detnet-link-info-attributes;
}

augment "/nw:networks/nw:network/nt:link" {
  when "../nw:network-types/tet:te-topology"
  {
    description
      "";
  }
  description

```



```
        "Advertised DetNet node information attributes.";
    uses detnet-node-info-attributes;
}
}
<CODE ENDS>
```

#### 4.2. DetNet Static Configuration YANG Model

```
<CODE BEGINS> file "ietf-detnet-static @2018-01-15.yang"
module ietf-detnet-static {
    namespace "urn:ietf:params:xml:ns:yang:ietf-detnet-static";
    prefix "detnet-static";

    import ietf-routing {
        prefix "rt";
    }

    import ietf-yang-types{
        prefix "yang";
    }

    import ietf-inet-types{
        prefix "inet";
    }

    import ietf-routing-types {
        prefix "rt-types";
    }

    organization
        "IETF Deterministic Networking(detnet)Working Group";

    contact
        "WG Web:    <http://tools.ietf.org/wg/detnet/>
        WG List:    <mailto: detnet@ietf.org>

        WG Chair: Lou Berger
                  <mailto:lberger@labn.net>

        Editor:    Xuesong Geng
                  <mailto:gengxuesong@huawei.com>

        Editor:    Mach Chen
                  <mailto:mach.chen@huawei.com>";

    description
        "This YAGN module augments the 'ietf-routing' module
        with detnet flow configuration attribute";
```

```
revision "2018-01-15" {
  description "Initial revision";
  reference "RFC XXXX: YANG Data Model for DetNet Topologies";
  //RFC Ed.: replace XXXX with actual RFC number and remove
  // this note
}

grouping flow-identification {
  description
    "DetNet flow identification";
  reference
    "draft-farkas-detnet-flow-information-model";
  leaf source-ip-address {
    type inet:ip-address;
    description
      "Source IP address";
  }
  leaf destination-ip-address {
    type inet:ip-address;
    description
      "Destination IP address";
  }
  leaf source-mac-address {
    type yang:mac-address;
    description
      "Source MAC address";
  }
  leaf destination-mac-address {
    type yang:mac-address;
    description
      "Destination MAC address";
  }
  leaf ipv6-flow-label {
    type uint32;
    description
      "ipv6 flow label";
  }
  leaf mpls-label {
    type rt-types:mpls-label;
    description
      "MPLS Label";
  }
}

grouping traffic-specification{
  description
    "traffic-specification specifies how the Source
    transmits packets for the flow. This is the
```

```
        promise/request of the Source to the network.
        The network uses this traffic specification
        to allocate resources and adjust queue
        parameters in network nodes.";
reference
  "draft-farkas-detnet-flow-information-model";
leaf max-packets-per-interval{
  type uint16;
  description
    "max-packets-per-interval specifies the maximum
    number of packets that the application shall
    transmit in one Interval.";
}
leaf max-packet-size{
  type uint16;
  description
    "max-packet-size specifies maximum packet size
    that the Source will transmit";
}
leaf queuing-algorithm-selection{
  type uint8;
  description
    "";
}
}

grouping routing-configuration{
  description
    "configuration parameters direct data plane
    operations";
  container flow-identification{
    description
      "flow identification";
    uses flow-identification;
  }
  leaf operation{
    type enumeration{
      enum transmission{
        description
          "Operation: transmit ";
      }
      enum replication{
        description
          "Operation: packet replication";
      }
      enum elimination{
        description
          "Operation: packet elimination";
      }
    }
  }
}
```

```
    }
    enum elimination-and-replication{
      description
        "Operation: packet elimination and
        replication";
    }
  }
  description
    "The operation will be done to the
    packet";
}

grouping queuing-parameters{
  description
    "The paramters used to configure
    queuing managment algorithm";
}

grouping replication-function{
  description
    "The paramters used to configure
    packet replication";
}

grouping elimination-function{
  description
    "The paramters used to configure
    packet elimination";
}

augment "/rt:routing"{
  description
    "DetNet node static configuration
    attributes.";
  uses flow-identification;
  uses traffic-specification;
  uses routing-configuration;
  uses queuing-parameters;
  uses replication-function;
  uses elimination-function;
}
}
<CODE ENDS>
```

## 5. DetNet Configuration Model Classification

This section defines three classes of DetNet configuration model: fully distributed configuration model, fully centralized configuration model, hybrid configuration model, based on different network architectures, showing how configuration information exchanges between various entities in the network.

### 5.1. Fully Distributed Configuration Model

In a fully distributed configuration model, UNI information is transmitted over DetNet UNI protocol from the user side to the network side; then UNI information and network configuration information propagate in the network over distributed control plane protocol. For example:

- 1) IGP collects topology information and DetNet capabilities of network([I-D.geng-detnet-info-distribution]);
- 2) Control Plane of the Edge Node(Ingress) receives a flow establishment request from UNI and calculates a/some valid path(s);
- 3) Using RSVP-TE, Edge Node(Ingress) sends a PATH message with explicit route. After receiving the PATH message, the other Edge Node(Egress) sends a Resv message with distributed label and resource reservation request.

Current distributed control plane protocol, e.g., RSVP-TE[RFC3209], SRP[IEEE802.1Qcc], can only reserve bandwidth along the path, while the configuration of a fine-grained schedule, e.g., Time Aware Shaping(TAS) defined in [IEEE802.1Qbv], is not supported.

The fully distributed configuration model is not covered by this draft. It should be discussed in the future DetNet control plane work.

### 5.2. Fully Centralized Configuration Model

In the fully centralized configuration model, UNI information is transmitted from Centralized User Configuration (CUC) to Centralized Network Configuration(CNC). Configurations of routers for DetNet flows are performed by CNC with network management protocol. For example:

- 1) CNC collects topology information and DetNet capability of network through Netconf;

2) CNC receives a flow establishment request from UNI and calculates a/some valid path(s);

3) CNC configures the devices along the path for flow transmission.

### 5.3. Hybrid Configuration Model

In the hybrid configuration model, controller and control plane protocols work together to offer DetNet service, and there are a lot of possible combinations. For example:

1) CNC collects topology information and DetNet capability of network through IGP/BGP-LS;

2) CNC receives a flow establishment request from UNI and calculates a/some valid path(s);

3) Based on the calculation result, CNC distributes flow path information to Edge Node(Ingress) and other information(e.g. replication/elimination) to the relevant nodes.

4) Using RSVP-TE, Edge Node(Ingress) sends a PATH message with explicit route. After receiving the PATH message, the other Edge Node(Egress) sends a Resv message with distributed label and resource reservation request.

or

1) Controller collects topology information and DetNet capability of network through IGP/BGP-LS;

2) Control Plane of Edge Node(Ingress) receives a flow establishment request from UNI;

3) Edge Node(Ingress) sends the path establishment request to CNC through PCEP;

4) After Calculation, CNC sends back the path information of the flow to the Edge Node(Ingress) through PCEP;

5) Using RSVP-TE, Edge Node(Ingress) sends a PATH message with explicit route. After receiving the PATH message, the other Edge Node(Egress) sends a Resv message with distributed label and resource reservation request.

There are also other variations that can be included in the hybrid model. This draft can not cover all the control plane data needed

in hybrid configuration models. Every solution has there own mechanism and corresponding parameters to make it work.

Editor's Note:

1. There are a lot of optional DetNet configuration models, and different scenario in different use case can choose one of them based on its conditions. Maybe next step of the work is to pick up one or more typical scenarios and give a practical solution.

2. [IEEE802.1Qcc] also defines three TSN configuration models: fully-centralized model, fully-distributed model, centralized Network / distributed User Model. This section defines the configuration model roughly the same, to keep the design of L2 and L3 in the same structure. Hybrid configuration model is slightly different from the 'centralized Network / distributed User Model'. The hybrid configuration model intends to contain more variations.

## 6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

## 7. Security Considerations

## 8. Acknowledgements

## 9. References

### 9.1. Normative References

[I-D.dt-detnet-dp-sol]

Korhonen, J., Andersson, L., Jiang, Y., Finn, N., Varga, B., Farkas, J., Bernardos, C., Mizrahi, T., and L. Berger, "DetNet Data Plane Encapsulation", draft-dt-detnet-dp-sol-02 (work in progress), September 2017.

[I-D.ietf-detnet-architecture]

Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-04 (work in progress), October 2017.

[I-D.ietf-detnet-flow-information-model]  
Farkas, J., Varga, B., rodney.cummings@ni.com, r., Jiang, Y., and Y. Zha, "DetNet Flow Information Model", draft-ietf-detnet-flow-information-model-00 (work in progress), January 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

## 9.2. Informative References

[I-D.geng-detnet-info-distribution]  
Geng, X. and M. Chen, "IGP-TE Extensions for DetNet Information Distribution", draft-geng-detnet-info-distribution-01 (work in progress), September 2017.

[I-D.ietf-detnet-use-cases]  
Grossman, E., "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-14 (work in progress), February 2018.

[I-D.ietf-teas-yang-te]  
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-12 (work in progress), February 2018.

[I-D.ietf-teas-yang-te-topo]  
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-15 (work in progress), February 2018.

[I-D.thubert-tsvwg-detnet-transport]  
Thubert, P., "A Transport Layer for Deterministic Networks", draft-thubert-tsvwg-detnet-transport-01 (work in progress), October 2017.

[I-D.varga-detnet-service-model]  
Varga, B. and J. Farkas, "DetNet Service Model", draft-varga-detnet-service-model-02 (work in progress), May 2017.



- [IEEE802.1CB]  
"IEEE, "Frame Replication and Elimination for Reliability (IEEE Draft P802.1CB)", 2017,  
<<http://www.ieee802.org/1/files/private/cb-drafts/>>.", 2016.
- [IEEE802.1Q-2014]  
"IEEE, "IEEE Std 802.1Q Bridges and Bridged Networks", 2014, <<http://ieeexplore.ieee.org/document/6991462/>>.", 2014.
- [IEEE802.1Qbu]  
"IEEE, "IEEE Std 802.1Qbu Bridges and Bridged Networks - Amendment 26: Frame Preemption", 2016,  
<<http://ieeexplore.ieee.org/document/7553415/>>.", 2016.
- [IEEE802.1Qbv]  
"IEEE, "IEEE Std 802.1Qbu Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", 2015,  
<<http://ieeexplore.ieee.org/document/7572858/>>.", 2016.
- [IEEE802.1Qcc]  
"IEEE, "Stream Reservation Protocol (SRP) Enhancements and Performance Improvements (IEEE Draft P802.1Qcc)", 2017,  
<<http://www.ieee802.org/1/files/private/cc-drafts/>>.",
- [IEEE802.1Qch]  
"IEEE, "Cyclic Queuing and Forwarding (IEEE Draft P802.1Qch)", 2017,  
<<http://www.ieee802.org/1/files/private/ch-drafts/>>.", 2016.
- [IEEE802.1Qci]  
"IEEE, "Per-Stream Filtering and Policing (IEEE Draft P802.1Qci)", 2016,  
<<http://www.ieee802.org/1/files/private/ci-drafts/>>.", 2016.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.

[RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.

#### Authors' Addresses

Xuesong Geng  
Huawei

Email: [gengxuesong@huawei.com](mailto:gengxuesong@huawei.com)

Mach(Guoyi) Chen  
Huawei

Email: [mach.chen@huawei.com](mailto:mach.chen@huawei.com)

Zhenqiang  
China Mobile

Email: [lizhenqiang@chinamobile.com](mailto:lizhenqiang@chinamobile.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 24, 2019

X. Geng  
M. Chen  
Huawei  
Z. Li  
China Mobile  
R. Rahman  
Cisco Systems  
October 21, 2018

DetNet Configuration YANG Model  
draft-geng-detnet-conf-yang-06

Abstract

This document defines a YANG data model for Deterministic Networking (DetNet), which includes the DetNet topology YANG module, DetNet flow configuration YANG module and DetNet Transport QoS YANG Module.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA).

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 24, 2019.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminologies . . . . .	4
3. DetNet Topology Model . . . . .	4
3.1. DetNet Node Attributes . . . . .	5
3.2. DetNet Link Attributes . . . . .	6
3.3. DetNet Link Terminate Point Attributes . . . . .	7
4. DetNet Flow Configuration Model . . . . .	9
4.1. DetNet Service Proxy Configuration Attributes . . . . .	9
4.2. DetNet Service Layer Configuration Attributes . . . . .	10
4.3. DetNet Transport Layer Configuration Attributes . . . . .	12
4.4. DetNet Flow Configuration Example . . . . .	13
5. DetNet Transport QoS Model . . . . .	14
6. DetNet Yang Structure . . . . .	15
6.1. DetNet Topology Model Tree Diagram . . . . .	15
6.2. DetNet Flow Configuration Model Tree Diagram . . . . .	17
7. DetNet YANG Model . . . . .	22
7.1. DetNet Topology YANG Model . . . . .	22
7.2. DetNet Flow Configuration YANG Model . . . . .	28
8. DetNet Configuration Model Classification . . . . .	45
8.1. Fully Distributed Configuration Model . . . . .	45
8.2. Fully Centralized Configuration Model . . . . .	46
8.3. Hybrid Configuration Model . . . . .	46
9. Open issues . . . . .	47
10. IANA Considerations . . . . .	48
11. Security Considerations . . . . .	48
12. Acknowledgements . . . . .	48
13. References . . . . .	48
13.1. Normative References . . . . .	48
13.2. Informative References . . . . .	49
Authors' Addresses . . . . .	51

## 1. Introduction

Deterministic Networking (DetNet) [I-D.ietf-detnet-architecture] is defined to provide high-quality network service with extremely low packet loss rate, bounded low latency and jitter.

DetNet flow information is defined in [I-D.ietf-detnet-flow-information-model], and the DetNet models are categorized as:

- o Flow models: describe characteristics of data flows. These models describe in detail all relevant aspects of a flow that are needed to support the flow properly by the network between the source and the destination(s).
- o Service models: describe characteristics of services being provided for data flows over a network. These models can be treated as a network operator independent information model.
- o Configuration models: describe in detail the settings required on network nodes to serve a data flow properly. Service and flow information models are used between the user and the network operator. Configuration information models are used between the management/control plane entity of the network and the network nodes.

They are shown in the Figure 1.

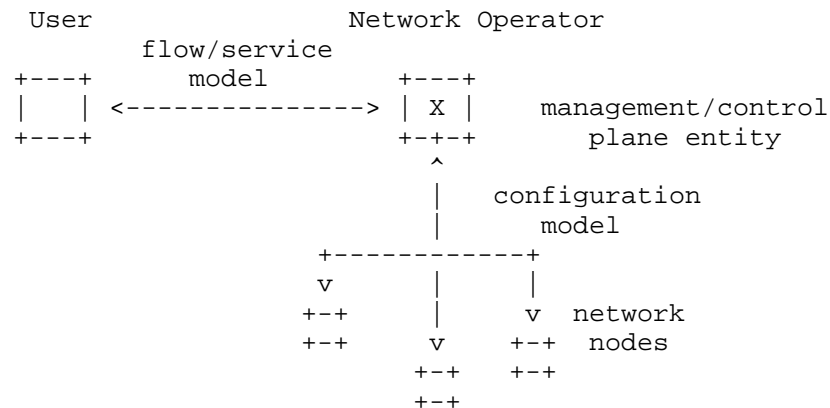


Figure 1. Three Information Models

This document defines DetNet configuration YANG [RFC7950] [RFC6991] data models, which include:

- o DetNet topology model

DetNet topology model is designed for DetNet topology/capability discovery and device configuration, it is an augmentation of the ietf-te-topology model [I-D.ietf-teas-yang-te-topo]. The detail of DetNet topology model is defined in Section 3.

- o DetNet flow configuration model

DetNet flow model is designed for DetNet flow path configuration and flow status reporting. The detail of DetNet flow configuration model is defined in Section 4.

- o DetNet transport QoS model

DetNet transport QoS model is designed for QoS attributes configuration of transport tunnels to achieve end-to-end bounded latency and zero congestion loss. The detail of DetNet transport QoS model is defined in Section 5.

## 2. Terminologies

This documents uses the terminologies defined in [I-D.ietf-detnet-architecture].

## 3. DetNet Topology Model

A DetNet topology is composed of a set of DetNet nodes and DetNet links. DetNet nodes represent the network devices that can transport DetNet services, which are connected by DetNet links. A DetNet Link Terminate Point(LTP) is the connection point between a DetNet node and a DetNet link, which represents the port or interface of a network node. The concept of DetNet node/link/LTP are similar as TE node/link/LTP that are defined in [I-D.ietf-teas-yang-te-topo].

Figure 2 shows a simple DetNet topology: A is a DetNet node, B is DetNet a LTP, and C is a DetNet link.

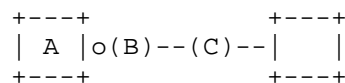


Figure 2. An example of DetNet Topology

DetNet topology model (ietf-detnet-topology) augments ietf-te-topology model [I-D.ietf-teas-yang-te-topo] to cover the following

attributes, which are necessary for supporting DetNet congestion protection and service protection functions:

- o Bandwidth related attributes (e.g., bandwidth reserved for DetNet);
- o Buffer/queue management related attributes (e.g., queue management algorithm, etc.);
- o PREOF (Packet Replication, Ordering and Elimination Function) capabilities and parameters (e.g., maximum out-of-order packets, etc.);
- o Delay related attributes (e.g., node processing delay, queuing delay, link delay, etc.);

The above attributes are categorized into three types: node attributes, link attributes and LTP attributes. The detailed descriptions and model definitions are specified in section 4.1, 4.2 and 4.3, respectively.

### 3.1. DetNet Node Attributes

Section 4.3 of [I-D.finn-detnet-bounded-latency] gives a DetNet time model, which defines that the delay within a node includes five parts: processing delay, regulation delay, queuing delay, output delay and preemption delay. The processing delay, queuing delay and regulation delay are variable in general, but for DetNet, these delays should be bounded, which is the basic assumption of deterministic networking. These bounded delay parameters are necessary to perform DetNet path computation. Among this delay attributes, processing delay and regulation delay are node relevant, and the queuing delay is LTP relevant. In addition, in order to simplify the model and implementation, the processing delay and regulation delay are combined as processing delay, and the preemption delay is included in queuing delay. [Editor notes: more comments and inputs need here].

For the DetNet node attributes, the following variables are introduced:

- o Maximum DetNet packet processing delay
- o Minimum DetNet packet processing delay
- o Maximum DetNet packet processing delay variation

The modeling structure is shown below:

```

augment /nw:networks/nw:network/nw:node/tet:te/tet:te-node-attributes:
  +--rw detnet-node-attributes
    +--rw minimum-packet-processing-delay?      uint32
    +--rw maximum-packet-processing-delay?      uint32
    +--rw maximum-packet-processing-delay-variation?  uint32

```

### 3.2. DetNet Link Attributes

DetNet link attributes include link delay and link bandwidth for DetNet. This document introduces the following link related attributes:

- o Link delay: link delay is a constant that only depends on the physical connection. It has been defined in ietf-te-topology [I-D.ietf-teas-yang-te-topo], and DetNet can reuse it directly.
- o Maximum DetNet reservable bandwidth: the maximum reservable bandwidth that is allocated to DetNet. For a 10G link, if 50% of the bandwidth is allocated to DetNet, then the maximum DetNet reservable bandwidth is 5G. That means there are 5G bandwidth that can be used by DetNet flows.
- o Reserved DetNet bandwidth: the bandwidth that has been reserved for DetNet flows.
- o Available DetNet bandwidth: the bandwidth that is available for new DetNet flows.

The DetNet link attributes are modeled within a link, and the YANG module structure is shown below:

```

augment /nw:networks/nw:network/nt:link/tet:te/tet:te-link-attributes:
  +--rw detnet-link-attributes
    +--rw maximum-reservable-bandwidth
      +--rw te-bandwidth
        +--rw (technology)?
          +--:(generic)
            +--rw generic?   te-bandwidth
      +--rw reserved-detnet-bandwidth
        +--rw te-bandwidth
          +--rw (technology)?
            +--:(generic)
              +--rw generic?   te-bandwidth
      +--rw available-detnet-bandwidth
        +--rw te-bandwidth
          +--rw (technology)?
            +--:(generic)
              +--rw generic?   te-bandwidth

```



### 3.3. DetNet Link Terminate Point Attributes

The concept of LTP is introduced in [I-D.ietf-teas-yang-te-topo], and this section introduces attributes for DetNet LTP.

PREOF (Packet Replication/Elimination/Ordering Function) is for DetNet service protection, which includes :

- o In-order delivery function: defined in Section 3.2.2.1 of [I-D.ietf-detnet-architecture]
- o Packet replication function: defined in Section 3.2.2.2 of [I-D.ietf-detnet-architecture]
- o Packet elimination function: defined in Section 3.2.2.3 of [I-D.ietf-detnet-architecture]

The above functions are modeled as a set of capabilities and relevant parameters, which are listed below:

- o in-order-capability: indicates whether a LTP has the in-order delivery capability.
- o maximum-number-of-out-of-order-packets: indicates the maximum number of out-of-order packets that an LTP can support, it depends on the reserved buffer size for packet reordering.
- o replication-capability: indicates whether a LTP has the packet replication capability.
- o elimination-capability: indicates whether a LTP has the packet elimination capability.

In addition, DetNet LTP also includes queuing management algorithms and queuing delay attributes. In the context of DetNet, the delay of queuing is bounded, and the bound depends on what queuing management method is used and how many buffers are allocated. More information can be found in [I-D.finn-detnet-bounded-latency]. Queuing related attributes are listed below:

- o queuing-algorithm-capabilities: it is modeled as a list that includes all queuing algorithms that a LTP supports.
- o detnet-queues: it's modeled as a list that includes all queues of a DetNet LTP. For each queue, it has the following attributes:
- o queue-identifier: an identifier of a queue. It could be an internal identifier that is only used within a node. Or it could

be used by a centralized controller to specify in which specific queue a flow/packet is required to enter.

- o queue-buffer-size: the size of a queue with unit of bytes.
- o enabled-queuing-algorithm: indicates what queuing management algorithm is enabled.
- o maximum-queuing-delay: the maximum queuing delay that a packet will undergo when transmitted through the queue.
- o minimum-queuing-delay: the minimum queuing delay that a packet will undergo when transmitted through the queue.
- o maximum-queuing-delay-variation: the maximum queuing delay variation that a packet will undergo when transmitted the queue.

The DetNet LTP attributes are modeled with a LTP, the YANG module structure is shown below:

```
augment /nw:networks/nw:network/nw:node/nt:termination-point/tet:te:
  +--rw detnet-terminate-point-attributes
    +--rw elimination-capability?          boolean
    +--rw replication-capability?          boolean
    +--rw in-ordering-capability
      |   +--rw in-ordering-capability?    boolean
      |   +--rw maximum-out-of-order-packets? uint32
    +--rw queuing-algorithm-capabilities
      |   +--rw credit-based-shaping?      boolean
      |   +--rw time-aware-shaping?        boolean
      |   +--rw cyclic-queuing-and-forwarding? boolean
      |   +--rw asynchronous-traffic-shaping? boolean
    +--rw queues* [queue-identifier]
      +--rw queue-identifier                uint32
      +--rw queue-buffer-size?              uint32
      +--rw enabled-queuing-algorithm
        |   +--rw credit-based-shaping?    boolean
        |   +--rw time-aware-shaping?      boolean
        |   +--rw cyclic-queuing-and-forwarding? boolean
        |   +--rw asynchronous-traffic-shaping? boolean
      +--rw minimum-queuing-delay?          uint32
      +--rw maximum-queuing-delay?          uint32
      +--rw maximum-queuing-delay-variation? uint32
```

#### 4. DetNet Flow Configuration Model

DetNet flow configuration includes DetNet Service Proxy configuration, DetNet Service Layer configuration and DetNet Transport Layer configuration. The corresponding attributes used in different layers are defined in Section 4.1, 4.2, 4.3, respectively. Section 4.4 gives a simple example on how to use these attributes for DetNet flow configuration.

##### 4.1. DetNet Service Proxy Configuration Attributes

DetNet service proxy is responsible for mapping between application flows and DetNet flows at the edge node (egress/ingress node). Where the application flows can be either layer 2 or layer 3 flows. To identify a flow at the User Network Interface (UNI), as defined in [I-D.ietf-detnet-flow-information-model], the following flow attributes are introduced:

- o DetNet L3 Flow Identification, refers to Section 7.1.1 of [I-D.ietf-detnet-flow-information-model]
- o DetNet L2 Flow Identification, refers to Section 7.1.2 of [I-D.ietf-detnet-flow-information-model]

DetNet service proxy can also do flow filtering and policing at the ingress to prevent the misbehaved flows from going into the network, which needs:

- o Traffic Specification, refers to Section 7.2 of [I-D.ietf-detnet-flow-information-model]

The YANG module structure is shown below:

```

+--rw client-flow* [flow-id]
|   +--rw flow-id                               uint32
|   +--rw (flow-type)?
|   |   +--:(l2-flow-identification)
|   |   |   +--rw source-mac-address?          yang:mac-address
|   |   |   +--rw destination-mac-address?     yang:mac-address
|   |   |   +--rw ethertype?                   eth:ethertype
|   |   |   +--rw vlan-id?                     uint16
|   |   |   +--rw pcp
|   |   +--:(l3-flow-identification)
|   |   |   +--rw (ip-flow-type)?
|   |   |   |   +--:(ipv4)
|   |   |   |   |   +--rw src-ipv4-address?      inet:ipv4-address
|   |   |   |   |   +--rw dest-ipv4-address?     inet:ipv4-address
|   |   |   |   |   +--rw dscp?                  uint8
|   |   |   |   +--:(ipv6)
|   |   |   |   |   +--rw src-ipv6-address?      inet:ipv6-address
|   |   |   |   |   +--rw dest-ipv6-address?     inet:ipv6-address
|   |   |   |   |   +--rw traffic-class?         uint8
|   |   |   |   |   +--rw flow-label?            inet:ipv6-flow-label
|   |   |   +--rw source-port?                  inet:port-number
|   |   |   +--rw destination-port?             inet:port-number
|   |   |   +--rw protocol?                     uint8
|   +--rw traffic-specification
|   |   +--rw interval?                         uint32
|   |   +--rw max-packets-per-interval?         uint32
|   |   +--rw max-payload-size?                 uint32
|   |   +--rw average-packets-per-interval?     uint32
|   |   +--rw average-payload-size?             uint32

```

#### 4.2. DetNet Service Layer Configuration Attributes

DetNet service functions, e.g., DetNet tunnel initialization/termination and service protection, are provided in DetNet service layer. To support these functions, the following service attributes need to be configured:

- o DetNetwork flow identification, refers to Section 7.1.3 of [I-D.ietf-detnet-flow-information-model].
- o Service function indication, indicates which service function will be invoked at a DetNet edge, relay node or end station. (DetNet tunnel initialization or termination are default functions in DetNet service layer, so there is no need for explicit indication.)
- o Flow Rank, refers to Section 7.3 of [I-D.ietf-detnet-flow-information-model].

- o Service Rank, refers to Section 7.4 of [I-D.ietf-detnet-flow-information-model].
- o Service encapsulation, refers to Section 6.2 of [I-D.ietf-detnet-dp-sol-mpls]
- o Transport encapsulation, refers to Section 6.2 of [I-D.ietf-detnet-dp-sol-mpls] and Section 3 of [I-D.ietf-detnet-dp-sol-ip]

The YANG module structure is shown below:

```

+--rw relay-node
  +--rw name? string
  +--rw flow-rank
  +--rw service-rank
  +--rw in-segment* [in-segment-id]
    +--rw in-segment-id uint32
    +--rw (flow-type)?
      +--:(IP)
        +--rw (ip-flow-type)?
          +--:(ipv4)
            +--rw src-ipv4-address? inet:ipv4-address
            +--rw dest-ipv4-address? inet:ipv4-address
            +--rw dscp? uint8
          +--:(ipv6)
            +--rw src-ipv6-address? inet:ipv6-address
            +--rw dest-ipv6-address? inet:ipv6-address
            +--rw traffic-class? uint8
            +--rw flow-label? inet:ipv6-flow-label
        +--rw source-port? inet:port-number
        +--rw destination-port? inet:port-number
        +--rw protocol? uint8
      +--:(MPLS)
        +--rw service-label uint32
    +--rw service-function? service-function-type
  +--rw out-segment* [out-segment-id]
    +--rw out-segment-id uint32
    +--rw detnet-service-encapsulation
      +--rw service-label uint32
      +--rw control-word uint32
    +--rw detnet-transport-encapsulation
      +--rw (tunnel-type)?
        +--:(IPv4)
          +--rw ipv4-encapsulation
            +--rw src-ipv4-address inet:ipv4-address
            +--rw dest-ipv4-address inet:ipv4-address
            +--rw protocol uint8

```

```

|
|
|      +--rw ttl?                uint8
|      +--rw dscp?               uint8
|  +--:(IPv6)
|      +--rw ipv6-encap          uint8
|      +--rw src-ipv6-address    inet:ipv6-address
|      +--rw dest-ipv6-address   inet:ipv6-address
|      +--rw next-header         uint8
|      +--rw traffic-class?      uint8
|      +--rw flow-label?         inet:ipv6-flow-label
|      +--rw hop-limit?          uint8
|  +--:(MPLS)
|      +--rw mpls-encap          uint8
|      +--rw label-operations* [label-oper-id]
|          +--rw label-oper-id    uint32
|          +--rw (label-actions)?
|              +--:(label-push)
|                  +--rw label-push
|                      +--rw label        uint32
|                      +--rw s-bit?       boolean
|                      +--rw tc-value?    uint8
|                      +--rw ttl-value?   uint8
|              +--:(label-swap)
|                  +--rw label-swap
|                      +--rw out-label    uint32
|                      +--rw ttl-action?  ttl-action-definition
|  +--rw interval?               uint32
|  +--rw max-packets-per-interval? uint32
|  +--rw max-payload-size?       uint32
|  +--rw average-packets-per-interval? uint32
|  +--rw average-payload-size?   uint32

```

#### 4.3. DetNet Transport Layer Configuration Attributes

As defined in [I-D.ietf-detnet-architecture], DetNet transport layer optionally provides congestion protection for DetNet flows over paths provided by the underlying network. Explicit route is another mechanism that is used by DetNet to avoid temporary interruptions caused by the convergence of routing or bridging protocols, and it is also implemented at the DetNet transport layer.

To support congestion protection and explicit route, the following transport layer related attributes are necessary:

- o Traffic Specification, refers to Section 7.2 of [I-D.ietf-detnet-flow-information-model]. It may be used for bandwidth reservation, flow shaping, filtering and policing.

- o Explicit path, existing explicit route mechanisms can be reused. For example, if Segment Routing (SR) tunnel is used as the transport tunnel, the configuration is mainly at the ingress node of the transport layer; if the static MPLS tunnel is used as the transport tunnel, the configurations need to be at every transit node along the path; for pure IP based transport tunnel, it's similar to the static MPLS case.

The YANG module structure is shown below:

```

|  +--rw transit-node
|      +--rw interval?                               uint32
|      +--rw max-packets-per-interval?                uint32
|      +--rw max-payload-size?                        uint32
|      +--rw average-packets-per-interval?            uint32
|      +--rw average-payload-size?                    uint32

```

The parameters for DetNet transport QoS are defined in Section 5.

#### 4.4. DetNet Flow Configuration Example

This section gives an example about how to implement an end-2-end DetNet service with the collaboration of DetNet proxy, service and transport layers.

To simplify the explanation, several terms are introduced. This document defines DetNet Service Proxy Instance (DSPI), DetNet Service Instance (DSI) and DetNet Transport Instance for end-to-end DetNet flow configuration as showed in Figure 4. DSPI 1 at Edge Node 1 (E1) maps an application flow to a DetNet Flow (DF1), which is transmitted over a DetNet tunnel (Tn1). In DSI 2 of Relay Node 1 (R1), DetNet Flow 1(DF1) was replicated into two member flows: DetNet Flow 2 (DF2) transmitted by DetNet Tunnel 2 (Tn12) and DetNet Flow 3 (DF3) by DetNet Tunnel 3(Tn13). In DSPI 3 of Edge Node 2 (E2), DetNet Flow 2 (DF2) and DetNet Flow 3(DF3) were merged and mapped to application flow used by CE2.

DF: DetNet Flow  
 DSPI: DetNet Service Proxy Instance  
 DSI: DetNet Service Instance  
 DTI: DetNet Transport Instance  
 Tnl: Tunnel

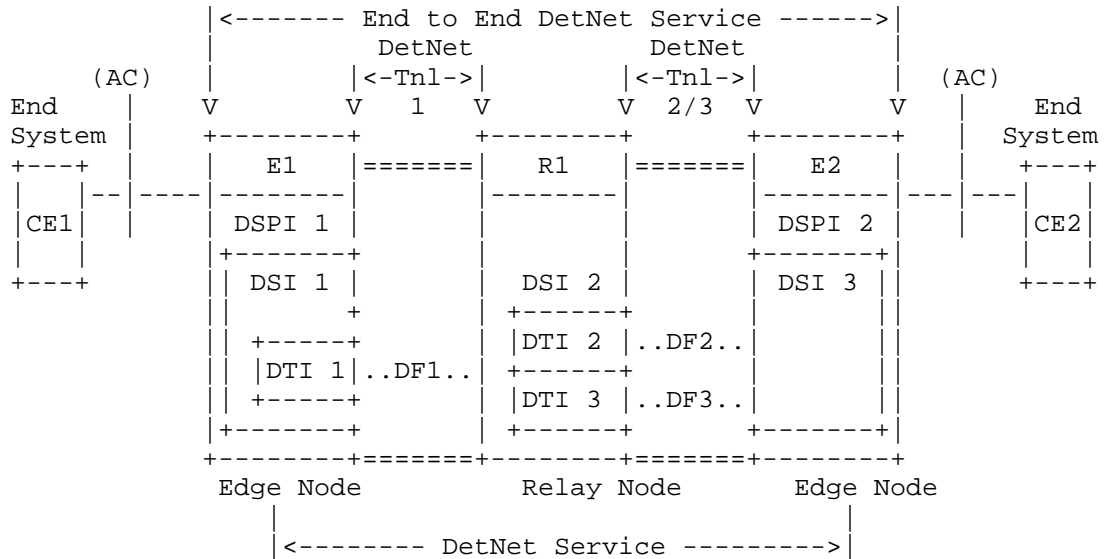


Figure 3. End-to-end DetNet Flow Configuration

## 5. DetNet Transport QoS Model

The YANG data model of transport QoS is very important to achieve end-to-end bounded latency and zero congestion loss. There are three possible methods to deal with the DetNet transport QoS YANG:

1. DetNet service is not aware of any QoS/queuing/bounded-latency information, and all relative parameters are defined in separate YANG models;
2. DetNet service is not aware of any of Qos/queuing/bounded-latency information, but it should maintain an interface to the corresponding YANG models;
3. DetNet service should be aware of the Qos/queuing/bounded-latency information, because some Qos/queuing/bounded-latency mechanisms are required to be configured with flow information;



How to define transport QoS YANG is still under discussion and the transport QoS YANG model is not included in the current version of the draft.

[Editor notes: more comments and inputs need here].

## 6. DetNet Yang Structure

### 6.1. DetNet Topology Model Tree Diagram

```

module: ietf-detnet-topology
augment /nw:networks/nw:network/nw:network-types/tet:te-topology:
  +--rw detnet-topology!
augment /nw:networks/nw:network/nw:node/tet:te/tet:te-node-attributes:
  +--rw detnet-node-attributes
    +--rw minimum-packet-processing-delay?          uint32
    +--rw maximum-packet-processing-delay?          uint32
    +--rw maximum-packet-processing-delay-variation? uint32
augment /nw:networks/nw:network/nt:link/tet:te/tet:te-link-attributes:
  +--rw detnet-link-attributes
    +--rw maximum-reservable-bandwidth
      +--rw te-bandwidth
        +--rw (technology)?
        +---:(generic)
          +--rw generic?   te-bandwidth
    +--rw reserved-detnet-bandwidth
      +--rw te-bandwidth
        +--rw (technology)?
        +---:(generic)
          +--rw generic?   te-bandwidth
    +--rw available-detnet-bandwidth
      +--rw te-bandwidth
        +--rw (technology)?
        +---:(generic)
          +--rw generic?   te-bandwidth
augment /nw:networks/nw:network/nw:node/nt:termination-point/tet:te:
  +--rw detnet-terminate-point-attributes
    +--rw elimination-capability?          boolean
    +--rw replication-capability?          boolean
    +--rw in-ordering-capability
      +--rw in-ordering-capability?        boolean
      +--rw maximum-out-of-order-packets?  uint32
    +--rw queuing-algorithm-capabilities
      +--rw credit-based-shaping?          boolean
      +--rw time-aware-shaping?            boolean
      +--rw cyclic-queuing-and-forwarding?  boolean
      +--rw asynchronous-traffic-shaping?   boolean
    +--rw queues* [queue-identifier]
      +--rw queue-identifier                uint32
      +--rw queue-buffer-size?              uint32
      +--rw enabled-queuing-algorithm
        +--rw credit-based-shaping?        boolean
        +--rw time-aware-shaping?          boolean
        +--rw cyclic-queuing-and-forwarding? boolean
        +--rw asynchronous-traffic-shaping? boolean
      +--rw minimum-queuing-delay?          uint32
      +--rw maximum-queuing-delay?          uint32
      +--rw maximum-queuing-delay-variation? uint32

```

## 6.2. DetNet Flow Configuration Model Tree Diagram

```

module: ietf-detnet-flow-config
  +--rw detnet-flow
    +--rw (detnet-node-role)?
      +---:(transit-node)
        +--rw transit-node
          +--rw interval?                               uint32
          +--rw max-packets-per-interval?               uint32
          +--rw max-payload-size?                       uint32
          +--rw average-packets-per-interval?            uint32
          +--rw average-payload-size?                   uint32
          +---:(relay-node)
            +--rw relay-node
              +--rw name?                               string
              +--rw flow-rank
              +--rw service-rank
              +--rw in-segment* [in-segment-id]
                +--rw in-segment-id                     uint32
                +--rw (flow-type)?
                  +---:(IP)
                    +--rw (ip-flow-type)?
                      +---:(ipv4)
                        +--rw src-ipv4-address?          inet:ipv4-address
                        +--rw dest-ipv4-address?          inet:ipv4-address
                        +--rw dscp?                       uint8
                      +---:(ipv6)
                        +--rw src-ipv6-address?          inet:ipv6-address
                        +--rw dest-ipv6-address?          inet:ipv6-address
                        +--rw traffic-class?              uint8
                        +--rw flow-label?                 inet:ipv6-flow-label
                    +--rw source-port?                   inet:port-number
                    +--rw destination-port?              inet:port-number
                    +--rw protocol?                      uint8
                  +---:(MPLS)
                    +--rw service-label                  uint32
                +--rw service-function?                  service-function-type
              +--rw out-segment* [out-segment-id]
                +--rw out-segment-id                     uint32
                +--rw detnet-service-encapsulation
                  +--rw service-label                    uint32
                  +--rw control-word                     uint32
                +--rw detnet-transport-encapsulation
                  +--rw (tunnel-type)?
                    +---:(IPv4)
                      +--rw ipv4-encapsulation
                        +--rw src-ipv4-address            inet:ipv4-address
                        +--rw dest-ipv4-address            inet:ipv4-address

```

```

+--rw protocol                               uint8
+--rw ttl?                                   uint8
+--rw dscp?                                  uint8
+---:(IPv6)
+--rw ipv6-encaplustion
+--rw src-ipv6-address                       inet:ipv6-address
+--rw dest-ipv6-address                     inet:ipv6-address
+--rw next-header                           uint8
+--rw traffic-class?                        uint8
+--rw flow-label?                           inet:ipv6-flow-label
+--rw hop-limit?                            uint8
+---:(MPLS)
+--rw mpls-encaplustion
+--rw label-operations* [label-oper-id]
+--rw label-oper-id                         uint32
+--rw (label-actions)?
+---:(label-push)
+--rw label-push
+--rw label                                 uint32
+--rw s-bit?                               boolean
+--rw tc-value?                            uint8
+--rw ttl-value?                           uint8
+---:(label-swap)
+--rw label-swap
+--rw out-label                             uint32
+--rw ttl-action?                          ttl-action-definition
+--rw interval?                            uint32
+--rw max-packets-per-interval?             uint32
+--rw max-payload-size?                    uint32
+--rw average-packets-per-interval?         uint32
+--rw average-payload-size?                uint32
+---:(edge-node)
+--rw edge-node
+--rw client-flow* [flow-id]
+--rw flow-id                               uint32
+--rw (flow-type)?
+---:(l2-flow-identification)
+--rw source-mac-address?                  yang:mac-address
+--rw destination-mac-address?            yang:mac-address
+--rw ethertype?                           eth:ethertype
+--rw vlan-id?                             uint16
+--rw pcp
+---:(l3-flow-identification)
+--rw (ip-flow-type)?
+---:(ipv4)
+--rw src-ipv4-address?                    inet:ipv4-address
+--rw dest-ipv4-address?                   inet:ipv4-address
+--rw dscp?                                uint8

```

```

| | | | | +---:(ipv6)
| | | | |   +--rw src-ipv6-address?          inet:ipv6-address
| | | | |   +--rw dest-ipv6-address?         inet:ipv6-address
| | | | |   +--rw traffic-class?             uint8
| | | | |   +--rw flow-label?                inet:ipv6-flow-la
bel
| | | | |   +--rw source-port?                inet:port-number
| | | | |   +--rw destination-port?          inet:port-number
| | | | |   +--rw protocol?                  uint8
| | | | | +--rw traffic-specification
| | | | |   +--rw interval?                    uint32
| | | | |   +--rw max-packets-per-interval?   uint32
| | | | |   +--rw max-payload-size?           uint32
| | | | |   +--rw average-packets-per-interval? uint32
| | | | |   +--rw average-payload-size?       uint32
| | | | | +--rw detnet-service-instance
| | | | |   +--rw name?                        string
| | | | |   +--rw flow-rank
| | | | |   +--rw service-rank
| | | | |   +--rw in-segment* [in-segment-id]
| | | | |     +--rw in-segment-id             uint32
| | | | |     +--rw (flow-type)?
| | | | |       +---:(IP)
| | | | |         +--rw (ip-flow-type)?
| | | | |           +---:(ipv4)
| | | | |             +--rw src-ipv4-address?  inet:ipv4-address
| | | | |             +--rw dest-ipv4-address? inet:ipv4-address
| | | | |             +--rw dscp?              uint8
| | | | |           +---:(ipv6)
| | | | |             +--rw src-ipv6-address?  inet:ipv6-address
| | | | |             +--rw dest-ipv6-address? inet:ipv6-address
| | | | |             +--rw traffic-class?     uint8
| | | | |             +--rw flow-label?        inet:ipv6-flow-label
| | | | |           +--rw source-port?          inet:port-number
| | | | |           +--rw destination-port?     inet:port-number
| | | | |           +--rw protocol?            uint8
| | | | |       +---:(MPLS)
| | | | |         +--rw service-label           uint32
| | | | |         +--rw service-function?       service-function-type
| | | | | +--rw out-segment* [out-segment-id]
| | | | |   +--rw out-segment-id                uint32
| | | | |   +--rw detnet-service-encapsulation
| | | | |     +--rw service-label              uint32
| | | | |     +--rw control-word               uint32
| | | | | +--rw detnet-transport-encapsulation
| | | | |   +--rw (tunnel-type)?
| | | | |     +---:(IPv4)
| | | | |       +--rw ipv4-encaplustion
| | | | |         +--rw src-ipv4-address        inet:ipv4-address

```

```

+--rw dest-ipv4-address          inet:ipv4-address
+--rw protocol                   uint8
+--rw ttl?                      uint8
+--rw dscp?                     uint8
+---:(IPv6)
+--rw ipv6-encaplustion
+--rw src-ipv6-address           inet:ipv6-address
+--rw dest-ipv6-address         inet:ipv6-address
+--rw next-header                uint8
+--rw traffic-class?            uint8
+--rw flow-label?               inet:ipv6-flow-label
+--rw hop-limit?                uint8
+---:(MPLS)
+--rw mpls-encaplustion
+--rw label-operations* [label-oper-id]
+--rw label-oper-id             uint32
+--rw (label-actions)?
+---:(label-push)
+--rw label-push
+--rw label                     uint32
+--rw s-bit?                    boolean
+--rw tc-value?                 uint8
+--rw ttl-value?                uint8
+---:(label-swap)
+--rw label-swap
+--rw out-label                  uint32
+--rw ttl-action?               ttl-action-defi
nition
+--rw interval?                  uint32
+--rw max-packets-per-interval?  uint32
+--rw max-payload-size?          uint32
+--rw average-packets-per-interval?  uint32
+--rw average-payload-size?      uint32
+---:(end-station)
+--rw end-station
+--rw client-flow* [flow-id]
+--rw flow-id                    uint32
+--rw (flow-type)?
+---:(l2-flow-identfication)
+--rw source-mac-address?        yang:mac-address
+--rw destination-mac-address?   yang:mac-address
+--rw ethertype?                 eth:ethertype
+--rw vlan-id?                   uint16
+--rw pcp
+---:(l3-flow-identification)
+--rw (ip-flow-type)?
+---:(ipv4)
+--rw src-ipv4-address?          inet:ipv4-address
+--rw dest-ipv4-address?         inet:ipv4-address

```

```

| | | | | +-rw dscp?                               uint8
| | | | | +--:(ipv6)
| | | | |   +-rw src-ipv6-address?                 inet:ipv6-address
| | | | |   +-rw dest-ipv6-address?                inet:ipv6-address
| | | | |   +-rw traffic-class?                     uint8
| | | | |   +-rw flow-label?                        inet:ipv6-flow-label
| | | | | +-rw source-port?                         inet:port-number
| | | | | +-rw destination-port?                    inet:port-number
| | | | | +-rw protocol?                            uint8
| | | +-rw traffic-specification
| | |   +-rw interval?                              uint32
| | |   +-rw max-packets-per-interval?              uint32
| | |   +-rw max-payload-size?                      uint32
| | |   +-rw average-packets-per-interval?          uint32
| | |   +-rw average-payload-size?                  uint32
| +-rw detnet-service-instance
| |   +-rw name?                                    string
| |   +-rw flow-rank
| |   +-rw service-rank
| |   +-rw in-segment* [in-segment-id]
| | |   +-rw in-segment-id                          uint32
| | |   +-rw (flow-type)?
| | | |   +--:(IP)
| | | | |   +-rw (ip-flow-type)?
| | | | | |   +--:(ipv4)
| | | | | | |   +-rw src-ipv4-address?              inet:ipv4-address
| | | | | | |   +-rw dest-ipv4-address?              inet:ipv4-address
| | | | | | |   +-rw dscp?                           uint8
| | | | | | |   +--:(ipv6)
| | | | | | |   +-rw src-ipv6-address?              inet:ipv6-address
| | | | | | |   +-rw dest-ipv6-address?              inet:ipv6-address
| | | | | | |   +-rw traffic-class?                  uint8
| | | | | | |   +-rw flow-label?                     inet:ipv6-flow-label
| | | | | | +-rw source-port?                         inet:port-number
| | | | | | +-rw destination-port?                    inet:port-number
| | | | | | +-rw protocol?                            uint8
| | | | | +--:(MPLS)
| | | | |   +-rw service-label                      uint32
| | | | | +-rw service-function?                    service-function-type
| +-rw out-segment* [out-segment-id]
| |   +-rw out-segment-id                          uint32
| |   +-rw detnet-service-encapsulation
| | |   +-rw service-label                          uint32
| | |   +-rw control-word                          uint32
| +-rw detnet-transport-encapsulation
| |   +-rw (tunnel-type)?
| | |   +--:(IPv4)
| | | |   +-rw ipv4-encapsulation

```

```

+---rw src-ipv4-address          inet:ipv4-address
+---rw dest-ipv4-address         inet:ipv4-address
+---rw protocol                  uint8
+---rw ttl?                      uint8
+---rw dscp?                     uint8
+---:(IPv6)
+---rw ipv6-encaplustion
+---rw src-ipv6-address          inet:ipv6-address
+---rw dest-ipv6-address         inet:ipv6-address
+---rw next-header               uint8
+---rw traffic-class?           uint8
+---rw flow-label?              inet:ipv6-flow-label
+---rw hop-limit?               uint8
+---:(MPLS)
+---rw mpls-encaplustion
+---rw label-operations* [label-oper-id]
+---rw label-oper-id            uint32
+---rw (label-actions)?
+---:(label-push)
+---rw label-push
+---rw label                    uint32
+---rw s-bit?                   boolean
+---rw tc-value?                uint8
+---rw ttl-value?               uint8
+---:(label-swap)
+---rw label-swap
+---rw out-label                 uint32
+---rw ttl-action?               ttl-action-defi
+---rw interval?                uint32
+---rw max-packets-per-interval? uint32
+---rw max-payload-size?         uint32
+---rw average-packets-per-interval? uint32
+---rw average-payload-size?     uint32

```

## 7. DetNet YANG Model

### 7.1. DetNet Topology YANG Model

```
<CODE BEGINS> file "ietf-detnet-topology@20180910.yang"
  module ietf-detnet-topology {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-detnet-topology";
    prefix "detnet-topology";

    import ietf-te-types {
      prefix "te-types";
    }
  }
<CODE ENDS>
```



```
import ietf-te-topology {
  prefix "tet";
}

import ietf-network {
  prefix "nw";
}

import ietf-network-topology {
  prefix "nt";
}

organization
  "IETF Deterministic Networking(DetNet)Working Group";

contact
  "WG Web:    <http://tools.ietf.org/wg/detnet/>
  WG List:    <mailto:detnet@ietf.org>

  WG Chair:   Lou Berger
              <mailto:lberger@labn.net>

              Janos Farkas
              <janos.farkas@ericsson.com>

  Editor:     Xuesong Geng
              <mailto:gengxuesong@huawei.com>

  Editor:     Mach Chen
              <mailto:mach.chen@huawei.com>

  Editor:     Zhenqiang Li
              <lizhenqiang@chinamobile.com>

  Editor:     Reshad Rahman
              <rrahman@cisco.com>";

description
  "This YANG module augments the 'ietf-te-topology'
  module with DetNet related capabilities and
  parameters.";

revision "2018-09-10" {
  description "Initial revision";
  reference "RFC XXXX: draft-geng-detnet-config-yang-05";
}
```

```
grouping detnet-queuing-algorithms {
  description
    "Relationship with IEEE 802.1 TSN YANG models is TBD.";
}

grouping detnet-node-attributes{
  description
    "DetNet node related attributes.";
  leaf minimum-packet-processing-delay{
    type uint32;
    description
      "Minimum packet processing delay
       in a node. The unit of the delay
       is microsecond(us)";
  }
  leaf maximum-packet-processing-delay{
    type uint32;
    description
      "Maximum packet processing delay
       in a node. The unit of the delay
       is microsecond(us)";
  }
  leaf maximum-packet-processing-delay-variation{
    type uint32;
    description
      "Maximum packet processing delay
       variation in a node. The unit of
       the delay variation is microsecond(us)";
  }
}

grouping detnet-link-attributes{
  description
    "DetNet link related attributes.";

  container maximum-reservable-bandwidth{
    uses te-types:te-bandwidth;
    description
      "This container specifies the maximum bandwidth
       that is reserved for DetNet on this link.";
  }

  container reserved-detnet-bandwidth{
    uses te-types:te-bandwidth;
    description
      "This container specifies the bandwidth that has
       been reserved for DetNet on this link.";
  }
}
```

```
    container available-detnet-bandwidth{
      uses te-types:te-bandwidth;
      description
        "This container specifies the bandwidth that is
        available for new DetNet flows on this link.";
    }
  }

  grouping detnet-terminate-point-attributes{
    description
      "DetNet terminate point related attributes.";

    leaf elimination-capability{
      type boolean;
      description
        "Indicates whether a node is able to do packet
        elimination.";
      reference
        "Section 3.2.2.3 of
        draft-ietf-detnet-architecture";
    }

    leaf replication-capability{
      type boolean;
      description
        "Indicates whether a node is able to do packet
        replication.";
      reference
        "Section 3.2.2.2 of
        draft-ietf-detnet-architecture";
    }
  }

  container in-ordering-capability {
    description
      "Indicates the parameters needed for
      packet in-ordering.";
    reference
      "Section 3.2.2.1 of
      draft-ietf-detnet-architecture";

    leaf in-ordering-capability {
      type boolean;
      description
        "Indicates whether a node is able to do packet
        in-ordering.";
    }
    leaf maximum-out-of-order-packets {
      type uint32;
      description
```

```
        "The maximum number of out-of-order packets.";
    }
}

container queuing-algorithm-capabilities {
    description
        "All queuing algorithms that a LTP supports.";
    uses detnet-queuing-algorithms;
}

list queues {
    key "queue-identifier";
    description
        "A list of DetNet queues.";
    leaf queue-identifier {
        type uint32;
        description
            "The identifier of the queue.";
    }
    leaf queue-buffer-size {
        type uint32;
        description
            "The size of the queue with unit of bytes.";
    }
}

container enabled-queuing-algorithm {
    description
        "The queuing algorithms that are enabled on the queue.";
    uses detnet-queuing-algorithms;
}

leaf minimum-queuing-delay{
    type uint32;
    description
        "The minimum queuing delay of the queue.
        The unit of the delay is microsecond(us)";
}
leaf maximum-queuing-delay{
    type uint32;
    description
        "The maximum queuing delay of the queue.
        The unit of the delay is microsecond(us)";
}
leaf maximum-queuing-delay-variation{
    type uint32;
    description
        "The maximum queuing delay variation of the queue.
        The unit of the delay variation is microsecond(us)";
}
```

```

    }
  }
}

augment "/nw:networks/nw:network/nw:network-types/tet:te-topology"{
  description
    "Introduce new network type for TE topology.";
  container detnet-topology {
    presence "Indicates DetNet topology.";
    description
      "Its presence identifies the DetNet topology type";
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes" {
  when "../../../nw:network-types/tet:te-topology/"
  + "detnet-topology:detnet-topology" {
    description
      "Augmentation parameters apply only for networks with
      DetNet topology type.";
  }
  description
    "Augmentation parameters apply for DetNet node attributes.";
  container detnet-node-attributes {
    description
      "Attributes for DetNet node.";
    uses detnet-node-attributes;
  }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:te-link-attributes" {
  when "../../../nw:network-types/tet:te-topology/"
  + "detnet-topology:detnet-topology" {
    description
      "Augmentation parameters apply only for networks with
      DetNet topology type.";
  }
  description
    "Augmentation parameters apply for DetNet link attributes.";
  container detnet-link-attributes {
    description
      "Attributes for DetNet link.";
    uses detnet-link-attributes;
  }
}

```

```

augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
  + "tet:te" {
  when "../../../nw:network-types/tet:te-topology/"
    + "detnet-topology:detnet-topology" {
    description
      "Augmentation parameters apply only for networks with
      DetNet topology type.";
  }
  description
    "Augmentation parameters apply for DetNet
    link termination point.";
  container detnet-terminate-point-attributes {
    description
      "Attributes for DetNet link terminate point.";
    uses detnet-terminate-point-attributes;
  }
}
} //topology module

```

<CODE ENDS>

## 7.2. DetNet Flow Configuration YANG Model

```

<CODE BEGINS> file "ietf-detnet-flow@20180910.yang"
module ietf-detnet-flow-config {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-detnet-flow-config";
  prefix "detnet-flow";

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-inet-types{
    prefix "inet";
  }

  import ietf-ethertypes {
    prefix "eth";
  }

  organization "IETF DetNet Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/detnet/>
    WG List:  <mailto:detnet@ietf.org>
    WG Chair: Lou Berger

```

```
<mailto:lberger@labn.net>

Janos Farkas
<janos.farkas@ericsson.com>

Editor:   Xuesong Geng
<mailto:gengxuesong@huawei.com>

Editor:   Mach Chen
<mailto:mach.chen@huawei.com>

Editor:   Zhenqiang Li
<lizhenqiang@chinamobile.com>

Editor:   Reshad Rahman
<rrahman@cisco.com>";
description
  "This YANG module describes the parameters needed
  for DetNet flow configuration and flow status
  reporting.";

revision "2018-09-10" {
  description "initial revision";
  reference "RFC XXXX: draft-geng-detnet-config-yang-05";
}

identity detnet-node-role {
  description
    "base detnet-node-role";
}

identity end-station {
  base detnet-node-role;
  description
    "Commonly called a 'host' in IETF documents,
    and an 'end station' is IEEE 802 documents.
    End systems of interest to this document
    are either sources or destinations of DetNet
    flows. And end system may or may not be
    DetNet transport layer aware or DetNet
    service layer aware.";
}

identity edge-node {
  base detnet-node-role;
  description
    "An instance of a DetNet relay node that
    includes either a DetNet service layer proxy
```

```
        function for DetNet service protection (e.g.
        the addition or removal of packet sequencing
        information) for one or more end systems, or
        starts or terminate congestion protection at
        the DetNet transport layer, analogous to a
        Label Edge Router (LER).";
    }

identity relay-node {
    base detnet-node-role;
    description
        "A DetNet node including a service layer
        function that interconnects different DetNet
        transport layer paths to provide service
        protection. A DetNet relay node can be a bridge,
        a router, a firewall, or any other system that
        participates in the DetNet service layer. It
        typically incorporates DetNet transport layer
        functions as well, in which case it is
        collocated with a transit node.";
}

identity transit-node {
    base detnet-node-role;
    description
        "A node operating at the DetNet transport layer,
        that utilizes link layer and/or network layer
        switching across multiple links and/or
        sub-networks to provide paths for DetNet
        service layer functions. Optionally provides
        congestion protection over those paths. An MPLS
        LSR is an example of a DetNet transit node.";
}

identity ttl-action {
    description
        "Base identity from which all TTL
        actions are derived.";
}

identity no-action {
    base "ttl-action";
    description
        "Do nothing regarding the TTL.";
}

identity copy-to-inner {
    base "ttl-action";
```



```
    description
      "Copy the TTL of the outer header
       to the inner header.";
  }

  identity decrease-and-copy-to-inner {
    base "ttl-action";
    description
      "Decrease TTL by one and copy the TTL
       to the inner header.";
  }

  typedef ttl-action-definition {
    type identityref {
      base "ttl-action";
    }
    description
      "TTL action definition.";
  }

  identity detnet-transport-layer {
    description
      "The layer that optionally provides congestion
       protection for DetNet flows over paths provided
       by the underlying network.";
  }

  identity detnet-service-layer {
    description
      "The layer at which service protection is
       provided, either packet sequencing, replication,
       and elimination or packet encoding";
  }

  typedef service-function-type {
    type enumeration {
      enum replication {
        description
          "A Packet Replication Function (PRF) replicates
           DetNet flow packets and forwards them to one or
           more next hops in the DetNet domain. The number
           of packet copies sent to each next hop is a
           DetNet flow specific parameter at the node doing
           the replication. PRF can be implemented by an
           edge node, a relay node, or an end system";
      }
      enum elimination {
        description
```

```
        "A Packet Elimination Function (PEF) eliminates
        duplicate copies of packets to prevent excess
        packets flooding the network or duplicate
        packets being sent out of the DetNet domain.
        PEF can be implemented by an edge node, a relay
        node, or an end system.";
    }
    enum ordering {
        description
        "A Packet Ordering Function (POF) re-orders
        packets within a DetNet flow that are received
        out of order. This function can be implemented
        by an edge node, a relay node, or an end system.";
    }
    enum elimination-ordering {
        description
        "A combination of PEF and POF that can be
        implemented by an edge node, a relay node, or
        an end system.";
    }
    enum elimination-replication {
        description
        "A combination of PEF and PRF that can be
        implemented by an edge node, a relay node, or
        an end system";
    }
    enum elimination-ordering-replicaiton {
        description
        "A combination of PEF, POF and PRF that can be
        implemented by an edge node, a relay node, or
        an end system";
    }
    }
    description
    "DetNet service function and function combination
    types.";
}

grouping detnet-transport-qos {
    description
    "DetNet transport tunnel QoS attributes.";
    uses traffic-specification;
}

grouping ipv4-header {
    description
    "The IPv4 header encapsulation information.";
    leaf src-ipv4-address {
```

```
    type inet:ipv4-address;
    mandatory true;
    description
        "The source IP address of the header.";
}
leaf dest-ipv4-address {
    type inet:ipv4-address;
    mandatory true;
    description
        "The destination IP address of the header.";
}
leaf protocol {
    type uint8;
    mandatory true;
    description
        "The protocol id of the header.";
}
leaf ttl {
    type uint8;
    description
        "The TTL of the header.";
}
leaf dscp {
    type uint8;
    description
        "The DSCP field of the header.";
}
}

grouping ipv6-header {
    description
        "The IPv6 header encapsulation information.";
    leaf src-ipv6-address {
        type inet:ipv6-address;
        mandatory true;
        description
            "The source IP address of the header.";
    }
    leaf dest-ipv6-address {
        type inet:ipv6-address;
        mandatory true;
        description
            "The destination IP address of the header.";
    }
    leaf next-header {
        type uint8;
        mandatory true;
        description
```

```
        "The next header of the IPv6 header.";
    }
    leaf traffic-class {
        type uint8;
        description
            "The traffic class value of the header.";
    }
    leaf flow-label {
        type inet:ipv6-flow-label;
        description
            "The flow label of the header.";
    }
    leaf hop-limit {
        type uint8 {
            range "1..255";
        }
        description
            "The hop limit of the header.";
    }
}

grouping mpls-header {
    description
        "The MPLS encapsulation header information.";
    list label-operations {
        key "label-oper-id";
        description
            "Label operations.";
        leaf label-oper-id {
            type uint32;
            description
                "An optional identifier that points
                 to a label operation.";
        }
        choice label-actions {
            description
                "Label action options.";
            case label-push {
                container label-push {
                    description
                        "Label push operation.";
                    leaf label {
                        type uint32;
                        mandatory true;
                        description
                            "The label to be pushed.";
                    }
                    leaf s-bit {
```

```

        type boolean;
        description
            "The s-bit of the label to be pushed.";
    }
    leaf tc-value {
        type uint8;
        description
            "The traffic class value of the label
            to be pushed.";
    }
    leaf ttl-value {
        type uint8;
        description
            "The TTL value of the label to be
            pushed.";
    }
}
}
case label-swap {
    container label-swap {
        description
            "Label swap operation.";
        leaf out-label {
            type uint32;
            mandatory true;
            description
                "The out MPLS label.";
        }
        leaf ttl-action {
            type ttl-action-definition;
            description
                "The label ttl actions:
                - No-action, or
                - Copy to inner label, or
                - Decrease (the in label) by 1 and
                copy to the out label.";
        }
    }
}
}
}
}
}
}

grouping mpls-detnet-header {
    description
        "The MPLS DetNet encapsulation header information.";
    leaf service-label {
        type uint32;
    }
}

```

```

        mandatory true;
        description
            "The service label.";
    }
    leaf control-word {
        type uint32;
        mandatory true;
        description
            "The control word of the DetNet header.";
    }
}

grouping transport-tunnel-encap{
    description
        "Defines the transport tunnel encapsulation
        header.";
    choice tunnel-type {
        description
            "Tunnel type includes: IPv4, IPv6, MPLS.";
        case IPv4 {
            description
                "IPv4 tunnel.";
            container ipv4-encapsulation {
                description
                    "IPv4 encapsulation.";
                uses ipv4-header;
            }
        }
        case IPv6 {
            description
                "IPv6 tunnel.";
            container ipv6-encapsulation {
                description
                    "IPv6 encapsulation.";
                uses ipv6-header;
            }
        }
        case MPLS {
            description
                "MPLS tunnel.";
            container mpls-encapsulation {
                description
                    "MPLS encapsulation.";
                uses mpls-header;
            }
        }
    }
}

```

```
grouping detnet-transport-instance {
  description
    "An instance of the DetNet transport layer, which
    depends on the specific data plane that is used
    as the underlay tunnel.";
  uses transport-tunnel-encap;
  uses detnet-transport-qos;
}

grouping ip-flow-identification {
  description
    "IP flow identification.";
  choice ip-flow-type {
    description
      "IP flow types: IPv4, IPv6.";
    case ipv4 {
      description
        "IPv4 flow identification.";
      leaf src-ipv4-address {
        type inet:ipv4-address;
        description
          "The source IP address of the header.";
      }
      leaf dest-ipv4-address {
        type inet:ipv4-address;
        description
          "The destination IP address of the header.";
      }
      leaf dscp {
        type uint8;
        description
          "The DSCP field of the header.";
      }
    }
    case ipv6 {
      description
        "IPv6 flow identification.";
      leaf src-ipv6-address {
        type inet:ipv6-address;
        description
          "The source IP address of the header.";
      }
      leaf dest-ipv6-address {
        type inet:ipv6-address;
        description
          "The destination IP address of the header.";
      }
      leaf traffic-class {
```

```
        type uint8;
        description
            "The traffic class value of the header.";
    }
    leaf flow-label {
        type inet:ipv6-flow-label;
        description
            "The flow label of the header.";
    }
}
}
leaf source-port {
    type inet:port-number;
    description
        "The source port number.";
}
leaf destination-port {
    type inet:port-number;
    description
        "The destination port number.";
}
leaf protocol {
    type uint8;
    description
        "The protocol id of the header.";
}
}

grouping l3-flow-identification {
    description
        "Layer 3 flow identification in the DetNet
        domain.";
    choice flow-type {
        description
            "L3 DetNet flow types: IP and MPLS.";
        case IP {
            description
                "IP (IPv4 or IPv6) flow identification.";
            uses ip-flow-identification;
        }
        case MPLS {
            description
                "MPLS flow identification.";
            leaf service-label {
                type uint32;
                mandatory true;
                description
                    "The service label.";
            }
        }
    }
}
```



```
    }  
  }  
}  
} //l3-flow-identification  
  
grouping in-segments {  
  description  
    "From a receiving node point of view, In-segments  
    are a set of instances of a DetNet flow at the  
    receiving node. This occurs when Packet Replication  
    Function (PRF) is enabled at an upstream node or  
    multiple flows map/aggregate to a single DetNet  
    flow.";  
  list in-segment {  
    key "in-segment-id";  
  
    description  
      "A list of in segments, there will be  
      multiple in-segments for a DetNet flow  
      when PRF and PEF enabled.";  
  
    leaf in-segment-id {  
      type uint32;  
      description  
        "in-segment identifier.";  
    }  
  
    uses l3-flow-identification;  
  
    leaf service-function {  
      type service-function-type;  
      description  
        "DetNet service function indication.";  
    }  
  }  
}  
  
grouping out-segments {  
  description  
    "Out-segments are a set of instances of  
    a DetNet flow, this occurs when implement  
    packet replication function, where an  
    in-segment of a DetNet flow is replicated  
    to multiple out-segments.";  
  
  list out-segment {  
    key "out-segment-id";  
    description
```

```
        "A list of segments, there will be multiple
        out-segments when perform PRF.";
    leaf out-segment-id {
        type uint32;
        description
            "The out-segment identifier";
    }

    container detnet-service-encapsulation {
        description
            "Only MPLS based DetNet defines DetNet
            service layer. The service encapsulation
            includes service label and control word.";
        uses mpls-detnet-header;
    }

    container detnet-transport-encapsulation {
        description
            "Each out-segment corresponds to a
            transport instance.";
        uses detnet-transport-instance;
    }
}

grouping detnet-service-instance{
    description
        "An end-2-end DetNet service is consisted of
        multiple segments. The concept of segment is
        similar to PW segment. For DetNet, since the
        existing of PREOF, there could be three cases:
        1 - One in-segment maps to multiple
            out-segments, when implement PRF;
        2 - Multiple in-segments map to one
            out-segment, when implement PEF;
        3 - Multiple in-segments map to multiple
            out-segments, when implement a combination
            of PEF and PRF.";

    leaf name {
        type string;
        description
            "The name of the service instance. This MUST
            be unique across all service instances in
            a given network device.";
    }
    container flow-rank{
        description
```

```
        "TBD based on the data plane solution.";
    }
    container service-rank{
        description
            "TBD based on the data plane solution.";
    }
    uses in-segments;
    uses out-segments;
}

grouping l2-flow-identification-at-uni {
    description
        "Layer 2 flow identification at UNI.";
    leaf source-mac-address {
        type yang:mac-address;
        description
            "The source MAC address used for
            flow identification.";
    }
    leaf destination-mac-address {
        type yang:mac-address;
        description
            "The destination MAC address used for
            flow identification.";
    }
}

leaf ethertype {
    type eth:ethertype;
    description
        "The Ethernet Type (or Length) value represented
        in the canonical order defined by IEEE 802.
        The canonical representation uses lowercase
        characters.";
    reference
        "IEEE 802-2014 Clause 9.2";
}

leaf vlan-id {
    type uint16 {
        range "1..4094";
    }
    description
        "Vlan Identifier used for L2 flow identification.";
}

container pcip {
    //Todo
    description
        "PCP used for L2 flow identification.";
```

```
    }  
  }  
  
  grouping l3-flow-identification-at-uni {  
    description  
      "Layer 3 flow identification at UNI.";  
    uses ip-flow-identification;  
  }  
  
  grouping traffic-specification {  
    description  
      "traffic-specification specifies how the Source  
      transmits packets for the flow. This is the  
      promise/request of the Source to the network.  
      The network uses this traffic specification  
      to allocate resources and adjust queue  
      parameters in network nodes.";  
    reference  
      "draft-ietf-detnet-flow-information-model";  
  
    leaf interval {  
      type uint32;  
      description  
        "The period of time in which the traffic  
        specification cannot be exceeded";  
    }  
    leaf max-packets-per-interval {  
      type uint32;  
      description  
        "The maximum number of packets that the  
        source will transmit in one Interval.";  
    }  
    leaf max-payload-size {  
      type uint32;  
      description  
        "The maximum payload size that the source  
        will transmit.";  
    }  
    leaf average-packets-per-interval {  
      type uint32;  
      description  
        "The average number of packets that the  
        source will transmit in one Interval";  
    }  
    leaf average-payload-size {  
      type uint32;  
      description  
        "The average payload size that the
```

```
        source will transmit.";
    }
}

grouping client-flows-at-uni {
    description
        "The attributes of the client flow at UNI. When
        flow aggregation is enabled at ingress, multiple
        client flows map to a DetNet service instance.";
    list client-flow {
        key "flow-id";
        description
            "A list of client flows.";
        leaf flow-id {
            type uint32;
            description
                "Flow identifier that is unique in a network
                device for client flow identification";
        }
        choice flow-type{
            description
                "Client flow type: layer 2 flow, layer 3
                flow.";
            case l2-flow-identification {
                description
                    "Ethernet flow identification.";
                uses l2-flow-identification-at-uni;
            }
            case l3-flow-identification {
                description
                    "layer 3 flow identification, including
                    IPv4,IPv6 and MPLS.";
                uses l3-flow-identification-at-uni;
            }
        }
        container traffic-specification {
            description
                "The traffic specification of the client flow.";
            uses traffic-specification;
        }
    }
}

grouping detnet-service-proxy-instance {
    description
        "Maps between App-flows and DetNet flows";
    uses client-flows-at-uni;
    container detnet-service-instance {
```

```
        description
            "A DetNet service instance.";
        uses detnet-service-instance;
    }
}

container detnet-flow{
    description
        "DetNet flow configuration and status reporting.";
    choice detnet-node-role{
        description
            "Depends on the role of a node to configure
            corresponding flow parameters.";

        case transit-node{
            description
                "DetNet flow configuration parameters for
                transit nodes.";
            container transit-node {
                description
                    "transit node container.";
                uses detnet-transport-qos;
            }
        }
        case relay-node{
            description
                "DetNet flow configuration parameters for
                relay nodes.";
            container relay-node {
                description
                    "Relay node container.";
                uses detnet-service-instance;
            }
        }
        case edge-node{
            description
                "DetNet flow configuration parameters for
                edge nodes.";
            container edge-node {
                description
                    "Edge node container.";
                uses detnet-service-proxy-instance;
            }
        }
        case end-station {
            description
                "DetNet flow configuration parameters for
                end stations.";
        }
    }
}
```

```
        container end-station {  
            description  
                "End station container.";  
            uses detnet-service-proxy-instance;  
        }  
    }  
}  
}  
}  
<CODE ENDS>
```

## 8. DetNet Configuration Model Classification

This section defines three classes of DetNet configuration model: fully distributed configuration model, fully centralized configuration model, hybrid configuration model, based on different network architectures, showing how configuration information exchanges between various entities in the network.

### 8.1. Fully Distributed Configuration Model

In a fully distributed configuration model, UNI information is transmitted over DetNet UNI protocol from the user side to the network side; then UNI information and network configuration information propagate in the network over distributed control plane protocol. For example:

- 1) IGP collects topology information and DetNet capabilities of network([I-D.geng-detnet-info-distribution]);
- 2) Control Plane of the Edge Node(Ingress) receives a flow establishment request from UNI and calculates a/some valid path(s);
- 3) Using RSVP-TE, Edge Node(Ingress) sends a PATH message with explicit route. After receiving the PATH message, the other Edge Node(Egress) sends a Resv message with distributed label and resource reservation request.

Current distributed control plane protocol, e.g., RSVP-TE[RFC3209], SRP[IEEE802.1Qcc], can only reserve bandwidth along the path, while the configuration of a fine-grained schedule, e.g., Time Aware Shaping(TAS) defined in [IEEE802.1Qbv], is not supported.

The fully distributed configuration model is not covered by this draft. It should be discussed in the future DetNet control plane work.

## 8.2. Fully Centralized Configuration Model

In the fully centralized configuration model, UNI information is transmitted from Centralized User Configuration (CUC) to Centralized Network Configuration(CNC). Configurations of routers for DetNet flows are performed by CNC with network management protocol. For example:

- 1) CNC collects topology information and DetNet capability of network through Netconf;
- 2) CNC receives a flow establishment request from UNI and calculates a/some valid path(s);
- 3) CNC configures the devices along the path for flow transmission.

## 8.3. Hybrid Configuration Model

In the hybrid configuration model, controller and control plane protocols work together to offer DetNet service, and there are a lot of possible combinations. For example:

- 1) CNC collects topology information and DetNet capability of network through IGP/BGP-LS;
- 2) CNC receives a flow establishment request from UNI and calculates a/some valid path(s);
- 3) Based on the calculation result, CNC distributes flow path information to Edge Node(Ingress) and other information(e.g. replication/elimination) to the relevant nodes.
- 4) Using RSVP-TE, Edge Node(Ingress) sends a PATH message with explicit route. After receiving the PATH message, the other Edge Node(Egress) sends a Resv message with distributed label and resource reservation request.

or

- 1) Controller collects topology information and DetNet capability of network through IGP/BGP-LS;
- 2) Control Plane of Edge Node(Ingress) receives a flow establishment request from UNI;
- 3) Edge Node(Ingress) sends the path establishment request to CNC through PCEP;



4) After Calculation, CNC sends back the path information of the flow to the Edge Node(Ingress) through PCEP;

5) Using RSVP-TE, Edge Node(Ingress) sends a PATH message with explicit route. After receiving the PATH message, the other Edge Node(Egress) sends a Resv message with distributed label and resource reservation request.

There are also other variations that can be included in the hybrid model. This draft can not cover all the control plane data needed in hybrid configuration models. Every solution has there own mechanism and corresponding parameters to make it work.

Editor's Note:

1. There are a lot of optional DetNet configuration models, and different scenario in different use case can choose one of them based on its conditions. Maybe next step of the work is to pick up one or more typical scenarios and give a practical solution.

2. [IEEE802.1Qcc] also defines three TSN configuration models: fully-centralized model, fully-distributed model, centralized Network / distributed User Model. This section defines the configuration model roughly the same, to keep the design of L2 and L3 in the same structure. Hybrid configuration model is slightly different from the 'centralized Network / distributed User Model'. The hybrid configuration model intends to contain more variations.

## 9. Open issues

There are some open issues that are still under discusion:

- o The Relationship with 802.1 TSN YANG models is TBD. TSN YANG models include: P802.1Qcw, which defines TSN YANG for Qbv, Qbu, and Qci, and P802.1CBcv, which defines YANG for 802.1CB. The possible problem here is how to avoid possible overlap among yang models defined in IETF and IEEE. A common YANG model may be defined in the future to shared by both TSN and DetNet. More discussion are needed here.
- o How to support DetNet OAM is TBD.

These issues will be resolved in the following versions of the draft.

## 10. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

## 11. Security Considerations

<TBD>

## 12. Acknowledgements

## 13. References

### 13.1. Normative References

- [I-D.finn-detnet-bounded-latency]  
Finn, N., Boudec, J., Mohammadpour, E., Varga, B., and J. Farkas, "DetNet Bounded Latency", draft-finn-detnet-bounded-latency-01 (work in progress), July 2018.
- [I-D.ietf-detnet-architecture]  
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-08 (work in progress), September 2018.
- [I-D.ietf-detnet-dp-sol-ip]  
Korhonen, J. and B. Varga, "DetNet IP Data Plane Encapsulation", draft-ietf-detnet-dp-sol-ip-00 (work in progress), July 2018.
- [I-D.ietf-detnet-dp-sol-mpls]  
Korhonen, J. and B. Varga, "DetNet MPLS Data Plane Encapsulation", draft-ietf-detnet-dp-sol-mpls-00 (work in progress), July 2018.
- [I-D.ietf-detnet-flow-information-model]  
Farkas, J., Varga, B., rodney.cummings@ni.com, r., Jiang, Y., and Y. Zha, "DetNet Flow Information Model", draft-ietf-detnet-flow-information-model-01 (work in progress), March 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

### 13.2. Informative References

- [I-D.geng-detnet-info-distribution]  
Geng, X., Chen, M., and Z. Li, "IGP-TE Extensions for DetNet Information Distribution", draft-geng-detnet-info-distribution-02 (work in progress), March 2018.
- [I-D.ietf-detnet-use-cases]  
Grossman, E., "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-19 (work in progress), October 2018.
- [I-D.ietf-teas-yang-te]  
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-16 (work in progress), July 2018.
- [I-D.ietf-teas-yang-te-topo]  
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-18 (work in progress), June 2018.
- [I-D.thubert-tsvwg-detnet-transport]  
Thubert, P., "A Transport Layer for Deterministic Networks", draft-thubert-tsvwg-detnet-transport-01 (work in progress), October 2017.
- [I-D.varga-detnet-service-model]  
Varga, B. and J. Farkas, "DetNet Service Model", draft-varga-detnet-service-model-02 (work in progress), May 2017.
- [IEEE802.1CB]  
"IEEE, "Frame Replication and Elimination for Reliability (IEEE Draft P802.1CB)", 2017, <<http://www.ieee802.org/1/files/private/cb-drafts/>>.", 2016.

- [IEEE802.1Q-2014]  
"IEEE, "IEEE Std 802.1Q Bridges and Bridged Networks",  
2014, <<http://ieeexplore.ieee.org/document/6991462/>>.",  
2014.
- [IEEE802.1Qbu]  
"IEEE, "IEEE Std 802.1Qbu Bridges and Bridged Networks -  
Amendment 26: Frame Preemption", 2016,  
<<http://ieeexplore.ieee.org/document/7553415/>>.", 2016.
- [IEEE802.1Qbv]  
"IEEE, "IEEE Std 802.1Qbu Bridges and Bridged Networks -  
Amendment 25: Enhancements for Scheduled Traffic", 2015,  
<<http://ieeexplore.ieee.org/document/7572858/>>.", 2016.
- [IEEE802.1Qcc]  
"IEEE, "Stream Reservation Protocol (SRP) Enhancements and  
Performance Improvements (IEEE Draft P802.1Qcc)", 2017,  
<<http://www.ieee802.org/1/files/private/cc-drafts/>>.".
- [IEEE802.1Qch]  
"IEEE, "Cyclic Queuing and Forwarding (IEEE Draft  
P802.1Qch)", 2017,  
<<http://www.ieee802.org/1/files/private/ch-drafts/>>.",  
2016.
- [IEEE802.1Qci]  
"IEEE, "Per-Stream Filtering and Policing (IEEE Draft  
P802.1Qci)", 2016,  
<<http://www.ieee802.org/1/files/private/ci-drafts/>>.",  
2016.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V.,  
and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP  
Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001,  
<<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S.  
Yasukawa, Ed., "Extensions to Resource Reservation  
Protocol - Traffic Engineering (RSVP-TE) for Point-to-  
Multipoint TE Label Switched Paths (LSPs)", RFC 4875,  
DOI 10.17487/RFC4875, May 2007,  
<<https://www.rfc-editor.org/info/rfc4875>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K.,  
and R. Wilton, "Network Management Datastore Architecture  
(NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018,  
<<https://www.rfc-editor.org/info/rfc8342>>.

Authors' Addresses

Xuesong Geng  
Huawei

Email: gengxuesong@huawei.com

Mach(Guoyi) Chen  
Huawei

Email: mach.chen@huawei.com

Zhenqiang Li  
China Mobile

Email: lizhenqiang@chinamobile.com

Reshad Rahman  
Cisco Systems

Email: rrahman@cisco.com

Interdomain Routing Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 6, 2018

X. Geng  
M. Chen  
Huawei  
Z. Li  
China Mobile  
March 05, 2018

IGP-TE Extensions for DetNet Information Distribution  
draft-geng-detnet-info-distribution-02

Abstract

There are requirements in diverse industries to establish multi-hop paths for characterized flows with bounded end-to-end latency and extremely low packet loss rate. Deterministic Networking (DetNet) can provide service satisfying the requirements.

This document describes extensions to IGP-TE, including OSPF-TE and ISIS-TE to distribute information of DetNet, which can be used for DetNet path computation/selection.

This document only covers the mechanisms by which DetNet information is distributed. The mechanisms for measuring, calculating or configuring DetNet capabilities, resources and other relevant parameters are out of the scope.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

#### Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	4
3. DetNet Extensions to OSPF TE . . . . .	4
3.1. Congestion Protection Method sub-TLV . . . . .	4
3.2. Maximum DetNet Reservable Bandwidth sub-TLV . . . . .	5
3.3. Available DetNet Bandwidth sub-TLV . . . . .	6
3.4. Min/Max Queuing Delay sub-TLV . . . . .	6
4. DetNet Extensions to ISIS TE . . . . .	7
4.1. Congestion Protection Method . . . . .	7
4.2. Maximum DetNet Reservable Bandwidth . . . . .	8
4.3. Available DetNet Bandwidth . . . . .	9
4.4. Min/Max Queuing Delay . . . . .	9
5. IANA Considerations . . . . .	10
5.1. Sub-TLVs for Link TLV . . . . .	10
5.2. Sub-TLVs for TLVs 22, 23, 141, 222, and 223 . . . . .	10
6. Security Considerations . . . . .	11
7. Acknowledgements . . . . .	11
8. References . . . . .	11
8.1. Normative References . . . . .	11
8.2. Informative References . . . . .	11
Authors' Addresses . . . . .	12

#### 1. Introduction

There are many use cases from diverse industries which have the need in common for deterministic service, for example: audio video production, industrial process control and mobile access networks. The requirements can be summarized as:

Deterministic minimum and maximum end-to-end latency from source to destination

Extremely low packet loss rate

Deterministic Networking (DetNet) can satisfy the requirements by the following techniques:

- o Congestion Protection by reserving data plane resources for DetNet flows in intermediate nodes along the path
- o Explicit Route that do not rapidly change with the network topology
- o Seamless Redundant which can distribute DetNet flow packets over multi paths to ensure delivery of each packet spite of the loss of a path

To make the above techniques work, it's necessary to know the capabilities (e.g., DetNet capable or not, which congestion protection algorithms are supported, etc.), resources (e.g, dedicated bandwidth for DetNet, buffers, etc.), performance (e.g., device/queue/link delay etc.) and other relevant information of each DetNet capable node. Then, a DetNet path computation element (e.g., PCE or ingress of a DetNet flow) can use these information to compute a path that satisfies the requirement of a specific DetNet flow. Specifically, according to the requirements stated in DetNet architecture, the information should include:

- o Whether a node is DetNet capable
- o Congestion protection methods supported by a DetNet capable node;
- o Dedicated bandwidth for DetNet flows;
- o Device and link delay;

Some of information (e.g., Link delay/loss ) can be distributed and collected through the Traffic Engineering (TE) metric extensions [RFC7471], [RFC7810].

This document defines extensions to OSPF and ISIS to distribute the above DetNet information that can not distributed by the existing protocols.



## 2. Terminology

All the DetNet related terminologies used in this document conform to the DetNet architecture [I-D.ietf-detnet-architecture].

## 3. DetNet Extensions to OSPF TE

This document defines new OSPF TE sub-TLVs for Link TLV to distribute the DetNet required information as stated in Section 1. These sub-TLVs includes:

Type	Length	Value
TBD1	4	Congestion Control Method
TBD2	4	Max DetNet Reservable Bandwidth
TBD3	4	Available DetNet Bandwidth
TBD4	8	Min/Max Queuing Delay

### 3.1. Congestion Protection Method sub-TLV

This Congestion Protection (CP) Method sub-TLV is used to advertise the DetNet flow congestion protection methods used in transit nodes. It may be required by some DetNet flows that all the transit nodes along the path SHOULD use the same congestion protection method. Some typical congestion protection methods are listed as below:

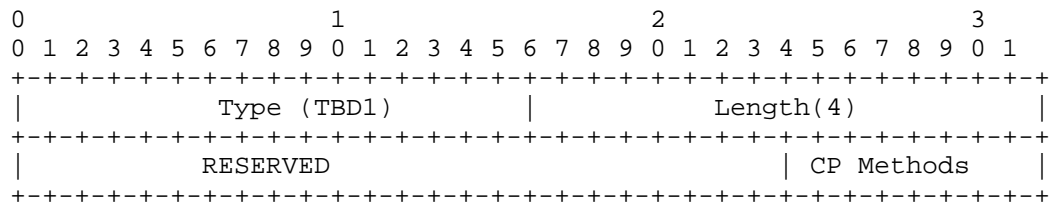
Time Aware Shaping [IEEE802.1Qbv]

Credit Based Shaper [IEEE802.1Q-2014]

Cyclic Queuing and Forwarding [IEEE802.1Qch]

Asynchronous Traffic Shaping [IEEE802.1Qcr]

The format of this sub-TLV is shown in the following diagram:



The Type field is 2 octets in length, and the value is TBD1.

The Length field is 2 octets in length and its value is 4.

The RESERVED field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

This Congestion Control Method field presents the congestion protection method used in the transit node.

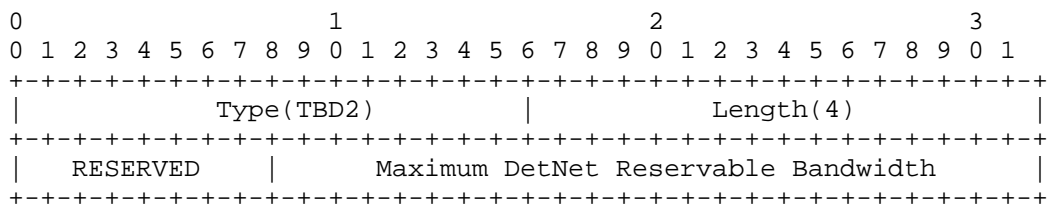
Five congestion protection methods are introduced in this document:

Value	Congestion Control Mechanisms
0	Reserved
1	Time Aware Shaper
2	Credit Based Shaper
3	Time Aware Shaper and Credit Based Shaper
4	Cyclic Queuing and Forwarding
5	Asynchronous Traffic Shaping
6-254	Unassigned
255	Reserved

### 3.2. Maximum DetNet Reservable Bandwidth sub-TLV

This sub-TLV specifies the maximum amount of bandwidth that is reserved for DetNet on this link. Note that this value SHOULD be smaller than the value of Maximum Reservable Bandwidth sub-TLV [RFC3630]. The value normally depends on the Congestion Protection Method and is user-configurable. In some particular Congestion Protection Method (e.g. Credit Based shaper in AVB), this value will affect the calculation of maximum queuing delay of the DetNet flow. The units are bytes per second.

The format of this sub-TLV is shown in the following diagram:



The Type field is 2 octets in length, and the value is TBD2.

The Length field is 2 octets in length and its value is 4.

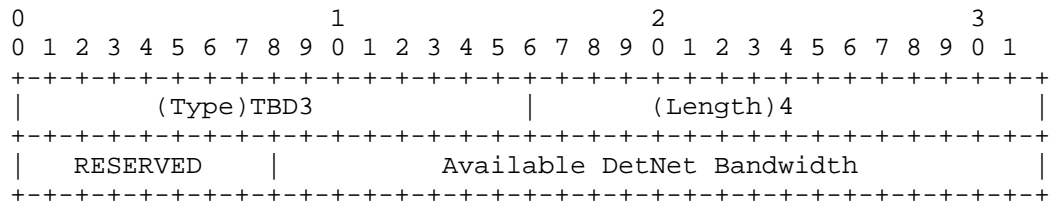
The RESERVED field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

This Maximum DetNet Reservable Bandwidth presents the maximum bandwidth that may be reserved for DetNet. The units are bytes per second.

### 3.3. Available DetNet Bandwidth sub-TLV

This sub-TLV specifies the available bandwidth that can be reserved for DetNet flow on this link for now. Considering that there is no generally accepted DetNet traffic classification, this value contains all the available DetNet Bandwidth from different DetNet traffic classes (if there is any), which differs from the Unreserved Bandwidth defined in [RFC3630].

The format of this sub-TLV is shown in the following diagram:



The Type field is 2 octets in length, and the value is TBD3.

The Length field is 2 octets in length and its value is 4.

The RESERVED field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

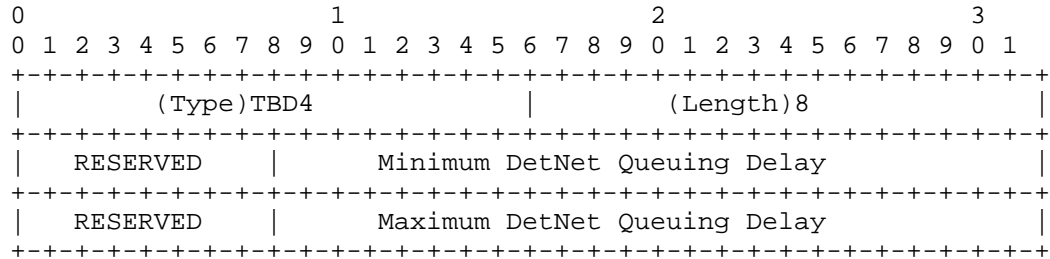
This Available DetNet Bandwidth field presents the available bandwidth for DetNet in this link. The units are bytes per second.

### 3.4. Min/Max Queuing Delay sub-TLV

[Editor Notes: more consideration and inputs are needed for these queue delays]

This sub-TLV advertises the minimum and maximum queuing delay values of specific DetNet flow in the link. Max/Min Unidirectional Link Delay Sub-TLV [RFC7471] excludes the queuing delay because of its instability. With the techniques used in DetNet, the queuing delay can be limited to a reasonable range, which means that the queuing delay bound is stable enough to be defined as a sub-TLV and advertised over the network.

The format of this sub-TLV is shown in the following diagram:



The Type field is 2 octets in length, and the value is TBD4.

The Length field is 2 octets in length and its value is 4.

The RESERVED field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

Minimum DetNet Queuing Delay is 24-bit field carrying minimum queuing delay value (in microseconds) encoded as an integer value. Implementations may also add this to the value of Min Delay Unidirectional Link Delay Sub-TLV [RFC7471] in order to advertise the minimum delay of this link. Min Queuing Delay can be the same with the Max Queuing Delay.

The RESERVED field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

Maximum DetNet Queuing Delay is 24-bit field carrying the maximum queuing delay value (in microseconds) encoded as an integer value. Implementations may also add this to the value of Max Delay Unidirectional Link Delay Sub-TLV [RFC7471] to order to advertise the maximum delay of this link.

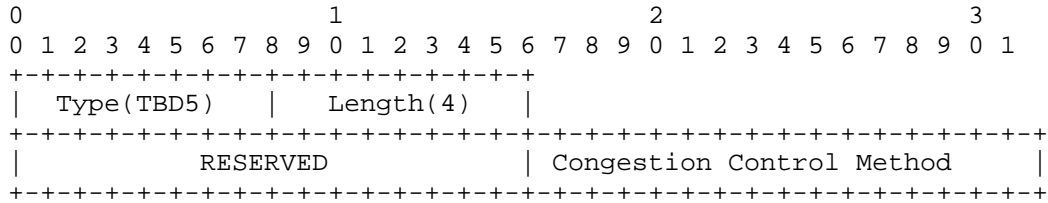
#### 4. DetNet Extensions to ISIS TE

This document defines new IS-IS TE sub-TLVs that can be announced in the TLVs 22, 23, 141, 222, and 223 in order to distribute DetNet information. The sub-TLV extensions below build on the ones provided in [RFC5305], [RFC5316] and [RFC7310].

##### 4.1. Congestion Protection Method

This Congestion Protection (CP) Method sub-TLV is used to advertise the DetNet flow congestion protection methods used in transit nodes. The reader can know more about this sub-TLV referring to section 3.1.

The format of this sub-TLV is shown in the following diagram:



The Type field is 1 octet in length, and the value is TBD5.

The Length field is 1 octet in length and its value is 4.

The RESERVED field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

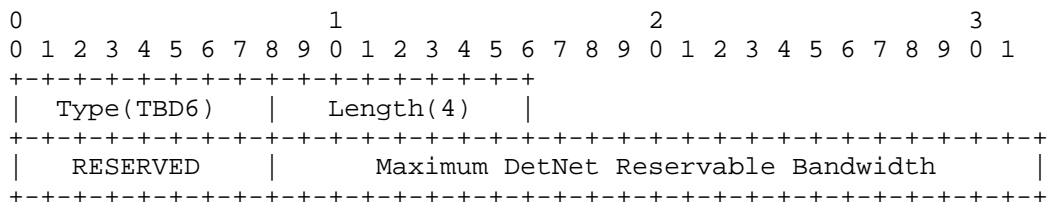
This Congestion Control Method field presents the congestion protection method used in the transit node.

Five congestion protection methods are introduced in this document:

#### 4.2. Maximum DetNet Reservable Bandwidth

This sub-TLV specifies the maximum amount of bandwidth that is reserved for DetNet on this link. Note that this value SHOULD be smaller than the value of Maximum Reservable Link Bandwidth [RFC5305]. The reader can know more about this sub-TLV referring to section 3.2.

The format of this sub-TLV is shown in the following diagram:



The Type field is 1 octet in length, and the value is TBD6.

The Length field is 1 octet in length and its value is 4.

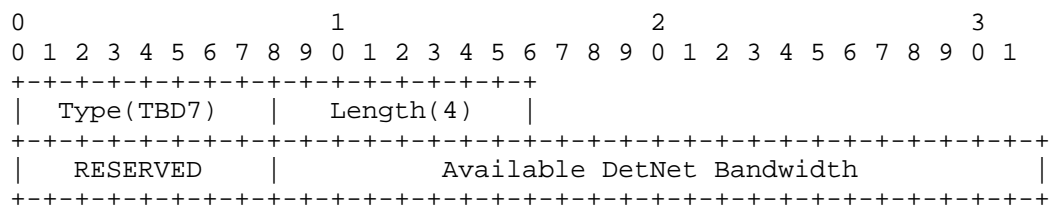
The RESERVED field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

This Maximum DetNet Reservable Bandwidth presents the maximum bandwidth that may be reserved for DetNet. The units are bytes per second.

### 4.3. Available DetNet Bandwidth

This sub-TLV specifies the available bandwidth that can be reserved for DetNet flow on this link for now. It is different from the Unreserved Bandwidth sub-TLV defined in [RFC5305] referring to section 3.3.

The format of this sub-TLV is shown in the following diagram:



The Type field is 1 octet in length, and the value is TBD7.

The Length field is 1 octet in length and its value is 4.

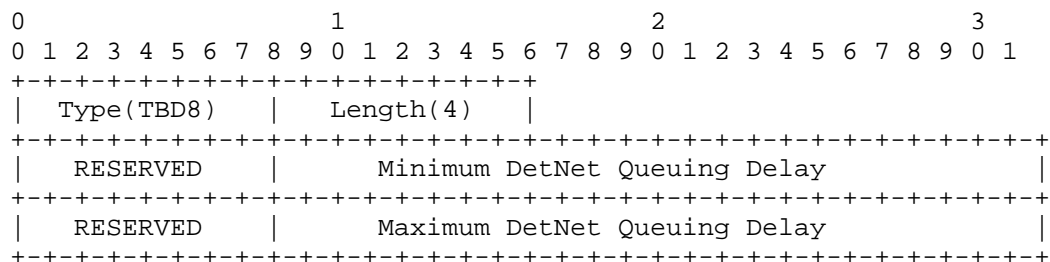
The RESERVED field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

This Available DetNet Bandwidth field presents the available bandwidth for DetNet in this link. The units are bytes per second.

#### 4.4. Min/Max Queuing Delay

The reader can know more about this sub-TLV referring to section 3.4.

The format of this sub-TLV is shown in the following diagram:



The Type field is 1 octet in length, and the value is TBD4.

The Length field is 1 octet in length and it's value is 4.

The RESERVED field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

Minimum DetNet Queuing Delay is 24-bit field carrying minimum queuing delay value (in microseconds) encoded as an integer value. Implementations may also add this to the value of Min Unidirectional Link Delay [RFC7810] in order to advertise the minimum delay of this link. Min Queuing Delay can be the same with the Max Queuing Delay.

The RESERVED field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

Maximum DetNet Queuing Delay is 24-bit field carrying the maximum queuing delay value (in microseconds) encoded as an integer value. Implementations may also add this to the value of Max Delay Unidirectional Link Delay Sub-TLV [RFC7810] to order to advertise the maximum delay of this link.

## 5. IANA Considerations

### 5.1. Sub-TLVs for Link TLV

IANA is requested to register the OSPF sub-TLVs defined in this document in the sub-TLVs for Link TLV registry.

Type	Description
-----	-----
TBD1	Congestion Protection Method
TBD2	Maximum DetNet Reservable Bandwidth
TBD3	Available DetNet Bandwidth
TBD4	Min/Max Queuing Delay

### 5.2. Sub-TLVs for TLVs 22, 23, 141, 222, and 223

IANA is requested to register the ISIS sub-TLVs defined in this document in the Sub-TLVs for TLVs 22, 23, 141, 222, and 223 registry.

Type	Description
-----	-----
TBD5	Congestion Protection Method
TBD6	Maximum DetNet Reservable Bandwidth
TBD7	Available DetNet Bandwidth
TBD8	Min/Max Queuing Delay

## 6. Security Considerations

This document does not introduce security issues beyond those discussed in [RFC7471] and [RFC7810].

## 7. Acknowledgements

## 8. References

### 8.1. Normative References

- [I-D.ietf-detnet-architecture]  
Finn, N., Thubert, P., Varga, B., and J. Farkas,  
"Deterministic Networking Architecture", draft-ietf-  
detnet-architecture-04 (work in progress), October 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S.  
 Previdi, "OSPF Traffic Engineering (TE) Metric  
Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015,  
<<https://www.rfc-editor.org/info/rfc7471>>.
- [RFC7810] Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and  
Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions",  
RFC 7810, DOI 10.17487/RFC7810, May 2016,  
<<https://www.rfc-editor.org/info/rfc7810>>.

### 8.2. Informative References

- [IEEE802.1Q-2014]  
"MAC Bridges and VLANs (IEEE 802.1Q-2014)", 2014.
- [IEEE802.1Qch]  
"Cyclic Queuing and Forwarding", 2016.
- [IEEE802.1Qcr]  
"Asynchronous Traffic Shaping", 2016.
- [IEEE802.1Qbv]  
"Enhancements for Scheduled Traffic", 2016.



- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, DOI 10.17487/RFC5316, December 2008, <<https://www.rfc-editor.org/info/rfc5316>>.
- [RFC7310] Lindsay, J. and H. Foerster, "RTP Payload Format for Standard apt-X and Enhanced apt-X Codecs", RFC 7310, DOI 10.17487/RFC7310, July 2014, <<https://www.rfc-editor.org/info/rfc7310>>.

## Authors' Addresses

Xuesong Geng  
Huawei

Email: [gengxuesong@huawei.com](mailto:gengxuesong@huawei.com)

Mach(Guoyi) Chen  
Huawei

Email: [mach.chen@huawei.com](mailto:mach.chen@huawei.com)

Zhenqiang  
China Mobile

Email: [lizhenqiang@chinamobile.com](mailto:lizhenqiang@chinamobile.com)

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: September 6, 2018

R. Huang  
T. Eckert  
N. Wei  
Huawei  
P. Thubert  
Cisco  
March 5, 2018

Encapsulation for BIER-TE  
draft-huang-bier-te-encapsulation-00

Abstract

This document proposes an enhanced encapsulation for BIER to support BIER, BIER-TE and a control word.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Terminology . . . . .	2
2. BIER-TE Encapsulation (normative) . . . . .	3
2.1. BT bit - Simultaneous support for BIER and BIER-TE . . . .	3
2.2. BIFT-ID . . . . .	3
2.3. Control Word and flows . . . . .	3
2.4. Header format & fields . . . . .	4
3. BIER-TE based resilience operations (informational) . . . . .	5
4. BitStringLength (BSL) considerations (informational) . . . . .	6
4.1. IPTV . . . . .	7
4.2. Multicast in L3VPN . . . . .	8
5. Acknowledgements . . . . .	9
6. Security Considerations . . . . .	9
7. IANA Considerations . . . . .	9
8. References . . . . .	9
8.1. Normative References . . . . .	9
8.2. Informative References . . . . .	10
Authors' Addresses . . . . .	10

## 1. Introduction

[BIER-TE-ARCH] specifies BIER-TE: Traffic Engineering for Bit Index Explicit Replication (BIER). It builds on the BIER architecture as described in RFC8279 [RFC8279], but uses every BitPosition of the BitString of a BIER-TE packet to indicate one or more adjacencies instead of a BFER as in BIER.

This document proposes an enhanced version of the MPLS and non-MPLS encapsulation for BIER packets to support both BIER and BIER-TE. It is based on RFC8296 [RFC8296].

This enhanced encapsulation also adds support for a control word in the header and discusses it. Finally, the document discusses BitStringLength (BSL) size requirements in implementations for informational reasons to help aide implementors to determine an appropriate BSL.

## 1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [RFC2119].

## 2. BIER-TE Encapsulation (normative)

### 2.1. BT bit - Simultaneous support for BIER and BIER-TE

This document supports mixed BIER and BIER-TE forwarding in a domain. Either or both of them may be used in a domain. The overall solution to support this depends on additional signaling such as existing BIER ISIS/BGP signaling. Architecturally, every SD SHOULD only use a single Type of BIER: BIER or BIER-TE. Note that this document will use the abbreviation BT to refer to the Bier Type.

In the presence of BIER and BIER-TE together in the network, there is always a risk of receiving a packet which is meant to be of one BT and processing it through a BIFT of the other BT. This can come from misconfiguration even in the face of signalling via IGP/BGP. The risk increases also when packets are generated modular from applications on PE or other sources and could use both BTs. To resolve this, the header includes a bit to indicate the BT. If the BT of a packet is inconsistent with the BT of the BIFT on the BFR, the BFR MUST NOT forward it. OAM actions MAY be triggered (subject to future work).

Note that the TTL field of the existing BIER packet header (or of IP packets) spends 7 bits on loop prevention. One bit for the BT is a comparably low cost to protect against a similar degree of problems.

Indicating the BT explicitly through a bit in the encapsulation is called the "explicit" option. Relying solely on the BT of the BIFTs is called "implicit" option. In this version of the document, we choose the explicit option for reasons outlined above.

### 2.2. BIFT-ID

Like in the original BIER header, the semantic of the BIFT-id of the header is that it is representing a <SI,SD,BSL> on the BFR receiving the packet. In the case of MPLS forwarding, the expectation is that the protocols to signal label ranges would be extended to also signal label ranges for the SD using BIER-TE. This is subject to the work of other documents. In the case of non-MPLS forwarding, no additional signaling may be necessary, and BIER and BIER-TE packets using the encapsulation of this document would equally use the BIFT-ID encoding as described in [BIER-non-MPLS].

### 2.3. Control Word and flows

This document adds a "control word" to the BIER packet header to allow that BIER or BIER-TE packets with this header could be used as

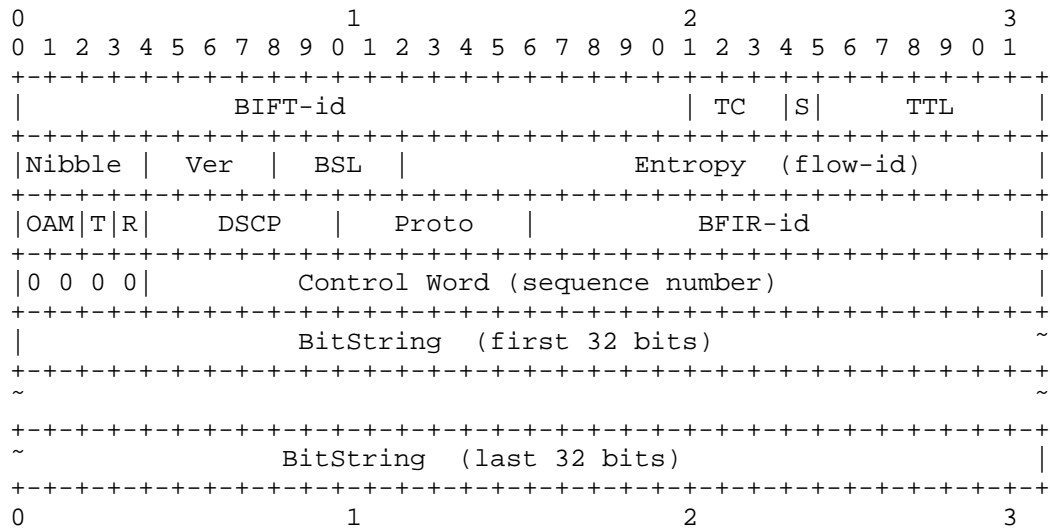
a DetNet Data Plane, independent of MPLS encapsulation, see [I-D.ietf-detnet-dp-sol], section 5.3 (in revision 01).

The control word provides a sequence number, therefore allowing to correct reordering and discover packet loss. The primary use though is resilient dual-path transmission of two copies of the same packet via disjoint paths. This is specifically a desirable use-case with BIER-TE because it allows the engineering of such disjoint paths. The flow to which the sequence number in the control word applies is <BFR-id,entropy>.

Note: The justification to carry a control word in the BIER encapsulation is similar to carrying the BFIR-ID in it. Initially, both could be seen as primarily required on BIER domain edge-nodes as part of the overlay using BIER, but not by BIER/BIER-TE itself directly. See Section 3 for more explanation how the resilience mechanism requiring the control word would work. Compared to the BFIR-ID, there is also the option to leverage it within BIER-TE itself. The details of that operations is subject to other specifications.

The authors think that the overhead of the control word is always acceptable for BIER-TE. For BIER, the use of this extended header version is optional, therefore BIER packets that need a control word would use this version of the header, those that do not need it would use version 1. If this overhead is considered to not be acceptable for all BIER-TE packets, the encoding could make those 32 bits optional through the use of one of the reserved bits or version numbers or by using a bit in the header to indicate whether the control word is present or not.

#### 2.4. Header format & fields



All header fields not described below are left unchanged from [BIER-TE-ARCH]

T: This 1-bit field identifies that the packet is to be forwarded as a BIER packet (0) or a BIER-TE packet(1).

Ver: The version of this header format is 2.

R: Reserved - unchanged (just reduced by one bit from version). Must be set to 0.

Entropy: Unchanged, but double-used as part of the flow-identifier together with the control word

Control Word: The control word in the terminology of MPLS pseudowires (where it originates from) is the full 32 bits. For detnet, the current target is 28 bits of sequence number and 4 bits 0 preceding it.

### 3. BIER-TE based resilience operations (informational)

This section discusses how resilience operations with the help of the sequence number in the control word of the header in this document can be operated as an overlay (BFIR-BFER) function but also points out that it could become an integral (optional) part of BIER-TE itself. This section is solely informational. The planned document

to describe the BIER-TE forwarding aspects of resilience operations is [I-D.thubert-bier-replication-elimination].

The BFIR determines - potentially with the help of a BIER-TE Controller Host (controller) - a bitstring that forms two disjoint DAGs (Directed Acyclic Graphs) through the BIER-TE domain towards the same set of BFER. In addition, an entropy value is decided (by BFIR and/or controller) and signalled to the BFER. The BFER can therefore set up "duplicate elimination state": The BFIR increments the sequence number with every packet of the flow it sends. The BFER assign packets to a flow by <bfir-id,entropy> and perform duplicate elimination on them.

Note that the bitstring as seen on the receiving BFER can provide additional diagnostics, for example the bits not reset by forwarding in BIER-TE give an indication about which path the BIER-TE packet was forwarded.

Instead of simply considering this protected mode of operations solely an end-to-end (BFIR/BFER) function, it could also be more flexibly embedded into BIER-TE itself, allow to provide in-BIER-TE segmented Packet Replication and (duplicate) Elimination Functions (PREF) definable by the bitstring of a BIER-TE packet. This could be achieved by adding to BIER-TE forwarding functions new adjacency types for duplication with sequence-number generation and duplicate-elimination. The ability to perform such processing as part of BIER-TE itself is the primary reason to ensure that all the necessary elements for such operations are part of the BIER-TE header itself.

#### 4. BitStringLength (BSL) considerations (informational)

BIER-TE uses each BitPosition to indicate the adjacencies instead of a BFER as in BIER, it therefore consumes more BitPositions than BIER. In BIER-TE, the number of adjacencies passed by one BIER-TE packet MUST be less than the value of BitString length (BSL). The BIER-TE architecture discusses a range of options to reduce the number of bits for intermediate hops through various BIER-TE adjacencies and how to use them.

The maximum supported BSL has a different impact in BIER-TE than it has in BIER: A smaller maximum supported BSL in BIER primarily leads to less replication efficiency: With a BSL of 256, BIER can be up to 256 more efficient than unicast (1 packet for 256 receivers). In BIER-TE, the BSL also limits the size of the topology towards BFER and the alternative paths that can explicitly be engineered to reach the BFER. One simple guess is that 50% of bits in a bitstring may be required for intermediate hops, therefore requiring about double the amount of bits as BIER - as the cost of being able to engineer paths.

So far, there is no comprehensive analysis of the number of required bits for specific scenarios in BIER-TE. The following subsections give two examples of such scenarios and how to use and save BIER-TE bits for intermedia hops.

#### 4.1. IPTV

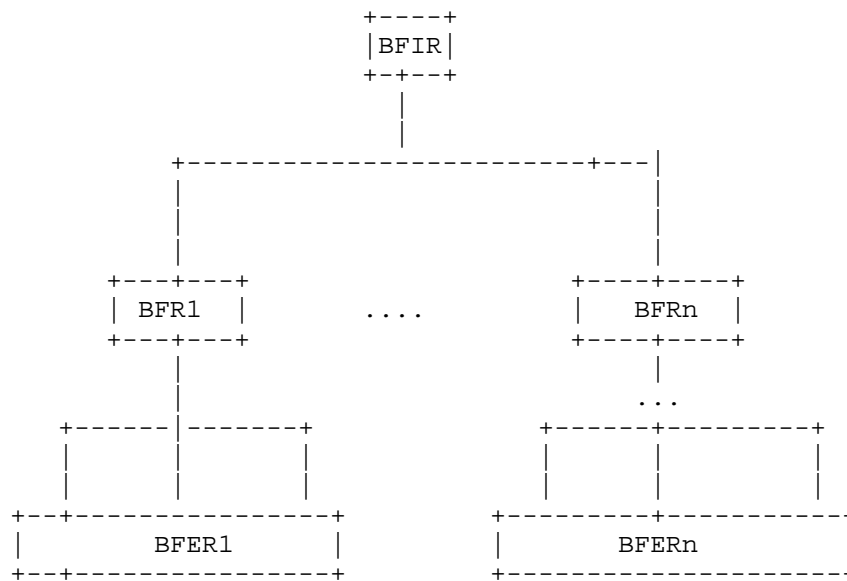
Multicast is widely used for IPTV services by simultaneously delivering a single stream of video to thousands of recipients. Currently, PIM is widely used to provide multicast capability usually from core router(CR) to provider edges (PEs). And the multicast tree is usually constructed in the hierarchical way. The end users using PIM/IGMP to request the multicast data. BIER can be well used in from CR to those PEs. The number of hops from multicast source (CR) which could be BFIRs, to the multicast receivers (PE) which can be regarded as BRERs is usually no more than 10. BIER-TE will be useful in the cases where different video channels can have different transport paths to achieve load balancing.

To save the bit consumption, 2 ways could be used:

1. Multiple BFRs and routes are required to receive the same data. These BFRs or links can share one bit.
2. Different bits can be used for pruning. But these bits can be reused in similar but different groups.

Considering an example illustrated as following:





BFR1 and BFRn, and other BFRs in between, can share one bit because they are receiving all the content and don't need pruning. From BFR1 to BFER1, there are 3 ways to reach BFER1. So these 3 ways can be assigned with different bits. But these 3 bits can be reused in the group from BFRn to BFERn, and other groups in between, which share the same topology as the group from BFR1 to BFER1.

BIER-TE can be well implemented using these 2 ways to save the bit consumption in IPTV networks with the similar topologies like the above example.

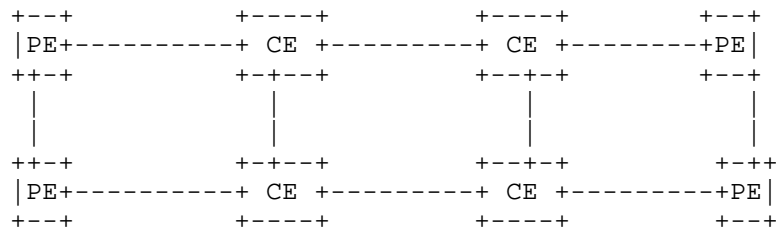
#### 4.2. Multicast in L3VPN

MVPN is a technology to deploy multicast service in an existing VPN or as part of a transport infrastructure. Multicast data is transmitted between private networks over a VPN infrastructure by encapsulating the original multicast packets. PE routers are connected to these private networks either containing receivers or senders.

There are several multicast applications widely using the MVPN deployment. For example, L3VPN multicast service offered by service providers to enterprise customers, and video transport applications for separation between different customers: One content provider may provide video wholesale service to another, or multiple content providers may share one network to transport video from headend. Especially the latter case, network SLAs should be guaranteed as the

original video content is precious. Thus, traffic engineering is required.

According to the current implementations, the scale of a MVPN network usually contains less than several hundreds of PEs, and hundreds of core routers which are connected in full mesh, like following figure illustrated.



In such a case, the ways in Section 7.5.2 of [BIER-TE-ARCH] can be used by regarding the CE area as the Core. Based on this, current BIER design is sufficient to be reused in BIER-TE.

## 5. Acknowledgements

TBD.

## 6. Security Considerations

The security considerations are in compliance with BIER-TE architecture [BIER-TE-ARCH] and BIER encapsulation RFC8296 [RFC8296]. And the content in this document does not create any other attacks or security concerns.

## 7. IANA Considerations

TBD.

## 8. References

### 8.1. Normative References

- [BIER-non-MPLS]  
Wijnands, I., Xu, X., and H. Bidgoli, "An Optional Encoding of the BIFT-id Field in the non-MPLS BIER Encapsulation", ID draft-wijnandsxu-bier-non-mpls-bift-encoding-01 (work in progress), August 2017.

## [BIER-TE-ARCH]

Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication BIER-TE", ID draft-ietf-bier-te-arch-00 (work in progress), January 2018.

## [I-D.thubert-bier-replication-elimination]

Thubert, P., Eckert, T., Brodard, Z., and H. Jiang, "BIER-TE extensions for Packet Replication and Elimination Function (PREF) and OAM", draft-thubert-bier-replication-elimination-03 (work in progress), March 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.

[RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.

## 8.2. Informative References

## [I-D.ietf-detnet-dp-sol]

Korhonen, J., Andersson, L., Jiang, Y., Finn, N., Varga, B., Farkas, J., Bernardos, C., Mizrahi, T., and L. Berger, "DetNet Data Plane Encapsulation", draft-ietf-detnet-dp-sol-01 (work in progress), January 2018.

## Authors' Addresses

Rachel Huang  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing 210012  
China

Email: [rachel.huang@huawei.com](mailto:rachel.huang@huawei.com)

Toerless Eckert  
Huawei USA - Futurewei Technologies Inc.  
2330 Central Expy  
Santa Clara 95050  
USA

Email: tte+ietf@cs.fau.de

Naiwen Wei  
Huawei

Email: weinaiwen@huawei.com

Pascal Thubert  
Cisco Systems  
Village d'Entreprises Green Side  
400, Avenue de Roumanille  
Batiment T3  
Biot - Sophia Antipolis 06410  
FRANCE

Phone: +33 4 97 23 26 34  
Email: pthubert@cisco.com

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: July 27, 2018

T. Eckert, Ed.  
Huawei  
G. Cauchie  
Bouygues Telecom  
W. Braun  
M. Menth  
University of Tuebingen  
January 23, 2018

Traffic Engineering for Bit Index Explicit Replication (BIER-TE)  
draft-ietf-bier-te-arch-00

Abstract

This document proposes an architecture for BIER-TE: Traffic Engineering for Bit Index Explicit Replication (BIER).

BIER-TE shares part of its architecture with BIER as described in [RFC8279]. It also proposes to share the packet format with BIER.

BIER-TE forwards and replicates packets like BIER based on a BitString in the packet header but it does not require an IGP. It does support traffic engineering by explicit hop-by-hop forwarding and loose hop forwarding of packets. It does support Fast ReRoute (FRR) for link and node protection and incremental deployment. Because BIER-TE like BIER operates without explicit in-network tree-building but also supports traffic engineering, it is more similar to SR than RSVP-TE.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 27, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Overview . . . . .	3
1.2. Requirements Language . . . . .	4
2. Layering . . . . .	4
2.1. The Multicast Flow Overlay . . . . .	5
2.2. The BIER-TE Controller Host . . . . .	5
2.2.1. Assignment of BitPositions to adjacencies of the network topology . . . . .	6
2.2.2. Changes in the network topology . . . . .	6
2.2.3. Set up per-multicast flow BIER-TE state . . . . .	6
2.2.4. Link/Node Failures and Recovery . . . . .	6
2.3. The BIER-TE Forwarding Layer . . . . .	7
2.4. The Routing Underlay . . . . .	7
3. BIER-TE Forwarding . . . . .	7
3.1. The Bit Index Forwarding Table (BIFT) . . . . .	7
3.2. Adjacency Types . . . . .	8
3.2.1. Forward Connected . . . . .	8
3.2.2. Forward Routed . . . . .	9
3.2.3. ECMP . . . . .	9
3.2.4. Local Decap . . . . .	9
3.3. Encapsulation considerations . . . . .	10
3.4. Basic BIER-TE Forwarding Example . . . . .	10
3.5. Forwarding comparison with BIER . . . . .	12
4. BIER-TE Controller Host BitPosition Assignments . . . . .	13
4.1. P2P Links . . . . .	13
4.2. BFER . . . . .	14
4.3. Leaf BFERs . . . . .	14
4.4. LANs . . . . .	14
4.5. Hub and Spoke . . . . .	15
4.6. Rings . . . . .	15
4.7. Equal Cost MultiPath (ECMP) . . . . .	16

4.8.	Routed adjacencies . . . . .	18
4.8.1.	Reducing BitPositions . . . . .	18
4.8.2.	Supporting nodes without BIER-TE . . . . .	18
5.	Avoiding loops and duplicates . . . . .	18
5.1.	Loops . . . . .	18
5.2.	Duplicates . . . . .	19
6.	BIER-TE Forwarding Pseudocode . . . . .	19
7.	Managing SI, subdomains and BFR-ids . . . . .	20
7.1.	Why SI and sub-domains . . . . .	21
7.2.	Bit assignment comparison BIER and BIER-TE . . . . .	22
7.3.	Using BFR-id with BIER-TE . . . . .	22
7.4.	Assigning BFR-ids for BIER-TE . . . . .	23
7.5.	Example bit allocations . . . . .	24
7.5.1.	With BIER . . . . .	24
7.5.2.	With BIER-TE . . . . .	25
7.6.	Summary . . . . .	26
8.	BIER-TE and Segment Routing . . . . .	26
9.	Security Considerations . . . . .	27
10.	IANA Considerations . . . . .	27
11.	Acknowledgements . . . . .	27
12.	Change log [RFC Editor: Please remove] . . . . .	27
13.	References . . . . .	29
	Authors' Addresses . . . . .	29

## 1. Introduction

### 1.1. Overview

This document specifies the architecture for BIER-TE: traffic engineering for Bit Index Explicit Replication BIER.

BIER-TE shares architecture and packet formats with BIER as described in [RFC8279].

BIER-TE forwards and replicates packets like BIER based on a BitString in the packet header but it does not require an IGP. It does support traffic engineering by explicit hop-by-hop forwarding and loose hop forwarding of packets. It does support incremental deployment and a Fast ReRoute (FRR) extension for link and node protection is given in [I-D.eckert-bier-te-frr]. Because BIER-TE like BIER operates without explicit in-network tree-building but also supports traffic engineering, it is more similar to Segment Routing (SR) than RSVP-TE.

The key differences over BIER are:

- o BIER-TE replaces in-network autonomous path calculation by explicit paths calculated offpath by the BIER-TE controller host.

- o In BIER-TE every BitPosition of the BitString of a BIER-TE packet indicates one or more adjacencies - instead of a BFER as in BIER.
- o BIER-TE in each BFR has no routing table but only a BIER-TE Forwarding Table (BIFT) indexed by SI:BitPosition and populated with only those adjacencies to which the BFR should replicate packets to.

BIER-TE headers use the same format as BIER headers.

BIER-TE forwarding does not require/use the BFIR-ID. The BFIR-ID can still be useful though for coordinated BFIR/BFER functions, such as the context for upstream assigned labels for MPLS payloads in MVPN over BIER-TE.

If the BIER-TE domain is also running BIER, then the BFIR-ID in BIER-TE packets can be set to the same BFIR-ID as used with BIER packets.

If the BIER-TE domain is not running full BIER or does not want to reduce the need to allocate bits in BIER bitstrings for BFIR-ID values, then the allocation of BFIR-ID values in BIER-TE packets can be done through other mechanisms outside the scope of this document, as long as this is appropriately agreed upon between all BFIR/BFER.

## 1.2. Requirements Language

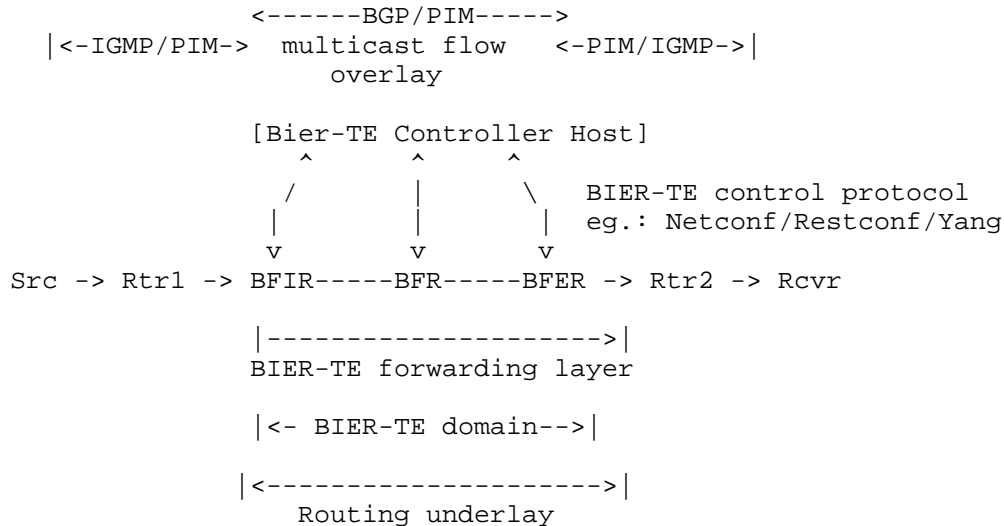
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 2. Layering

End to end BIER-TE operations consists of four components: The "Multicast Flow Overlay", the "BIER-TE Controller Host", the "Routing Underlay" and the "BIER-TE forwarding layer".



Picture 2: Layers of BIER-TE



## 2.1. The Multicast Flow Overlay

The Multicast Flow Overlay operates as in BIER. See [RFC8279]. Instead of interacting with the BIER layer, it interacts with the BIER-TE Controller Host

## 2.2. The BIER-TE Controller Host

The BIER-TE controller host is representing the control plane of BIER-TE. It communicates two sets of information with BFRs:

During bring-up or modifications of the network topology, the controller discovers the network topology, assigns BitPositions to adjacencies and signals the resulting mapping of BitPositions to adjacencies to each BFR connecting to the adjacency.

During day-to-day operations of the network, the controller signals to BFIRs what multicast flows are mapped to what BitStrings.

Communications between the BIER-TE controller host to BFRs is ideally via standardized protocols and data-models such as Netconf/Retconf/Yang. This is currently outside the scope of this document. Vendor-specific CLI on the BFRs is also a possible stopgap option (as in many other SDN solutions lacking definition of standardized data model).

For simplicity, the procedures of the BIER-TE controller host are described in this document as if it is a single, centralized automated entity, such as an SDN controller. It could equally be an operator setting up CLI on the BFRs. Distribution of the functions of the BIER-TE controller host is currently outside the scope of this document.

#### 2.2.1. Assignment of BitPositions to adjacencies of the network topology

The BIER-TE controller host tracks the BFR topology of the BIER-TE domain. It determines what adjacencies require BitPositions so that BIER-TE explicit paths can be built through them as desired by operator policy.

The controller then pushes the BitPositions/adjacencies to the BIFT of the BFRs, populating only those SI:BitPositions to the BIFT of each BFR to which that BFR should be able to send packets to - adjacencies connecting to this BFR.

#### 2.2.2. Changes in the network topology

If the network topology changes (not failure based) so that adjacencies that are assigned to BitPositions are no longer needed, the controller can re-use those BitPositions for new adjacencies. First, these BitPositions need to be removed from any BFIR flow state and BFR BIFT state, then they can be repopulated, first into BIFT and then into the BFIR.

#### 2.2.3. Set up per-multicast flow BIER-TE state

The BIER-TE controller host tracks the multicast flow overlay to determine what multicast flow needs to be sent by a BFIR to which set of BFER. It calculates the desired distribution tree across the BIER-TE domain based on algorithms outside the scope of this document (eg.: CSFP, Steiner Tree,...). It then pushes the calculated BitString into the BFIR.

#### 2.2.4. Link/Node Failures and Recovery

When link or nodes fail or recover in the topology, BIER-TE can quickly respond with the optional FRR procedures described in [I-D.eckert-bier-te-frr]. It can also more slowly react by recalculating the BitStrings of affected multicast flows. This reaction is slower than the FRR procedure because the controller needs to receive link/node up/down indications, recalculate the desired BitStrings and push them down into the BFIRs. With FRR, this

is all performed locally on a BFR receiving the adjacency up/down notification.

### 2.3. The BIER-TE Forwarding Layer

When the BIER-TE Forwarding Layer receives a packet, it simply looks up the BitPositions that are set in the BitString of the packet in the Bit Index Forwarding Table (BIFT) that was populated by the BIER-TE controller host. For every BP that is set in the BitString, and that has one or more adjacencies in the BIFT, a copy is made according to the type of adjacencies for that BP in the BIFT. Before sending any copy, the BFR resets all BitPositions in the BitString of the packet to which it can create a copy. This is done to inhibit that packets can loop.

### 2.4. The Routing Underlay

BIER-TE is sending BIER packets to directly connected BIER-TE neighbors as L2 (unicasted) BIER packets without requiring a routing underlay. BIER-TE forwarding uses the Routing underlay for forward\_routed adjacencies which copy BIER-TE packets to not-directly-connected BFRs (see below for adjacency definitions).

If the BFR intends to support FRR for BIER-TE, then the BIER-TE forwarding plane needs to receive fast adjacency up/down notifications: Link up/down or neighbor up/down, eg.: from BFD. Providing these notifications is considered to be part of the routing underlay in this document.

## 3. BIER-TE Forwarding

### 3.1. The Bit Index Forwarding Table (BIFT)

The Bit Index Forwarding Table (BIFT) exists in every BFR. For every subdomain in use, it is a table indexed by SI:BitPosition and is populated by the BIER-TE control plane. Each index can be empty or contain a list of one or more adjacencies.

BIER-TE can support multiple subdomains like BIER. Each one with a separate BIFT

In the BIER architecture, indices into the BIFT are explained to be both BFR-id and SI:BitString (BitPosition). This is because there is a 1:1 relationship between BFR-id and SI:BitString - every bit in every SI is/can be assigned to a BFIR/BFER. In BIER-TE there are more bits used in each BitString than there are BFIR/BFER assigned to the bitstring. This is because of the bits required to express the (traffic engineered) path through the topology. The BIER-TE

forwarding definitions do therefore not use the term BFR-id at all. Instead, BFR-ids are only used as required by routing underlay, flow overlay of BIER headers. Please refer to Section 7 for explanations how to deal with SI, subdomains and BFR-id in BIER-TE.

Index: SI:BitPosition	Adjacencies: <empty> or one or more per entry
0:1	forward_connected(interface,neighbor,DNR)
0:2	forward_connected(interface,neighbor,DNR) forward_connected(interface,neighbor,DNR)
0:3	local_decap([VRF])
0:4	forward_routed([VRF,l3-neighbor])
0:5	<empty>
0:6	ECMP({adjacency1,...adjacencyN}, seed)
...	
BitStringLength	...

Bit Index Forwarding Table

The BIFT is programmed into the data plane of BFRs by the BIER-TE controller host and used to forward packets, according to the rules specified in the BIER-TE Forwarding Procedures.

Adjacencies for the same BP when populated in more than one BFR by the controller do not have to have the same adjacencies. This is up to the controller. BPs for p2p links are one case (see below).

### 3.2. Adjacency Types

#### 3.2.1. Forward Connected

A "forward\_connected" adjacency is towards a directly connected BFR neighbor using an interface address of that BFR on the connecting interface. A forward\_connected adjacency does not route packets but only L2 forwards them to the neighbor.

Packets sent to an adjacency with "DoNotReset" (DNR) set in the BIFT will not have the BitPosition for that adjacency reset when the BFR creates a copy for it. The BitPosition will still be reset for

copies of the packet made towards other adjacencies. The can be used for example in ring topologies as explained below.

### 3.2.2. Forward Routed

A "forward\_routed" adjacency is an adjacency towards a BFR that is not a forward\_connected adjacency: towards a loopback address of a BFR or towards an interface address that is non-directly connected. Forward\_routed packets are forwarded via the Routing Underlay.

If the Routing Underlay has multiple paths for a forward\_routed adjacency, it will perform ECMP independent of BIER-TE for packets forwarded across a forward\_routed adjacency.

If the Routing Underlay has FRR, it will perform FRR independent of BIER-TE for packets forwarded across a forward\_routed adjacency.

### 3.2.3. ECMP

The ECMP mechanisms in BIER are tied to the BIER BIFT and are are therefore not directly useable with BIER-TE. The following procedures describe ECMP for BIER-TE that we consider to be lightweight but also well manageable. It leverages the existing entropy parameter in the BIER header to keep packets of the flows on the same path and it introduces a "seed" parameter to allow engineering traffic to be polarized or randomized across multiple hops.

An "Equal Cost Multipath" (ECMP) adjacency has a list of two or more adjacencies included in it. It copies the BIER-TE to one of those adjacencies based on the ECMP hash calculation. The BIER-TE ECMP hash algorithm must select the same adjacency from that list for all packets with the same "entropy" value in the BIER-TE header if the same number of adjacencies and same seed are given as parameters. Further use of the seed parameter is explained below.

### 3.2.4. Local Decap

A "local\_decap" adjacency passes a copy of the payload of the BIER-TE packet to the packets NextProto within the BFR (IPv4/IPv6, Ethernet,...). A local\_decap adjacency turns the BFR into a BFER for matching packets. Local\_decap adjacencies require the BFER to support routing or switching for NextProto to determine how to further process the packet.

### 3.3. Encapsulation considerations

Specifications for BIER-TE encapsulation are outside the scope of this document. This section gives explanations and guidelines.

Because a BFR needs to interpret the BitString of a BIER-TE packet differently from a BIER packet, it is necessary to distinguish BIER from BIER-TE packets. This is subject to definitions in BIER encapsulation specifications.

MPLS encapsulation [RFC8296] for example assigns one label by which BFRs recognizes BIER packets for every (SI,subdomain) combination. If it is desirable that every subdomain can forward only BIER or BIER-TE packets, then the label allocation could stay the same, and only the forwarding model (BIER/BIER-TE) would have to be defined per subdomain. If it is desirable to support both BIER and BIER-TE forwarding in the same subdomain, then additional labels would need to be assigned for BIER-TE forwarding.

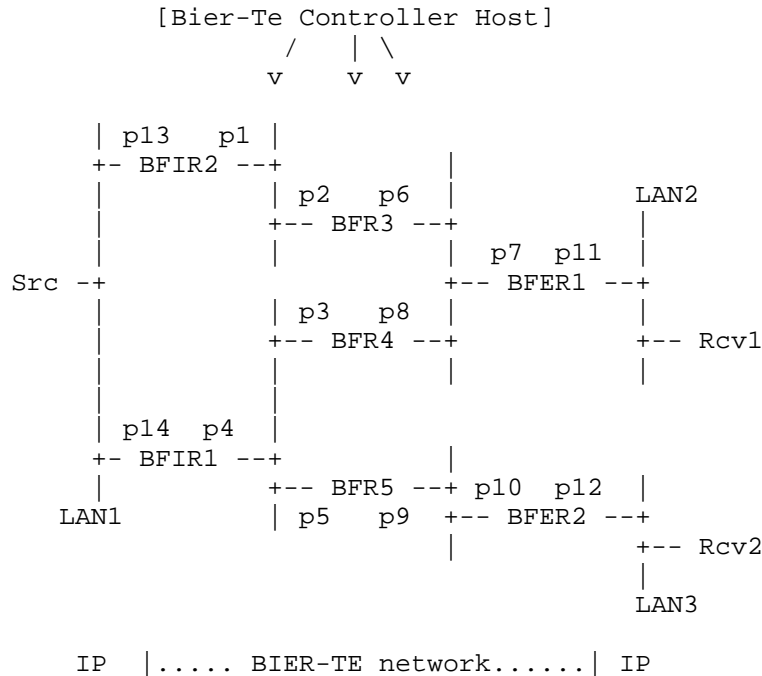
"forward\_routed" requires an encapsulation permitting to unicast BIER-TE packets to a specific interface address on a target BFR. With MPLS encapsulation, this can simply be done via a label stack with that addresses label as the top label - followed by the label assigned to (SI,subdomain) - and if necessary (see above) BIER-TE. With non-MPLS encapsulation, some form of IP tunneling (IP in IP, LISP, GRE) would be required.

The encapsulation used for "forward\_routed" adjacencies can equally support existing advanced adjacency information such as "loose source routes" via eg: MPLS label stacks or appropriate header extensions (eg: for IPv6).

### 3.4. Basic BIER-TE Forwarding Example

Step by step example of basic BIER-TE forwarding. This does not use ECMP or forward\_routed adjacencies nor does it try to minimize the number of required BitPositions for the topology.

Picture 1: Forwarding Example



pXX indicate the BitPositions number assigned by the BIER-TE controller host to adjacencies in the BIER-TE topology. For example, p9 is the adjacency towards BFR9 on the LAN connecting to BFER2.

```

BIFT BFIR2:
  p13: local_decap()
  p2: forward_connected(BFR3)

BIFT BFR3:
  p1: forward_connected(BFIR2)
  p7: forward_connected(BFER1)
  p8: forward_connected(BFR4)

BIFT BFER1:
  p11: local_decap()
  p6: forward_connected(BFR3)
  p8: forward_connected(BFR4)
  
```

...and so on.

Traffic needs to flow from BFIR2 towards Rcv1, Rcv2. The controller determines it wants it to pass across the following paths:

```
        -> BFER1 -----> Rcv1
BFIR2 -> BFR3
        -> BFR4 -> BFR5 -> BFER2 -> Rcv2
```

These paths equal to the following BitString: p2, p5, p7, p8, p10, p11, p12.

This BitString is set up in BFIR2. Multicast packets arriving at BFIR2 from Src are assigned this BitString.

BFIR2 forwards based on that BitString. It has p2 and p13 populated. Only p13 is in BitString which has an adjacency towards BFR3. BFIR2 resets p2 in BitString and sends a copy towards BFR2.

BFR3 sees a BitString of p5,p7,p8,p10,p11,p12. It is only interested in p1,p7,p8. It creates a copy of the packet to BFER1 (due to p7) and one to BFR4 (due to p8). It resets p7, p8 before sending.

BFER1 sees a BitString of p5,p10,p11,p12. It is only interested in p6,p7,p8,p11 and therefore considers only p11. p11 is a "local\_decap" adjacency installed by the BIER-TE controller host because BFER1 should pass packets to IP multicast. The local\_decap adjacency instructs BFER1 to create a copy, decapsulate it from the BIER header and pass it on to the NextProtocol, in this example IP multicast. IP multicast will then forward the packet out to LAN2 because it did receive PIM or IGMP joins on LAN2 for the traffic.

Further processing of the packet in BFR4, BFR5 and BFER2 accordingly.

### 3.5. Forwarding comparison with BIER

Forwarding of BIER-TE is designed to allow common forwarding hardware with BIER. Like BIER, the core of BIER-TE forwarding are BIFTs with bitstring size number of entries: One for each bit of the bitstring in the processed packet (consider that 256 is the most common size).

When a packet is received, the BIFT to process needs to be selected. This is based on SI and subdomain like in BIER. How SI and subdomain are indicated is subject to the BIER-TE encapsulation, but not BIER-T itself. It is expected that the mechanisms for encapsulation will be very similar if not the same to BIER, but this is subject to followup work.

There are some key difference between the BIFT in BIER and BIER-TE:

In BIER-TE, each entry in the BIFT can have a list of 0 or more adjacencies. A separate copy of the packet is made for each adjacency. In BIER, each BIFT entry has at most one adjacency (BFR-



NBR). In BIER, different bits can not be processed independently directly: Only one packet copy is to be sent for all bits in the packet with the same adjacency, which is why the forwarding procedure specifies how to sequentially identify those bits and avoid duplication. In BIER-TE there are no mutual dependencies between bit adjacencies, so all bits of a BIER-TE bitstring could be processed independently in parallel.

In BIER the BIFT has adjacencies for all BFR-ids assigned to BFER and reachable in the IGP. In BIER-TE the BIFT only has adjacencies for bits that are adjacent hops - intermediate or BFER. In forwarding, this can be treated via the same lookup logic except that in BIER-TE there is no step modifying the original packet and the packet copy bitstring with the FBM. Instead, all the bits locally processed are reset in the original packet before looking up bits in the BIFT (~MyBitsOfInterest). Only for an adjacency with the "DNR" (Do Not Reset) bit set would the bit in the bitstring not be set again as part of processing of the adjacency.

In summary, implementations of BIER forwarding that are to be extended to also support BIER-TE forwarding primarily need to consider how they can ensure that individual bit lookups can result in a sequence of more than one copy to be made (as opposed to one in BIER), and they need to see that they can accordingly reset bits in the bitstring differently for BIER (per-packet) vs. BIER-TE (per-packet-copy).

#### 4. BIER-TE Controller Host BitPosition Assignments

This section describes how the BIER-TE controller host can use the different BIER-TE adjacency types to define the BitPositions of a BIER-TE domain.

Because the size of the BitString is limiting the size of the BIER-TE domain, many of the options described exist to support larger topologies with fewer BitPositions (4.1, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8).

##### 4.1. P2P Links

Each P2p link in the BIER-TE domain is assigned one unique BitPosition with a forward\_connected adjacency pointing to the neighbor on the p2p link.

#### 4.2. BFER

Every BFER is given a unique BitPosition with a local\_decap adjacency.

#### 4.3. Leaf BFERs

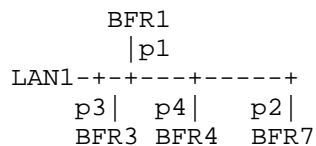
Leaf BFERs are BFERs where incoming BIER-TE packets never need to be forwarded to another BFR but are only sent to the BFER to exit the BIER-TE domain. For example, in networks where PEs are spokes connected to P routers, those PEs are Leaf BFERs unless there is a U-turn between two PEs.

All leaf-BFER in a BIER-TE domain can share a single BitPosition. This is possible because the BitPosition for the adjacency to reach the BFER can be used to distinguish whether or not packets should reach the BFER.

This optimization will not work if an upstream interface of the BFER is using a BitPosition optimized as described in the following two sections (LAN, Hub and Spoke).

#### 4.4. LANs

In a LAN, the adjacency to each neighboring BFR on the LAN is given a unique BitPosition. The adjacency of this BitPosition is a forward\_connected adjacency towards the BFR and this BitPosition is populated into the BIFT of all the other BFRs on that LAN.



If Bandwidth on the LAN is not an issue and most BIER-TE traffic should be copied to all neighbors on a LAN, then BitPositions can be saved by assigning just a single BitPosition to the LAN and populating the BitPosition of the BIFTs of each BFRs on the LAN with a list of forward\_connected adjacencies to all other neighbors on the LAN.

This optimization does not work in the face of BFRs redundantly connected to more than one LANs with this optimization because these BFRs would receive duplicates and forward those duplicates into the opposite LANs. Adjacencies of such BFRs into their LANs still need a separate BitPosition.

#### 4.5. Hub and Spoke

In a setup with a hub and multiple spokes connected via separate p2p links to the hub, all p2p links can share the same BitPosition. The BitPosition on the hubs BIFT is set up with a list of forward\_connected adjacencies, one for each Spoke.

This option is similar to the BitPosition optimization in LANs: Redundantly connected spokes need their own BitPositions.

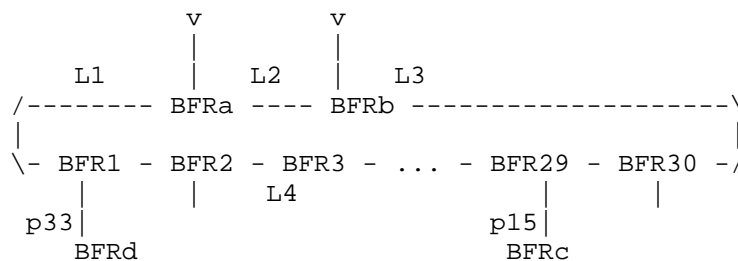
#### 4.6. Rings

In L3 rings, instead of assigning a single BitPosition for every p2p link in the ring, it is possible to save BitPositions by setting the "Do Not Reset" (DNR) flag on forward\_connected adjacencies.

For the rings shown in the following picture, a single BitPosition will suffice to forward traffic entering the ring at BFRa or BFRb all the way up to BFR1:

On BFRa, BFRb, BFR30,... BFR3, the BitPosition is populated with a forward\_connected adjacency pointing to the clockwise neighbor on the ring and with DNR set. On BFR2, the adjacency also points to the clockwise neighbor BFR1, but without DNR set.

Handling DNR this way ensures that copies forwarded from any BFR in the ring to a BFR outside the ring will not have the ring BitPosition set, therefore minimizing the chance to create loops.



Note that this example only permits for packets to enter the ring at BFRa and BFRb, and that packets will always travel clockwise. If packets should be allowed to enter the ring at any ring BFR, then one would have to use two ring BitPositions. One for clockwise, one for counterclockwise.

Both would be set up to stop rotating on the same link, eg: L1. When the ingress ring BFR creates the clockwise copy, it will reset the counterclockwise BitPosition because the DNR bit only applies to the

bit for which the replication is done. Likewise for the clockwise BitPosition for the counterclockwise copy. In result, the ring ingress BFR will send a copy in both directions, serving BFRs on either side of the ring up to L1.

#### 4.7. Equal Cost MultiPath (ECMP)

The ECMP adjacency allows to use just one BP per link bundle between two BFRs instead of one BP for each p2p member link of that link bundle. In the following picture, one BP is used across L1,L2,L3 and BFR1/BFR2 have for the BP

```

      --L1-----
BFR1 --L2----- BFR2
      --L3-----

```

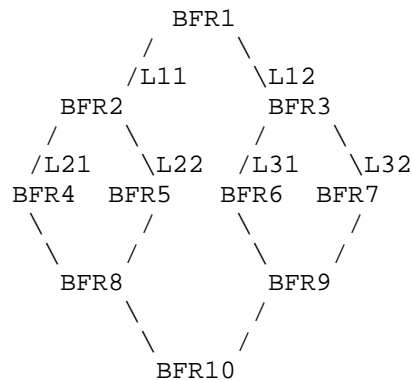
BIFT entry in BFR1:

Index	Adjacencies
0:6	ECMP({L1-to-BFR2,L2-to-BFR2,L3-to-BFR2}, seed)

BIFT entry in BFR2:

Index	Adjacencies
0:6	ECMP({L1-to-BFR1,L2-to-BFR1,L3-to-BFR1}, seed)

In the following example, all traffic from BFR1 towards BFR10 is intended to be ECMP load split equally across the topology. This example is not mean as a likely setup, but to illustrate that ECMP can be used to share BPs not only across link bundles, and it explains the use of the seed parameter.



BIFT entry in BFR1:

0:6	ECMP({L11-to-BFR2,L12-to-BFR3}, seed)
-----	---------------------------------------

BIFT entry in BFR2:

0:6	ECMP({L21-to-BFR4,L22-to-BFR5}, seed)
-----	---------------------------------------

BIFT entry in BFR3:

0:6	ECMP({L31-to-BFR6,L32-to-BFR7}, seed)
-----	---------------------------------------

With the setup of ECMP in above topology, traffic would not be equally load-split. Instead, links L22 and L31 would see no traffic at all: BFR2 will only see traffic from BFR1 for which the ECMP hash in BFR1 selected the first adjacency in a list of 2 adjacencies: link L11-to-BFR2. When forwarding in BFR2 performs again an ECMP with two adjacencies on that subset of traffic, then it will again select the first of its two adjacencies to it: L21-to-BFR4. And therefore L22 and BFR5 sees no traffic.

To resolve this issue, the ECMP adjacency on BFR1 simply needs to be set up with a different seed than the ECMP adjacencies on BFR2/BFR3

This issue is called polarization. It depends on the ECMP hash. It is possible to build ECMP that does not have polarization, for example by taking entropy from the actual adjacency members into account, but that can make it harder to achieve evenly balanced load-splitting on all BFR without making the ECMP hash algorithm potentially too complex for fast forwarding in the BFRs.

## 4.8. Routed adjacencies

### 4.8.1. Reducing BitPositions

Routed adjacencies can reduce the number of BitPositions required when the traffic engineering requirement is not hop-by-hop explicit path selection, but loose-hop selection.

```

      .....
BFR1--... Redundant ...--L1-- BFR2... Redundant ...--
      \--... Network   ...--L2--/   ... Network   ...--
BFR4--... Segment 1 ...--L3-- BFR3... Segment 2 ...--
      .....

```

Assume the requirement in above network is to explicitly engineer paths such that specific traffic flows are passed from segment 1 to segment 2 via link L1 (or via L2 or via L3).

To achieve this, BFR1 and BFR4 are set up with a forward\_routed adjacency BitPosition towards an address of BFR2 on link L1 (or link L2 BFR3 via L3).

For paths to be engineered through a specific node BFR2 (or BFR3), BFR1 and BFR4 are set up with a forward\_routed adjacency BitPosition towards a loopback address of BFR2 (or BFR3).

### 4.8.2. Supporting nodes without BIER-TE

Routed adjacencies also enable incremental deployment of BIER-TE. Only the nodes through which BIER-TE traffic needs to be steered - with or without replication - need to support BIER-TE. Where they are not directly connected to each other, forward\_routed adjacencies are used to pass over non BIER-TE enabled nodes.

## 5. Avoiding loops and duplicates

### 5.1. Loops

Whenever BIER-TE creates a copy of a packet, the BitString of that copy will have all BitPositions cleared that are associated with adjacencies in the BFR. This inhibits looping of packets. The only exception are adjacencies with DNR set.

With DNR set, looping can happen. Consider in the ring picture that link L4 from BFR3 is plugged into the L1 interface of BFRa. This creates a loop where the rings clockwise BitPosition is never reset for copies of the packets traveling clockwise around the ring.

To inhibit looping in the face of such physical misconfiguration, only `forward_connected` adjacencies are permitted to have DNR set, and the link layer destination address of the adjacency (eg.: MAC address) protects against closing the loop. Link layers without port unique link layer addresses should not be used with the DNR flag set.

## 5.2. Duplicates

Duplicates happen when the topology of the BitString is not a tree but redundantly connects BFRs with each other. The controller must therefore ensure to only create BitStrings that are trees in the topology.

When links are incorrectly physically re-connected before the controller updates BitStrings in BFIRs, duplicates can happen. Like loops, these can be inhibited by link layer addressing in `forward_connected` adjacencies.

If interface or loopback addresses used in `forward_routed` adjacencies are moved from one BFR to another, duplicates can equally happen. Such re-addressing operations must be coordinated with the controller.

## 6. BIER-TE Forwarding Pseudocode

The following sections of Pseudocode are meant to illustrate the BIER-TE forwarding plane. This code is not meant to be normative but to serve both as a potentially easier to read and more precise representation of the forwarding functionality and to illustrate how simple BIER-TE forwarding is and that it can be efficiently be implemented.

The following procedure is executed on a BFR whenever the BIFT is changed by the BIER-TE controller host:

```
global MyBitsOfInterest

void BIFTChanged()
{
    for (Index = 0; Index++ ; Index <= BitStringLength)
        if(BIFT[Index] != <empty>)
            MyBitsOfInterest != 2<<(Index-1)
}
```

The following procedure is executed whenever a BIER-TE packet is to be forwarded:

```

void ForwardBierTePacket (Packet)
{
    // We calculate in BitMask the subset of BPs of the BitString
    // for which we have adjacencies. This is purely an
    // optimization to avoid to replicate for every BP
    // set in BitString only to discover that for most of them,
    // the BIFT has no adjacency.

    local BitMask = Packet->BitString
    Packet->BitString &= ~MyBitsOfInterest
    BitMask &= MyBitsOfInterest

    // Replication
    for (Index = GetFirstBitPosition(BitMask); Index ;
        Index = GetNextBitPosition(BitMask, Index))
        foreach adjacency BIFT[Index]

            if(adjacency == ECMP(ListOfAdjacencies, seed) )
                I = ECMP_hash(sizeof(ListOfAdjacencies),
                               Packet->Entropy, seed)
                adjacency = ListOfAdjacencies[I]

            PacketCopy = Copy(Packet)

            switch(adjacency)
            case forward_connected(interface,neighbor,DNR):
                if(DNR)
                    PacketCopy->BitString |= 2<<(Index-1)
                    SendToL2Unicast(PacketCopy,interface,neighbor)

            case forward_routed([VRF],neighbor):
                SendToL3(PacketCopy,[VRF],l3-neighbor)

            case local_decap([VRF],neighbor):
                DecapBierHeader(PacketCopy)
                PassTo(PacketCopy,[VRF],Packet->NextProto)
    }
}

```

## 7. Managing SI, subdomains and BFR-ids

When the number of bits required to represent the necessary hops in the topology and BFER exceeds the supported bitstring length, multiple SI and/or subdomains must be used. This section discusses how.

BIER-TE forwarding does not require the concept of BFR-id, but routing underlay, flow overlay and BIER headers may. This section also discusses how BFR-id can be assigned to BFIR/BFER for BIER-TE.



### 7.1. Why SI and sub-domains

For BIER and BIER-TE forwarding, the most important result of using multiple SI and/or subdomains is the same: Packets that need to be sent to BFER in different SI or subdomains require different BIER packets: each one with a bitstring for a different (SI,subdomain) bitstring. Each such bitstring uses one bitstring length sized SI block in the BIFT of the subdomain. We call this a BIFT:SI (block).

For BIER and BIER-TE forwarding itself there is also no difference whether different SI and/or sub-domains are chosen, but SI and subdomain have different purposes in the BIER architecture shared by BIER-TE. This impacts how operators are managing them and how especially flow overlays will likely use them.

By default, every possible BFIR/BFER in a BIER network would likely be given a BFR-id in subdomain 0 (unless there are > 64k BFIR/BFER).

If there are different flow services (or service instances) requiring replication to different subsets of BFER, then it will likely not be possible to achieve the best replication efficiency for all of these service instances via subdomain 0. Ideal replication efficiency for N BFER exists in a subdomain if they are split over not more than  $\text{ceiling}(N/\text{bitstring-length})$  SI.

If service instances justify additional BIER:SI state in the network, additional subdomains will be used: BFIR/BFER are assigned BFIR-id in those subdomains and each service instance is configured to use the most appropriate subdomain. This results in improved replication efficiency for different services.

Even if creation of subdomains and assignment of BFR-id to BFIR/BFER in those subdomains is automated, it is not expected that individual service instances can deal with BFER in different subdomains. A service instance may only support configuration of a single subdomain it should rely on.

To be able to easily reuse (and modify as little as possible) existing BIER procedures including flow-overlay and routing underlay, when BIER-TE forwarding is added, we therefore reuse SI and subdomain logically in the same way as they are used in BIER: All necessary BFIR/BFER for a service use a single BIER-TE BIFT and are split across as many SI as necessary (see below). Different services may use different subdomains that primarily exist to provide more efficient replication (and for BIER-TE desirable traffic engineering) for different subsets of BFIR/BFER.

## 7.2. Bit assignment comparison BIER and BIER-TE

In BIER, bitstrings only need to carry bits for BFER, which lead to the model that BFR-ids map 1:1 to each bit in a bitstring.

In BIER-TE, bitstrings need to carry bits to indicate not only the receiving BFER but also the intermediate hops/links across which the packet must be sent. The maximum number of BFER that can be supported in a single bitstring or BIFT:SI depends on the number of bits necessary to represent the desired topology between them.

"Desired" topology because it depends on the physical topology, and on the desire of the operator to allow for explicit traffic engineering across every single hop (which requires more bits), or reducing the number of required bits by exploiting optimizations such as unicast (`forward_route`), ECMP or flood (DNR) over "uninteresting" sub-parts of the topology - eg: parts where different trees do not need to take different paths due to traffic-engineering reasons.

The total number of bits to describe the topology in a BIFT:SI can therefore easily be as low as 20% or as high as 80%. The higher the percentage, the higher the likelihood, that those topology bits are not just BIER-TE overhead without additional benefit, but instead they will allow to express the desired traffic-engineering alternatives.

## 7.3. Using BFR-id with BIER-TE

Because there is no 1:1 mapping between bits in the bitstring and BFER, BIER-TE can not simply rely on the BIER 1:1 mapping between bits in a bitstring and BFR-id.

In BIER, automatic schemes could assign all possible BFR-ids sequentially to BFERs. This will not work in BIER-TE. In BIER-TE, the operator or BIER-TE controller host has to determine a BFR-id for each BFER in each required subdomain. The BFR-id may or may not have a relationship with a bit in the bitstring. Suggestions are detailed below. Once determined, the BFR-id can then be configured on the BFER and used by flow overlay, routing underlay and the BIER header almost the same as the BFR-id in BIER.

The one exception are application/flow-overlays that automatically calculate the bitstring(s) of BIER packets by converting BFR-id to bits. In BIER-TE, this operation can be done in two ways:

"Independent branches": For a given application or (set of) trees, the branches from a BFIR to every BFER are independent of the

branches to any other BFER. For example, shortest path trees have independent branches.

"Interdependent branches": When a BFER is added or deleted from a particular distribution tree, branches to other BFER still in the tree may need to change. Steiner tree are examples of dependent branch trees.

If "independent branches" are sufficient, the BIER-TE controller host can provide to such applications for every BFR-id a SI:bitstring with the BIER-TE bits for the branch towards that BFER. The application can then independently calculate the SI:bitstring for all desired BFER by OR'ing their bitstrings.

If "interdependent branches" are required, the application could call a BIER-TE controller host API with the list of required BFER-id and get the required bitstring back. Whenever the set of BFER-id changes, this is repeated.

Note that in either case (unlike in BIER), the bits in BIER-TE may need to change upon link/node failure/recovery, network expansion and network load by other traffic (as part of traffic engineering goals). Interactions between such BFIR applications and the BIER-TE controller host do therefore need to support dynamic updates to the bitstrings.

#### 7.4. Assigning BFR-ids for BIER-TE

For non-leaf BFER, there is usually a single bit  $k$  for that BFER with a `local_decap()` adjacency on the BFER. The BFR-id for such a BFER is therefore most easily the one it would have in BIER:  $SI * \text{bitstring-length} + k$ .

As explained earlier in the document, leaf BFER do not need such a separate bit because the fact alone that the BIER-TE packet is forwarded to the leaf BFER indicates that the BFER should decapsulate it. Such a BFER will have one or more bits for the links leading only to it. The BFR-id could therefore most easily be the BFR-id derived from the lowest bit for those links.

These two rules are only recommendations for the operator or BIER-TE controller assigning the BFR-ids. Any allocation scheme can be used, the BFR-ids just need to be unique across BFRs in each subdomain.

It is not currently determined if a single subdomain could or should be allowed to forward both BIER and BIER-TE packets. If this should be supported, there are two options:

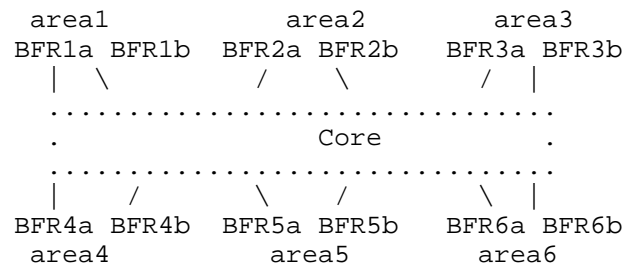
A. BIER and BIER-TE have different BFR-id in the same subdomain. This allows higher replication efficiency for BIER because their BFR-id can be assigned sequentially, while the bitstrings for BIER-TE will have also the additional bits for the topology. There is no relationship between a BFR BIER BFR-id and BIER-TE BFR-id.

B. BIER and BIER-TE share the same BFR-id. The BFR-id are assigned as explained above for BIER-TE and simply reused for BIER. The replication efficiency for BIER will be as low as that for BIER-TE in this approach. Depending on topology, only the same 20%..80% of bits as possible for BIER-TE can be used for BIER.

## 7.5. Example bit allocations

### 7.5.1. With BIER

Consider a network setup with a bitstring length of 256 for a network topology as shown in the picture below. The network has 6 areas, each with ca. 180 BFR, connecting via a core with some larger (core) BFR. To address all BFER with BIER, 4 SI are required. To send a BIER packet to all BFER in the network, 4 copies need to be sent by the BFIR. On the BFIR it does not make a difference how the BFR-id are allocated to BFER in the network, but for efficiency further down in the network it does make a difference.



With random allocation of BFR-id to BFER, each receiving area would (most likely) have to receive all 4 copies of the BIER packet because there would be BFR-id for each of the 4 SI in each of the areas. Only further towards each BFER would this duplication subside - when each of the 4 trees runs out of branches.

If BFR-id are allocated intelligently, then all the BFER in an area would be given BFR-id with as few as possible different SI. Each area would only have to forward one or two packets instead of 4.

Given how networks can grow over time, replication efficiency in an area will also easily go down over time when BFR-id are network wide allocated sequentially over time. An area that initially only has

BFR-id in one SI might end up with many SI over a longer period of growth. Allocating SIs to areas with initially sufficiently many spare bits for growths can help to alleviate this issue. Or renumber BFR-id after network expansion. In this example one may consider to use 6 SI and assign one to each area.

This example shows that intelligent BFR-id allocation within at least subdomain 0 can even be helpful or even necessary in BIER.

#### 7.5.2. With BIER-TE

In BIER-TE one needs to determine a subset of the physical topology and attached BFER so that the "desired" representation of this topology and the BFER fit into a single bitstring. This process needs to be repeated until the whole topology is covered.

Once bits/SIs are assigned to topology and BFER, BFR-id is just a derived set of identifiers from the operator/BIER-TE controller as explained above.

Every time that different sub-topologies have overlap, bits need to be repeated across the bitstrings, increasing the overall amount of bits required across all bitstring/SIs. In the worst case, random subsets of BFER are assigned to different SI. This is much worse than in BIER because it not only reduces replication efficiency with the same number of overall bits, but even further - because more bits are required due to duplication of bits for topology across multiple SI. Intelligent BFER to SI assignment and selecting specific "desired" subtopologies can minimize this problem.

To set up BIER-TE efficiently for above topology, the following bit allocation methods can be used. This method can easily be expanded to other, similarly structured larger topologies.

Each area is allocated one or more SI depending on the number of future expected BFER and number of bits required for the topology in the area. In this example, 6 SI, one per area.

In addition, we use 4 bits in each SI: bia, bib, bea, beb: bit ingress a, bit ingress b, bit egress a, bit egress b. These bits will be used to pass BIER packets from any BFIR via any combination of ingress area a/b BFR and egress area a/b BFR into a specific target area. These bits are then set up with the right forward\_routed adjacencies on the BFIR and area edge BFR:

On all BFIR in an area j, bia in each BIFT:SI is populated with the same forward\_routed(BFRja), and bib with forward\_routed(BFRjb). On all area edge BFR, bea in BIFT:SI=k is populated with

forward\_routed(BFRka) and beb in BIFT:SI=k with forward\_routed(BFRkb).

For BIER-TE forwarding of a packet to some subset of BFER across all areas, a BFIR would create at most 6 copies, with SI=1...SI=6. In each packet, the bits indicate bits for topology and BFER in that topology plus the four bits to indicate whether to pass this packet via the ingress area a or b border BFR and the egress area a or b border BFR, therefore allowing path engineering for those two "unicast" legs: 1) BFIR to ingress area edge and 2) core to egress area edge. Replication only happens inside the egress areas. For BFER in the same area as in the BFIR, these four bits are not used.

#### 7.6. Summary

BIER-TE can like BIER support multiple SI within a sub-domain to allow re-using the concept of BFR-id and therefore minimize BIER-TE specific functions in underlay routing, flow overlay methods and BIER headers.

The number of BFIR/BFER possible in a subdomain is smaller than in BIER because BIER-TE uses additional bits for topology.

Subdomains can in BIER-TE be used like in BIER to create more efficient replication to known subsets of BFER.

Assigning bits for BFER intelligently into the right SI is more important in BIER-TE than in BIER because of replication efficiency and overall amount of bits required.

#### 8. BIER-TE and Segment Routing

Segment Routing aims to achieve lightweight path engineering via loose source routing. Compared for example to RSVP-TE, it does not require per-path signaling to each of these hops.

BIER-TE supports the same design philosophy for multicast. Like in SR, it relies on source-routing - via the definition of a BitString. Like SR, it only requires to consider the "hops" on which either replication has to happen, or across which the traffic should be steered (even without replication). Any other hops can be skipped via the use of routed adjacencies.

Instead of defining BitPositions for non-replicating hops, it is equally possible to use segment routing encapsulations (eg: MPLS label stacks) for "forward\_routed" adjacencies.

## 9. Security Considerations

The security considerations are the same as for BIER with the following differences:

BFR-ids and BFR-prefixes are not used in BIER-TE, nor are procedures for their distribution, so these are not attack vectors against BIER-TE.

## 10. IANA Considerations

This document requests no action by IANA.

## 11. Acknowledgements

The authors would like to thank Greg Shepherd, Ijsbrand Wijnands and Neale Ranns for their extensive review and suggestions.

## 12. Change log [RFC Editor: Please remove]

draft-ietf-bier-te-arch:

00: Changed target state to experimental (WG conclusion), updated references, mod auth association.

- Source now on <http://www.github.com/toerless/bier-te-arch>
- Please open issues on the github for change/improvement requests to the document - in addition to posting them on the list (bier@ietf.). Thanks!.

draft-eckert-bier-te-arch:

06: Added overview of forwarding differences between BIER, BIER-TE.

05: Author affiliation change only.

04: Added comparison to Live-Live and BFIR to FRR section (Eckert).

04: Removed FRR content into the new FRR draft [I-D.eckert-bier-te-frr] (Braun).

- Linked FRR information to new draft in Overview/Introduction
- Removed BTAFT/FRR from "Changes in the network topology"

- Linked new draft in "Link/Node Failures and Recovery"
- Removed FRR from "The BIER-TE Forwarding Layer"
- Moved FRR section to new draft
- Moved FRR parts of Pseudocode into new draft
- Left only non FRR parts
- removed `FrrUpDown(..)` and `//FRR` operations in `ForwardBierTePacket(..)`
- New draft contains `FrrUpDown(..)` and `ForwardBierTePacket(Packet)` from bier-arch-03
- Moved "BIER-TE and existing FRR to new draft"
- Moved "BIER-TE and Segment Routing" section one level up
- Thus, removed "Further considerations" that only contained this section
- Added Changes for version 04

03: Updated the FRR section. Added examples for FRR key concepts. Added BIER-in-BIER tunneling as option for tunnels in backup paths. BIFT structure is expanded and contains an additional match field to support full node protection with BIER-TE FRR.

03: Updated FRR section. Explanation how BIER-in-BIER encapsulation provides P2MP protection for node failures even though the routing underlay does not provide P2MP.

02: Changed the definition of BIFT to be more inline with BIER. In revs. up to -01, the idea was that a BIFT has only entries for a single bitstring, and every SI and subdomain would be a separate BIFT. In BIER, each BIFT covers all SI. This is now also how we define it in BIER-TE.

02: Added Section 7 to explain the use of SI, subdomains and BFR-id in BIER-TE and to give an example how to efficiently assign bits for a large topology requiring multiple SI.

02: Added further detailed for rings - how to support input from all ring nodes.



01: Fixed BFIR -> BFER for section 4.3.

01: Added explanation of SI, difference to BIER ECMP, consideration for Segment Routing, unicast FRR, considerations for encapsulation, explanations of BIER-TE controller host and CLI.

00: Initial version.

### 13. References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.

### Authors' Addresses

Toerless Eckert (editor)  
Huawei USA - Futurewei Technologies Inc.  
2330 Central Expy  
Santa Clara 95050  
USA

Email: [tte+ietf@cs.fau.de](mailto:tte+ietf@cs.fau.de)

Gregory Cauchie  
Bouygues Telecom

Email: [GCAUCHIE@bouyguestelecom.fr](mailto:GCAUCHIE@bouyguestelecom.fr)

Wolfgang Braun  
University of Tuebingen

Email: [wolfgang.braun@uni-tuebingen.de](mailto:wolfgang.braun@uni-tuebingen.de)

Michael Menth  
University of Tuebingen

Email: [menth@uni-tuebingen.de](mailto:menth@uni-tuebingen.de)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 27 October 2022

T.T.E. Eckert, Ed.  
Futurewei  
M.M. Menth  
University of Tuebingen  
G.C. Cauchie  
KOEVOO  
April 2022

Tree Engineering for Bit Index Explicit Replication (BIER-TE)  
draft-ietf-bier-te-arch-13

Abstract

This memo describes per-packet stateless strict and loose path steered replication and forwarding for "Bit Index Explicit Replication" (BIER, RFC8279) packets. It is called BIER Tree Engineering (BIER-TE) and is intended to be used as the path steering mechanism for Traffic Engineering with BIER.

BIER-TE introduces a new semantic for "bit positions" (BP). They indicate adjacencies of the network topology, as opposed to (non-TE) BIER in which BPs indicate "Bit-Forwarding Egress Routers" (BFER). A BIER-TE packets BitString therefore indicates the edges of the (loop-free) tree that the packet is forwarded across by BIER-TE. BIER-TE can leverage BIER forwarding engines with little changes. Co-existence of BIER and BIER-TE forwarding in the same domain is possible, for example by using separate BIER "sub-domains" (SDs). Except for the optional routed adjacencies, BIER-TE does not require a BIER routing underlay, and can therefore operate without depending on an "Interior Gateway Routing protocol" (IGP).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 October 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Overview . . . . .	3
1.1. Requirements Language . . . . .	5
2. Introduction . . . . .	5
2.1. Basic Examples . . . . .	5
2.2. BIER-TE Topology and adjacencies . . . . .	8
2.3. Relationship to BIER . . . . .	9
2.4. Accelerated/Hardware forwarding comparison . . . . .	11
3. Components . . . . .	11
3.1. The Multicast Flow Overlay . . . . .	12
3.2. The BIER-TE Control Plane . . . . .	12
3.2.1. The BIER-TE Controller . . . . .	14
3.2.1.1. BIER-TE Topology discovery and creation . . . . .	14
3.2.1.2. Engineered Trees via BitStrings . . . . .	15
3.2.1.3. Changes in the network topology . . . . .	16
3.2.1.4. Link/Node Failures and Recovery . . . . .	16
3.3. The BIER-TE Forwarding Plane . . . . .	16
3.4. The Routing Underlay . . . . .	17
3.5. Traffic Engineering Considerations . . . . .	17
4. BIER-TE Forwarding . . . . .	18
4.1. The BIER-TE Bit Index Forwarding Table (BIFT) . . . . .	18
4.2. Adjacency Types . . . . .	20
4.2.1. Forward Connected . . . . .	21
4.2.2. Forward Routed . . . . .	21
4.2.3. ECMP . . . . .	21
4.2.4. Local Decapsulation . . . . .	22
4.3. Encapsulation / Co-existence with BIER . . . . .	22
4.4. BIER-TE Forwarding Pseudocode . . . . .	23
4.5. BFR Requirements for BIER-TE forwarding . . . . .	26
5. BIER-TE Controller Operational Considerations . . . . .	27
5.1. Bit Position Assignments . . . . .	27
5.1.1. P2P Links . . . . .	27
5.1.2. BFER . . . . .	27

5.1.3.	Leaf BFERs . . . . .	27
5.1.4.	LANs . . . . .	29
5.1.5.	Hub and Spoke . . . . .	30
5.1.6.	Rings . . . . .	30
5.1.7.	Equal Cost MultiPath (ECMP) . . . . .	31
5.1.8.	Forward Routed adjacencies . . . . .	34
5.1.8.1.	Reducing bit positions . . . . .	34
5.1.8.2.	Supporting nodes without BIER-TE . . . . .	35
5.1.9.	Reuse of bit positions (without DNC) . . . . .	35
5.1.10.	Summary of BP optimizations . . . . .	36
5.2.	Avoiding duplicates and loops . . . . .	37
5.2.1.	Loops . . . . .	38
5.2.2.	Duplicates . . . . .	38
5.3.	Managing SI, sub-domains and BFR-ids . . . . .	39
5.3.1.	Why SI and sub-domains . . . . .	39
5.3.2.	Assigning bits for the BIER-TE topology . . . . .	40
5.3.3.	Assigning BFR-id with BIER-TE . . . . .	41
5.3.4.	Mapping from BFR to BitStrings with BIER-TE . . . . .	42
5.3.5.	Assigning BFR-ids for BIER-TE . . . . .	43
5.3.6.	Example bit allocations . . . . .	43
5.3.6.1.	With BIER . . . . .	43
5.3.6.2.	With BIER-TE . . . . .	44
5.3.7.	Summary . . . . .	45
6.	Security Considerations . . . . .	46
7.	IANA Considerations . . . . .	47
8.	Acknowledgements . . . . .	47
9.	Change log [RFC Editor: Please remove] . . . . .	48
10.	References . . . . .	61
10.1.	Normative References . . . . .	61
10.2.	Informative References . . . . .	61
Appendix A.	BIER-TE and Segment Routing (SR) . . . . .	64
Authors' Addresses	. . . . .	65

## 1. Overview

BIER-TE is based on the (non-TE) BIER architecture, terminology and packet formats as described in [RFC8279] and [RFC8296]. This document describes BIER-TE in the expectation that the reader is familiar with these two documents.

BIER-TE introduces a new semantic for "bit positions" (BP). They indicate adjacencies of the network topology, as opposed to (non-TE) BIER in which BPs indicate "Bit-Forwarding Egress Routers" (BFER). A BIER-TE packets BitString therefore indicates the edges of the (loop-free) tree that the packet is forwarded across by BIER-TE. With BIER-TE, the "Bit Index Forwarding Table" (BIFT) of each "Bit Forwarding Router" (BFR) is only populated with BP that are adjacent to the BFR in the BIER-TE Topology. Other BPs are empty in the BIFT.

The BFR replicate and forwards BIER packets to adjacent BPs that are set in the packet. BPs are normally also cleared upon forwarding to avoid duplicates and loops.

BIER-TE can leverage BIER forwarding engines with little or no changes. It can also co-exist with BIER forwarding in the same domain, for example by using separate BIER sub-domains. Except for the optional routed adjacencies, BIER-TE does not require a BIER routing underlay, and can therefore operate without depending on an "Interior Gateway Routing protocol" (IGP).

This document is structured as follows:

- \* Section 2 introduces BIER-TE with two forwarding examples, followed by an introduction of the new concepts of the BIER-TE (overlay) topology and finally a summary of the relationship between BIER and BIER-TE and a discussion of accelerated hardware forwarding.
- \* Section 3 describes the components of the BIER-TE architecture, Flow overlay, BIER-TE layer with the BIER-TE control plane (including the BIER-TE controller) and BIER-TE forwarding plane, and the routing underlay.
- \* Section 4 specifies the behavior of the BIER-TE forwarding plane with the different type of adjacencies and possible variations of BIER-TE forwarding pseudocode, and finally the mandatory and optional requirements.
- \* Section 5 describes operational considerations for the BIER-TE controller, foremost how the BIER-TE controller can optimize the use of BP by using specific type of BIER-TE adjacencies for different type of topological situations, but also how to assign bits to avoid loops and duplicates (which in BIER-TE does not come for free), and finally how "Set Identifier" (SI), "sub-domain" (SD) and BFR-ids can be managed by a BIER-TE controller, examples and summary.
- \* Appendix A concludes the technology specific sections of the document by further relating BIER-TE to Segment Routing (SR).

Note that related work, [I-D.ietf-roll-ccast] uses Bloom filters [Bloom70] to represent leaves or edges of the intended delivery tree. Bloom filters in general can support larger trees/topologies with fewer addressing bits than explicit BitStrings, but they introduce the heuristic risk of false positives and cannot clear bits in the BitString during forwarding to avoid loops. For these reasons, BIER-TE uses explicit BitStrings like BIER. The explicit BitStrings of BIER-TE can also be seen as a special type of Bloom filter, and this is how related work [ICC] describes it.

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

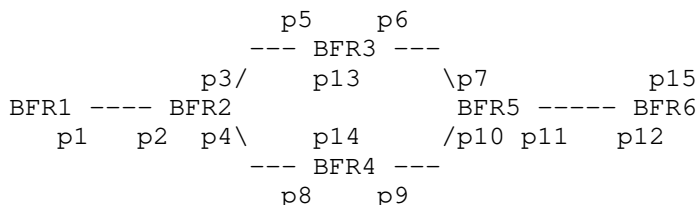
## 2. Introduction

### 2.1. Basic Examples

BIER-TE forwarding is best introduced with simple examples. These examples use formal terms defined later in the document (Figure 4), including `forward_connected()`, `forward_routed()` and `local_decap()`.

## BIER-TE Topology:

Diagram:



(simplified) BIER-TE Bit Index Forwarding Tables (BIFT):

```

BFR1:  p1  -> local_decap()
        p2  -> forward_connected() to BFR2

BFR2:  p1  -> forward_connected() to BFR1
        p5  -> forward_connected() to BFR3
        p8  -> forward_connected() to BFR4

BFR3:  p3  -> forward_connected() to BFR2
        p7  -> forward_connected() to BFR5
        p13 -> local_decap()

BFR4:  p4  -> forward_connected() to BFR2
        p10 -> forward_connected() to BFR5
        p14 -> local_decap()

BFR5:  p6  -> forward_connected() to BFR3
        p9  -> forward_connected() to BFR4
        p12 -> forward_connected() to BFR6

BFR6:  p11 -> forward_connected() to BFR5
        p15 -> local_decap()

```

Figure 1: BIER-TE basic example

Consider the simple network in the above BIER-TE overview example picture with 6 BFRs. p1...p15 are the bit positions used. All BFRs can act as an ingress BFR (BFIR), BFR1, BFR3, BFR4 and BFR6 can also be BFERs. Forward\_connected() is the name for adjacencies that are representing subnet adjacencies of the network. Local\_decap() is the name of the adjacency to decapsulate BIER-TE packets and pass their payload to higher layer processing.



Assume a packet from BFR1 should be sent via BFR4 to BFR6. This requires a BitString (p2,p8,p10,p12,p15). When this packet is examined by BIER-TE on BFR1, the only bit position from the BitString that is also set in the BIFT is p2. This will cause BFR1 to send the only copy of the packet to BFR2. Similarly, BFR2 will forward to BFR4 because of p8, BFR4 to BFR5 because of p10 and BFR5 to BFR6 because of p12. p15 finally makes BFR6 receive and decapsulate the packet.

To send a copy to BFR6 via BFR4 and also a copy to BFR3, the BitString needs to be (p2,p5,p8,p10,p12,p13,p15). When this packet is examined by BFR2, p5 causes one copy to be sent to BFR3 and p8 one copy to BFR4. When BFR3 receives the packet, p13 will cause it to receive and decapsulate the packet.

If instead the BitString was (p2,p6,p8,p10,p12,p13,p15), the packet would be copied by BFR5 towards BFR3 because of p6 instead of being copied by BFR2 to BFR3 because of p5 in the prior case. This is showing the ability of the shown BIER-TE Topology to make the traffic pass across any possible path and be replicated where desired.

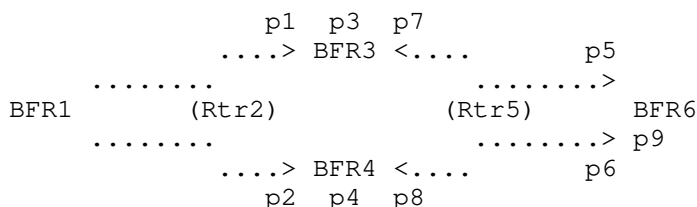
BIER-TE has various options to minimize BP assignments, many of which are based on out-of-band knowledge about the required multicast traffic paths and bandwidth consumption in the network, such as from pre-deployment planning.

Figure 2 shows a modified example, in which Rtr2 and Rtr5 are assumed not to support BIER-TE, so traffic has to be unicast encapsulated across them. To emphasize non-L2, but routed/tunneled forwarding of BIER-TE packets, these adjacencies are called "forward\_routed". Otherwise, there is no difference in their processing over the aforementioned forward\_connected() adjacencies.

In addition, bits are saved in the following example by assuming that BFR1 only needs to be BFIR but not BFER or transit BFR.

## BIER-TE Topology:

Diagram:



(simplified) BIER-TE Bit Index Forwarding Tables (BIFT):

```

BFR1:  p1  -> forward_routed() to BFR3
       p2  -> forward_routed() to BFR4

BFR3:  p3  -> local_decap()
       p5  -> forward_routed() to BFR6

BFR4:  p4  -> local_decap()
       p6  -> forward_routed() to BFR6

BFR6:  p7  -> forward_routed() to BFR3
       p8  -> forward_routed() to BFR4
       p9  -> local_decap()
  
```

Figure 2: BIER-TE basic overlay example

To send a BIER-TE packet from BFR1 via BFR3 to be received by BFR6, the BitString is (p1,p5,p9). From BFR1 via BFR4 to be received by BFR6, the BitString is (p2,p6,p9). A packet from BFR1 to be received by BFR3,BFR4 and from BFR3 to be received by BFR6 uses (p1,p2,p3,p4,p5,p9). A packet from BFR1 to be received by BFR3,BFR4 and from BFR4 to be received by BFR6 uses (p1,p2,p3,p4,p6,p9). A packet from BFR1 to be received by BFR4, and from BFR4 to be received by BFR6 and from there to be received by BFR3 uses (p2,p3,p4,p6,p7,p9). A packet from BFR1 to be received by BFR3, and from BFR3 to be received by BFR6 there to be received by BFR4 uses (p1,p3,p4,p5,p8,p9).

## 2.2. BIER-TE Topology and adjacencies

The key new component in BIER-TE compared to (non-TE) BIER is the BIER-TE topology as introduced through the two examples in Section 2.1. It is used to control where replication can or should happen and how to minimize the required number of BP for adjacencies.

The BIER-TE Topology consists of the BIFTs of all the BFR and can also be expressed as a directed graph where the edges are the adjacencies between the BFRs labelled with the BP used for the adjacency. Adjacencies are naturally unidirectional. BP can be reused across multiple adjacencies as long as this does not lead to undesired duplicates or loops as explained in Section 5.2.

If the BIER-TE topology represents (a subset of) the underlying (layer 2) topology of the network as shown in the first example, this may be called a "native" BIER-TE topology. A topology consisting only of "forward\_routed" adjacencies as shown in the second example may be called an "overlay" BIER-TE topology. A BIER-TE topology with both forward\_connected() and forward\_routed() adjacencies may be called a "hybrid" BIER-TE topology.

### 2.3. Relationship to BIER

BIER-TE is designed so that its forwarding plane is a simple extension to the (non-TE) BIER forwarding plane, hence allowing for it to be added to BIER deployments where it can be beneficial.

BIER-TE is also intended as an option to expand the BIER architecture into deployments where (non-TE) BIER may not be the best fit, such as statically provisioned networks with needs for path steering but without desire for distributed routing protocols.

#### 1. BIER-TE inherits the following aspects from BIER unchanged:

1. The fundamental purpose of per-packet signaled replication and delivery via a BitString.
2. The overall architecture consisting of three layers, flow overlay, BIER(-TE) layer and routing underlay.
3. The supported encapsulations [RFC8296].
4. The semantic of all [RFC8296] header elements used by the BIER-TE forwarding plane other than the semantic of the BP in the BitString.
5. The BIER forwarding plane, except for how bits have to be cleared during replication.

#### 2. BIER-TE has the following key changes with respect to BIER:

1. In BIER, bits in the BitString of a BIER packet header indicate a BFER and bits in the BIFT indicate the BIER control plane calculated next-hop toward that BFER. In BIER-

TE, a bit in the BitString of a BIER packet header indicates an adjacency in the BIER-TE topology, and only the BFR that is the upstream of that adjacency has its BP populated with the adjacency in its BIFT.

2. In BIER, the implied reference options for the core part of the BIER layer control plane are the BIER extensions for distributed routing protocols. This includes ISIS/OSPF extensions for BIER, [RFC8401] and [RFC8444].
  3. The reference option for the core part of the BIER-TE control plane is the BIER-TE controller. Nevertheless, both the BIER and BIER-TE BIFTs forwarding plane state could equally be populated by any mechanism.
  4. Assuming the reference options for the control plane, BIER-TE replaces in-network autonomous path calculation by explicit paths calculated by the BIER-TE controller.
3. The following elements/functions described in the BIER architecture are not required by the BIER-TE architecture:
1. "Bit Index Routing Tables" (BIRTs) are not required on BFRs for BIER-TE when using a BIER-TE controller because the controller can directly populate the BIFTs. In BIER, BIRTs are populated by the distributed routing protocol support for BIER, allowing BFRs to populate their BIFTs locally from their BIRTs. Other BIER-TE control plane or management plane options may introduce requirements for BIRTs for BIER-TE BFRs.
  2. The BIER-TE layer forwarding plane does not require BFRs to have a unique BP and therefore also no unique BFR-id. See Section 5.1.3.
  3. Identification of BFRs by the BIER-TE control plane is outside the scope of this specification. Whereas the BIER control plane uses BFR-ids in its BFR to BFR signaling, a BIER-TE controller may choose any form of identification deemed appropriate.
  4. BIER-TE forwarding does not require the BFIR-id field of the BIER packet header.
4. Co-existence of BIER and BIER-TE in the same network requires the following:

1. The BIER/BIER-TE packet header needs to allow addressing both BIER and BIER-TE BIFTs. Depending on the encapsulation option, the same SD may or may not be reusable across BIER and BIER-TE. See Section 4.3. In either case, a packet is always only forwarded end-to-end via BIER or via BIER-TE (ships in the nights forwarding).
2. BIER-TE deployments will have to assign BFR-ids to BFRs and insert them into the BFIR-id field of BIER packet headers as BIER does, whenever the deployment uses (unchanged) components developed for BIER that use BFR-id, such as multicast flow overlays or BIER layer control plane elements. See also Section 5.3.3.

#### 2.4. Accelerated/Hardware forwarding comparison

BIER-TE forwarding rules, especially the BitString parsing are designed to be as close as possible to those of BIER in the expectation that this eases the programming of BIER-TE forwarding code and/or BIER-TE forwarding hardware on platforms supporting BIER. The pseudocode in Section 4.4 shows how existing (non-TE) BIER/BIFT forwarding can be modified to support the required BIER-TE forwarding functionality (Section 4.5), by using BIER BIFT's "Forwarding Bit Mask" (F-BM): Only the clearing of bits to avoid duplicate packets to a BFR's neighbor is skipped in BIER-TE forwarding because it is not necessary and could not be done when using BIER F-BM.

Whether to use BIER or BIER-TE forwarding is simply a choice of the mode of the BIFT indicated by the packet (BIER or BIER-TE BIFT). This is determined by the BFR configuration for the encapsulation, see Section 4.3.

#### 3. Components

BIER-TE can be thought of being constituted from the same three layers as BIER: The "multicast flow overlay", the "BIER layer" and the "routing underlay". The following picture also shows how the "BIER layer" is constituted from the "BIER-TE forwarding plane" and the "BIER-TE control plane" represent by the "BIER-TE Controller".

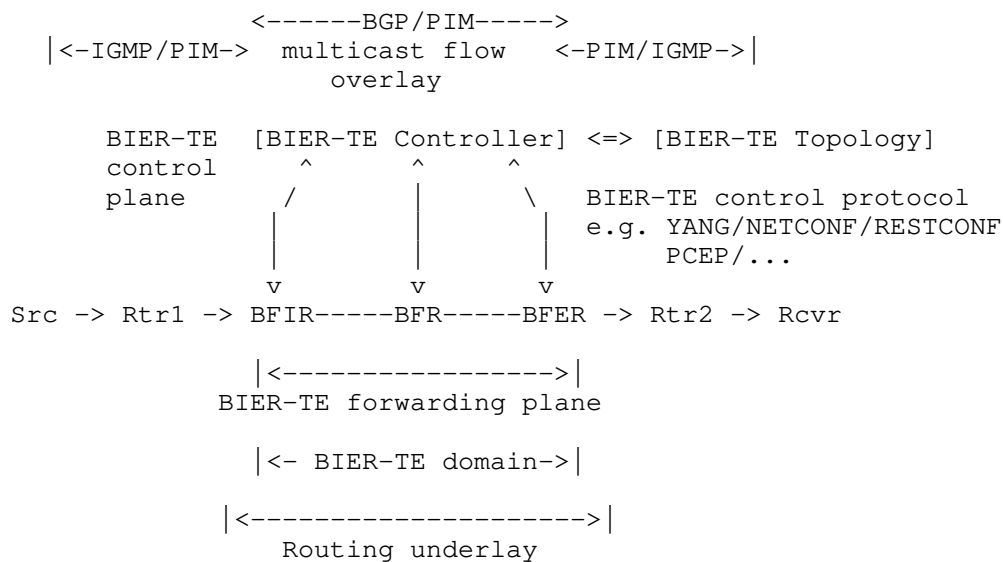


Figure 3: BIER-TE architecture

### 3.1. The Multicast Flow Overlay

The Multicast Flow Overlay has the same role as described for BIER in [RFC8279], Section 4.3. See also Section 3.2.1.2.

When a BIER-TE controller is used, then the signaling for the Multicast Flow Overlay may also be preferred to operate through a central point of control. For BGP based overlay flow services such as "Multicast VPN Using BIER" ([RFC8556]) this can be achieved by making the BIER-TE controller operate as a BGP Route Reflector ([RFC4456]) and combining it with signaling through BGP or a different protocol for the BIER-TE controller calculated BitStrings. See Section 3.2.1.2 and Section 5.3.4.

### 3.2. The BIER-TE Control Plane

In the (non-TE) BIER architecture [RFC8279], the BIER control plane is not explicitly separated from the BIER forwarding plane, but instead their functions are summarized together in Section 4.2. Example standardized options for the BIER control plane include ISIS/OSPF extensions for BIER, [RFC8401] and [RFC8444].

For BIER-TE, the control plane includes at minimum the following functionality.

1. BIER-TE topology control: During initial provisioning of the

network and/or during modifications of its topology and/or services, the protocols and/or procedures to establish BIER-TE BIFTs:

1. Determine the desired BIER-TE topology for a BIER-TE sub-domains: the native and/or overlay adjacencies that are assigned to BPs. Topology discovery is discussed in Section 3.2.1.1 and the various aspects of the BIER-TE controllers determinations about the topology are discussed throughout Section 5
  2. Determine the per-BFR BIFT from the BIER-TE topology. This is achieved by simply extracting the adjacencies of the BFR from the BIER-TE topology and populating the BFRs BIFT with them.
  3. Optionally assign BFR-ids to BFIRs for later insertion into BIER headers on BFIRs as BFIR-id. Alternatively, BFIR-id in BIER packet headers may be managed solely by the flow overlay layer and/or be unused. This is discussed in Section 5.3.3.
  4. Install/update the BIFTs into the BFRs and optionally BFR-ids into BFIRs. This is discussed in Section 3.2.1.1.
2. BIER-TE tree control: During operations of the network, protocols and/or procedures to support creation/change/removal of overlay flows on BFIRs:
1. Process the BIER-TE requirements for the multicast overlay flow: BFIR and BFERs of the flow as well as policies for the path selection of the flow. This is discussed in Section 3.5.
  2. Determine the BitStrings and optionally Entropy. This is discussed in Section 3.2.1.2, Section 3.5 and Section 5.3.4.
  3. Install state on the BFIR to impose the desired BIER packet header(s) for packets of the overlay flow. Different aspects of this and the next point are discussed throughout Section 3.2.1 and in Section 4.3, but the main responsibility of these two points is with the Multicast Flow Overlay (Section 3.1), which is architecturally inherited from BIER.
  4. Install the necessary state on the BFERs to decapsulate the BIER packet header and properly dispatch its payload.

### 3.2.1. The BIER-TE Controller

[RFC-Editor: the following text has three references to anchors topology-control, topology-control-1 and tree-control. Unfortunately, XMLv2 does not offer any tagging that reasonable references are generated (i had this problem already in RFCs last year. Please make sure there are useful-to-read cross-references in the RFC in these three places after you convert to XMLv3.)]

This architecture describes the BIER-TE control plane as shown in Figure 3 to consist of:

- \* A BIER-TE controller.
- \* BFR data-models and protocols to communicate between controller and BFRs in support of BIER-TE topology control (Section 3.2), such as YANG/NETCONF/RESTCONF ([RFC7950]/[RFC6241]/[RFC8040]).
- \* BFR data-models and protocols to communicate between controller and BFIR in support of BIER-TE tree control (Section 3.2), such as BIER-TE extensions for [RFC5440].

The single, centralized BIER-TE controller is used in this document as reference option for the BIER-TE control plane but other options are equally feasible. The BIER-TE control plane could equally be implemented without automated configuration/protocols, by an operator via CLI on the BFRs. In that case, operator configured local policy on the BFIR would have to determine how to set the appropriate BIER header fields. The BIER-TE control plane could also be decentralized and/or distributed, but this document does not consider any additional protocols and/or procedures which would then be necessary to coordinate its (distributed/decentralized) entities to achieve the above described functionality.

#### 3.2.1.1. BIER-TE Topology discovery and creation

The first item of BIER-TE topology control (Section 3.2, Paragraph 3, Item 2.2.1) includes network topology discovery and BIER-TE topology creation. The latter describes the process by which a Controller determines which routers are to be configured as BFRs and the adjacencies between them.

In statically managed networks, such as in industrial environments, both discovery and creation can be a manual/offline process.



In other networks, topology discovery may rely on protocols including extending a "Link-State-Protocol" based IGP into the BIER-TE controller itself, [RFC7752] (BGP-LS) or [RFC8345] (YANG topology) as well as BIER-TE specific methods, for example via [I-D.ietf-bier-te-yang]. These options are non-exhaustive.

Dynamic creation of the BIER-TE topology can be as easy as mapping the network topology 1:1 to the BIER-TE topology by assigning a BP for every network subnet adjacency. In larger networks, it likely involves more complex policy and optimization decisions including how to minimize the number of BPs required and how to assign BPs across different BitStrings to minimize the number of duplicate packets across links when delivering an overlay flow to BFER using different SIs/BitStrings. These topics are discussed in Section 5.

When the BIER-TE topology is determined, the BIER-TE Controller then pushes the BitPositions/adjacencies to the BIFT of the BFRs. On each BFR only those SI:BitPositions are populated that are adjacencies to other BFRs in the BIER-TE topology.

Communications between the BIER-TE Controller and BFRs for both BIER-TE topology control and BIER-TE tree control is ideally via standardized protocols and data-models such as NETCONF/RESTCONF/YANG/PCP. Vendor-specific CLI on the BFRs is also an option (as in many other SDN solutions lacking definition of standardized data models).

#### 3.2.1.2. Engineered Trees via BitStrings

In BIER, the same set of BFER in a single sub-domain is always encoded as the same BitString. In BIER-TE, the BitString used to reach the same set of BFER in the same sub-domain can be different for different overlay flows because the BitString encodes the paths towards the BFER, so the BitStrings from different BFIR to the same set of BFER will often be different. Likewise, the BitString from the same BFIR to the same set of BFER can be different for different overlay flows for policy reasons such as shortest path trees, Steiner trees (minimum cost trees), diverse path trees for redundancy and so on.

See also [I-D.ietf-bier-multicast-http-response] for an application leveraging BIER-TE engineered trees.

### 3.2.1.3. Changes in the network topology

If the network topology changes (not failure based) so that adjacencies that are assigned to bit positions are no longer needed, the BIER-TE Controller can re-use those bit positions for new adjacencies. First, these bit positions need to be removed from any BFIR flow state and BFR BIFT state, then they can be repopulated, first into BIFT and then into the BFIR.

### 3.2.1.4. Link/Node Failures and Recovery

When link or nodes fail or recover in the topology, BIER-TE could quickly respond with FRR procedures such as [I-D.eckert-bier-te-frr], the details of which are out of scope for this document. It can also more slowly react by recalculating the BitStrings of affected multicast flows. This reaction is slower than the FRR procedure because the BIER-TE Controller needs to receive link/node up/down indications, recalculate the desired BitStrings and push them down into the BFIRs. With FRR, this is all performed locally on a BFR receiving the adjacency up/down notification.

## 3.3. The BIER-TE Forwarding Plane

[RFC-editor Q: "is constituted from" / "consists of" / "composed from..." ???]

The BIER-TE Forwarding Plane is constituted from the following components:

1. On a BFIR, imposition of the BIER header for packets from overlay flows. This is driven by a combination of state established by the BIER-TE control plane and/or the multicast flow overlay as explained in Section 3.1.
2. On BFRs (including BFIR and BFER), forwarding/replication of BIER packets according to their SD, SI, "BitStringLength" (BSL), BitString and optionally Entropy fields as explained in Section 4. Processing of other BIER header fields such as DSCP is outside the scope of this document.
3. On BFERs, removal of the BIER header and dispatching of the payload according to state created by the BIER-TE control plane and/or overlay layer.

When the BIER-TE Forwarding Plane receives a packet, it simply looks up the bit positions that are set in the BitString of the packet in the BIFT that was populated by the BIER-TE Controller. For every BP that is set in the BitString, and that has one or more adjacencies in

the BIFT, a copy is made according to the type of adjacencies for that BP in the BIFT. Before sending any copy, the BFR clears all BPs in the BitString of the packet for which the BFR has one or more adjacencies in the BIFT. Clearing these bits inhibits packets from looping when the BitStrings erroneously includes a forwarding loop. When a `forward_connected()` adjacency has the "DoNotClear" (DNC) flag set, then this BP is re-set for the packet copied to that adjacency. See Section 4.2.1.

### 3.4. The Routing Underlay

For `forward_connected()` adjacencies, BIER-TE is sending BIER packets to directly connected BIER-TE neighbors as L2 (unicasted) BIER packets without requiring a routing underlay. For `forward_routed()` adjacencies, BIER-TE forwarding encapsulates a copy of the BIER packet so that it can be delivered by the forwarding plane of the routing underlay to the routable destination address indicated in the adjacency. See Section 4.2.2 for the adjacency definition.

BIER relies on the routing underlay to calculate paths towards BFRs and derive next-hop BFR adjacencies for those paths. This commonly relies on BIER specific extensions to the routing protocols of the routing underlay but may also be established by a controller. In BIER-TE, the next-hops of a packet are determined by the BitString through the BIER-TE Controller established adjacencies on the BFR for the BPs of the BitString. There is thus no need for BFR specific routing underlay extensions to forward BIER packets with BIER-TE semantics.

Encapsulation parameters can be provisioned by the BIER-TE controller into the `forward_connected()` or `forward_routed()` adjacencies directly without relying on a routing underlay.

If the BFR intends to support FRR for BIER-TE, then the BIER-TE forwarding plane needs to receive fast adjacency up/down notifications: Link up/down or neighbor up/down, e.g. from BFD. Providing these notifications is considered to be part of the routing underlay in this document.

### 3.5. Traffic Engineering Considerations

Traffic Engineering ([I-D.ietf-teas-rfc3272bis]) provides performance optimization of operational IP networks while utilizing network resources economically and reliably. The key elements needed to effect TE are policy, path steering and resource management. These elements require support at the control/controller level and within the forwarding plane.

Policy decisions are made within the BIER-TE control plane, i.e., within BIER-TE Controllers. Controllers use policy when composing BitStrings and BFR BIFT state. The mapping of user/IP traffic to specific BitStrings/BIER-TE flows is made based on policy. The specific details of BIER-TE policies and how a controller uses them are out of scope of this document.

Path steering is supported via the definition of a BitString. BitStrings used in BIER-TE are composed based on policy and resource management considerations. For example, when composing BIER-TE BitStrings, a Controller must take into account the resources available at each BFR and for each BP when it is providing congestion-loss-free services such as Rate Controlled Service Disciplines [RCSD94]. Resource availability could be provided for example via routing protocol information, but may also be obtained via a BIER-TE control protocol such as NETCONF or any other protocol commonly used by a Controller to understand the resources of the network it operates on. The resource usage of the BIER-TE traffic admitted by the BIER-TE controller can be solely tracked on the BIER-TE Controller based on local accounting as long as no `forward_routed()` adjacencies are used (see Section 4.2.1 for the definition of `forward_routed()` adjacencies). When `forward_routed()` adjacencies are used, the paths selected by the underlying routing protocol need to be tracked as well.

Resource management has implications on the forwarding plane beyond the BIER-TE defined steering of packets. This includes allocation of buffers to guarantee the worst case requirements of admitted RCSD traffic and potentially policing and/or rate-shaping mechanisms, typically done via various forms of queuing. This level of resource control, while optional, is important in networks that wish to support congestion management policies to control or regulate the offered traffic to deliver different levels of service and alleviate congestion problems, or those networks that wish to control latencies experienced by specific traffic flows.

#### 4. BIER-TE Forwarding

##### 4.1. The BIER-TE Bit Index Forwarding Table (BIFT)

The BIER-TE BIFT is the equivalent to the BIER BIFT for (non-TE) BIER. It exists on every BFR running BIER-TE. For every BIER sub-domain (SD) in use for BIER-TE, it is a table as shown in Figure 4. That example BIFT assumes a BSL of 8 bit positions (BPs) in the packets BitString. As in [RFC8279] this BSL is purely used for the example and not a BIER/BIER-TE supported BSL (minimum BSL is 64).

A BIER-TE BIFT compares to a BIER BIFT as shown in [RFC8279] as follows.

In both BIER and BIER-TE, BIFT rows/entries are indexed in their respective BIER pseudocode ([RFC8279] Section 6.5) and BIER-TE pseudocode (Section 4.4) by the BIFT-index derived from the packets SI, BSL and the one bit position of the packets BitString (BP) addressing the BIFT row:  $\text{BIFT-index} = \text{SI} * \text{BSL} + \text{BP} - 1$ . BP within a BitString are numbered from 1 to BSL, hence the - 1 offset when converting to a BIFT-index. This document also uses the notion SI:BP to indicate BIFT rows, [RFC8279] uses the equivalent notion SI:BitString, where the BitString is filled with only the BP for the BIFT row.

In BIER, each BIFT-index addresses one BFER by its BFR-id = BIFT-index + 1 and is populated on each BFR with the next-hop "BFR Neighbor" (BFR-NBR) towards that BFER.

In BIER-TE, each BIFT-index and therefore SI:BP indicates one or more adjacencies between BFRs in the topology and is only populated with those adjacencies forwarding entries on the BFR that is the upstream for these adjacencies. The BIFT entry are empty on all other BFRs.

In BIER, each BIFT row also requires a "Forwarding Bit Mask" (F-BM) entry for BIER forwarding rules. In BIER-TE forwarding, F-BM is not required, but can be used when implementing BIER-TE on forwarding hardware derived from BIER forwarding, that must use F-BM. This is discussed in the first BIER-TE forwarding pseudocode in Section 4.4.

BIFT-index (SI:BP)	(FBM)	Adjacencies: <empty> or one or more per entry
BIFT indices for Packets with SI=0		
0 (0:1)	...	forward_connected(interface,neighbor{,DNC})
1 (0:2)	...	forward_connected(interface,neighbor{,DNC})
	...	forward_connected(interface,neighbor{,DNC})
...	...	...
4 (0:5)	...	local_decap({VRF})
5 (0:6)	...	forward_routed({VRF},l3-neighbor)
6 (0:7)	...	<empty>
7 (0:8)	...	ECMP((adjacency1,...adjacencyN){,seed})
BIFT indices for BitString/Packet with SI=1		
9 (1:1)	...	...
...	...	...

BIER-TE Bit Index Forwarding Table (BIFT)

Figure 4: BIER-TE BIFT with different adjacencies

The BIFT is configured for the BIER-TE data plane of a BFR by the BIER-TE Controller through an appropriate protocol and data-model. The BIFT is then used to forward packets, according to the rules specified in the BIER-TE Forwarding Procedures.

Note that a BIFT index (SI:BP) may be populated in the BIFT of more than one BFR to save BPs. See Section 5.1.6 for an example of how a BIER-TE controller could assign BPs to (logical) adjacencies shared across multiple BFRs, Section 5.1.3 for an example of assigning the same BP to different adjacencies, and Section 5.1.9 for general guidelines regarding re-use of BPs across different adjacencies.

{VRF} indicates the Virtual Routing and Forwarding context into which the BIER payload is to be delivered. This is optional and depends on the multicast flow overlay.

#### 4.2. Adjacency Types

#### 4.2.1. Forward Connected

A "forward\_connected()" adjacency is towards a directly connected BFR neighbor using an interface address of that BFR on the connecting interface. A forward\_connected() adjacency does not route packets but only L2 forwards them to the neighbor.

Packets sent to an adjacency with "DoNotClear" (DNC) set in the BIFT MUST NOT have the bit position for that adjacency cleared when the BFR creates a copy for it. The bit position will still be cleared for copies of the packet made towards other adjacencies. This can be used for example in ring topologies as explained in Section 5.1.6.

For protection against loops from misconfiguration (see Section 5.2.1), DNC is only permissible for forward\_connected() adjacencies. No need or benefit of DNC for other type of adjacencies was identified and their risk was not analyzed.

#### 4.2.2. Forward Routed

A "forward\_routed()" adjacency is an adjacency towards a BFR that uses a (tunneling) encapsulation which will cause the packet to be forwarded by the routing underlay toward the adjacent BFR. This can leverage any feasible encapsulation, such as MPLS or tunneling over IP/IPv6, as long as the BIER-TE packet can be identified as a payload. This identification can either rely on the BIER/BIER-TE co-existence mechanisms described in Section 4.3, or by explicit support for a BIER-TE payload type in the tunneling encapsulation.

forward\_routed() adjacencies are necessary to pass BIER-TE traffic across non BIER-TE capable routers or to minimize the number of required BP by tunneling over (BIER-TE capable) routers on which neither replication nor path-steering is desired, or simply to leverage path redundancy and FRR of the routing underlay towards the next BFR. They may also be useful to a multi-subnet adjacent BFR to leverage the routing underlay ECMP independent of BIER-TE ECMP (Section 4.2.3).

#### 4.2.3. ECMP

(non-TE) BIER ECMP is tied to the BIER BIFT processing semantic and is therefore not directly usable with BIER-TE.

A BIER-TE "Equal Cost Multipath" (ECMP()) adjacency as shown in Figure 4 for BIFT-index 7 has a list of two or more non-ECMP adjacencies as parameters and an optional seed parameter. When a BIER-TE packet is copied onto such an ECMP() adjacency, an implementation specific so-called hash function will select one out

of the list's adjacencies to which the packet is forwarded. If the packet's encapsulation contains an entropy field, the entropy field SHOULD be respected; two packets with the same value of the entropy field SHOULD be sent on the same adjacency. The seed parameter allows to design hash functions that are easy to implement at high speed without running into polarization issues across multiple consecutive ECMP hops. See Section 5.1.7 for more explanations.

#### 4.2.4. Local Decap(sulation)

A "local\_decap()" adjacency passes a copy of the payload of the BIER-TE packet to the protocol ("NextProto") within the BFR (IPv4/IPv6, Ethernet,...) responsible for that payload according to the packet header fields. A local\_decap() adjacency turns the BFR into a BFER for matching packets. Local\_decap() adjacencies require the BFER to support routing or switching for NextProto to determine how to further process the packet.

#### 4.3. Encapsulation / Co-existence with BIER

Specifications for BIER-TE encapsulation are outside the scope of this document. This section gives explanations and guidelines.

Like [RFC8279], handling of "Maximum Transmission Unit" (MTU) limitations is outside the scope of this document and instead part of the BIER-TE packet encapsulation and/or flow overlay. See for example [RFC8296], Section 3. It applies equally to BIER-TE as it does to BIER.

Because a BFR needs to interpret the BitString of a BIER-TE packet differently from a (non-TE) BIER packet, it is necessary to distinguish BIER from BIER-TE packets. In the BIER encapsulation [RFC8296], the BIFT-id field of the packet indicates the BIFT of the packet. BIER and BIER-TE can therefore be run simultaneously, when the BIFT-id address space is shared across BIER BIFT and BIER-TE BIFT. Partitioning the BIFT-id address space is subject to BIER-TE/BIER control plane procedures.

When [RFC8296] is used for BIER with MPLS, BIFT-id address ranges can be dynamically allocated from MPLS label space only for the set of actually used SD:BSL BIFT. This allows to also allocate non-overlapping label ranges for BIFT-id that are to be used with BIER-TE BIFTs.

With MPLS, it is also possible to reuse the same SD space for both BIER-TE and BIER, so that the same SD has both a BIER BIFT with a corresponding range of BIFT-ids and disjoint BIER-TE BIFTs with a non-overlapping range of BIFT-ids.



When a fixed mapping from BSL, SD and SI to BIFT-id is used which does not explicitly partition the BIFT-id space between BIER and BIER-TE, such as proposed for non-MPLS forwarding with [RFC8296] encapsulation in [I-D.ietf-bier-non-mpls-bift-encoding] revision 04, section 5, then it is necessary to allocate disjoint SDs to BIER and BIER-TE BIFTs so that both can be addressed by the BIFT-ids. The encoding proposed in section 6. of the same document does not statically encode BSL or SD into the BIFT-id, but allows for a mapping, and hence could provide for the same freedom as when MPLS is being used (same or different SD for BIER/BIER-TE).

forward\_routed() requires an encapsulation that permits to direct unicast encapsulated BIER-TE packets to a specific interface address on a target BFR. With MPLS encapsulation, this can simply be done via a label stack with that addresses label as the top label - followed by the label assigned to the (BSL,SD,SI) BitString. With non-MPLS encapsulation, some form of IP encapsulation would be required (for example IP/GRE).

The encapsulation used for forward\_routed() adjacencies can equally support existing advanced adjacency information such as "loose source routes" via e.g. MPLS label stacks or appropriate header extensions (e.g. for IPv6).

#### 4.4. BIER-TE Forwarding Pseudocode

The following pseudocode, Figure 5, for BIER-TE forwarding is based on the (non-TE) BIER forwarding pseudocode of [RFC8279], section 6.5 with one modification.

```
void ForwardBitMaskPacket_withTE (Packet)
{
    SI=GetPacketSI(Packet);
    Offset=SI*BitStringLength;
    for (Index = GetFirstBitPosition(Packet->BitString); Index ;
        Index = GetNextBitPosition(Packet->BitString, Index)) {
        F-BM = BIFT[Index+Offset]->F-BM;
        if (!F-BM) continue;                                [3]
        BFR-NBR = BIFT[Index+Offset]->BFR-NBR;
        PacketCopy = Copy(Packet);
        PacketCopy->BitString &= F-BM;                        [2]
        PacketSend(PacketCopy, BFR-NBR);
        // The following must not be done for BIER-TE:
        // Packet->BitString &= ~F-BM;                          [1]
    }
}
```

Figure 5: BIER-TE Forwarding Pseudocode for required functions,  
based on BIER Pseudocode

In step [2], the F-BM is used to clear bit(s) in PacketCopy. This step exists in both BIER and BIER-TE, but the F-BMs need to be populated differently for BIER-TE than for BIER for the desired clearing.

In BIER, multiple bits of a BitString can have the same BFR-NBR. When a received packets BitString has more than one of those bits set, the BIER replication logic has to avoid that more than one PacketCopy is sent to that BFR-NBR ([1]). Likewise, the PacketCopy sent to a BFR-NBR must clear all bits in its BitString that are not routed across BFR-NBR. This protects against BIER replication on any possible further BFR to create duplicates ([2]).

To solve both [1] and [2] for BIER, the F-BM of each bit index needs to have all bits set that this BFR wants to route across BFR-NBR. [2] clears all other bits in PacketCopy->BitString, and [1] clears those bits from Packet->BitString after the first PacketCopy.

In BIER-TE, a BFR-NBR in this pseudocode is an adjacency, forward\_connected(), forward\_routed() or local\_decap(). There is no need for [2] to suppress duplicates in the way BIER does because in general, different BP would never have the same adjacency. If a BIER-TE controller actually finds some optimization in which this would be desirable, then the controller is also responsible to ensure that only one of those bits is set in any Packet->BitString, unless the controller explicitly wants for duplicates to be created.

The following points describe how the forwarding bit mask (F-BM) for each BP is configured in the BIFT and how this impacts the BitString of the packet being processed with that BIFT:

1. The F-BMs of all BIFT BPs without an adjacency have all their bits clear. This will cause [3] to skip further processing of such a BP.
2. All BIFT BPs with an adjacency (with DNC flag clear) have an F-BM that has only those BPs set for which this BFR does not have an adjacency. This causes [2] to clear all bits from PacketCopy->BitString for which this BFR does have an adjacency.
3. [1] is not performed for BIER-TE. All bit clearing required by BIER-TE is performed by [2].

This Forwarding Pseudocode can support the required BIER-TE forwarding functions (see Section 4.5), `forward_connected()`, `forward_routed()` and `local_decap()`, but not the recommended functions DNC flag and multiple adjacencies per bit nor the optional function, `ECMP()` adjacencies. The DNC flag cannot be supported when using only [1] to mask bits.

The modified and expanded Forwarding Pseudocode in Figure 6 specifies how to support all BIER-TE forwarding functions (required, recommended and optional):

- \* This pseudocode eliminates per-bit F-BM, therefore reducing the size of BIFT state by  $BSL^2 \cdot SI$  and eliminating the need for per-packet-copy BitString masking operations except for adjacencies with the DNC flag set:
  - `AdjacentBits[SI]` are bit positions with a non-empty list of adjacencies in this BFR BIFT. This can be computed whenever the BIER-TE Controller updates (add/removes) adjacencies in the BIFT.
  - The BFR needs to create packet copies for these adjacent bits when they are set in the packets BitString. This set of bits is calculated in `PktAdjacentBits`.
  - All bit positions to which the BFR creates copies have to be cleared in packet copies to avoid loops. This is done by masking the BitString of the packet with `~AdjacentBits[SI]`. When an adjacency has DNC set, this bit position is set again only for the packet copy towards that bit position.
- \* BIFT entries may contain more than one adjacency in support of specific configurations such as Section 5.1.5. The code therefore includes a loop over these adjacencies.
- \* The `ECMP()` adjacency is shown. Its parameters are a seed and a `ListOfAdjacencies` from which one is picked.
- \* The `forward_connected()`, `forward_routed()`, `local_decap()` adjacencies are shown with their parameters.

```

void ForwardBitMaskPacket_withTE (Packet)
{
    SI = GetPacketSI(Packet);
    Offset = SI * BitStringLength;
    // Determine adjacent bits in the Packets BitString
    PktAdjacentBits = Packet->BitString & AdjacentBits[SI];

    // Clear adjacent bits in Packet header to avoid loops
    Packet->BitString &= ~AdjacentBits[SI];

    // Loop over PktAdjacentBits to create packet copies
    for (Index = GetFirstBitPosition(PktAdjacentBits); Index ;
        Index = GetNextBitPosition(PktAdjacentBits, Index)) {
        for adjacency in BIFT[Index+Offset]->Adjacencies {
            if(adjacency.type == ECMP(ListOfAdjacencies,seed) ) {
                I = ECMP_hash(sizeof(ListOfAdjacencies),
                               Packet->Entropy,seed);
                adjacency = ListOfAdjacencies[I];
            }
            PacketCopy = Copy(Packet);
            switch(adjacency.type) {
                case forward_connected(interface,neighbor,DNC):
                    if(DNC)
                        PacketCopy->BitString |= 1<<(Index-1);
                    SendToL2Unicast(PacketCopy,interface,neighbor);

                case forward_routed({VRF},l3-neighbor):
                    SendToL3(PacketCopy,{VRF},l3-neighbor);

                case local_decap({VRF},neighbor):
                    DecapBierHeader(PacketCopy);
                    PassTo(PacketCopy,{VRF},Packet->NextProto);
            }
        }
    }
}

```

Figure 6: Complete BIER-TE Forwarding Pseudocode for required, recommended and optional functions

#### 4.5. BFR Requirements for BIER-TE forwarding

BFR that support BIER-TE and BIER MUST support configuration that enables BIER-TE instead of (non-TE) BIER forwarding rules for all BIFT of one or more BIER sub-domains. Every BP in a BIER-TE BIFT MUST support to have zero or one adjacency. BIER-TE forwarding MUST support the adjacency types `forward_connected()` with the DNC flag not set, `forward_routed()` and `local_decap()`. As explained in

Section 4.4, these required BIER-TE forwarding functions can be implemented via the same Forwarding Pseudocode as BIER forwarding except for one modification (skipping one masking with F-BM).

BIER-TE forwarding SHOULD support `forward_connected()` adjacencies with a set DNC flag, as this is highly useful to save bits in rings (see Section 5.1.6).

BIER-TE forwarding SHOULD support more than one adjacency on a bit. This allows to save bits in hub and spoke scenarios (see Section 5.1.5).

BIER-TE forwarding MAY support `ECMP()` adjacencies to save bits in ECMP scenarios, see Section 5.1.7 for an example. This is an optional requirement, because for ECMP deployments using BIER-TE one can also leverage ECMP of the routing underlay via `forwarded_routed` adjacencies and/or might prefer to have more explicit control of the path chosen via explicit BP/adjacencies for each ECMP path alternative.

## 5. BIER-TE Controller Operational Considerations

### 5.1. Bit Position Assignments

This section describes how the BIER-TE Controller can use the different BIER-TE adjacency types to define the bit positions of a BIER-TE domain.

Because the size of the BitString limits the size of the BIER-TE domain, many of the options described exist to support larger topologies with fewer bit positions.

#### 5.1.1. P2P Links

On a P2P link that connects two BFRs, the same bit position can be used on both BFRs for the adjacency to the neighboring BFR. A P2P link requires therefore only one bit position.

#### 5.1.2. BFER

Every non-Leaf BFER is given a unique bit position with a `local_decap()` adjacency.

#### 5.1.3. Leaf BFERs

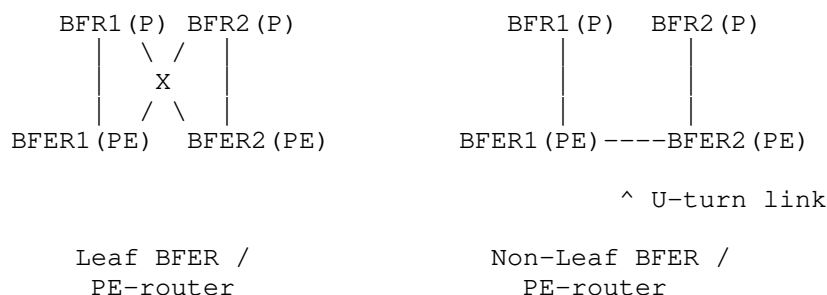


Figure 7: Leaf vs. non-Leaf BFER Example

A leaf BFER is one where incoming BIER-TE packets never need to be forwarded to another BFR but are only sent to the BFER to exit the BIER-TE domain. For example, in networks where Provider Edge (PE) router are spokes connected to Provider (P) routers, those PEs are Leaf BFERs unless there is a U-turn between two PEs.

Consider how redundant disjoint traffic can reach BFER1/BFER2 in Figure 7: When BFER1/BFER2 are Non-Leaf BFER as shown on the right-hand side, one traffic copy would be forwarded to BFER1 from BFR1, but the other one could only reach BFER1 via BFER2, which makes BFER2 a non-Leaf BFER. Likewise, BFER1 is a non-Leaf BFER when forwarding traffic to BFER2. Note that the BFERs in the left-hand picture are only guaranteed to be leaf-BFER by fitting routing configuration that prohibits transit traffic to pass through the BFERs, which is commonly applied in these topologies.

In most situations, leaf-BFER that are to be addressed via the same BitString can share a single bit position for their local\_decap() adjacency in that BitString and therefore save bit positions. On a non-leaf BFER, a received BIER-TE packet may only need to transit the BFER or it may need to also be decapsulated. Whether or not to decapsulate the packet therefore needs to be indicated by a unique bit position populated only on the BIFT of this BFER with a local\_decap() adjacency. On a leaf-BFER, packets never need to pass through; any packet received is therefore usually intended to be decapsulated. This can be expressed by a single, shared bit position that is populated with a local\_decap() adjacency on all leaf-BFER addressed by the BitString.

The possible exception from this leaf-BFER bit position optimization can be cases where the bit position on the prior BIER-TE BFR (which created the packet copy for the leaf-BFER in question) is populated with multiple adjacencies as an optimization, such as in Section 5.1.4 or Section 5.1.5. With either of these two optimizations, the sender of the packet could only control explicitly

whether the packet was to be decapsulated on the leaf-BFER in question, if the leaf-BFER has a unique bit position for its `local_decap()` adjacency.

However, if the bit position is shared across leaf-BFER, and packets are therefore decapsulated potentially unnecessarily, this may still be appropriate if the decapsulated payload of the BIER-TE packet indicates whether or not the packet needs to be further processed/received. This is typically true for example if the payload is IP multicast because IP multicast on a BFER would know the membership state of the IP multicast payload and be able to discard it if the packet was delivered unnecessarily by the BIER-TE layer. If the payload has no such membership indication, and the BFIR wants to have explicit control about which BFER are to receive and decapsulate a packet, then these two optimizations can not be used together with shared bit positions optimization for leaf-BFER.

#### 5.1.4. LANs

In a LAN, the adjacency to each neighboring BFR is given a unique bit position. The adjacency of this bit position is a `forward_connected()` adjacency towards the BFR and this bit position is populated into the BIFT of all the other BFRs on that LAN.

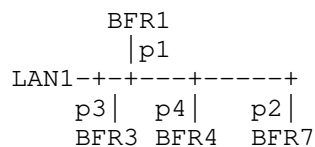


Figure 8: LAN Example

If Bandwidth on the LAN is not an issue and most BIER-TE traffic should be copied to all neighbors on a LAN, then bit positions can be saved by assigning just a single bit position to the LAN and populating the bit position of the BIFTs of each BFRs on the LAN with a list of `forward_connected()` adjacencies to all other neighbors on the LAN.

This optimization does not work in the case of BFRs redundantly connected to more than one LAN with this optimization because these BFRs would receive duplicates and forward those duplicates into the opposite LANs. Adjacencies of such BFRs into their LAN still need a separate bit position.

#### 5.1.5. Hub and Spoke

In a setup with a hub and multiple spokes connected via separate p2p links to the hub, all p2p adjacencies from the hub to the spokes links can share the same bit position. The bit position on the hub's BIFT is set up with a list of `forward_connected()` adjacencies, one for each Spoke.

This option is similar to the bit position optimization in LANs: Redundantly connected spokes need their own bit positions, unless they are themselves Leaf-BFER.

This type of optimized BP could be used for example when all traffic is "broadcast" traffic (very dense receiver set) such as live-TV or many-to-many telemetry including situation-awareness (SA). This BP optimization can then be used to explicitly steer different traffic flows across different ECMP paths in Data-Center or broadband-aggregation networks with minimal use of BPs.

#### 5.1.6. Rings

In L3 rings, instead of assigning a single bit position for every p2p link in the ring, it is possible to save bit positions by setting the "DoNotClear" (DNC) flag on `forward_connected()` adjacencies.

For the rings shown in Figure 9, a single bit position will suffice to forward traffic entering the ring at BFRa or BFRb all the way up to BFR1:

On BFRa, BFRb, BFR30,... BFR3, the bit position is populated with a `forward_connected()` adjacency pointing to the clockwise neighbor on the ring and with DNC set. On BFR2, the adjacency also points to the clockwise neighbor BFR1, but without DNC set.

Handling DNC this way ensures that copies forwarded from any BFR in the ring to a BFR outside the ring will not have the ring bit position set, therefore minimizing the chance to create loops.

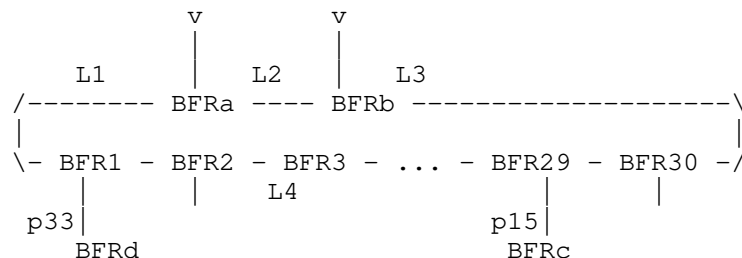




Figure 9: Ring Example

Note that this example only permits for packets intended to make it all the way around the ring to enter it at BFRa and BFRb, and that packets will always travel clockwise. If packets should be allowed to enter the ring at any ring BFR, then one would have to use two ring bit positions. One for each direction: clockwise and counterclockwise.

Both would be set up to stop rotating on the same link, e.g. L1. When the ingress ring BFR creates the clockwise copy, it will clear the counterclockwise bit position because the DNC bit only applies to the bit for which the replication is done. Likewise for the clockwise bit position for the counterclockwise copy. As a result, the ring ingress BFR will send a copy in both directions, serving BFRs on either side of the ring up to L1.

#### 5.1.7. Equal Cost MultiPath (ECMP)

[RFC-Editor: A reviewer (Lars Eggert) noted that the infinite "to use" in the following sentence is not correct. The same was also noted for several other similar instances. The following URL seems to indicate though that this is a per-case decision, which seems undefined: <https://writingcenter.gmu.edu/guides/choosing-between-infinite-and-gerund-to-do-or-doing>. What exactly should be done about this ?].

An ECMP() adjacency allows to use just one BP to deliver packets to one of N adjacencies instead of one BP for each adjacency. In the common example case Figure 10, a link-bundle of three links L1,L2,L3 connects BFR1 and BFR2, and only one BP is used instead of three BP to deliver packets from BFR1 to BFR2.

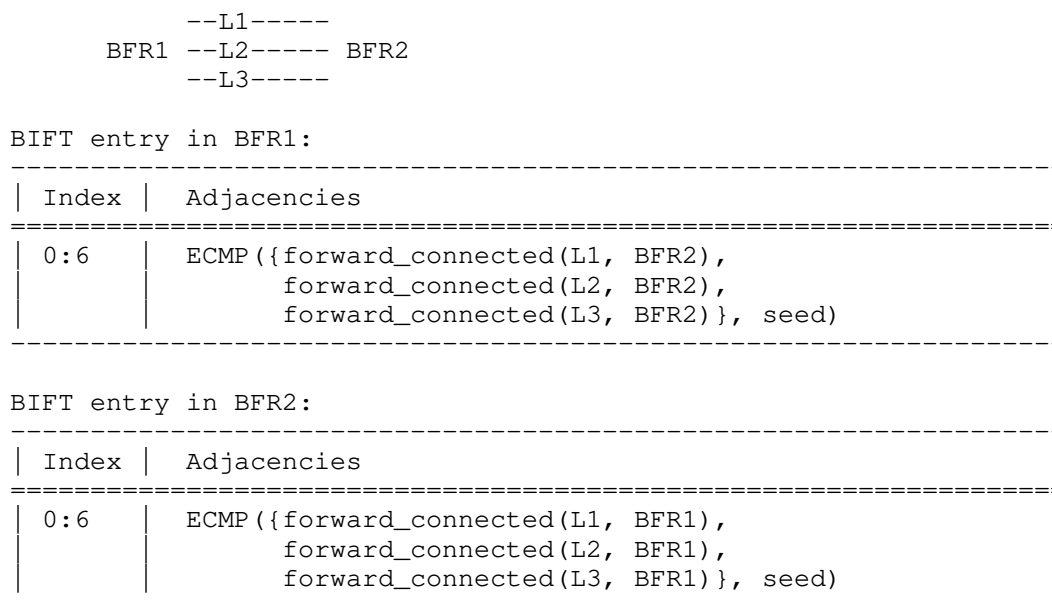


Figure 10: ECMP Example

This document does not standardize any ECMP algorithm because it is sufficient for implementations to document their freely chosen ECMP algorithm. Figure 11 shows an example ECMP algorithm, and would double as its documentation: A BIER-TE controller could determine which adjacency is chosen based on the seed and adjacencies parameters and the packet entropy.

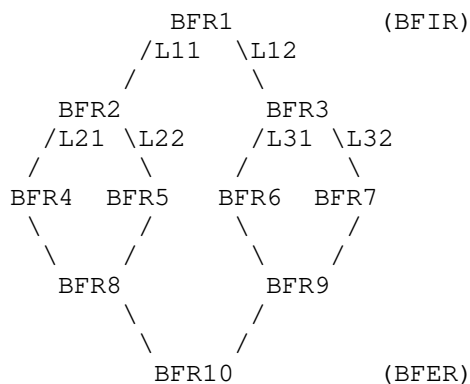
```

forward(packet, ECMP(adj(0), adj(1),... adj(N-1), seed)):
  i = (packet(bier-header-entropy) XOR seed) % N
  forward packet to adj(i)

```

Figure 11: ECMP algorithm Example

In the following example, all traffic from BFR1 towards BFR10 is intended to be ECMP load split equally across the topology. This example is not meant as a likely setup, but to illustrate that ECMP can be used to share BPs not only across link bundles, but also across alternative paths across different transit BFR, and it explains the use of the seed parameter.



BIFT entry in BFR1:

0:6	ECMP({forward_connected(L11, BFR2), forward_connected(L12, BFR3)}, seed1)
-----	--

BIFT entry in BFR2:

0:7	ECMP({forward_connected(L21, BFR4), forward_connected(L22, BFR5)}, seed1)
-----	--

BIFT entry in BFR3:

0:7	ECMP({forward_connected(L31, BFR6), forward_connected(L32, BFR7)}, seed1)
-----	--

BIFT entry in BFR4, BFR5:

0:8	forward_connected(Lxx, BFR8)	xx differs on BFR4/BFR5
-----	------------------------------	-------------------------

BIFT entry in BFR6, BFR7:

0:8	forward_connected(Lxx, BFR9)	xx differs on BFR6/BFR7
-----	------------------------------	-------------------------

BIFT entry in BFR8, BFR9:

0:9	forward_connected(Lxx, BFR10)	xx differs on BFR8/BFR9
-----	-------------------------------	-------------------------

Figure 12: Polarization Example

Note that for the following discussion of ECMP, only the BIFT ECMP adjacencies on BFR1, BFR2, BFR3 are relevant. The re-use of BP across BFR in this example is further explained in Section 5.1.9 below.

With the setup of ECMP in the topology above, traffic would not be equally load-split. Instead, links L22 and L31 would see no traffic at all: BFR2 will only see traffic from BFR1 for which the ECMP hash in BFR1 selected the first adjacency in the list of 2 adjacencies given as parameters to the ECMP. It is link L11-to-BFR2. BFR2 performs again ECMP with two adjacencies on that subset of traffic using the same seed1, and will therefore again select the first of its two adjacencies: L21-to-BFR4. And therefore L22 and BFR5 sees no traffic. Likewise for L31 and BFR6.

This issue in BFR2/BFR3 is called polarization. It results from the re-use of the same hash function across multiple consecutive hops in topologies like these. To resolve this issue, the ECMP() adjacency on BFR1 can be set up with a different seed2 than the ECMP() adjacencies on BFR2/BFR3. BFR2/BFR3 can use the same hash because packets will not sequentially pass across both of them. Therefore, they can also use the same BP 0:7.

Note that ECMP solutions outside of BIER often hide the seed by auto-selecting it from local entropy such as unique local or next-hop identifiers. Allowing the BIER-TE Controller to explicitly set the seed gives the ability for it to control same/different path selection across multiple consecutive ECMP hops.

#### 5.1.8. Forward Routed adjacencies

##### 5.1.8.1. Reducing bit positions

Forward\_routed() adjacencies can reduce the number of bit positions required when the path steering requirement is not hop-by-hop explicit path selection, but loose-hop selection. Forward\_routed() adjacencies can also allow to operate BIER-TE across intermediate hop routers that do not support BIER-TE.

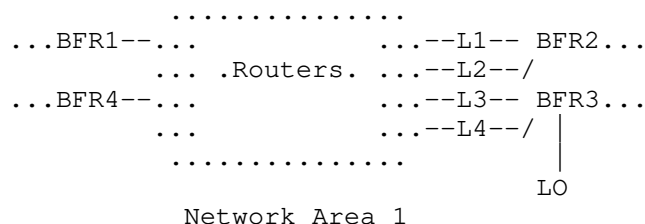


Figure 13: Forward Routed Adjacencies Example

Assume the requirement in Figure 13 is to explicitly steer traffic flows that have arrived at BFR1 or BFR4 via a path in the routing underlay "Network Area 1" to one of the following three next segments: (1) BFR2 via link L1, (2) BFR2 via link L2, or (3) via BFR3 and then not caring whether the packet is forwarded via L3 or L4.

To enable this, both BFR1 and BFR4 are set up with a `forward_routed` adjacency bit position towards an address of BFR2 on link L1, another `forward_routed()` bit position towards an address of BFR2 on link L2 and a third `forward_routed()` bit position towards a node address L0 of BFR3.

#### 5.1.8.2. Supporting nodes without BIER-TE

`Forward_routed()` adjacencies also enable incremental deployment of BIER-TE. Only the nodes through which BIER-TE traffic needs to be steered - with or without replication - need to support BIER-TE. Where they are not directly connected to each other, `forward_routed` adjacencies are used to pass over non BIER-TE enabled nodes.

#### 5.1.9. Reuse of bit positions (without DNC)

Bit positions can be re-used across multiple BFRs to minimize the number of BP needed. This happens when adjacencies on multiple BFRs use the DNC flag as described above, but it can also be done for non-DNC adjacencies. This section only discusses this non-DNC case.

Because BP are cleared when passing a BFR with an adjacency for that BP, reuse of BP across multiple BFRs does not introduce any problems with duplicates or loops that do not also exist when every adjacency has a unique BP. Instead, the challenge when reusing BP is whether it allows to still achieve the desired Tree Engineering goals.

BP cannot be reused across two BFRs that would need to be passed sequentially for some path: The first BFR will clear the BP, so those paths cannot be built. BP can be set across BFR that would (A) only occur across different paths or (B) across different branches of the same tree.

An example of (A) was given in Figure 12, where BP 0:7, BP 0:8 and BP 0:9 are each reused across multiple BFRs because a single packet/path would never be able to reach more than one BFR sharing the same BP.

Assume the example was changed: BFR1 has no `ECMP()` adjacency for BP 0:6, but instead BP 0:5 with `forward_connected()` to BFR2 and BP 0:6 with `forward_connected()` to BFR3. Packets with both BP 0:5 and BP

0:6 would now be able to reach both BFR2 and BFR3 and the still existing re-use of BP 0:7 between BFR2 and BFR3 is a case of (B) where reuse of BP is perfect because it does not limit the set of useful path choices:

If instead of reusing BP 0:7, BFR3 used a separate BP 0:10 for its ECMP() adjacency, no useful additional path steering options would be enabled. If duplicates at BFR10 where undesirable, this would be done by not setting BP 0:5 and BP 0:6 for the same packet. If the duplicates where desirable (e.g.: resilient transmission), the additional BP 0:10 would also not render additional value.

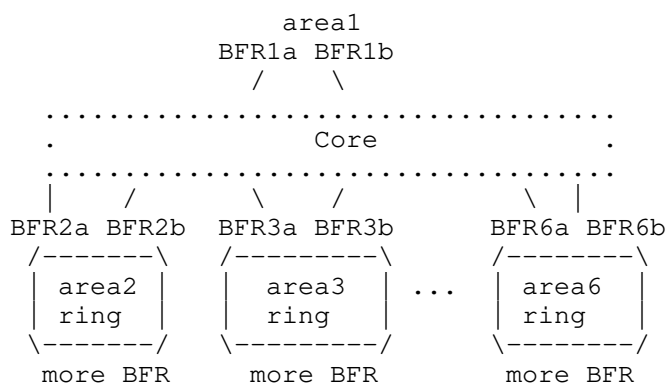


Figure 14: Reuse of BP

Reuse may also save BPs in larger topologies. Consider the topology shown in Figure 14. A BFIR/sender (e.g.: video headend) is attached to area 1, and area 2...6 contain receivers/BFER. Assume each area had a distribution ring, each with two BPs to indicate the direction (as explained before). These two BPs could be reused across the 5 areas. Packets would be replicated through other BPs for the Core to the desired subset of areas, and once a packet copy reaches the ring of the area, the two ring BPs come into play. This reuse is a case of (B), but it limits the topology choices: Packets can only flow around the same direction in the rings of all areas. This may or may not be acceptable based on the desired path steering options: If resilient transmission is the path engineering goal, then it is likely a good optimization, if the bandwidth of each ring was to be optimized separately, it would not be a good limitation.

#### 5.1.10. Summary of BP optimizations

This section reviewed a range of techniques by which a BIER-TE Controller can create a BIER-TE topology in a way that minimizes the number of necessary BPs.

Without any optimization, a BIER-TE Controller would attempt to map the network subnet topology 1:1 into the BIER-TE topology and every subnet adjacent neighbor requires a `forward_connected()` BP and every BFER requires a `local_decap()` BP.

The optimizations described are then as follows:

- \* P2P links require only one BP (Section 5.1.1).
- \* All leaf-BFER can share a single `local_decap()` BP (Section 5.1.3).
- \* A LAN with N BFR needs at most N BP (one for each BFR). It only needs one BP for all those BFR that are not redundantly connected to multiple LANs (Section 5.1.4).
- \* A hub with p2p connections to multiple non-leaf-BFER spokes can share one BP to all spokes if traffic can be flooded to all spokes, e.g.: because of no bandwidth concerns or dense receiver sets (Section 5.1.5).
- \* Rings of BFR can be built with just two BP (one for each direction) except for BFR with multiple ring connections - similar to LANs (Section 5.1.6).
- \* ECMP() adjacencies to N neighbors can replace N BP with 1 BP. Multihop ECMP can avoid polarization through different seeds of the ECMP algorithm (Section 5.1.7).
- \* `Forward_routed()` adjacencies allow to "tunnel" across non-BIER-TE capable routers and across BIER-TE capable routers where no traffic-steering or replications are required (Section 5.1.8).
- \* BP can generally be reused across a set of nodes where it can be guaranteed that no path will ever need to traverse more than one node of the set. Depending on scenario, this may limit the feasible path steering options (Section 5.1.9).

Note that the described list of optimizations is not exhaustive. Especially when the set of required path steering choices is limited and the set of possible subsets of BFERs that should be able to receive traffic is limited, further optimizations of BP are possible. The hub and spoke optimization is a simple example of such traffic pattern dependent optimizations.

## 5.2. Avoiding duplicates and loops

## 5.2.1. Loops

Whenever BIER-TE creates a copy of a packet, the BitString of that copy will have all bit positions cleared that are associated with adjacencies on the BFR. This inhibits looping of packets. The only exception are adjacencies with DNC set.

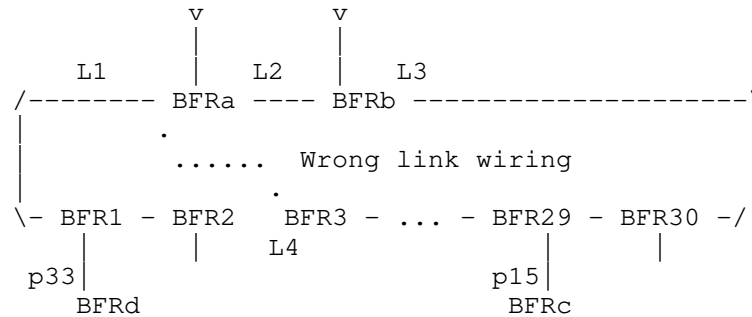


Figure 15: Miswired Ring Example

With DNC set, looping can happen. Consider in Figure 15 that link L4 from BFR3 is (inadvertently) plugged into the L1 interface of BFRa (instead of BFR2). This creates a loop where the rings clockwise bit position is never cleared for copies of the packets traveling clockwise around the ring.

To inhibit looping in the face of such physical misconfiguration, only `forward_connected()` adjacencies are permitted to have DNC set, and the link layer port unique unicast destination address of the adjacency (e.g. MAC address) protects against closing the loop. Link layers without port unique link layer addresses should not be used with the DNC flag set.

## 5.2.2. Duplicates

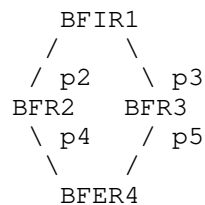


Figure 16: Duplicates Example



Duplicates happen when the graph expressed by a BitString is not a tree but redundantly connecting BFRs with each other. In Figure 16, a BitString of p2,p3,p4,p5 would result in duplicate packets to arrive on BFER4. The BIER-TE Controller must therefore ensure to only create BitStrings that are trees.

When links are incorrectly physically re-connected before the BIER-TE Controller updates BitStrings in BFIRs, duplicates can happen. Like loops, these can be inhibited by link layer addressing in `forward_connected()` adjacencies.

If interface or loopback addresses used in `forward_routed()` adjacencies are moved from one BFR to another, duplicates can equally happen. Such re-addressing operations must be coordinated with the BIER-TE Controller.

### 5.3. Managing SI, sub-domains and BFR-ids

When the number of bits required to represent the necessary hops in the topology and BFER exceeds the supported BitStringLength (BSL), multiple SIs and/or sub-domains must be used. This section discusses how.

BIER-TE forwarding does not require the concept of BFR-id, but routing underlay, flow overlay and BIER headers may. This section also discusses how BFR-ids can be assigned to BFIR/BFER for BIER-TE.

#### 5.3.1. Why SI and sub-domains

For (non-TE) BIER and BIER-TE forwarding, the most important result of using multiple SI and/or sub-domains is the same: Packets that need to be sent to BFERs in different SIs or sub-domains require different BIER packets: each one with a BitString for a different (SI,sub-domain) combination. Each such BitString uses one BSL sized SI block in the BIFT of the sub-domain. We call this a BIFT:SI (block).

For BIER and BIER-TE forwarding themselves there is also no difference whether different SIs and/or sub-domains are chosen, but SI and sub-domain have different purposes in the BIER architecture shared by BIER-TE. This impacts how operators are managing them and how especially flow overlays will likely use them.

By default, every possible BFIR/BFER in a BIER network would likely be given a BFR-id in sub-domain 0 (unless there are > 64k BFIR/BFER).

If there are different flow services (or service instances) requiring replication to different subsets of BFERs, then it will likely not be possible to achieve the best replication efficiency for all of these service instances via sub-domain 0. Ideal replication efficiency for N BFER exists in a sub-domain if they are split over not more than  $\text{ceiling}(N/\text{BitStringLength})$  SI.

If service instances justify additional BIER:SI state in the network, additional sub-domains will be used: BFIR/BFER are assigned BFR-id in those sub-domains and each service instance is configured to use the most appropriate sub-domain. This results in improved replication efficiency for different services.

Even if creation of sub-domains and assignment of BFR-id to BFIR/BFER in those sub-domains is automated, it is not expected that individual service instances can deal with BFER in different sub-domains. A service instance may only support configuration of a single sub-domain it should rely on.

To be able to easily reuse (and modify as little as possible) existing BIER procedures including flow-overlay and routing underlay, when BIER-TE forwarding is added, we therefore reuse SI and sub-domain logically in the same way as they are used in BIER: All necessary BFIR/BFER for a service use a single BIER-TE BIFT and are split across as many SIs as necessary (see Section 5.3.2). Different services may use different sub-domains that primarily exist to provide more efficient replication (and for BIER-TE desirable path steering) for different subsets of BFIR/BFER.

#### 5.3.2. Assigning bits for the BIER-TE topology

In BIER, BitStrings only need to carry bits for BFERs, which leads to the model that BFR-ids map 1:1 to each bit in a BitString.

In BIER-TE, BitStrings need to carry bits to indicate not only the receiving BFER but also the intermediate hops/links across which the packet must be sent. The maximum number of BFER that can be supported in a single BitString or BIFT:SI depends on the number of bits necessary to represent the desired topology between them.

"Desired" topology because it depends on the physical topology, and on the desire of the operator to allow for explicit path steering across every single hop (which requires more bits), or reducing the number of required bits by exploiting optimizations such as unicast (`forward_routed()`), ECMP() or flood (DNC) over "uninteresting" sub-parts of the topology - e.g. parts where different trees do not need to take different paths due to path steering reasons.

The total number of bits to describe the topology vs. the number of BFERs in a BIFT:SI can range widely based on the size of the topology and the amount of alternative paths in it. In a BIER-TE topology crafted by a BIER-TE expert, the higher the percentage of non-BFER bits, the higher the likelihood, that those topology bits are not just BIER-TE overhead without additional benefit, but instead that they will allow to express desirable path steering alternatives.

### 5.3.3. Assigning BFR-id with BIER-TE

BIER-TE forwarding does not use the BFR-id, nor does it require for the BFIR-id field of the BIER header to be set to a particular value. However, other parts of a BIER-TE deployment may need a BFR-id, specifically multicast flow overlay signaling and multicast flow overlay packet disposition, and in that case BFRs need to also have BFR-ids for BIER-TE SDs.

For example, for BIER overlay signaling, BFIRs need to have a BFR-id, because this BFIR BFR-id is carried in the BFIR-id field of the BIER header to indicate to the overlay signaling on the receiving BFER which BFIR originated the packet.

In BIER,  $\text{BFR-id} = \text{SI} * \text{BSL} + \text{BP}$ , such that the SI and BP of a BFER can be calculated from the BFR-id and vice versa. This also means that every BFR with a BFR-id has a reserved BP in an SI, even if that is not necessary for BIER forwarding, because the BFR may never be a BFER but only a BFIR.

In BIER-TE, for a non-leaf BFER, there is usually a single BP for that BFER with a `local_decap()` adjacency on the BFER. The BFR-id for such a BFER can therefore be determined using the same procedure as in (non-TE) BIER:  $\text{BFR-id} = \text{SI} * \text{BSL} + \text{BP}$ .

As explained in Section 5.1.3, leaf BFERs do not need such a unique `local_decap()` adjacency. Likewise, BFIRs that are not also BFERs may not have a unique `local_decap()` adjacency either. For all those BFIRs and (leaf) BFERs, the controller needs to determine unique BFR-ids that do not collide with the BFR-ids derived from the non-leaf BFER `local_decap()` BPs.

While this document defines no requirements on how to allocate such BFR-id, a simple option is to derive it from the (SI,BP) of an adjacency that is unique to the BFR in question. For a BFIR this can be the first adjacency only populated on this BFIR, for a leaf-BFER, this could be the first BP with an adjacency towards that BFER.

#### 5.3.4. Mapping from BFR to BitStrings with BIER-TE

In BIER, applications of the flow overlay on a BFIR can calculate the (SI,BP) of a BFER from the BFR-id of the BFER and can therefore easily determine the BitStrings for a BIER packet to a set of BFERs with known BFR-ids.

In BIER-TE this mapping needs to be equally supported for flow overlays. This section outlines two core options, based on what type of Tree Engineering the BIER-TE controller needs to perform for a particular application.

"Independent branches": For a given flow overlay instance, the branches from a BFIR to every BFER are calculated by the BIER-TE controller to be independent of the branches to any other BFER. Shortest path trees are the most common examples of trees with independent branches.

"Interdependent branches": When a BFER is added or deleted from a particular distribution tree, the BIER-TE controller has to recalculate the branches to other BFER, because they may need to change. Steiner trees are examples of interdependent branch trees.

If "independent branches" are used, the BIER-TE Controller can signal to the BFIR flow overlay for every BFER an SI:BitString that represents the branch to that BFER. The flow overlay on the BFIR can then independently of the controller calculate the SI:BitString for all desired BFERs by OR'ing their BitStrings. This allows for flow overlay applications to operate independently of the controller whenever it needs to determine which subset of BFERs need to receive a particular packet.

If "interdependent branches" are required, the application would need to inquire the SI:BitString for a given set of BFER whenever the set changes.

Note that in either case (unlike in BIER), the bits may need to change upon link/node failure/recovery, network expansion and network resource consumption by other traffic as part of traffic engineering goals (e.g.: re-optimization of lower priority traffic flows). Interactions between such BFIR applications and the BIER-TE Controller do therefore need to support dynamic updates to the SI:BitStrings.

Communications between the BFIR flow overlay and the BIER-TE controller requires some way to identify the BFER. If BFR-ids are used in the deployment, as outlined in Section 5.3.3, then those are the natural BFR identifier. If BFR-ids are not used, then any other unique identifier, such as the BFR-prefix of the BFR ([RFC8279]) could be used.

#### 5.3.5. Assigning BFR-ids for BIER-TE

It is not currently determined if a single sub-domain could or should be allowed to forward both (non-TE) BIER and BIER-TE packets. If this should be supported, there are two options:

- A. BIER and BIER-TE have different BFR-id in the same sub-domain. This allows higher replication efficiency for BIER because their BFR-id can be assigned sequentially, while the BitStrings for BIER-TE will have also the additional bits for the topology. There is no relationship between a BFR BIER BFR-id and its BIER-TE BFR-id.
- B. BIER and BIER-TE share the same BFR-id. The BFR-ids are assigned as explained above for BIER-TE and simply reused for BIER. The replication efficiency for BIER will be as low as that for BIER-TE in this approach.

#### 5.3.6. Example bit allocations

##### 5.3.6.1. With BIER

Consider a network setup with a BSL of 256 for a network topology as shown in Figure 17. The network has 6 areas, each with 170 BFERs, connecting via a core with 4 (core) BFRs. To address all BFERs with BIER, 4 SIs are required. To send a BIER packet to all BFER in the network, 4 copies need to be sent by the BFIR. On the BFIR it does not make a difference how the BFR-ids are allocated to BFER in the network, but for efficiency further down in the network it does make a difference.

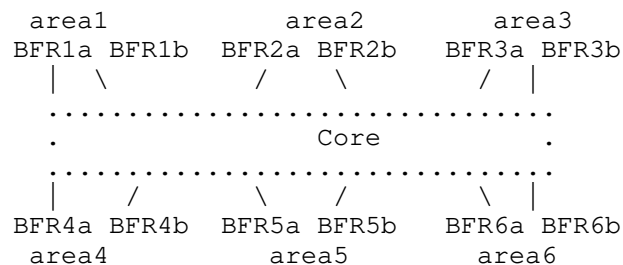


Figure 17: Scaling BIER-TE bits by reuse

With random allocation of BFR-id to BFER, each receiving area would (most likely) have to receive all 4 copies of the BIER packet because there would be BFR-id for each of the 4 SIs in each of the areas. Only further towards each BFER would this duplication subside - when each of the 4 trees runs out of branches.

If BFR-ids are allocated intelligently, then all the BFER in an area would be given BFR-id with as few as possible different SIs. Each area would only have to forward one or two packets instead of 4.

Given how networks can grow over time, replication efficiency in an area will then also go down over time when BFR-ids are only allocated sequentially, network wide. An area that initially only has BFR-id in one SI might end up with many SIs over a longer period of growth. Allocating SIs to areas with initially sufficiently many spare bits for growths can help to alleviate this issue. Or renumber BFERs after network expansion. In this example one may consider to use 6 SIs and assign one to each area.

This example shows that intelligent BFR-id allocation within at least sub-domain 0 can even be helpful or even necessary in BIER.

#### 5.3.6.2. With BIER-TE

In BIER-TE one needs to determine a subset of the physical topology and attached BFERs so that the "desired" representation of this topology and the BFER fit into a single BitString. This process needs to be repeated until the whole topology is covered.

Once bits/SIs are assigned to topology and BFERs, BFR-id is just a derived set of identifiers from the operator/BIER-TE Controller as explained above.

Every time that different sub-topologies have overlap, bits need to be repeated across the BitStrings, increasing the overall amount of bits required across all BitString/SIs. In the worst case, one assigns random subsets of BFERs to different SIs. This will result in an outcome much worse than in (non-TE) BIER: It maximizes the amount of unnecessary topology overlap across SI and therefore reduces the number of BFER that can be reached across each individual SI. Intelligent BFER to SI assignment and selecting specific "desired" subtopologies can minimize this problem.

To set up BIER-TE efficiently for the topology of Figure 17, the following bit allocation method can be used. This method can easily be expanded to other, similarly structured larger topologies.

Each area is allocated one or more SIs depending on the number of future expected BFERs and number of bits required for the topology in the area. In this example, 6 SIs, one per area.

In addition, we use 4 bits in each SI: bia, bib, bea, beb: (b)it (i)ngress (a), (b)it (i)ngress (b), (b)it (e)gress (a), (b)it (e)gress (b). These bits will be used to pass BIER packets from any BFIR via any combination of ingress area a/b BFR and egress area a/b BFR into a specific target area. These bits are then set up with the right forward\_routed() adjacencies on the BFIR and area edge BFR:

On all BFIRs in an area  $j | j=1...6$ , bia in each BIFT:SI is populated with the same forward\_routed(BFRja), and bib with forward\_routed(BFRjb). On all area edge BFR, bea in BIFT:SI= $k | k=1...6$  is populated with forward\_routed(BFRka) and beb in BIFT:SI= $k$  with forward\_routed(BFRkb).

For BIER-TE forwarding of a packet to a subset of BFERs across all areas, a BFIR would create at most 6 copies, with SI=1...SI=6. In each packet, the bits indicate bits for topology and BFER in that topology plus the four bits to indicate whether to pass this packet via the ingress area a or b border BFR and the egress area a or b border BFR, therefore allowing path steering for those two "unicast" legs: 1) BFIR to ingress area edge and 2) core to egress area edge. Replication only happens inside the egress areas. For BFER in the same area as in the BFIR, these four bits are not used.

#### 5.3.7. Summary

BIER-TE can, like BIER, support multiple SIs within a sub-domain. This allows to apply the mapping  $\text{BFR-id} = \text{SI} * \text{BSL} + \text{BP}$ . This allows to re-use the BIER architecture concept of BFR-id and therefore minimize BIER-TE specific functions in possible BIER layer control plane mechanisms with BIER-TE, including flow overlay methods and BIER header fields.

The number of BFIR/BFER possible in a sub-domain is smaller than in BIER because BIER-TE uses additional bits for topology.

Sub-domains (SDs) in BIER-TE can be used like in BIER to create more efficient replication to known subsets of BFERs.

Assigning bits for BFERs intelligently into the right SI is more important in BIER-TE than in BIER because of replication efficiency and overall amount of bits required.

## 6. Security Considerations

If [RFC8296] is used, BIER-TE shares its security considerations.

BIER-TE shares the security considerations of BIER, [RFC8279], with the following overriding or additional considerations.

BIER-TE forwarding explicitly supports unicast "tunneling" of BIER packets via `forward_routed()` adjacencies. The BIER domain security model is based on a subset of interfaces on a BFR that connect to other BFRs of the same BIER domain. For BIER-TE, this security model equally applies to such unicast "tunneled" BIER packets. This does not only include the need to filter received unicast "tunneled" BIER packets to prohibit injection of such "tunneled" BIER packets from outside the BIER domain, but also prohibiting `forward_routed()` adjacencies to leak BIER packets from the BIER domain. It SHOULD be possible to configure interfaces to be part of a BIER domain solely for sending and receiving of unicast "tunneled" BIER packets even if the interface can not send/receive BIER encapsulated packets.

In BIER, the standardized methods for the routing underlays are IGPs with extensions to distribute BFR-ids and BFR-prefixes. [RFC8401] specifies the extensions for IS-IS and [RFC8444] specifies the extensions for OSPF. Attacking the protocols for the BIER routing underlay or (non-TE) BIER layer control plane, or impairment of any BFR in a domain may lead to successful attacks against the results of the routing protocol, enabling DoS attacks against paths or the addressing (BFR-id, BFR-prefixes) used by BIER.

The reference model for the BIER-TE layer control plane is a BIER-TE controller. When such a controller is used, impairment of an individual BFR in a domain causes no impairment of the BIER-TE control plane on other BFRs. If a routing protocol is used to support `forward_routed()` adjacencies, then this is still an attack vector as in BIER, but only for BIER-TE `forward_routed()` adjacencies, and not other adjacencies.

Whereas IGP routing protocols are most often not well secured through cryptographic authentication and confidentiality, communications between controllers and routers such as those to be considered for the BIER-TE controller/control-plane can be and are much more commonly secured with those security properties, for example by using Secure Shell (SSH), [RFC4253] for NETCONF ([RFC6242]), or via Transport Layer Security (TLS), such as [RFC8253] for PCEP, [RFC5440], or [RFC7589] for NETCONF. BIER-TE controllers SHOULD use security equal to or better than these mechanisms.



When any of these security mechanisms/protocols are used for communications between a BIER-TE controller and BFRs, their security considerations apply to BIER-TE. In addition, the security considerations of PCE, [RFC4655] apply.

The most important attack vector in BIER-TE is misconfiguration, either on the BFR themselves or via the BIER-TE controller. Forwarding entries with DNC could be set up to create persistent loops, in which packets only expire because of TTL. To minimize the impact of such attacks (or more likely unintentional misconfiguration by operators and/or bad BIER-TE controller software), the BIER-TE forwarding rules are defined to be as strict in clearing bits as possible. The clearing of all bits with an adjacency on a BFR prohibits that a looping packet creates additional packet amplification through the misconfigured loop on the packet's second or further times around the loop, because all relevant adjacency bits would have been cleared on the first round through the loop. In result, BIER-TE has the same degree of looping packets as possible with unintentional or malicious loops in the routing underlay with BIER or even with unicast traffic.

Deployments where BIER-TE would likely be beneficial may include operational models where actual configuration changes from the controller are only required during non-production phases of the network's life-cycle, such as in embedded networks or in manufacturing networks during e.g. plant reworking/repairs. In these type of deployments, configuration changes could be locked out when the network is in production state and could only be (re-)enabled through reverting the network/installation into non-production state. Such security designs would not only allow to provide additional layers of protection against configuration attacks, but would foremost protect the active production process from such configuration attacks.

## 7. IANA Considerations

This document requests no action by IANA.

## 8. Acknowledgements

The authors would like to thank Greg Shepherd, Ijsbrand Wijnands, Neale Ranns, Dirk Trossen, Sandy Zheng, Lou Berger, Jeffrey Zhang, Carsten Borman and Wolfgang Braun for their reviews and suggestions.

Special thanks to Xuesong Geng for shepherding the document and for IESG review/suggestions by Alvaro Retana (responsible AD/RTG), Benjamin Kaduk (SEC), Tommy Pauly (TSV), Zaheduzzaman Sarker (TSV), Eric Vyncke (INT), Martin Vigoureux (RTG), Robert Wilton (OPS), Eric

Kline (INT), Lars Eggert (GEN), Roman Danyliv (SEC), Ines Robles (RTGDIR), Robert Sparks (Gen-ART), Yingzhen Qu (RTGdir), Martin Duke (TSV).

9. Change log [RFC Editor: Please remove]

draft-ietf-bier-te-arch:

13:

Changed Gregs author association/email.

Fixed Nits in -12 from Ben Kaduk.

Fixed Alvaro's concerns: (1) Removed references to SR in Abstract/Overview (2) removed section 4.5.

12:

AD review Alvaro Retana.

Various textual/editorial nits including adding () to all instances of forwarding adjacency name instances.

3.1 Added new paragraph outlining possible use of BGP as RR in BIER-TE controller as core of multicast flow overlay component of BIER-TE.

3.2 added xref's to relevant sections to the listed control plane points.

4.1 rewrote paragraphs of 4.1 leading up to Figure 4. to eliminate any confusion in how the BIFT work and how it compares to the notions in rfc8279, as well as better linking it to the Pseudocode.

Moved SR section into appendix.

TSV review Martin Duke.

Text/editorial nits.

4.4 improved text describing handling of F-BM.

RTGdir review Yingzhen Qu.

Various text/editorial nits.

Added notion that BitStrings represent loop free tree for packet to abstract and intro.

Various text nit and editorial improvements.

Fixed some BFR-id field -> BFIR-id field mistakes.

Capitalized NETCONF/RESTCONF/YANG, added RFC references.

Improved Figure 16 with explicitly two links into BFR3 and explanatory text.

Gen-ART review Robert Sparks.

Various textual nits, editorial improvements.

3.2 Introduced terms "BIER-TE topology control" and "BIER-TE tree control" for the two functional components of the control plane.

3.2.1 - 3.2 change introduces the open RFC-editor issue of appropriate xrfs (to be resolved by RFC-editor).

3.3 Rewrote last paragraph to better describe loop prevention through clearing of bits in BitString.

4.1 Fixed up text/formula describing mapping between bfr-id, SI:BP and SI,BSL and BP. Fix offset bug.

5.3.6.2 Improved description paragraph explaining overlap of topology for different SI.

5.3.7 Improved first summary paragraph.

7. Rephrased applicability statement of control plane protocol security considerations to BIER-TE security.

RTGDIR review Ines Robles.

Fixed up adjacencies in Example 2 and explanation text to be explicit about which BFR not only passes, but also receives the packet.

7. (security considerations). Added paragraph about forward\_routed() and prohibiting BIER packet leaking in/out of domain.

IESG review Roman Danyliv (SEC).

Several textual/sentence nits/editorials.

IESG review Lars Eggert (GEN).

Various good editorial word fixed.

Pointer to non-false-positive bloom filter work that looks like it happened after our IETF discussions documented in this doc, so will not add it to doc, but here is URL for folks interested: <https://ieeexplore.ieee.org/document/8486415>.

Did not change "native" to a different word for inclusivity because of my worry there is no established single replacement word, making reading/searching/understanding more difficult.

IESG review Martin Vigoureux (RTG).

Added back reference to RFC8402. Textual fixes.

IESG review Eric Kline (INT).

2.1 Fixed typo in BFR\* explanations.

4.3 Added explanatio about MTU handling.

IESG review Eric Vyncke (INT).

Fixed up initial text to introduce various abbreviations.

2.4 refined wording to "with the `_intent_` to easily build common forwarding planes...".

4.2.3 refined text about entropy in ECMP - now taken text from rfc8279.

IESG review Zaheduzzaman Sarker (TSV).

5.1.7 Refined text explaining documentation of ECMP algorithm.

5.3.6.2. fixed range of areas/SI over which to build the example large network BPs - removed explanation of the large network shown to be only used for sources in area 1 (IPTV), because it was a stale explanation.

IESG review Ben Kaduk (round 2):

4.4 Advanced pseudocode still had one wrong "~". Root cause seems to have been day 0 problem in pseudocode written for -01, "~" was inserted in the wrong one of two code lines. Also enhanced textual description and comments in pseudocode, changed variable name AdjacentBits to PktAdjacentBits to avoid confusion with AdjacentBits[SI].

5.1.3 Rewrote last two paragraphs explaining the sharing of bit positions for lead-BFER hopefully better. Also detailed how it interacts with other optimizations and the type of payload BIER-TE packets may carry.

4.4 (from Carsten Borman) changed spacing in pseudocode to be consistent. Fixed {VRF}, clarified pseudocode object syntax, typos.

11: IESG review Ben Kaduk, summary:

One discuss for bug in pseudocode. turned out to be one cahrcrter typo.

Added (non-TE) prefix in places where BIER by itsels had to be better disambiguated.

enhanced text for hub-and-spoke to indicate we're only talking about hub to spoke traffic.

long list ot language fixes/improvement (nits). Thanks a lot!.

add suggestion to SHOULD use known confidentiality protocols between controller and BFR.

10: AD review Alvaro Retana, summary:

Note: rfcdiff shows more changes than actually exist because text moved around.

Summary:

1. restructuring: merged all controller sections under common controller ops main section, moved unfitting stuff out to other parts of doc. Split Intro section into Overview and Intro. Shortened Abstract, moved text into Overview, added sections overview.
2. enhanced/rewrote: 2.3 Comparison with -> Relationship to BIER-TE

3. enhanced/rewrote: 3.2 BIER-TE controller -> BIER-TE control plane, 3.2.1 BIER-TE controller, for consistency with rfc8279
4. additional subsections for Alvaros asks
5. added to: 3.3 BIER-TE forwarding plane (consistency with rfc8279)
6. Enhanced description of 4.3/encap considerations to better explain how BIER/BIER-TE can run together.

Notation: Markers (a), (b), ... at end of points are references from the review discussion with Alvaro to the changes made.

Details:.

Throughout text: changed term spelling to rfc8279 - bit positions, sub-domain, ... (i).

Reset changed to clear, also DNR changed to DNC (Do Not Clear) (q).

Abstract: Shortened. Removed name explanation note (Tree Engineering), (a).

1. Introduction -> Overview: Moved important explanation paragraph from abstract to Introduction. Fixed text, (a).

Added bullet point list explanation of structure of document (e).

Renamed to Overview because that is now more factually correct.

1.1. Fixed bug in example adding bit p15.(l).

2. (New - Introduction): Moved section 1.1 - 1.3 (examples, comparison with BIER-TE) from Introduction into new "Overview" section. Primarily so that "requirements language" section (at end of Introduction) is not in middle of document after all the Introduction.

2.1 Removed discussion of encap, moved to 4.2.2 (m).

2.2 enhanced paragraph suggesting native/overlay topology types, also suggest type hybrid (n).

2.3 Overhauled comparison text BIER/BIER-TE, structured into common, different, not-required-by-te, integration-bier-bier-te. Changed title to "Relationship" to allow including last point. (f).

2.4 moved Hardware forwarding comparison section into section 2 to allow coalescing of sections into section 5 about the controller operations (hardware forwarding was in the middle of it, wrong place). Shortened/improved third paragraph by pointing to BIFT as deciding element for selection between BIER/BIER-TE. Removed notion of experimentation (this now targets standard) (g).

3. (Components): Aligned component name and descriptions better with RFC8279. Now describe exactly same three layers. BIER layer constituted from BIER-TE control plane and BIER-TE forwarding plane. BIER-TE controller is now simply component of BIER-TE control plane. (b).

3.1. shortened/improved paragraph explaining use of SI:BP instead of also bfr-id as index into BIFT, rewrote paragraph talking about reuse of BPs(o).

3.2. rewrote explanation of BIER-TE control plane in the style of RFC8729 Section 4.2 (BIER layer) with numbered points. Note that RFC8729 mixes control and forwarding plane bullet points (this doc does not). Merged text from old sections 2.2.1 and 2.2.3 into list. (b).

3.2.1. Expanded/improved explanation of BIER-TE Controller (b).

3.2.1.1. Added subsection for topology discovery and creation (d).

3.2.1.2. Added subsection for engineered BitStrings as key novel aspect not found in BIER. (X).

3.3. Added numbered list for components of BIER-TE forwarding plane (completing the comparable text from RFC8729 Section 4.2).

3.4 Alvaro does not mind additional example, fixed bugs.

3.5 Removed notion about using IGP BIER extensions for BIER-TE, such as BIFT address ranges. After -10 making use of BIFT clearer, it now looks to authors as if use of IGP extensions would not be beneficial, as long as we do need to use the BIER-TE controller, e.g. unlike in BIER, a BFR could not learn from the IGP information what traffic to send towards a particular BIFT-ID, but instead that is the core of what the controller needs to provide.

4.2.2 Improved text to explain requirement to identify BIER-TE in the tunnel encap and compress description of use-cases (m).

4.2.3 enhanced ECMP text (p).

4.3. rewrote most of Encapsulation Considerations to better explain to Alvaros question re sharing or not sharing SD via BIER/BIER-TE. Added reference to I-D.ietf-bier-non-mpls-bift-encoding as a very helpful example. (f).

4.3 Renamed title to "...Co-Existence with BIER" as this is what it is about and to help finding it from abstract/intro ("co-exist") (j).

4.4. Moved BIER-TE Forwarding Pseudocode here to coalesce text logically. Changed text to better compare with BIER pseudo forwarding code. Numerical list of how F-BM works for BIER-TE. Removed efficiency comparison with BIER (too difficult to provide sufficient justification, derails from focus of section) (j).

4.6. (Requirements) Restructured: Removed notion of "basic" BIER-TE forwarding, simply referring to it now as "mandatory" BIER-TE forwarding. Cleaned up text to have requirements for different adjacencies in different paragraphs. (c).

5. Created new main section "BIER-TE Controller operational considerations", coalesced old sections 4., 5., 7. into this new main section. No text changes. (k).

5.1.9 Added new separate picture instead of referring to a picture later in text, adjusted text (r).

5.3.2 Changed title to not include word "comparison" to avoid this being accounted against Alvaros concern about scattering comparison (IMHO text already has little comparison, so title was misleading) (h).

co-authors internal review:



4.4 Added xref to Figure 5.

5.2.1 Duplicated ring picture, added visuals for described miswiring (s).

5.2.2 replace "topology" with graph (wrong word).

5.3.3 rewrote explanation of how to map BFR-id to SI:BP and assign them, clarified BFR-id is option. Retitled to better explain scope of section.

5.3.4 Removed considerations in 5.3.4 for sharing BFR-id across BIER/BIER-TE (t), changed title to explain how BFIR/BIER-TE controller interactions need some form of identifying BFR but this does not have to be BFR-id.

7. Added new security considerations (u).

09: Incorporated fixes for feedback from Shepherd (Xuesong Geng).

Added references for Bloom Filters and Rate Controlled Service Disciplines.

1.1 Fixed numbering of example 1 topology explanation. Improved language on second example (less abbreviating to avoid confusion about meaning).

1.2 Improved explanation of BIER-TE topology, fixed terminology of graphs (BIER-TE topology is a directed graph where the edges are the adjacencies).

2.4 Fixed and amended routing underlay explanations: detailed why no need for BFER routing underlay routing protocol extensions, but potential to re-use BIER routing underlay routing protocol extensions for non-BFER related extensions.

3.1 Added explanation for VRF and its use in adjacencies.

08: Incorporated (with hopefully acceptable fixes) for Lou suggested section 2.5, TE considerations.

Fixes are primarily to the point to a) emphasize that BIER-TE does not depend on the routing underlay unless `forward_routed()` adjacencies are used, and b) that the allocation and tracking of resources does not explicitly have to be tied to BPs, because they are just steering labels. Instead, it would ideally come from per-hop resource management that can be maintained only via local accounting in the controller.

07: Further reworking text for Lou.

Renamed BIER-PE to BIER-TE standing for "Tree Engineering" after votes from BIER WG.

Removed section 1.1 (introduced by version 06) because not considered necessary in this doc by Lou (for framework doc).

Added [RFC editor pls. remove] Section to explain name change to future reviewers.

06: Concern by Lou Berger re. BIER-TE as full traffic engineering solution.

Changed title "Traffic Engineering" to "Path Engineering"

Added intro section of relationship BIER-PE to traffic engineering.

Changed "traffic engineering" term in text to "path engineering", where appropriate

Other:

Shortened "BIER-TE Controller Host" to "BIER-TE Controller".  
Fixed up all instances of controller to do this.

05: Review Jeffrey Zhang.

Part 2:

4.3 added note about leaf-BFER being also a property of routing setup.

4.7 Added missing details from example to avoid confusion with routed adjacencies, also compressed explanatory text and better justification why seed is explicitly configured by controller.

4.9 added section discussing generic reuse of BP methods.

4.10 added section summarizing BP optimizations of section 4.

6. Rewrote/compressed explanation of comparison BIER/BIER-TE forwarding difference. Explained benefit of BIER-TE per-BP forwarding being independent of forwarding for other BPs.

Part 1:

Explicitly use forwarded\_connected adjacency in ECMP adjacency examples to avoid confusion.

4.3 Add picture as example for leaf vs. non-leaf BFR in topology. Improved description.

4.5 Example for traffic that can be broadcast -> for single BP in hub&spoke.

4.8.1 Simplified example picture for routed adjacency, explanatory text.

Review from Dirk Trossen:

Fixed up explanation of ICC paper vs. bloom filter.

04: spell check run.

Added remaining fixes for Sandys (Zhang Zheng) review:

4.7 Enhance ECMP explanations:

example ECMP algorithm, highlight that doc does not standardize ECMP algorithm.

Review from Dirk Trossen:

1. Added mentioning of prior work for traffic engineered paths with bloom filters.

2. Changed title from layers to components and added "BIER-TE control plane" to "BIER-TE Controller" to make it clearer, what it does.

2.2.3. Added reference to I-D.ietf-bier-multicast-http-response as an example solution.

2.3. clarified sentence about resetting BPs before sending copies (also forgot to mention DNR here).

3.4. Added text saying this section will be removed unless IESG review finds enough redeeming value in this example given how -03 introduced section 1.1 with basic examples.

7.2. Removed explicit numbers 20%/80% for number of topology bits in BIER-TE, replaced with more vague (high/low) description, because we do not have good reference material Added text saying this section will be removed unless IESG review finds enough redeeming value in this example given how -03 introduced section 1.1 with basic examples.

many typos fixed. Thanks a lot.

03: Last call textual changes by authors to improve readability:

removed Wolfgang Braun as co-authors (as requested).

Improved abstract to be more explanatory. Removed mentioning of FRR (not concluded on so far).

Added new text into Introduction section because the text was too difficult to jump into (too many forward pointers). This primarily consists of examples and the early introduction of the BIER-TE Topology concept enabled by these examples.

Amended comparison to SR.

Changed syntax from [VRF] to {VRF} to indicate its optional and to make idnits happy.

Split references into normative / informative, added references.

02: Refresh after IETF104 discussion: changed intended status back to standard. Reasoning:

Tighter review of standards document == ensures arch will be better prepared for possible adoption by other WGs (e.g. DetNet) or std. bodies.

Requirement against the degree of existing implementations is self defined by the WG. BIER WG seems to think it is not necessary to apply multiple interoperating implementations against an architecture level document at this time to make it qualify to go to standards track. Also, the levels of support introduced in -01 rev. should allow all BIER forwarding engines to also be able to support the base level BIER-TE forwarding.

01: Added note comparing BIER and SR to also hopefully clarify BIER-TE vs. BIER comparison re. SR.

- added requirements section mandating only most basic BIER-TE forwarding features as MUST.

- reworked comparison with BIER forwarding section to only summarize and point to pseudocode section.

- reworked pseudocode section to have one pseudocode that mirrors the BIER forwarding pseudocode to make comparison easier and a second pseudocode that shows the complete set of BIER-TE forwarding options and simplification/optimization possible vs. BIER forwarding. Removed MyBitsOfInterest (was pure optimization).

- Added captions to pictures.

- Part of review feedback from Sandy (Zhang Zheng) integrated.

00: Changed target state to experimental (WG conclusion), updated references, mod auth association.

- Source now on <https://www.github.com/toerless/bier-te-arch>

- Please open issues on the github for change/improvement requests to the document - in addition to posting them on the list (bier@ietf.). Thanks!.

draft-eckert-bier-te-arch:

06: Added overview of forwarding differences between BIER, BIER-TE.

05: Author affiliation change only.

04: Added comparison to Live-Live and BFIR to FRR section (Eckert).

04: Removed FRR content into the new FRR draft [I-D.eckert-bier-te-frr] (Braun).

- Linked FRR information to new draft in Overview/Introduction

- Removed BTAFT/FRR from "Changes in the network topology"

- Linked new draft in "Link/Node Failures and Recovery"

- Removed FRR from "The BIER-TE Forwarding Layer"
- Moved FRR section to new draft
- Moved FRR parts of Pseudocode into new draft
- Left only non FRR parts
- removed `FrrUpDown(..)` and `//FRR` operations in `ForwardBierTePacket(..)`
- New draft contains `FrrUpDown(..)` and `ForwardBierTePacket(Packet)` from `bier-arch-03`
- Moved "BIER-TE and existing FRR to new draft
- Moved "BIER-TE and Segment Routing" section one level up
- Thus, removed "Further considerations" that only contained this section
- Added Changes for version 04

03: Updated the FRR section. Added examples for FRR key concepts. Added BIER-in-BIER tunneling as option for tunnels in backup paths. BIFT structure is expanded and contains an additional match field to support full node protection with BIER-TE FRR.

03: Updated FRR section. Explanation how BIER-in-BIER encapsulation provides P2MP protection for node failures even though the routing underlay does not provide P2MP.

02: Changed the definition of BIFT to be more inline with BIER. In revs. up to -01, the idea was that a BIFT has only entries for a single BitString, and every SI and sub-domain would be a separate BIFT. In BIER, each BIFT covers all SI. This is now also how we define it in BIER-TE.

02: Added Section 5.3 to explain the use of SI, sub-domains and BFR-id in BIER-TE and to give an example how to efficiently assign bits for a large topology requiring multiple SI.

02: Added further detailed for rings - how to support input from all ring nodes.

01: Fixed BFIR -> BFER for section 4.3.

01: Added explanation of SI, difference to BIER ECMP, consideration for Segment Routing, unicast FRR, considerations for encapsulation, explanations of BIER-TE Controller and CLI.

00: Initial version.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.

### 10.2. Informative References

- [Bloom70] Bloom, B. H., "Space/time trade-offs in hash coding with allowable errors", Comm. ACM 13(7):422-6, July 1970, <<https://dl.acm.org/doi/10.1145/362686.362692>>.
- [I-D.eckert-bier-te-frr] Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Protection Methods for BIER-TE", Work in Progress, Internet-Draft, draft-eckert-bier-te-frr-03, 5 March 2018, <<https://www.ietf.org/archive/id/draft-eckert-bier-te-frr-03.txt>>.
- [I-D.ietf-bier-multicast-http-response] Trossen, D., Rahman, A., Wang, C., and T. Eckert, "Applicability of BIER Multicast Overlay for Adaptive Streaming Services", Work in Progress, Internet-Draft,

draft-ietf-bier-multicast-http-response-06, 10 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-multicast-http-response-06.txt>>.

[I-D.ietf-bier-non-mpls-bift-encoding]

Wijnands, I., Mishra, M., Xu, X., and H. Bidgoli, "An Optional Encoding of the BIFT-id Field in the non-MPLS BIER Encapsulation", Work in Progress, Internet-Draft, draft-ietf-bier-non-mpls-bift-encoding-04, 30 May 2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-non-mpls-bift-encoding-04.txt>>.

[I-D.ietf-bier-te-yang]

Zhang, Z., Wang, C., Chen, R., Hu, F., Sivakumar, M., and H. Chen, "A YANG data model for Tree Engineering for Bit Index Explicit Replication (BIER-TE)", Work in Progress, Internet-Draft, draft-ietf-bier-te-yang-04, 7 November 2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-te-yang-04.txt>>.

[I-D.ietf-roll-ccast]

Bergmann, O., Bormann, C., Gerdes, S., and H. Chen, "Constrained-Cast: Source-Routed Multicast for RPL", Work in Progress, Internet-Draft, draft-ietf-roll-ccast-01, 30 October 2017, <<https://www.ietf.org/archive/id/draft-ietf-roll-ccast-01.txt>>.

[I-D.ietf-teas-rfc3272bis]

Farrel, A., "Overview and Principles of Internet Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc3272bis-16, 24 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-rfc3272bis-16.txt>>.

[ICC]

Reed, M. J., Al-Naday, M., Thomos, N., Trossen, D., Petropoulos, G., and S. Spirou, "Stateless multicast switching in software defined networks", IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 2016, May 2016, <<https://ieeexplore.ieee.org/document/7511036>>.

[RCSD94]

Zhang, H. and D. Domenico, "Rate-Controlled Service Disciplines", Journal of High-Speed Networks, 1994, May 1994, <<https://dl.acm.org/doi/10.5555/2692227.2692232>>.

[RFC4253]

Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <<https://www.rfc-editor.org/info/rfc4253>>.



- [RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <<https://www.rfc-editor.org/info/rfc4456>>.
- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7589] Badra, M., Luchuk, A., and J. Schoenwaelder, "Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication", RFC 7589, DOI 10.17487/RFC7589, June 2015, <<https://www.rfc-editor.org/info/rfc7589>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC7988] Rosen, E., Ed., Subramanian, K., and Z. Zhang, "Ingress Replication Tunnels in Multicast VPN", RFC 7988, DOI 10.17487/RFC7988, October 2016, <<https://www.rfc-editor.org/info/rfc7988>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8401] Ginsberg, L., Ed., Przygienda, T., Aldrin, S., and Z. Zhang, "Bit Index Explicit Replication (BIER) Support via IS-IS", RFC 8401, DOI 10.17487/RFC8401, June 2018, <<https://www.rfc-editor.org/info/rfc8401>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8444] Psenak, P., Ed., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., Zhang, J., and S. Aldrin, "OSPFv2 Extensions for Bit Index Explicit Replication (BIER)", RFC 8444, DOI 10.17487/RFC8444, November 2018, <<https://www.rfc-editor.org/info/rfc8444>>.
- [RFC8556] Rosen, E., Ed., Sivakumar, M., Przygienda, T., Aldrin, S., and A. Dolganow, "Multicast VPN Using Bit Index Explicit Replication (BIER)", RFC 8556, DOI 10.17487/RFC8556, April 2019, <<https://www.rfc-editor.org/info/rfc8556>>.

#### Appendix A. BIER-TE and Segment Routing (SR)

SR ([RFC8402]) aims to enable lightweight path steering via loose source routing. Compared to its more heavy-weight predecessor RSVP-TE, SR does for example not require per-path signaling to each of these hops.

BIER-TE supports the same design philosophy for multicast. Like in SR, it relies on source-routing - via the definition of a BitString. Like SR, it only requires to consider the "hops" on which either replication has to happen, or across which the traffic should be steered (even without replication). Any other hops can be skipped via the use of routed adjacencies.

BIER-TE bit position (BP) can be understood as the BIER-TE equivalent of "forwarding segments" in SR, but they have a different scope than SR forwarding segments. Whereas forwarding segments in SR are global or local, BPs in BIER-TE have a scope that is the group of BFR(s) that have adjacencies for this BP in their BIFT. This can be called "adjacency" scoped forwarding segments.

Adjacency scope could be global, but then every BFR would need an adjacency for this BP, for example a `forward_routed()` adjacency with encapsulation to the global SR SID of the destination. Such a BP would always result in ingress replication though (as in [RFC7988]). The first BFR encountering this BP would directly replicate to it. Only by using non-global adjacency scope for BPs can traffic be steered and replicated on non-ingress BFR.

SR can naturally be combined with BIER-TE and help to optimize it. For example, instead of defining bit positions for non-replicating hops, it is equally possible to use segment routing encapsulations (e.g. SR-MPLS label stacks) for the encapsulation of "forward\_routed" adjacencies.

Note that (non-TE) BIER itself can also be seen to be similar to SR. BIER BPs act as global destination Node-SIDs and the BIER BitString is simply a highly optimized mechanism to indicate multiple such SIDs and let the network take care of effectively replicating the packet hop-by-hop to each destination Node-SID. What BIER does not allow is to indicate intermediate hops, or in terms of SR the ability to indicate a sequence of SID to reach the destination. This is what BIER-TE and its adjacency scoped BP enables.

#### Authors' Addresses

Toerless Eckert (editor)  
Futurewei Technologies Inc.  
2330 Central Expwy  
Santa Clara, 95050  
United States of America  
Email: tte+ietf@cs.fau.de

Michael Menth  
University of Tuebingen  
Email: menth@uni-tuebingen.de

Gregory Cauchie  
KOEVOO  
Email: gregory@koevoo.tech

DetNet  
Internet-Draft  
Intended status: Standards Track  
Expires: September 6, 2018

J. Korhonen, Ed.  
Nordic  
L. Andersson  
Y. Jiang  
N. Finn  
Huawei  
B. Varga  
J. Farkas  
Ericsson  
CJ. Bernardos  
UC3M  
T. Mizrahi  
Marvell  
L. Berger  
LabN  
March 5, 2018

DetNet Data Plane Encapsulation  
draft-ietf-detnet-dp-sol-03

Abstract

This document specifies Deterministic Networking data plane encapsulation solutions. The described data plane solutions can be applied over either IP or MPLS Packet Switched Networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
2.1. Terms used in this document . . . . .	4
2.2. Abbreviations . . . . .	5
3. Requirements language . . . . .	6
4. DetNet data plane overview . . . . .	6
4.1. DetNet data plane encapsulation requirements . . . . .	8
4.2. Packet replication and elimination considerations . . . . .	10
4.3. Packet reordering considerations . . . . .	10
5. DetNet encapsulation . . . . .	10
5.1. End-system specific considerations . . . . .	10
5.2. DetNet domain specific considerations . . . . .	12
5.2.1. DetNet Bridging Service . . . . .	13
5.2.2. DetNet Routing Service . . . . .	14
5.3. DetNet Inter-Working Function (DN-IWF) . . . . .	17
5.3.1. Networks with multiple technology segments . . . . .	17
5.3.2. DN-IWF related considerations . . . . .	18
6. MPLS-based DetNet data plane solution . . . . .	19
6.1. DetNet specific packet fields . . . . .	19
6.2. Data plane encapsulation . . . . .	19
6.3. DetNet control word . . . . .	20
6.4. Flow identification . . . . .	21
6.5. Service layer considerations . . . . .	21
6.5.1. Edge node processing . . . . .	22
6.5.2. Relay node processing . . . . .	23
6.5.3. End system processing . . . . .	25
6.6. Transport node considerations . . . . .	25
6.6.1. Congestion protection . . . . .	25
6.6.2. Explicit routes . . . . .	25
7. IPv6-based DetNet data plane solution . . . . .	25
7.1. Data plane encapsulation . . . . .	25

7.2.	DetNet destination option . . . . .	27
7.3.	Flow identification . . . . .	28
7.4.	Service layer considerations . . . . .	28
7.4.1.	Edge node processing . . . . .	29
7.4.2.	Relay node processing . . . . .	31
7.4.3.	End system processing . . . . .	31
7.5.	Transport node processing . . . . .	31
7.5.1.	Congestion protection . . . . .	31
7.5.2.	Explicit routes . . . . .	32
8.	Other DetNet data plane considerations . . . . .	32
8.1.	Class of Service . . . . .	32
8.2.	Quality of Service . . . . .	32
8.3.	Cross-DetNet flow resource aggregation . . . . .	34
8.4.	Bidirectional traffic . . . . .	35
8.5.	Layer 2 addressing and QoS Considerations . . . . .	35
8.6.	Interworking between MPLS- and IPv6-based encapsulations . . . . .	36
8.7.	IPv4 considerations . . . . .	36
9.	Time synchronization . . . . .	36
10.	Management and control considerations . . . . .	38
10.1.	MPLS-based data plane . . . . .	38
10.1.1.	S-Label assignment and distribution . . . . .	38
10.1.2.	Explicit routes . . . . .	38
10.2.	IPv6-based data plane . . . . .	38
10.2.1.	Flow Label assignment and distribution . . . . .	38
10.2.2.	Explicit routes . . . . .	39
10.3.	Packet replication and elimination . . . . .	39
10.4.	Congestion protection and latency control . . . . .	39
10.5.	Flow aggregation control . . . . .	39
11.	Security considerations . . . . .	39
12.	IANA considerations . . . . .	39
13.	Acknowledgements . . . . .	39
14.	References . . . . .	40
14.1.	Normative references . . . . .	40
14.2.	Informative references . . . . .	42
Appendix A.	Example of DetNet data plane operation . . . . .	43
Appendix B.	Example of pinned paths using IPv6 . . . . .	44
Authors' Addresses	. . . . .	44

## 1. Introduction

Deterministic Networking (DetNet) is a service that can be offered by a network to DetNet flows. DetNet provides these flows extremely low packet loss rates and assured maximum end-to-end delivery latency. General background and concepts of DetNet can be found in [I-D.ietf-detnet-architecture].

This document specifies the DetNet data plane and the on-wire encapsulation of DetNet flows. The specified encapsulation provides

the building blocks to enable the DetNet service layer functions and allow flow identification as described in the DetNet Architecture. Two data plane definitions are given.

1. MPLS-based: The encapsulation resembles PseudoWires (PW) with an MPLS Packet Switched Network (PSN) [RFC3985][RFC4385].
2. Native-IP-based: The encapsulating protocol is IPv6 and the solution relies on IP header fields, existing and DetNet specific IPv6 extension header options [RFC8200].

[Editor's note: MPLS- and IPv6-based solutions are likely to be split into different documents.]

It is worth noting that while MPLS-based solution can transport IP packets a native-IP solution is meant for deployments where the DetNet service layer functions are provided at the IP-layer rather than the underlying transport network. The primary reason for this is the benefit gained by enabling the use of a normal application stack, where transport protocols such as TCP or UDP are directly encapsulated in IP.

The DetNet transport layer functionality that provides congestion protection for DetNet flows is assumed to be in place in a DetNet node.

Furthermore, this document also describes how DetNet flows are identified, how a DetNet Relay/Edge/Transit nodes work, and how the Packet Replication and Elimination function (PREF) is implemented with the two data plane solutions.

This document does not define the associated control plane functions, or Operations, Administration, and Maintenance (OAM). It also does not specify traffic handling capabilities required to deliver congestion protection and latency control for DetNet flows at the DetNet transport layer.

## 2. Terminology

### 2.1. Terms used in this document

This document uses the terminology established in the DetNet architecture [I-D.ietf-detnet-architecture] and the DetNet Data Plane Solution Alternatives [I-D.ietf-detnet-dp-alt].

**T-Label**            A label used to identify the LSP used to transport a DetNet flow across an MPLS PSN, e.g., a hop-by-hop label used between label switching routers (LSR).

S-Label	A DetNet "service" label that is used between DetNet nodes that implement also the DetNet service layer functions. An S-Label is also used to identify a DetNet flow at DetNet service layer.
Flow Label	IPv6 header field that is used to identify a DetNet flow (together with the source IP address field).
Local-ID	A DetNet Edge and Relay node internal construct that uniquely identifies a DetNet flow within a node and never appear on-wire. It may be used to select proper forwarding and/or DetNet specific service function.
PREF	A Packet Replication and Elimination Function (PREF) does the replication and elimination processing of DetNet flow packets in edge or relay nodes. The replication function is essentially the existing 1+1 protection mechanism. The elimination function reuses and extends the existing duplicate detection mechanism to operate over multiple (separate) DetNet member flows of a DetNet compound flow.
DetNet Control Word	A control word used for sequencing and identifying duplicate packets at the DetNet service layer.

## 2.2. Abbreviations

The following abbreviations used in this document:

AC	Attachment Circuit.
CE	Customer Edge equipment.
CoS	Class of Service.
CW	Control Word.
d-CW	DetNet Control Word.
DetNet	Deterministic Networking.
DF	DetNet Flow.
L2VPN	Layer 2 Virtual Private Network.
LSR	Label Switching Router.



MPLS	Multiprotocol Label Switching.
MPLS-TP	Multiprotocol Label Switching - Transport Profile.
MS-PW	Multi-Segment PseudoWire (MS-PW).
NSP	Native Service Processing.
OAM	Operations, Administration, and Maintenance.
PE	Provider Edge.
PREF	Packet Replication and Elimination Function.
PSN	Packet Switched Network.
PW	PseudoWire.
QoS	Quality of Service.
TSN	Time-Sensitive Network.

### 3. Requirements language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

### 4. DetNet data plane overview

This document describes how to use IP and/or MPLS to support a data plane method of flow identification and packet forwarding over layer-3. Two different cases are covered: (i) the inter-connect scenario, in which IEEE802.1 TSN is routed over a layer-3 network (i.e., to enlarge the layer-2 domain), and (ii) native connectivity between DetNet-aware end systems.

Figure 1 illustrates how DetNet can provide services for IEEE 802.1TSN end systems over a DetNet enabled network. The edge nodes insert and remove required DetNet data plane encapsulation. The 'X' in the edge and relay nodes represents a potential DetNet flow packet replication and elimination point. This conceptually parallels L2VPN services, and could leverage existing related solutions as discussed below.

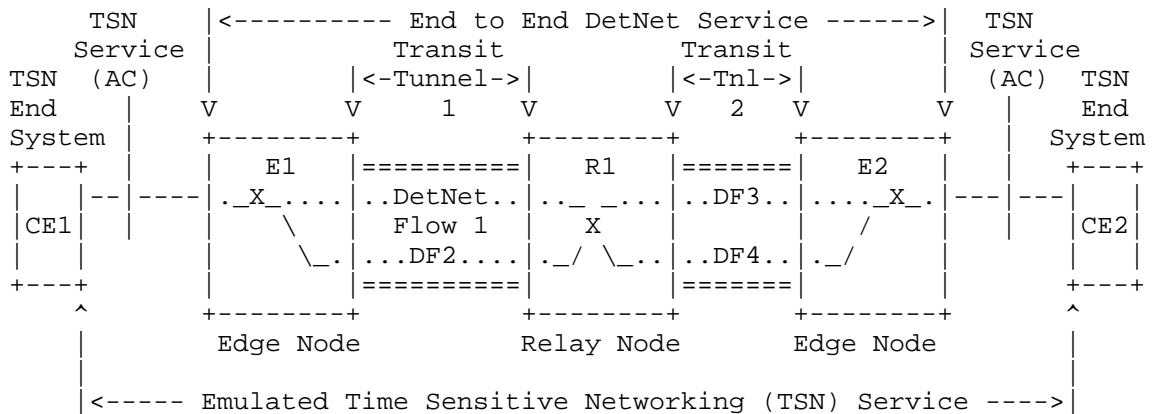


Figure 1: IEEE 802.1TSN over DetNet

Figure 2 illustrates how end to end MPLS-based DetNet service can be provided. In this case, the end systems are able to send and receive DetNet flows. For example, an end system sends data encapsulated in MPLS. Like earlier the 'X' in the end systems, edge and relay nodes represents potential DetNet flow packet replication and elimination points. Here the relay nodes may change the underlying transport, for example tunneling IP over MPLS, or simply interconnect network segments.

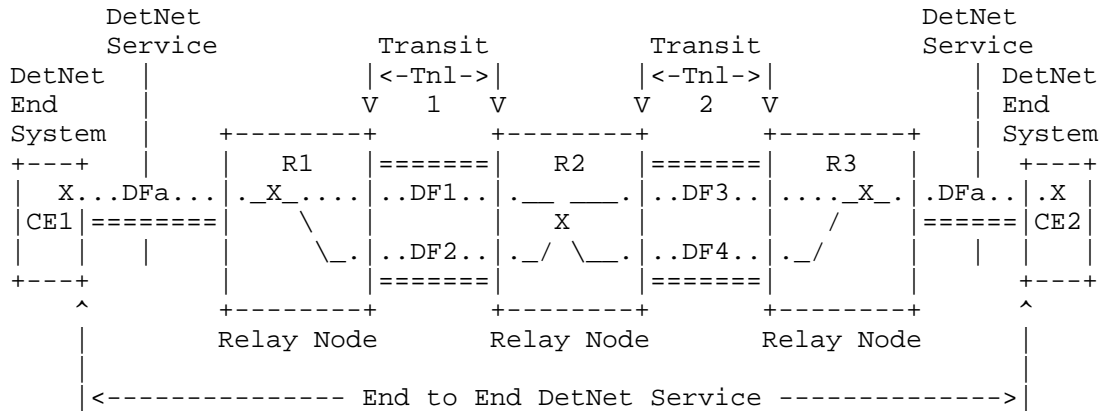


Figure 2: MPLS-Based Native DetNet

Figure 3 illustrates how end to end IP-based DetNet service can be provided. In this case, the end systems are able to send and receive DetNet flows. [Editor's note: TBD]

NOTE: This figures is TBD

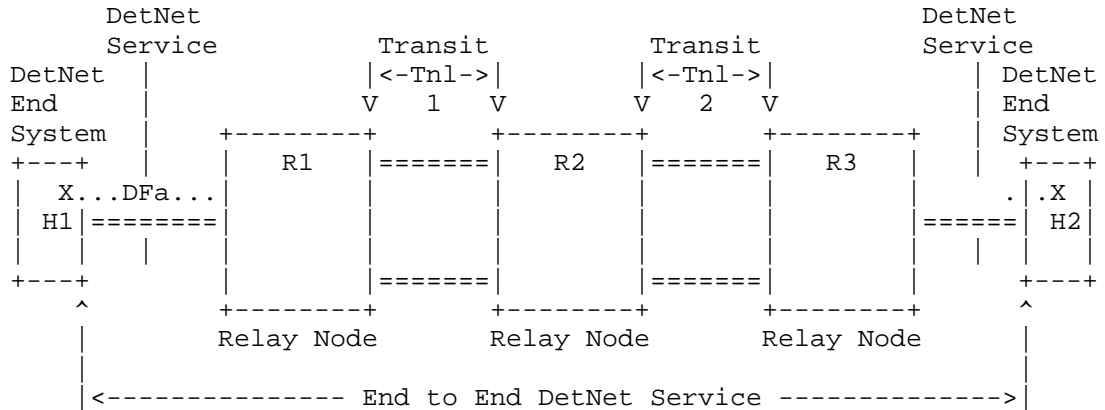


Figure 3: IP-Based Native DetNet

#### 4.1. DetNet data plane encapsulation requirements

Two major groups of scenarios can be distinguished which require flow identification during transport:

##### 1. DetNet function related scenarios:

- \* Congestion protection and latency control: usage of allocated resources (queuing, policing, shaping).
- \* Explicit routes: select/apply the flow specific path.
- \* Service protection: recognize DetNet compound and member flows for replication and elimination.

Comment #12 I am not sure whether the correct architectural construct is flow or flow group. Flow suggests that sharing/aggregation is not allowed but whether this is allowed or not is an application specific issue.

Discussion: Agree that a flow group would be a better characterization.

Comment #13 I think that there needs to be some clarification as to whether FG is understood by the DN system exclusively or whether there is an expectation that it is understood by the underlay.

Discussion: Agree that more detail is needed here. DetNet aware nodes need to understand flow groups. Underlay needs to be aware of flow groups at the resource allocation level.

2. OAM function related scenarios:

- \* troubleshooting (e.g., identify misbehaving flows, etc.)
- \* recognize flow(s) for analytics (e.g., increase counters, etc.)
- \* correlate events with flows (e.g., volume above threshold, etc.)
- \* etc.

Each DetNet node (edge, relay and transit) use an internal/implementation specific local-ID of the DetNet-(compound)-flow in order to accomplish its role during transport. Recognizing the DetNet flow is more relaxed for edge and relay nodes, as they are fully aware of both the DetNet service and transport layers. The primary DetNet role of intermediate transport nodes is limited to ensuring congestion protection and latency control for the above listed DetNet functions.

The DetNet data plane allows for the aggregation of DetNet flows, e.g., via MPLS hierarchical LSPs, to improved scaling. When DetNet flows are aggregated, transit nodes may have limited ability to provide service on per-flow DetNet identifiers. Therefore, identifying each individual DetNet flow on a transit node may not be achieved in some network scenarios, but DetNet service can still be assured in these scenarios through resource allocation and control.

Comment #14 You could introduce the concept of a flow group identified into the packet. You may also include a flow id at a lower layer.

Discussion: Agree on the identification properties. Adding a specific id into actual on-wire formats is not necessarily needed.

On each DetNet node dealing with DetNet flows, an internal local-ID is assumed to determine what local operation a packet goes through. Therefore, local-IDs has to be unique on each edge and relay nodes. Local-ID is unambiguously bound to the DetNet flow.

#### 4.2. Packet replication and elimination considerations

DetNet service layer introduces packet replication and elimination functionality (PREF) for use in DetNet edge and relay node and end system packet processing. PREF MAY be enabled in a DetNet node and the required processing is only applied to packets with a positive flow identification at the DetNet service layer. PREF utilizes a sequence number carried within a DetNet flow packets.

At a DetNet node level the output of the PREF elimination function is always a single packet. The output of the PREF replication function at a DetNet node level is always one or more packets (i.e., 1:M replication). The replicated packets MUST share the same d-CW i.e., the sequence number is the same for each member flow of the compound flow. The location and mechanism on the packet processing pipeline used for replication is implementation specific.

The complex part of the DetNet PREF processing is tracking the history of received packets for multiple DetNet member flows. These ingress DetNet member flows (to a node) MUST have the same local-ID if they belong to the same DetNet (compound) flow and share the same sequence number counter and the history information. The location of the packet elimination on the packet processing pipeline is implementation specific.

#### 4.3. Packet reordering considerations

DetNet service layer introduces also packet reordering functionality for use in DetNet edge and relay node and end system packet processing. The reordering functionality MAY be enabled in a DetNet node. The reordering functionality relies on a presence of sequence numbers in a DetNet (compound) flows. The reordering processing is only applied to packets with a positive flow identification at the DetNet service layer.

### 5. DetNet encapsulation

#### 5.1. End-system specific considerations

Data-flows requiring DetNet service are generated and terminated on end-systems. Encapsulation depends on application and its preferences. In a DetNet (or even a TSN) domain the DN (TSN) functions use at most two flow parameters, namely Flow-ID and Seq.Number. However, an application may exchange further flow related parameters (e.g., time-stamp), which are not considered by DN functions.

Two types of end-systems are distinguished:

- o L3 (IP) end-system: application over L3
- o L2 (Ethernet) end-system: application directly over L2

In case of Ethernet end-systems the application data is encapsulated directly in L2. From the DN domain perspective no upper layer protocols are visible. The Data-flow uses only Ethernet tag(s) and further flow specific parameters (if needed) are hidden inside the PDU.

The IP end-system scenario is different. Data-flows are encapsulated directly in L3 (i.e., IP) and the application may use further upper layer protocols (e.g., RTP). Many valid combinations exist, and it may be application specific how the IP header fields are used. Also, usage of further upper layer protocols depends on application requirements (e.g., time-stamp). Some examples for encoding of Flow-ID or Seq.Number attributes: IP address, IPv6-Flow-label, L4 ports, RTP-header, etc.

As a general rule, DetNet domains MUST be capable to forward any Data-flows and the DetNet domain MUST NOT intend to mandate end-system encapsulation format.

Furthermore, no application-level-proxy function is envisioned inside the DetNet domain, so end-systems peer with end-systems using the same application encapsulation format (see figure below):

- o L3 end-systems peer with L3 end-systems and
- o L2 end-systems peer with L2 end-systems

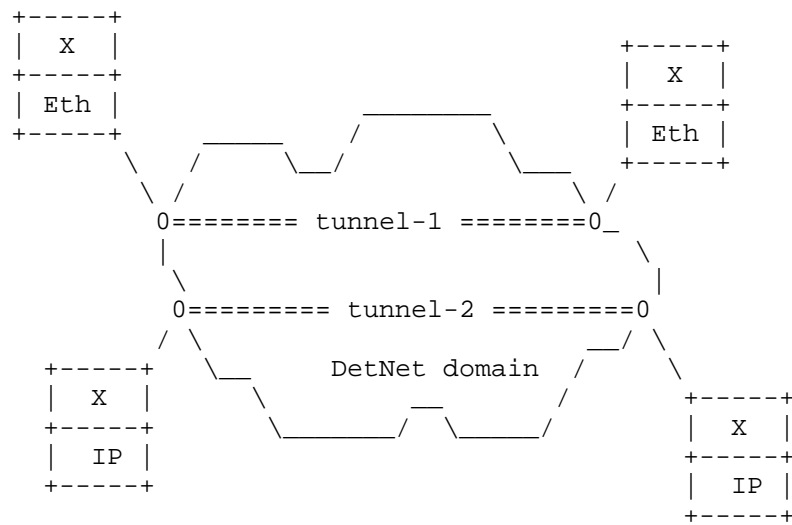


Figure 4: End-systems and the DetNet domain

## 5.2. DetNet domain specific considerations

From connection type perspective three scenarios are distinguished:

1. Directly attached: end-system is directly connected to an edge node
2. Indirectly attached: end-system is behind a (L2-TSN / L3-DetNet) sub-net
3. DN integrated: end-system is part of the DetNet domain

L3 end-systems may use any of these connection types, however L2 end-systems may use only the first two (directly or indirectly attached). DetNet domain MUST allow communication between any end-systems of the same type (L2-L2, L3-L3), independent of their connection type and DetNet capability. However directly attached and indirectly attached end-systems have no knowledge about the DetNet domain and its encapsulation format at all. See the figure below for L3 end-system scenarios.

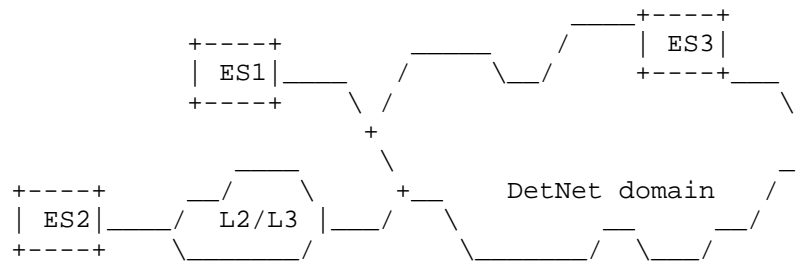
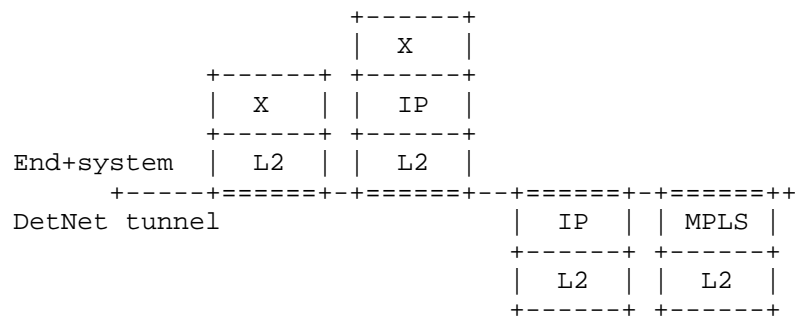


Figure 5: Connection types of L3 end-systems

#### 5.2.1. DetNet Bridging Service

The simplest DetNet service is to provide bridging (i.e., tunneling for L2), where the connected hosts are in the same broadcast (BC) domain. Forwarding over the DetNet domain is based on L2 (MAC) addresses (i.e. dst-MAC), so L2 headers MUST be kept. For both IP and MPLS PSN a DetNet specific tunnel encapsulation MUST be introduced.





## Examples

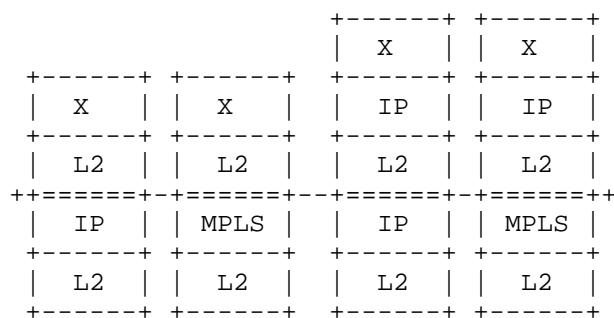


Figure 6: Encapsulation format for DetNet Bridging

As shown on the figure both L2 and L3 end-systems can be served by such a DetNet Bridging service.

### 5.2.2. DetNet Routing Service

DetNet Routing service provides routing, therefore available only for L3 hosts that are in different BC domains. Forwarding over the DetNet domain is based on L3 (IP) addresses (i.e. dst-IP).

#### 5.2.2.1. MPLS PSN

In case of an MPLS PSN at the ingress/egress (i.e., PE nodes of DetNet domain) the IP packets are encapsulated in MPLS. The data-flow IP header MUST be preserved as-is.

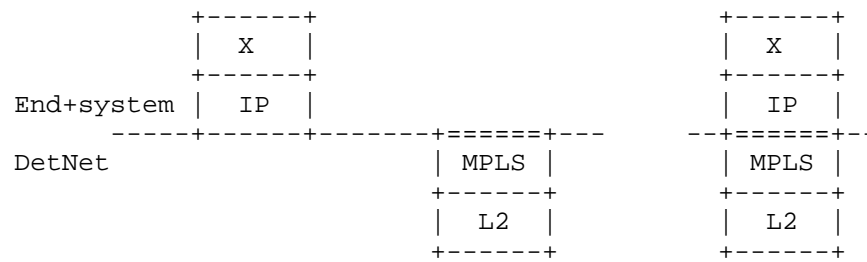


Figure 7: Encapsulation format for DetNet Routing in MPLS PSN for L3 end-systems

#### 5.2.2.2. IP PSN

In case of an IP PSN the same tunneling concept can be used as for an MPLS PSN, but the tunnel is constructed by a new IP header (and possible upper layer fields). The data-flow IP header **MUST** be preserved as-is.

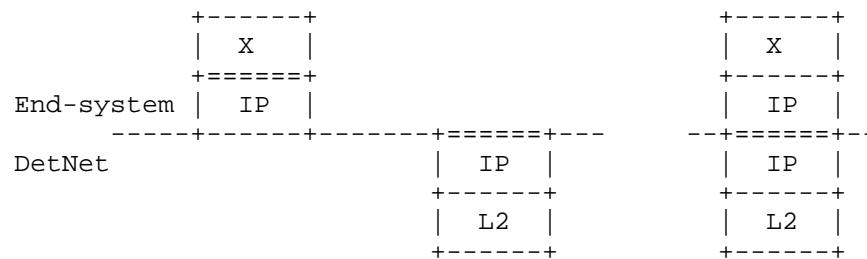


Figure 8: Encapsulation format for DetNet Routing in IP PSN for L3 end-systems

DetNet IP header contains the IP addresses of the ingress/egress PE nodes of DetNet domain. The End-system IP header contains the IP addresses of the end-systems.

Note: In case of IP PSN one may consider avoiding the additional IP encapsulation, however there are many issues with such an approach. First, the DetNet nodes **MUST** be able to extract from the IP header (and maybe upper layers) the attributes required by DetNet functions (i.e. Flow-ID, Seq.Number). The challenge is that encoding of those attributes may be application specific, so DetNet nodes **MUST** be prepared to handle all application specific formats. Second, adding further fields (e.g., explicit path information) to an existing IP header may be impossible (e.g., due to security/encryption).

Furthermore, DetNet domain IP-header format may collide with IP-header format used by the source of a flow. Implementing such an approach requires that source encapsulation is in-line with DetNet domain encapsulation format, however we do not intend to mandate end-systems' encapsulation format (see former text: As a general rule, DetNet domains MUST be capable to forward any Data-flows and the DetNet domain MUST NOT intend to mandate end-system encapsulation format.).

#### 5.2.2.3. Simplified IP Service

In this case there is no "tunneling" below the DetNet Service, but the DetNet Service flows are mapped to each link / sub net using its technology specific methods. The DetNet IP header contains the IP address destination DetNet end system. The data-flow IP header MUST be preserved as-is.

This solution provides end to end DetNet service consisting of congestion protection and latency control and the rouse allocation (queuing, policing, shaping) done using the underlying link / sub net specific mechanisms. Compared to previously described DetNet routing services, the service protections (packet replication and packet emilination functions) and not provided end to end, but per underlying layer-2 link / sub net.

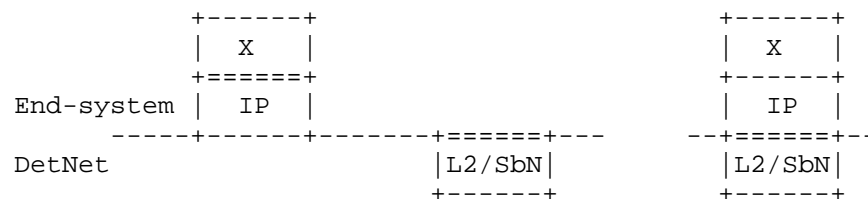


Figure 9: Encapsulation of DetNet Routing in simplified IP service L3 end-systems

Note: the DetNet Service Flow MUST be mapped to the link / sub net specific resources using an underlying system specific means. This implies each DetNet aware node on path MUST look into the transported DetNet Service Flow packet and utilize e.g., a five tuple to find out the required mapping in a node. As noted earlier, the Service Protection is done within each link / sub net independently using the domain specific mechanisms (due the lack of a unified end to end sequencing information that would be available for intermediate nodes). If end to end service protection is desired that can be implemented, for example, by the DetNet end systems using Layer-4

transport protocols or application protocols. However, these are out of scope of this document.

[Editor's note: the service protection to be clarified further.]

### 5.3. DetNet Inter-Working Function (DN-IWF)

#### 5.3.1. Networks with multiple technology segments

There are network scenarios, where the DetNet domain contains multiple technology segments (IP, MPLS) and all those segments are under the same administrative control. Furthermore, DetNet nodes may be interconnected via TSN segments.

An important aspect of DetNet network design is placement of DetNet functions across the domain. Designs based on segment-by-segment optimization can provide only suboptimal solutions. In order to achieve global optimum Inter-Working Functions (DN-IWF) can be placed at segment border nodes, which stitch together DetNet flows across connected segments.

DN-IWF may ensure that flow attributes are correlated across segment borders. For example, there are two DetNet functions which require Seq.Numbers: (1) Elimination: removes duplications from flows and (2) IOD: ensures in-order-delivery of packet in a flow. Stitching flows together and correlating attributes means for example that replication of packets can happen in one segment and elimination of duplicates in a different one.

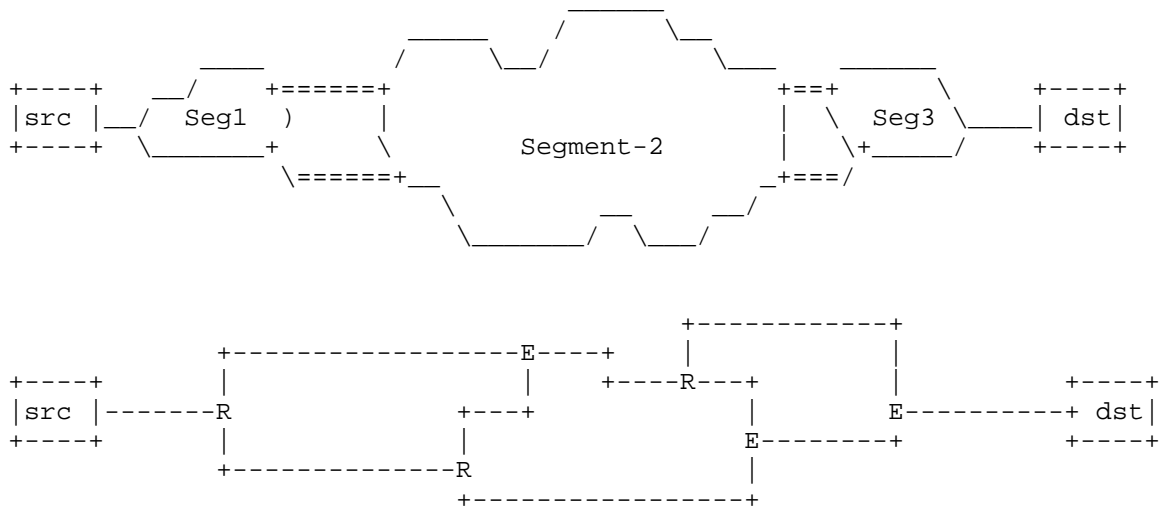


Figure 10: Optimal replication and elimination placement across technology segments example

### 5.3.2. DN-IWF related considerations

The ultimate goal of DN-IWF is to (1) match and (2) translate segment specific flow attributes. The DN-IWF ensures that segment specific attributes comprise per domain unique attributes for the whole DetNet domain. This characteristic can ensure that DetNet functions can be based on per domain attributes and not per segment attributes.

The two DetNet specific attributes have the following characteristics:

- o Flow-ID: it is same in all packets of a flow
- o Seq.Number: it is different packet-by-packet

For the Flow-ID the DN-IWF can implement a static mapping. The situation is more complicated for Seq.Number as it is different packet-by-packet, so it may need more sophisticated translation unless its format is exactly the same in the two technology segments. In this latter case the DN-IWF can simply copy the Seq.Number field between the tunneling encapsulation of the two technology segments.

In case of three technology segments (IP, MPLS and TSN) three DN-IWF functions can be specified. In the rest of this section the focus is on the (1) IP - MPLS network scenario. Note: the use-cases are out-

of-scope for (2) TSN - IP, (3) TSN - MPLS. Note2: incompatible format of Seq.Number with TSN.

Simplest implementation of DN-IWF is provided if the flow attributes have the same format. Such a common denominator of the tunnel encapsulation format is the pseudowire encapsulation over both IP and MPLS.

Placeholder

Figure 11: FIGURE Placeholder PW over X

## 6. MPLS-based DetNet data plane solution

### 6.1. DetNet specific packet fields

The DetNet data plane encapsulation MUST include two DetNet specific information elements in each packet of a DetNet flow: (1) a flow identification and (2) a sequence number.

The DetNet data plane encapsulation may consists further elements used for overlay tunneling, to distinguish between DetNet member flows of the same DetNet compound flow or to support OAM functions.

### 6.2. Data plane encapsulation

Figure 12 illustrates a DetNet data plane MPLS encapsulation. The MPLS-based encapsulation of the DetNet flows is a good fit for the Layer-2 interconnect deployment cases (see Figure 1). Furthermore, end to end DetNet service i.e., native DetNet deployment (see Figure 2) is also possible if DetNet end systems are capable of initiating and termination MPLS encapsulated packets. Transport of IP encapsulated DetNet flows, see Section 7, over MPLS-based DetNet data plane is also possible. Interworking between PW- and IPv6-based encapsulations is discussed further in Section 8.6.

The MPLS-based DetNet data plane encapsulation consists of:

- o DetNet control word (d-CW) containing sequencing information for packet replication and duplicate elimination purposes. There MUST a separate sequence number space for each DetNet flow.
- o DetNet Label that identifies a DetNet flow within a DetNet Edge or a Relay node. The DetNet label MUST be at the bottom of the label stack.

- o An optional DetNet service label (S-Label) that represents DetNet Service LSP used between DetNet Edge and/or Relay nodes. One possible use of an S-Label is to identify DetNet member flows used to provide protection to a DetNet compound flow, perhaps even when both LSPs appear on the same link for some reason.

One or more MPLS transport LSP label(s) (T-label) which may be a hop-by-hop label used between LSR and MUST appear higher in the label stack than S-labels. A top of stack T-label may be PHPed before arriving at a DetNet node. In general T-labels should be considered to be part of the underlying transport network rather than the actual DetNet data plane encapsulation.

DetNet MPLS-based encapsulation

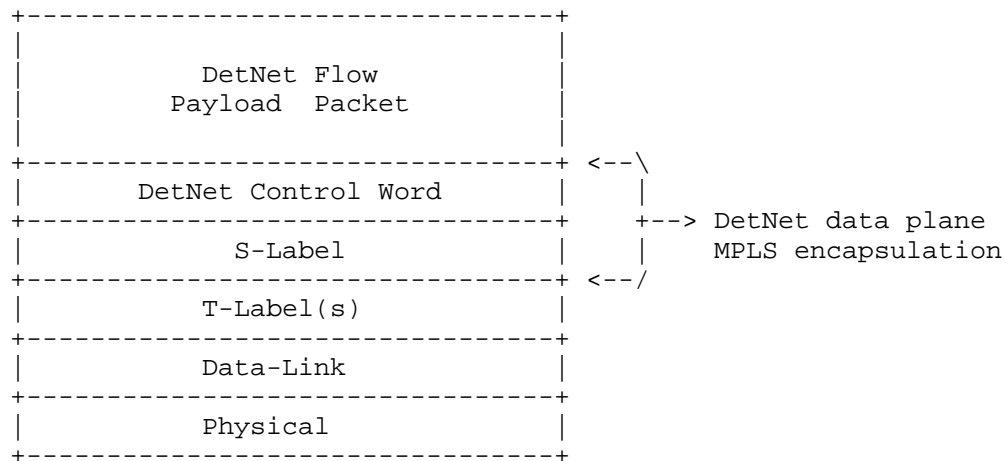


Figure 12: Encapsulation of a DetNet flow in an MPLS(-TP) PSN

### 6.3. DetNet control word

A DetNet control word (d-CW) conforms to the Generic PW MPLS Control Word (PWMCW) defined in [RFC4385] and is illustrated in Figure 13. The upper nibble of the d-CW MUST be set to zero (0). The effective sequence number bit length is between 0 and 28 bits, and configured either by a control plane or manually for each DetNet flow. The sequence number is aligned to the right (least significant bits) and unused bits MUST be set to zero (0). Each DetNet flow MUST have its own sequence number counter. The sequence number is incremented by one for each new packet.

The d-CW MUST always be present in a packet. In a case the sequence number is not used (e.g., for DetNet-t-flows) the control plane or the manual configuration has to define zero (0) bit length sequence number and the value of the sequence number MUST be set to zero (0).

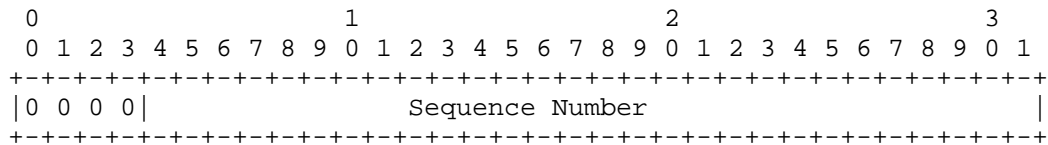


Figure 13: DetNet Control Word

#### 6.4. Flow identification

DetNet flow identification at a DetNet service layer is realized by an S-label. It maps a Detnet flow to a specific d-CW in a DetNet node. The S-label used for flow identification MUST be bottom label of the label stack for a DetNet-s- or DetNet-st-flow and MUST precede the d-CW.

An S-label for a single DetNet flow does not need to be unique DetNet domain wide. As long as two or more different DetNet flows do not erroneously map to a same d-CW in a DetNet node the labels may vary.

#### 6.5. Service layer considerations

[Editor's note: quite a bit of unfinished and old text in the following sections.]

The edge and relay node internal procedures of the PREF are implementation specific. The order of a packet elimination or replication is out of scope in this specification. However, care should be taken that the replication function does not actually loopback packets as "replicas". Looped back packets include artificial delay when the node that originally initiated the packet receives it again. Also, looped back packets may make the network condition to look healthier than it actually is (in some cases link failures are not reflected properly because looped back packets make the situation appear better than it actually is).

Comment #29: SB> There needs to be some text about preventing a node ever receiving its own replicated packets. Indeed that would suggest that the flow id should be changed and replication should only take place on configured flow IDs. I have a feeling that this would all be a lot safer if replication only happened at ingress and we managed the diversity of the paths.



Discussion: Agree on hardening the loopback text considerations.

#### 6.5.1.1. Edge node processing

TBD.

[Editor's note: Since we are not defining the inner workings and implementation of the DetNet Edge node - rather only what goes in and what comes out, and of course the on-wire details, then the figures shown in the coming section would not need to detail the inner architecture of a DetNet Node.]

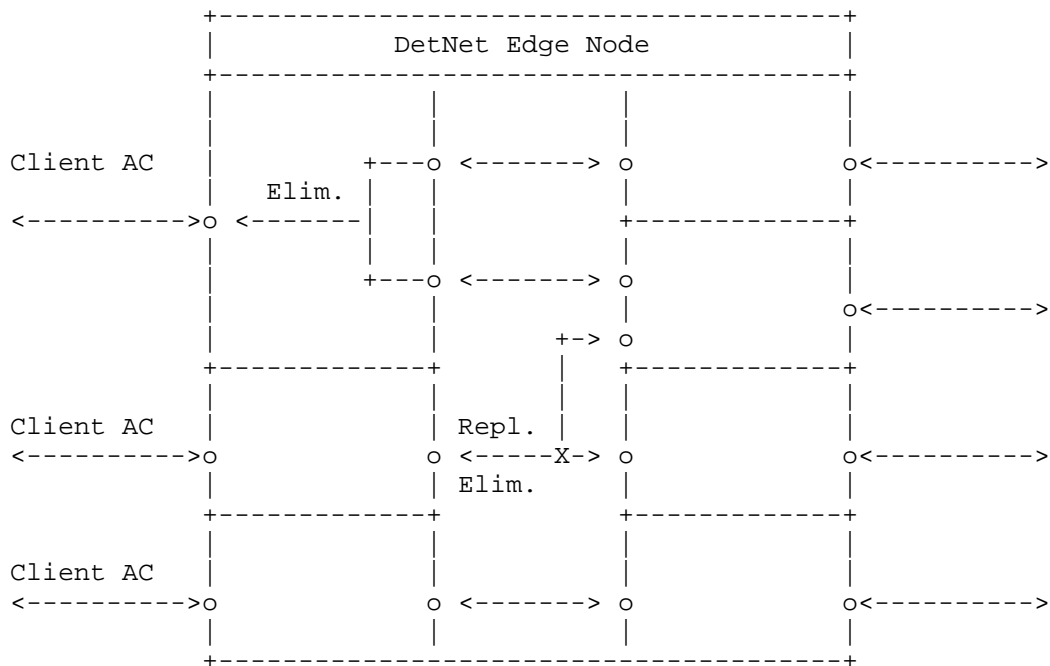


Figure 14: DetNet Edge Node processing

An edge node participates to the packet replication and duplication elimination. Required processing is done within an extended forwarder function. In the case the native service processing (NSP) is IEEE 802.1CB [IEEE8021CB] capable, the packet replication and duplicate elimination MAY entirely be done in the NSP and bypassing the DetNet flow encapsulation and logic entirely, and thus is able to operate over unmodified implementation and deployment. The NSP approach works only between edge nodes and cannot make use of relay nodes (see Section 6.5.2).

Comment #31 SB> This would be a fine way to operate the PW system - edge to edge.

Discussion: When it comes to use of NSPs, agree. Also for "island interconnect" this is a fine. However, when there is a need to do PREF in a middle, plain edge to edge is not enough.

The DetNet-aware extended forwarder selects the egress DetNet member flow based on the DetNet forwarding rules. In both "normal AC" and "Packet AC" cases there may be no DetNet encapsulation header available yet as it is the case with relay nodes (see Section 6.5.2). It is the responsibility of the extended forwarder within the edge node to push the DetNet specific encapsulation (if not already present) to the packet before forwarding it to the appropriate egress DetNet member flow instance(s).

Comment #32 SB> I am not convinced of the wisdom of having a mid-point node convert a flow into a DN flow, which is what you are implying here. This seems like an ingress function.

Discussion: OK. The text here has issues and seems to mix relay and edge.

The extended forwarder MAY copy the sequencing information from the native DetNet packet into the DetNet sequence number field and vice versa. If there is no existing sequencing information available in the native packet or the forwarder chose not to copy it from the native packet, then the extended forwarder MUST maintain a sequence number counter for each DetNet flow (indexed by the DetNet flow identification).

#### 6.5.2. Relay node processing

TBD.

A DetNet Relay node participates to the packet replication and duplication elimination. This processing is done within an extended forwarder function. Whether an ingress DetNet member flow receives DetNet specific processing depends on how the forwarding is programmed. For some DetNet member flows the relay node can act as a normal relay node and for some apply the DetNet specific processing (i.e., PREF).

Comment #34 SB> Again relay node is not a normal term, so am not sure what it does in the absence of a PREF function.

Discussion: Relay node was a DetNet aware S-PE originally, which is not explicitly stated here anymore, thus slightly confusing text

here. The text here needs to clarify the roles of PREF and switching functions. A DetNet relay is described in the architecture document. However, there is definitely room for termonology and text improvements.

It is also possible to treat the relay node as a transit node, see Section 8.3. Again, this is entirely up to how the forwarding has been programmed.

The DetNet-aware forwarder selects the egress DetNet member flow segment based on the flow identification. The mapping of ingress DetNet member flow segment to egress DetNet member flow segment may be statically or dynamically configured. Additionally the DetNet-aware forwarder does duplicate frame elimination based on the flow identification and the sequence number combination. The packet replication is also done within the DetNet-aware forwarder. During elimination and the replication process the sequence number of the DetNet member flow MUST be preserved and copied to the egress DetNet member flow.

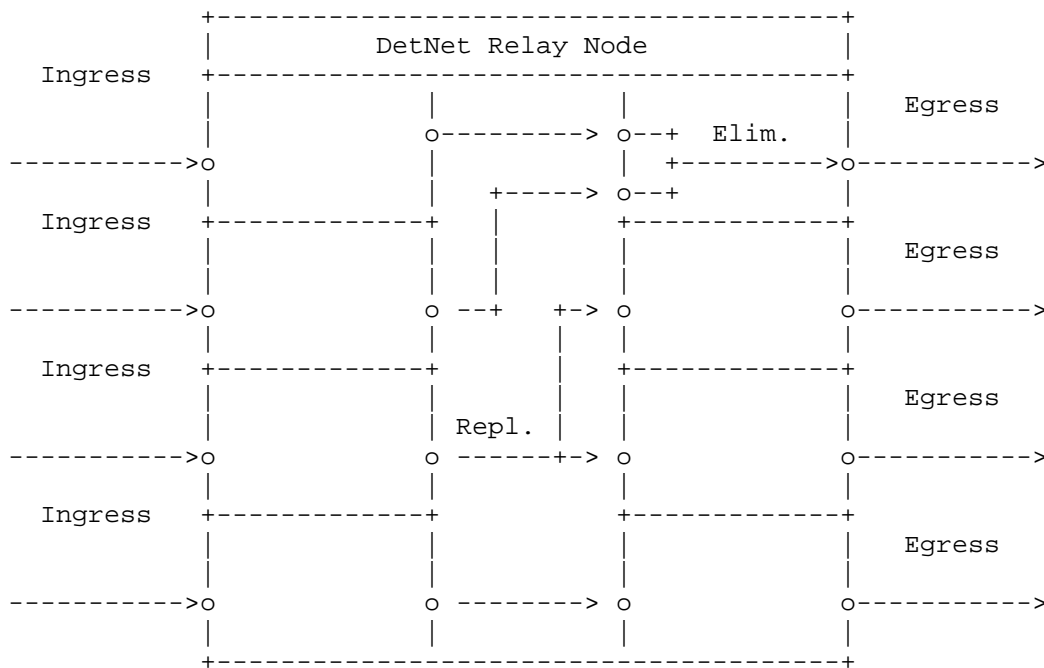


Figure 15: DetNet Relay Node processing

Comment #35 SB> Somewhere in the dp document there needs to be a note of the requirement for interfaces to do fast exchange of counter state, and a note to those planning the network and designing the control plane that they need to provide support for this.

Discussion: We kind of agree but also think the above exchange or synchronization of counter states is not in our scope to solve.

#### 6.5.3. End system processing

TBD.

#### 6.6. Transport node considerations

##### 6.6.1. Congestion protection

TBD.

##### 6.6.2. Explicit routes

TBD.

### 7. IPv6-based DetNet data plane solution

#### 7.1. Data plane encapsulation

Figure 16 illustrates a DetNet native IPv6 encapsulation. The native IPv6 encapsulation is meant for end to end Detnet service use cases, where the end stations are DetNet-aware (see Figure 3). Technically it is possible to use the IPv6 encapsulation to tunnel any traffic over a DetNet enabled network, which would make native IPv6 encapsulation also a valid data plane choice for an interconnect use case (see Figure 1).

The native IPv6-based DetNet data plane encapsulation consists of:

- o IPv6 header as the transport protocol.
- o IPv6 header Flow Label that is used to help to identify a DetNet flow (i.e., roughly an equivalent to an S-Label for the MPLS encapsulation). A Flow Label together with the IPv6 source address uniquely identifies a DetNet flow.

Comment #21 SB> Have we validated that it is unconditionally safe to make this assumption about the use of the FL?

Discussion: RFC6437 does not restrict such use and DetNet DP solution can always define their own use of flow label. It should be noted that a DetNet aware node will always contain new code and is not a load balancer.

- o Zero, one or two DetNet Destination Options containing sequencing information for packet replication and duplicate elimination function (PREF), and/or packet reordering purposes. The DetNet Destination Option is equivalent to the DetNet Control Word. If PREF or packet reordering is not needed for the DetNet flow then no DetNet Destination Option is inserted into the IPv6 header.

A DetNet-aware end station (a host) or an intermediate Detnet node initiating an (or adding a tunnelling) IPv6 packet is responsible for setting the Flow Label, adding the optional DetNet Destination Option(s) for DetNet-s- or DetNet-st-flows, and possibly adding a routing header such as the segment routing option (e.g., for pre-defined paths [I-D.ietf-6man-segment-routing-header]). If a routing header is inserted into the IPv6 packet for DetNet-s- or DetNet-st-flows then a second instance of the DetNet Destination Option MUST be added before the routing header (see Section 4.1 of [RFC8200]).

A DetNet-aware end station (a host) or an intermediate node receiving an IPv6 packet destined to it and containing a DetNet Destination Option does the appropriate processing of the packet. This may involve packet duplication and elimination (PREF processing), terminating a tunnel or delivering the packet to the upper layers/Applications.

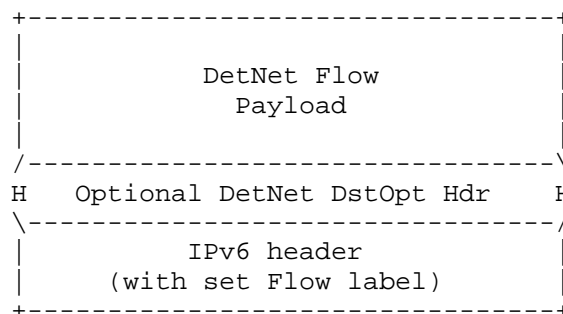


Figure 16: Encapsulation of a native IPv6 DetNet-s- or DetNet-st-flow without a routing header

Figure 17 illustrates an IPv6 packet for the case where a routing header has been added into the packet by a DetNet-aware end system (again assuming DetNet-s- or DetNet-st-flows). Note that the use of

routing header such as the one with the segment routing option is not mandatory for explicit routes. Similar functionality can be arranged using other means as well (e.g., using policy routing or layer-2 means).

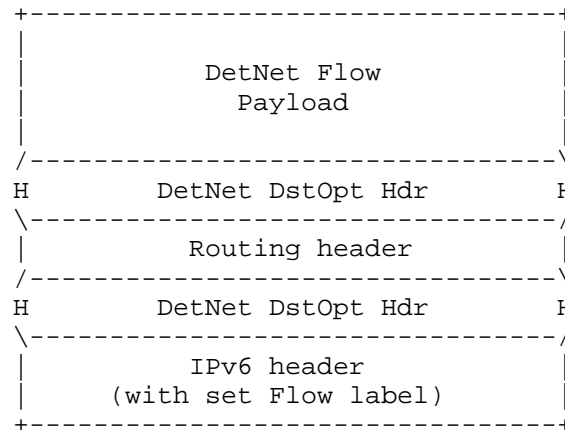


Figure 17: Encapsulation of a native IPv6 DetNet-s- or DetNet-st-flows with routing header

IPv6 extension headers can only be inserted by a node that initiated the IPv6 packet. IPv6 extension headers, except for the Hop-by-Hop Option headers, can only be processed by an IPv6 node that is identified by the Destination Address field of the IPv6 header (see Section 0 of [RFC8200]). Therefore, if a DetNet-aware end system only inserted the DetNet Destination Option into the IPv6 but e.g., a DetNet Edge node is configured to enforce an explicit route for the IPv6 packet using a source routing header, then it has no other possibility than add an outer tunneling IPv6 header with required extension headers in it. The processing of IPv6 packets in a DetNet Edge node is discussed further in Section 7.4.1.

## 7.2. DetNet destination option

A DetNet flow must carry sequencing information for packet replication and elimination function (PREF) purposes. This document specifies a new IPv6 Destination Option: the DetNet Destination Option for that purpose. The format of the option is illustrated in Figure 18.

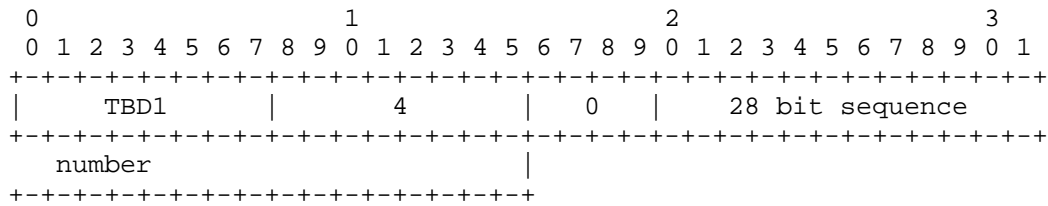


Figure 18: DetNet Destination Option

The Option Type for the DetNet Destination Option is set to TBD1. [To be removed from the final version of the document: The Option Type MUST have the two most significant bits set to 10b]

If an IPv6 packet gets dropped due the DetNet Service layer processing based on the DetNet Destination Option an ICMPv6 packet of any type MUST NOT be sent back to the source of the packet.

### 7.3. Flow identification

The DetNet flow identification is based on the IPv6 Flow Label and the source address combination. The two fields uniquely identify the end to end native IPv6 encapsulated DetNet flow. Obviously, the identification fails if any intermediate node modifies either the source address or the Flow Label.

Comment #27 SB> See earlier. If there are enough IPv6 addresses to address video fragments, why not DN flows? Then this problem goes away.

Discussion: See the earlier comment #25 discussion. If nodes get their addressies via DHCPv6 basically ruins this mechanism. Also the assumption for this to work is that the node has a full /64 to use, which is not always the case. Otherwise the idea is just fine.

### 7.4. Service layer considerations

[Editor's note: this section is TBD. It will detail the PREF functionality.]

- o PREF - requires both flow identification and sequence numbering.
- o Packet reordeing - requires both flow identification and sequence numbering.

A DetNet service layer processing can be done at each DetNet node that matches the IPv6 header's Destination Address. Then, if the DetNet flow identification provides a positive match for the DetNet flow that the node has a service layer state installed e.g., for PREF or packet reordering purposes, further service layer processing takes place. In a case of PREF or packet reordering that means processing the DetNet Destination Option for the identified DetNet flow.

#### 7.4.1. Edge node processing

[Editor's note: This is the start of the IPv6 handling text - there are errors and bad language. The founding assumption is the use of source routing when intermediate nodes (relays/edges) need to modify packets. This is due the text in RFC8200 and the fact that without hph options only routing+dsthdr is usable with intermediates under strict RFC8200.. ]

[Editor's note: Regrading the source routing and the "example" SRv6 approach. Current text is based on the assumption that intermediates cannot add/delete extension headers such as the SRv6. That said adding adding a header implies adding a tunneling outer IPv6 header and deleting a header implies a tunnel decapsulation. This is not probably desired due to the involved overhead and to be discussed whether it is possible/acceptable to just "process" the Application flow packets.]

For a DetNet Edge node there are several scenarios that involve modifications to the DetNet flow IPv6 packets. The assumption is that a DetNet-aware end system has always set the IPv6 header flow label properly for the flow identification purposes. A DetNet- or DetNet-t-flow does not include the DetNet Destination Option. Following cases have been identified:

1. A DetNet App-flow or a DetNet-t-flow packet arrives at an ingress DetNet Edge node and DetNet service layer functions are done only at DetNet Edge nodes. Possible explicit routes between edge nodes are arranged by other than IPv6 specific means.
2. A DetNet App-flow or a DetNet-t-flow packet arrives at an ingress DetNet Edge node and multiple DetNet Relay nodes may process DetNet flow packets before reaching an egress DetNet Edge node. Explicit routes between edge nodes has to be arranged by IPv6 specific means.
3. A DetNet-s- or a DetNet-st-flow packet arrives at an ingress DetNet Edge node and DetNet service layer functions are done only at DetNet Edge nodes. Possible explicit routes between edge nodes are arranged by other than IPv6 specific means.



4. A DetNet-s- or a DetNet-st-flow packet arrives at an ingress DetNet Edge node and multiple DetNet Relay nodes may process DetNet flow packets before reaching an egress DetNet Edge node. Explicit routes between edge nodes has to be arranged by IPv6 specific means.

A generic DetNet IPv6 encapsulation for a DetNet flow packet between DetNet Edge nodes is shown in Figure 19. Essentially every time an ingress DetNet Edge node has to insert something into the DetNet flow packet it has to add an outer tunneling IPv6 header, which then contain possible additional extension headers.

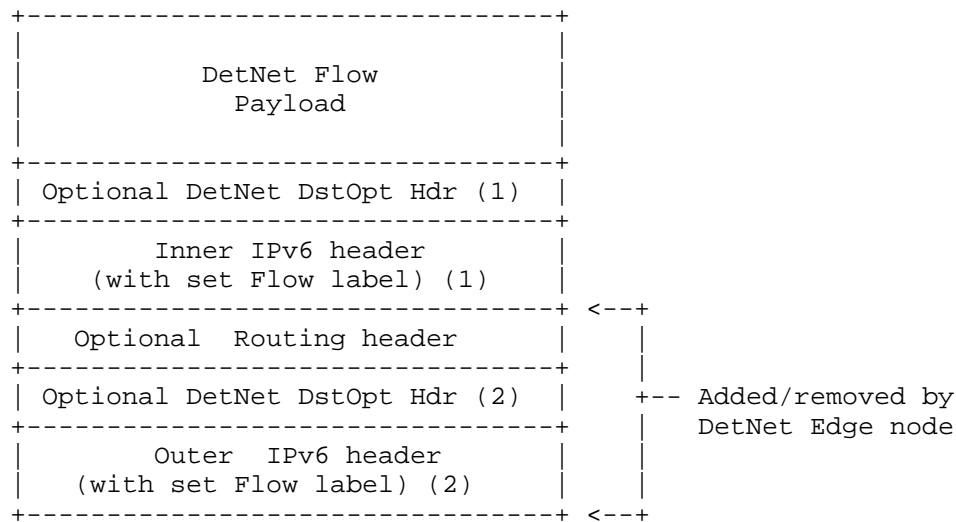


Figure 19: Encapsulation of a DetNet-flow IPv6 packet at the DetNet Edge node

#### 7.4.1.1. Ingress DetNet Edge node processing

Case 1) MAY require an addition of the DetNet Destination Option if packet reordering is requested at the egress DetNet Edge node. Otherwise, no modifications except rewriting the IPv6 header flow label to the packet is done. If modifications are required then:

- o The outer IPv6 header is added with the Source Address set to the ingress DetNet Edge node address and the Destination Address set to the egress DetNet Edge node address.
- o The flow label of the outer IPv6 header SHOULD be set to a value maintained by the edge node.

- o The DetNet Destination Option with the edge node managed per DetNet flow sequence number value is inserted into the outer IPv6 header.

Case 2) requires an addition of the DetNet Destination Option unless neither packet reordering or PREF is enable at any DetNet Edge/Relay node. A source routing header has to be added for the explicit route purposes. An example of the source routing header is the Segment Routing header. The following modifications to DetNet flow IPv6 packets are required:

- o An outer IPv6 header is added with the Source Address set to the ingress DetNet Edge node address and the Destination Address set to the egress DetNet Edge node address.
- o The flow label of the outer IPv6 header SHOULD be set to a value maintained by the edge node.
- o The DetNet Destination Option with the edge node managed per DetNet flow sequence number value MAY be inserted into the outer IPv6 header.
- o A source routing header with addresses of those DetNet Relay nodes that must be traversed is inserted into the outer IPv6 header.

Case 3) ...[Editor's note: is it OK if the sequence number added here by the edge node has only local significance between the edge nodes and not end to end between end systems? ]

Case 4) ...

#### 7.4.1.2. Egress DetNet Edge node processing

#### 7.4.2. Relay node processing

TBD.

#### 7.4.3. End system processing

TBD.

#### 7.5. Transport node processing

##### 7.5.1. Congestion protection

### 7.5.2. Explicit routes

## 8. Other DetNet data plane considerations

### 8.1. Class of Service

Class and quality of service, i.e., CoS and QoS, are terms that are often used interchangeably and confused. In the context of DetNet, CoS is used to refer to mechanisms that provide traffic forwarding treatment based on aggregate group basis and QoS is used to refer to mechanisms that provide traffic forwarding treatment based on a specific DetNet flow basis. Examples of existing network level CoS mechanisms include DiffServ which is enabled by IP header differentiated services code point (DSCP) field [RFC2474] and MPLS label traffic class field [RFC5462], and at Layer-2, by IEEE 802.1p priority code point (PCP).

CoS for DetNet flows carried in PWs and MPLS is provided using the existing MPLS Differentiated Services (DiffServ) architecture [RFC3270]. Both E-LSP and L-LSP MPLS DiffServ modes MAY be used to support DetNet flows. The Traffic Class field (formerly the EXP field) of an MPLS label follows the definition of [RFC5462] and [RFC3270]. The Uniform, Pipe, and Short Pipe DiffServ tunneling and TTL processing models are described in [RFC3270] and [RFC3443] and MAY be used for MPLS LSPs supporting DetNet flows. MPLS ECN MAY also be used as defined in ECN [RFC5129] and updated by [RFC5462].

CoS for DetNet flows carried in IPv6 is provided using the standard differentiated services code point (DSCP) field [RFC2474] and related mechanisms. The 2-bit explicit congestion notification (ECN) [RFC3168] field MAY also be used.

One additional consideration for DetNet nodes which support CoS services is that they MUST ensure that the CoS service classes do not impact the congestion protection and latency control mechanisms used to provide DetNet QoS. This requirement is similar to requirement for MPLS LSRs to that CoS LSPs do not impact the resources allocated to TE LSPs via [RFC3473].

### 8.2. Quality of Service

Quality of Service (QoS) mechanisms for flow specific traffic treatment typically includes a guarantee/agreement for the service, and allocation of resources to support the service. Example QoS mechanisms include discrete resource allocation, admission control, flow identification and isolation, and sometimes path control, traffic protection, shaping, policing and remarking. Example protocols that support QoS control include Resource ReSerVation

Protocol (RSVP) [RFC2205] (RSVP) and RSVP-TE [RFC3209] and [RFC3473]. The existing MPLS mechanisms defined to support CoS [RFC3270] can also be used to reserve resources for specific traffic classes.

In addition to explicit routes, and packet replication and elimination, described in Section 6 above, DetNet provides zero congestion loss and bounded latency and jitter. As described in [I-D.ietf-detnet-architecture], there are different mechanisms that maybe used separately or in combination to deliver a zero congestion loss service. These mechanisms are provided by the either the MPLS or IP layers, and may be combined with the mechanisms defined by the underlying network layer such as 802.1TSN.

A baseline set of QoS capabilities for DetNet flows carried in PWs and MPLS can provided by MPLS with Traffic Engineering (MPLS-TE) [RFC3209] and [RFC3473]. TE LSPs can also support explicit routes (path pinning). Current service definitions for packet TE LSPs can be found in "Specification of the Controlled Load Quality of Service", [RFC2211], "Specification of Guaranteed Quality of Service", [RFC2212], and "Ethernet Traffic Parameters", [RFC6003]. Additional service definitions are expected in future documents to support the full range of DetNet services. In all cases, the existing label-based marking mechanisms defined for TE-LSPs and even E-LSPs are use to support the identification of flows requiring DetNet QoS.

QoS for DetNet flows carried in IPv6 MUST be provided locally by the DetNet-aware hosts and routers supporting DetNet flows. Such support will leverage the underlying network layer such as 802.1TSN. The traffic control mechanisms used to deliver QoS for IP encapsulated DetNet flows are expected to be defined in a future document. From an encapsulation perspective, and as defined in Section 7, the combination of the Flow Label together with the IP source address uniquely identifies a DetNet flow.

Packets that are marked with a DetNet Class of Service value, but that have not been the subject of a completed reservation, can disrupt the QoS offered to properly reserved DetNet flows by using resources allocated to the reserved flows. Therefore, the network nodes of a DetNet network MUST:

- o Defend the DetNet QoS by discarding or remarking (to a non-DetNet CoS) packets received that are not the subject of a completed reservation.
- o Not use a DetNet reserved resource, e.g. a queue or shaper reserved for DetNet flows, for any packet that does not carry a DetNet Class of Service marker.

### 8.3. Cross-DetNet flow resource aggregation

The ability to aggregate individual flows, and their associated resource control, into a larger aggregate is an important technique for improving scaling of control in the data, management and control planes. This document identifies the traffic identification related aspects of aggregation of DetNet flows. The resource control and management aspects of aggregation (including the queuing/shaping/policing implications) will be covered in other documents. The data plane implications of aggregation are independent for PW/MPLS and IP encapsulated DetNet flows.

DetNet flows transported via MPLS can leverage MPLS-TE's existing support for hierarchical LSPs (H-LSPs), see [RFC4206]. H-LSPs are typically used to aggregate control and resources, they may also be used to provide OAM or protection for the aggregated LSPs. Arbitrary levels of aggregation naturally falls out of the definition for hierarchy and the MPLS label stack [RFC3032]. DetNet nodes which support aggregation (LSP hierarchy) map one or more LSPs (labels) into and from an H-LSP. Both carried LSPs and H-LSPs may or may not use the TC field, i.e., L-LSPs or E-LSPs. Such nodes will need to ensure that traffic from aggregated LSPs are placed (shaped/policed/enqueued) onto the H-LSPs in a fashion that ensures the required DetNet service is preserved.

DetNet flows transported via IP have more limited aggregation options, due to the available traffic flow identification fields of the IP solution. One available approach is to manage the resources associated with a DSCP identified traffic class and to map (remark) individually controlled DetNet flows onto that traffic class. This approach also requires that nodes support aggregation ensure that traffic from aggregated LSPs are placed (shaped/policed/enqueued) in a fashion that ensures the required DetNet service is preserved.

Comment #38 SB> I am sure we can do better than this with SR, or the use of routing techniques that map certain addresses to certain paths.

Discussion: --

In both the MPLS and IP cases, additional details of the traffic control capabilities needed at a DetNet-aware node may be covered in the new service descriptions mentioned above or in separate future documents. Management and control plane mechanisms will also need to ensure that the service required on the aggregate flow (H-LSP or DSCP) are provided, which may include the discarding or remarking mentioned in the previous sections.

#### 8.4. Bidirectional traffic

Some DetNet applications generate bidirectional traffic. Using MPLS definitions [RFC5654] there are associated bidirectional flows, and co-routed bidirectional flows. MPLS defines a point-to-point associated bidirectional LSP as consisting of two unidirectional point-to-point LSPs, one from A to B and the other from B to A, which are regarded as providing a single logical bidirectional transport path. This would be analogous of standard IP routing, or PWs running over two reciprocal unidirectional LSPs. MPLS defines a point-to-point co-routed bidirectional LSP as an associated bidirectional LSP which satisfies the additional constraint that its two unidirectional component LSPs follow the same path (in terms of both nodes and links) in both directions. An important property of co-routed bidirectional LSPs is that their unidirectional component LSPs share fate. In both types of bidirectional LSPs, resource allocations may differ in each direction. The concepts of associated bidirectional flows and co-routed bidirectional flows can be applied to DetNet flows as well whether IPv6 or MPLS is used.

While the IPv6 and MPLS data planes must support bidirectional DetNet flows, there are no special bidirectional features with respect to the data plane other than need for the two directions take the same paths. Fate sharing and associated vs co-routed bidirectional flows can be managed at the control level. Note, that there is no stated requirement for bidirectional DetNet flows to be supported using the same IPv6 Flow Labels or MPLS Labels in each direction. Control mechanisms will need to support such bidirectional flows for both IPv6 and MPLS, but such mechanisms are out of scope of this document. An example control plane solution for MPLS can be found in [RFC7551].

#### 8.5. Layer 2 addressing and QoS Considerations

The Time-Sensitive Networking (TSN) Task Group of the IEEE 802.1 Working Group have defined (and are defining) a number of amendments to IEEE 802.1Q [IEEE8021Q] that provide zero congestion loss and bounded latency in bridged networks. IEEE 802.1CB [IEEE8021CB] defines packet replication and elimination functions that should prove both compatible with and useful to, DetNet networks.

As is the case for DetNet, a Layer 2 network node such as a bridge may need to identify the specific DetNet flow to which a packet belongs in order to provide the TSN/DetNet QoS for that packet. It also will likely need a CoS marking, such as the priority field of an IEEE Std 802.1Q VLAN tag, to give the packet proper service.

Although the flow identification methods described in IEEE 802.1CB [IEEE8021CB] are flexible, and in fact, include IP 5-tuple

identification methods, the baseline TSN standards assume that every Ethernet frame belonging to a TSN stream (i.e. DetNet flow) carries a multicast destination MAC address that is unique to that flow within the bridged network over which it is carried. Furthermore, IEEE 802.1CB [IEEE8021CB] describes three methods by which a packet sequence number can be encoded in an Ethernet frame.

Ensuring that the proper Ethernet VLAN tag priority and destination MAC address are used on a DetNet/TSN packet may require further clarification of the customary L2/L3 transformations carried out by routers and edge label switches. Edge nodes may also have to move sequence number fields among Layer 2, PW, and IPv6 encapsulations.

#### 8.6. Interworking between MPLS- and IPv6-based encapsulations

[Editor's note: add considerations for interworking between MPLS-based and native IPv6-based DetNet encapsuations.]

#### 8.7. IPv4 considerations

[Editor's note: The fact is that there are and will be deployments using IPv4. Neglecting it entirely is not feasible.]

#### 9. Time synchronization

Comment #39 SB> This section should point the reader to RFC8169 (residence time in MPLS n/w. We need to consider if we need to introduce the same concept in IP.

Discussion: Agree. For IP we could reference to PTPv2 or v3 over UDP/IP, since it measures residence time among other things.

[Editor's note: describe a bit of issues and deployment considerations related to time-synchronization within DetNet. Refer to DT discussion and the slides that summarize different approaches and rough synchronization performance numbers. Finally, scope time-synchronization solution outside data plane.]

When DetNet is used, there is an underlying assumption that the application(s) require clock synchronization such as the Precision Time Protocol (PTP) [IEEE1588]. The relay nodes may or may not utilize clock synchronization in order to provide zero congestion loss and controlled latency delivery. In either case, there are a few possible approaches of how synchronization protocol packets are forwarded and handled by the network:

- o PTP packets can be sent either as DetNet flows or as high-priority best effort packets. Using DetNet for PTP packets requires

careful consideration to prevent unwanted interactions between clock-synchronized network nodes and the packets that synchronize the clocks.

- o PTP packets are sent as a normal DetNet flow through network nodes that are not time-synchronized: in this approach PTP traffic is forwarded as a DetNet flow, and as such it is forwarded in a way that allows a low delay variation. However, since intermediate nodes do not take part in the synchronization protocol, this approach provides a relatively low degree of accuracy.
- o PTP with on-path support: in this approach PTP packets are sent as ordinary or as DetNet flows, and intermediate nodes take part in the protocol as Transparent Clocks or Boundary Clocks [IEEE1588]. The on-path PTP support by intermediate nodes provides a higher degree of accuracy than the previous approach. The actual accuracy depends on whether all intermediate nodes are PTP-capable, or only a subset of them.
- o Time-as-a-service: in this approach accurate time is provided as-a-service to the DetNet source and destination, as well as the intermediate nodes. Since traffic between the source and destination is sent over a provider network, if the provider supports time-as-a-service, then accurate time can be provided to both the source and the destination of DetNet traffic. This approach can potentially provide the highest degree of accuracy.

It is expected that the latter approach will be the most common one, as it provides the highest degree of accuracy, and creates a layer separation between the DetNet data and the synchronization service.

It should be noted that in all four approaches it is not recommended to use replication and elimination for synchronization packets; the replication/elimination approach may in some cases reduce the synchronization accuracy, since the observed path delay will be bivalent.

Comment #40 SB> I am not sure why we should not use PREP. We should explain to the reader.

Discussion: Agree that a this can be opened a bit more in detail. The issue is explained briefly in the last sentence but it could be more clear.



## 10. Management and control considerations

[Editor's note: This section needs to be different for MPLS and IPv6 solutions. Most solutions are technology dependant,]

While management plane and control planes are traditionally considered separately, from the Data Plane perspective there is no practical difference based on the origin of flow provisioning information. This document therefore does not distinguish between information provided by a control plane protocol, e.g., RSVP-TE [RFC3209] and [RFC3473], or by a network management mechanisms, e.g., RestConf [RFC8040] and YANG [RFC7950].

[Editor's note: This section is a work in progress. discuss here what kind of enhancements are needed for DetNet and specifically for PREF and DetNet zero congest loss and latency control. Need to cover both traffic control (queuing) and connection control (control plane).]

### 10.1. MPLS-based data plane

#### 10.1.1. S-Label assignment and distribution

[Editor's note: Outdated and MPLS specific.. and needs more work.]

The DetNet S-Label distribution follows the same mechanisms specified for XYZ . The details of the control plane protocol solution required for the label distribution and the management of the label number space are out of scope of this document.

#### 10.1.2. Explicit routes

[Editor's note: Outdated.. and needs more work.]

[TBD: based on MPLS TE and possibly IPv6 SR]

### 10.2. IPv6-based data plane

#### 10.2.1. Flow Label assignment and distribution

[Editor's note: Outdated and IPv6 Specific.. and needs more work.]

The IPv6 Flow Label distribution and the label number space are out of scope of this document. However, it should be noted that the combination of the IPv6 source address and the IPv6 Flow Label is assumed to be unique within the DetNet-enabled network. Therefore, as long as each node is able to assign unique Flow Labels for the

source address(es) it is using the DetNet-enabled network wide flow identification uniqueness is guaranteed.

#### 10.2.2. Explicit routes

[Editor's note: Outdated.. and needs more work.]

[TBD: What we have there for IPv6 and explicit routes]

#### 10.3. Packet replication and elimination

[Editor's note: Outdated and at the functional level technology independent.. but needs more work.]

The control plane protocol solution required for managing the PREF processing is outside the scope of this document.

#### 10.4. Congestion protection and latency control

[TBD]

#### 10.5. Flow aggregation control

[TBD]

#### 11. Security considerations

The security considerations of DetNet in general are discussed in [I-D.ietf-detnet-architecture] and [I-D.sdt-detnet-security]. Other security considerations will be added in a future version of this draft.

#### 12. IANA considerations

TBD.

#### 13. Acknowledgements

The author(s) ACK and NACK.

The following people were part of the DetNet Data Plane Solution Design Team:

Jouni Korhonen

Janos Farkas

Norman Finn

Balazs Varga

Loa Andersson

Tal Mizrahi

David Mozes

Yuanlong Jiang

Carlos J. Bernardos

The DetNet chairs serving during the DetNet Data Plane Solution Design Team:

Lou Berger

Pat Thaler

## 14. References

### 14.1. Normative references

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2211] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", RFC 2211, DOI 10.17487/RFC2211, September 1997, <<https://www.rfc-editor.org/info/rfc2211>>.
- [RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, DOI 10.17487/RFC2212, September 1997, <<https://www.rfc-editor.org/info/rfc2212>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.

- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <<https://www.rfc-editor.org/info/rfc3270>>.
- [RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <<https://www.rfc-editor.org/info/rfc3443>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, DOI 10.17487/RFC4206, October 2005, <<https://www.rfc-editor.org/info/rfc4206>>.
- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<https://www.rfc-editor.org/info/rfc4385>>.
- [RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, DOI 10.17487/RFC5129, January 2008, <<https://www.rfc-editor.org/info/rfc5129>>.
- [RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <<https://www.rfc-editor.org/info/rfc5462>>.

- [RFC6003] Papadimitriou, D., "Ethernet Traffic Parameters", RFC 6003, DOI 10.17487/RFC6003, October 2010, <<https://www.rfc-editor.org/info/rfc6003>>.
- [RFC6073] Martini, L., Metz, C., Nadeau, T., Bocci, M., and M. Aissaoui, "Segmented Pseudowire", RFC 6073, DOI 10.17487/RFC6073, January 2011, <<https://www.rfc-editor.org/info/rfc6073>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

#### 14.2. Informative references

- [I-D.ietf-6man-segment-routing-header]  
Previdi, S., Filsfils, C., Raza, K., Dukes, D., Leddy, J., Field, B., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Matsushima, S., Leung, I., Linkova, J., Aries, E., Kosugi, T., Vyncke, E., Lebrun, D., Steinberg, D., and R. Raszuk, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-08 (work in progress), January 2018.
- [I-D.ietf-detnet-architecture]  
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-04 (work in progress), October 2017.
- [I-D.ietf-detnet-dp-alt]  
Korhonen, J., Farkas, J., Mirsky, G., Thubert, P., Zhuangyan, Z., and L. Berger, "DetNet Data Plane Protocol and Solution Alternatives", draft-ietf-detnet-dp-alt-00 (work in progress), October 2016.
- [I-D.sdt-detnet-security]  
Mizrahi, T., Grossman, E., Hacker, A., Das, S., "Deterministic Networking (DetNet) Security Considerations", draft-sdt-detnet-security, work in progress", 2017.
- [IEEE1588]  
IEEE, "IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", 2008.

## [IEEE8021CB]

Finn, N., "Draft Standard for Local and metropolitan area networks - Seamless Redundancy", IEEE P802.1CB /D2.1 P802.1CB, December 2015, <<http://www.ieee802.org/1/files/private/cb-drafts/d2/802-1CB-d2-1.pdf>>.

## [IEEE8021Q]

IEEE 802.1, "Standard for Local and metropolitan area networks--Bridges and Bridged Networks (IEEE Std 802.1Q-2014)", 2014, <<http://standards.ieee.org/about/get/>>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.

[RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <<https://www.rfc-editor.org/info/rfc3985>>.

[RFC5654] Niven-Jenkins, B., Ed., Brungard, D., Ed., Betts, M., Ed., Sprecher, N., and S. Ueno, "Requirements of an MPLS Transport Profile", RFC 5654, DOI 10.17487/RFC5654, September 2009, <<https://www.rfc-editor.org/info/rfc5654>>.

[RFC7551] Zhang, F., Ed., Jing, R., and R. Gandhi, Ed., "RSVP-TE Extensions for Associated Bidirectional Label Switched Paths (LSPs)", RFC 7551, DOI 10.17487/RFC7551, May 2015, <<https://www.rfc-editor.org/info/rfc7551>>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

## Appendix A. Example of DetNet data plane operation

[Editor's note: Add a simplified example of DetNet data plane and how labels etc work in the case of MPLS-based PSN and utilizing PREF. The figure is subject to change depending on the further DT decisions on the label handling..]

## Appendix B. Example of pinned paths using IPv6

TBD.

## Authors' Addresses

Jouni Korhonen (editor)  
Nordic Semiconductor  
  
Email: jouni.nospam@gmail.com

Loa Andersson  
Huawei  
  
Email: loa@pi.nu

Yuanlong Jiang  
Huawei  
  
Email: jiangyuanlong@huawei.com

Norman Finn  
Huawei  
3101 Rio Way  
Spring Valley, CA 91977  
USA  
  
Email: norman.finn@mail01.huawei.com

Balazs Varga  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary  
  
Email: balazs.a.varga@ericsson.com

Janos Farkas  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary

Email: [janos.farkas@ericsson.com](mailto:janos.farkas@ericsson.com)

Carlos J. Bernardos  
Universidad Carlos III de Madrid  
Av. Universidad, 30  
Leganes, Madrid 28911  
Spain

Phone: +34 91624 6236  
Email: [cjbc@it.uc3m.es](mailto:cjbc@it.uc3m.es)  
URI: <http://www.it.uc3m.es/cjbc/>

Tal Mizrahi  
Marvell  
6 Hamada st.  
Yokneam  
Israel

Email: [talmi@marvell.com](mailto:talmi@marvell.com)

Lou Berger  
LabN Consulting, L.L.C.

Email: [lberger@labn.net](mailto:lberger@labn.net)



DetNet  
Internet-Draft  
Intended status: Standards Track  
Expires: September 23, 2018

J. Korhonen, Ed.  
Nordic  
L. Andersson  
Y. Jiang  
N. Finn  
Huawei  
B. Varga  
J. Farkas  
Ericsson  
CJ. Bernardos  
UC3M  
T. Mizrahi  
Marvell  
L. Berger  
LabN  
March 22, 2018

DetNet Data Plane Encapsulation  
draft-ietf-detnet-dp-sol-04

Abstract

This document specifies Deterministic Networking data plane encapsulation solutions. The described data plane solutions can be applied over either IP or MPLS Packet Switched Networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 23, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
2.1. Terms used in this document . . . . .	4
2.2. Abbreviations . . . . .	5
3. Requirements language . . . . .	6
4. DetNet data plane overview . . . . .	6
4.1. DetNet data plane encapsulation requirements . . . . .	8
4.2. Packet replication and elimination considerations . . . . .	10
4.3. Packet reordering considerations . . . . .	10
5. DetNet encapsulation . . . . .	10
5.1. End-system specific considerations . . . . .	10
5.2. DetNet domain specific considerations . . . . .	12
5.2.1. DetNet Bridging Service . . . . .	13
5.2.2. DetNet Routing Service . . . . .	14
5.3. DetNet Inter-Working Function (DN-IWF) . . . . .	17
5.3.1. Networks with multiple technology segments . . . . .	17
5.3.2. DN-IWF related considerations . . . . .	18
6. MPLS-based DetNet data plane solution . . . . .	19
6.1. DetNet specific packet fields . . . . .	19
6.2. Data plane encapsulation . . . . .	19
6.3. DetNet control word . . . . .	20
6.4. Flow identification . . . . .	21
6.5. Service layer considerations . . . . .	21
6.5.1. Edge node processing . . . . .	22
6.5.2. Relay node processing . . . . .	23
6.5.3. End system processing . . . . .	25
6.6. Transport node considerations . . . . .	25
6.6.1. Congestion protection . . . . .	25
6.6.2. Explicit routes . . . . .	25
7. Simplified IP based DetNet data plane solution . . . . .	25
8. Other DetNet data plane considerations . . . . .	25

8.1.	Class of Service . . . . .	25
8.2.	Quality of Service . . . . .	26
8.3.	Cross-DetNet flow resource aggregation . . . . .	27
8.4.	Bidirectional traffic . . . . .	28
8.5.	Layer 2 addressing and QoS Considerations . . . . .	29
8.6.	Interworking between MPLS- and IPv6-based encapsulations . . . . .	29
8.7.	IPv4 considerations . . . . .	30
9.	Time synchronization . . . . .	30
10.	Management and control considerations . . . . .	31
10.1.	MPLS-based data plane . . . . .	32
10.1.1.	S-Label assignment and distribution . . . . .	32
10.1.2.	Explicit routes . . . . .	32
10.2.	IPv6-based data plane . . . . .	32
10.2.1.	Flow Label assignment and distribution . . . . .	32
10.2.2.	Explicit routes . . . . .	32
10.3.	Packet replication and elimination . . . . .	32
10.4.	Congestion protection and latency control . . . . .	33
10.5.	Flow aggregation control . . . . .	33
11.	Security considerations . . . . .	33
12.	IANA considerations . . . . .	33
13.	Acknowledgements . . . . .	33
14.	References . . . . .	34
14.1.	Normative references . . . . .	34
14.2.	Informative references . . . . .	36
Appendix A.	Example of DetNet data plane operation . . . . .	37
Appendix B.	Example of pinned paths using IPv6 . . . . .	38
Authors' Addresses	. . . . .	38

## 1. Introduction

Deterministic Networking (DetNet) is a service that can be offered by a network to DetNet flows. DetNet provides these flows extremely low packet loss rates and assured maximum end-to-end delivery latency. General background and concepts of DetNet can be found in [I-D.ietf-detnet-architecture].

This document specifies the DetNet data plane and the on-wire encapsulation of DetNet flows. The specified encapsulation provides the building blocks to enable the DetNet service layer functions and allow flow identification as described in the DetNet Architecture. Two data plane definitions are given.

1. MPLS-based: The encapsulation resembles PseudoWires (PW) with an MPLS Packet Switched Network (PSN) [RFC3985][RFC4385].
2. Native-IP-based: The encapsulating protocol is IPv6 and the solution relies on IP header fields, existing and DetNet specific IPv6 extension header options [RFC8200].

[Editor's note: MPLS- and IPv6-based solutions are likely to be split into different documents.]

It is worth noting that while MPLS-based solution can transport IP packets a native-IP solution is meant for deployments where the DetNet service layer functions are provided at the IP-layer rather than the underlying transport network. The primary reason for this is the benefit gained by enabling the use of a normal application stack, where transport protocols such as TCP or UDP are directly encapsulated in IP.

The DetNet transport layer functionality that provides congestion protection for DetNet flows is assumed to be in place in a DetNet node.

Furthermore, this document also describes how DetNet flows are identified, how a DetNet Relay/Edge/Transit nodes work, and how the Packet Replication and Elimination function (PREF) is implemented with the two data plane solutions.

This document does not define the associated control plane functions, or Operations, Administration, and Maintenance (OAM). It also does not specify traffic handling capabilities required to deliver congestion protection and latency control for DetNet flows at the DetNet transport layer.

## 2. Terminology

### 2.1. Terms used in this document

This document uses the terminology established in the DetNet architecture [I-D.ietf-detnet-architecture] and the DetNet Data Plane Solution Alternatives [I-D.ietf-detnet-dp-alt].

T-Label	A label used to identify the LSP used to transport a DetNet flow across an MPLS PSN, e.g., a hop-by-hop label used between label switching routers (LSR).
S-Label	A DetNet "service" label that is used between DetNet nodes that implement also the DetNet service layer functions. An S-Label is also used to identify a DetNet flow at DetNet service layer.
Flow Label	IPv6 header field that is used to identify a DetNet flow (together with the source IP address field).
Local-ID	A DetNet Edge and Relay node internal construct that uniquely identifies a DetNet flow within a node and

never appear on-wire. It may be used to select proper forwarding and/or DetNet specific service function.

**PREF** A Packet Replication and Elimination Function (PREF) does the replication and elimination processing of DetNet flow packets in edge or relay nodes. The replication function is essentially the existing 1+1 protection mechanism. The elimination function reuses and extends the existing duplicate detection mechanism to operate over multiple (separate) DetNet member flows of a DetNet compound flow.

**DetNet Control Word** A control word used for sequencing and identifying duplicate packets at the DetNet service layer.

## 2.2. Abbreviations

The following abbreviations used in this document:

AC	Attachment Circuit.
CE	Customer Edge equipment.
CoS	Class of Service.
CW	Control Word.
d-CW	DetNet Control Word.
DetNet	Deterministic Networking.
DF	DetNet Flow.
L2VPN	Layer 2 Virtual Private Network.
LSR	Label Switching Router.
MPLS	Multiprotocol Label Switching.
MPLS-TP	Multiprotocol Label Switching - Transport Profile.
MS-PW	Multi-Segment PseudoWire (MS-PW).
NSP	Native Service Processing.
OAM	Operations, Administration, and Maintenance.

PE	Provider Edge.
PREF	Packet Replication and Elimination Function.
PSN	Packet Switched Network.
PW	PseudoWire.
QoS	Quality of Service.
TSN	Time-Sensitive Network.

### 3. Requirements language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

### 4. DetNet data plane overview

This document describes how to use IP and/or MPLS to support a data plane method of flow identification and packet forwarding over layer-3. Two different cases are covered: (i) the inter-connect scenario, in which IEEE802.1 TSN is routed over a layer-3 network (i.e., to enlarge the layer-2 domain), and (ii) native connectivity between DetNet-aware end systems.

Figure 1 illustrates how DetNet can provide services for IEEE 802.1TSN end systems over a DetNet enabled network. The edge nodes insert and remove required DetNet data plane encapsulation. The 'X' in the edge and relay nodes represents a potential DetNet flow packet replication and elimination point. This conceptually parallels L2VPN services, and could leverage existing related solutions as discussed below.

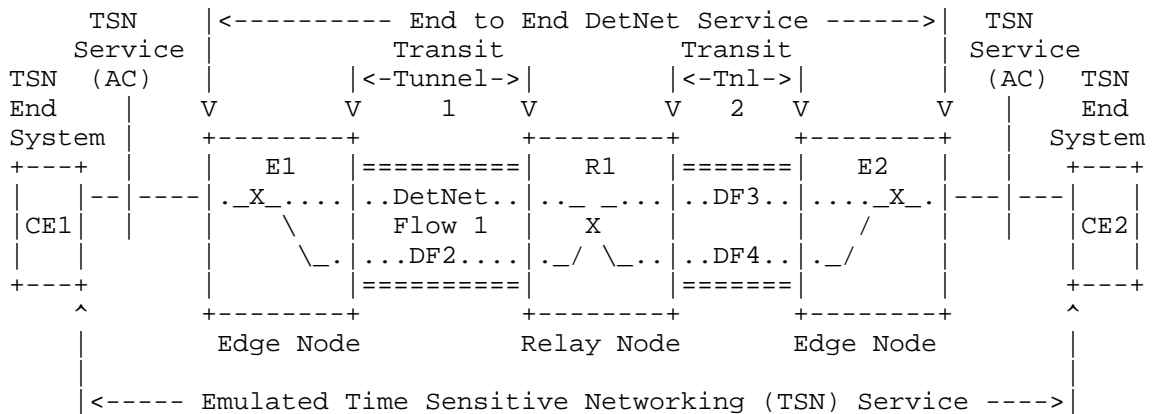


Figure 1: IEEE 802.1TSN over DetNet

Figure 2 illustrates how end to end MPLS-based DetNet service can be provided. In this case, the end systems are able to send and receive DetNet flows. For example, an end system sends data encapsulated in MPLS. Like earlier the 'X' in the end systems, edge and relay nodes represents potential DetNet flow packet replication and elimination points. Here the relay nodes may change the underlying transport, for example tunneling IP over MPLS, or simply interconnect network segments.

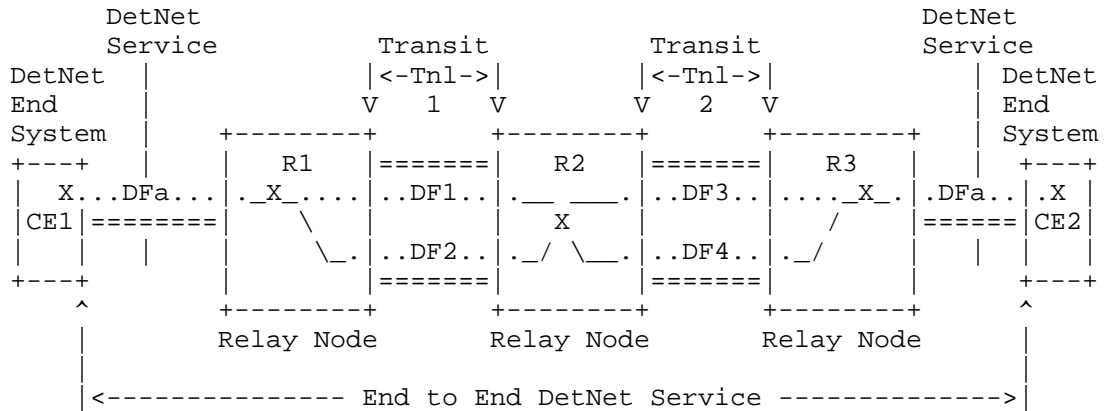


Figure 2: MPLS-Based Native DetNet

Figure 3 illustrates how end to end IP-based DetNet service can be provided. In this case, the end systems are able to send and receive DetNet flows. [Editor's note: TBD]

NOTE: This figures is TBD

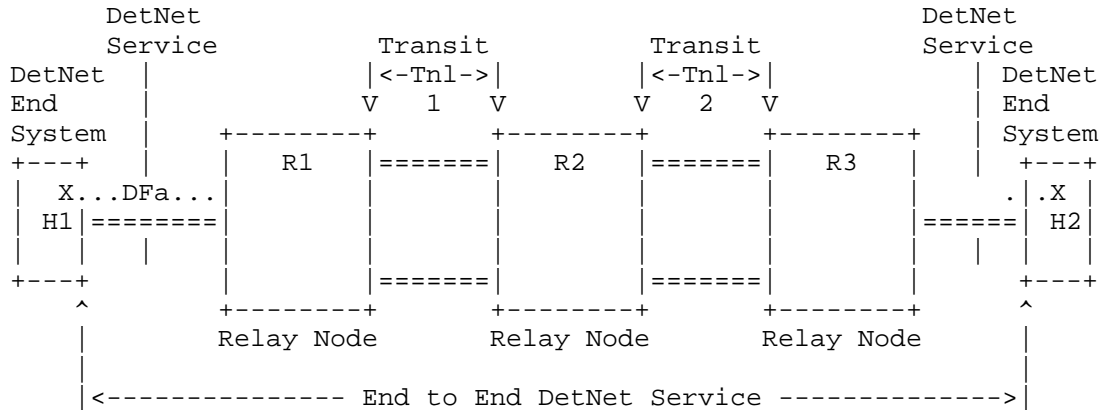


Figure 3: IP-Based Native DetNet

#### 4.1. DetNet data plane encapsulation requirements

Two major groups of scenarios can be distinguished which require flow identification during transport:

##### 1. DetNet function related scenarios:

- \* Congestion protection and latency control: usage of allocated resources (queuing, policing, shaping).
- \* Explicit routes: select/apply the flow specific path.
- \* Service protection: recognize DetNet compound and member flows for replication and elimination.

Comment #12 I am not sure whether the correct architectural construct is flow or flow group. Flow suggests that sharing/aggregation is not allowed but whether this is allowed or not is an application specific issue.

Discussion: Agree that a flow group would be a better characterization.

Comment #13 I think that there needs to be some clarification as to whether FG is understood by the DN system exclusively or whether there is an expectation that it is understood by the underlay.



Discussion: Agree that more detail is needed here. DetNet aware nodes need to understand flow groups. Underlay needs to be aware of flow groups at the resource allocation level.

2. OAM function related scenarios:

- \* troubleshooting (e.g., identify misbehaving flows, etc.)
- \* recognize flow(s) for analytics (e.g., increase counters, etc.)
- \* correlate events with flows (e.g., volume above threshold, etc.)
- \* etc.

Each DetNet node (edge, relay and transit) use an internal/implementation specific local-ID of the DetNet-(compound)-flow in order to accomplish its role during transport. Recognizing the DetNet flow is more relaxed for edge and relay nodes, as they are fully aware of both the DetNet service and transport layers. The primary DetNet role of intermediate transport nodes is limited to ensuring congestion protection and latency control for the above listed DetNet functions.

The DetNet data plane allows for the aggregation of DetNet flows, e.g., via MPLS hierarchical LSPs, to improved scaling. When DetNet flows are aggregated, transit nodes may have limited ability to provide service on per-flow DetNet identifiers. Therefore, identifying each individual DetNet flow on a transit node may not be achieved in some network scenarios, but DetNet service can still be assured in these scenarios through resource allocation and control.

Comment #14 You could introduce the concept of a flow group identified into the packet. You may also include a flow id at a lower layer.

Discussion: Agree on the identification properties. Adding a specific id into actual on-wire formats is not necessarily needed.

On each DetNet node dealing with DetNet flows, an internal local-ID is assumed to determine what local operation a packet goes through. Therefore, local-IDs has to be unique on each edge and relay nodes. Local-ID is unambiguously bound to the DetNet flow.

#### 4.2. Packet replication and elimination considerations

DetNet service layer introduces packet replication and elimination functionality (PREF) for use in DetNet edge and relay node and end system packet processing. PREF MAY be enabled in a DetNet node and the required processing is only applied to packets with a positive flow identification at the DetNet service layer. PREF utilizes a sequence number carried within a DetNet flow packets.

At a DetNet node level the output of the PREF elimination function is always a single packet. The output of the PREF replication function at a DetNet node level is always one or more packets (i.e., 1:M replication). The replicated packets MUST share the same d-CW i.e., the sequence number is the same for each member flow of the compound flow. The location and mechanism on the packet processing pipeline used for replication is implementation specific.

The complex part of the DetNet PREF processing is tracking the history of received packets for multiple DetNet member flows. These ingress DetNet member flows (to a node) MUST have the same local-ID if they belong to the same DetNet (compound) flow and share the same sequence number counter and the history information. The location of the packet elimination on the packet processing pipeline is implementation specific.

#### 4.3. Packet reordering considerations

DetNet service layer introduces also packet reordering functionality for use in DetNet edge and relay node and end system packet processing. The reordering functionality MAY be enabled in a DetNet node. The reordering functionality relies on a presence of sequence numbers in a DetNet (compound) flows. The reordering processing is only applied to packets with a positive flow identification at the DetNet service layer.

### 5. DetNet encapsulation

#### 5.1. End-system specific considerations

Data-flows requiring DetNet service are generated and terminated on end-systems. Encapsulation depends on application and its preferences. In a DetNet (or even a TSN) domain the DN (TSN) functions use at most two flow parameters, namely Flow-ID and Seq.Number. However, an application may exchange further flow related parameters (e.g., time-stamp), which are not considered by DN functions.

Two types of end-systems are distinguished:

- o L3 (IP) end-system: application over L3
- o L2 (Ethernet) end-system: application directly over L2

In case of Ethernet end-systems the application data is encapsulated directly in L2. From the DN domain perspective no upper layer protocols are visible. The Data-flow uses only Ethernet tag(s) and further flow specific parameters (if needed) are hidden inside the PDU.

The IP end-system scenario is different. Data-flows are encapsulated directly in L3 (i.e., IP) and the application may use further upper layer protocols (e.g., RTP). Many valid combinations exist, and it may be application specific how the IP header fields are used. Also, usage of further upper layer protocols depends on application requirements (e.g., time-stamp). Some examples for encoding of Flow-ID or Seq.Number attributes: IP address, IPv6-Flow-label, L4 ports, RTP-header, etc.

As a general rule, DetNet domains MUST be capable to forward any Data-flows and the DetNet domain MUST NOT intend to mandate end-system encapsulation format.

Furthermore, no application-level-proxy function is envisioned inside the DetNet domain, so end-systems peer with end-systems using the same application encapsulation format (see figure below):

- o L3 end-systems peer with L3 end-systems and
- o L2 end-systems peer with L2 end-systems

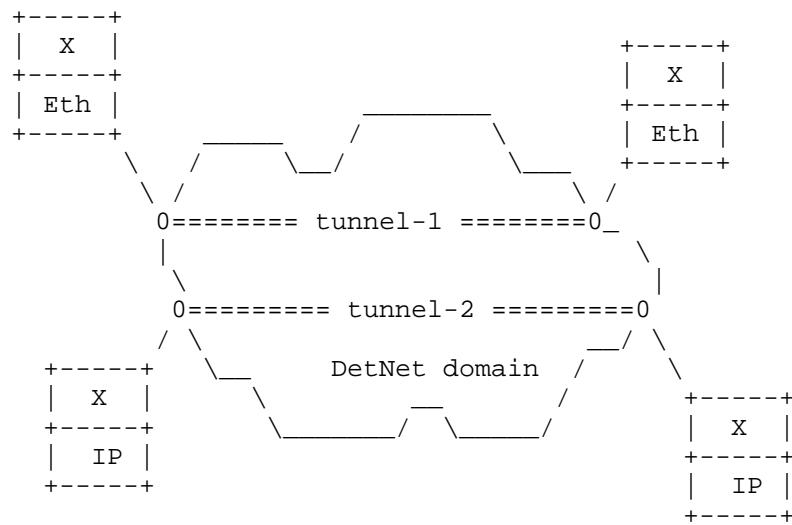


Figure 4: End-systems and the DetNet domain

## 5.2. DetNet domain specific considerations

From connection type perspective three scenarios are distinguished:

1. Directly attached: end-system is directly connected to an edge node
2. Indirectly attached: end-system is behind a (L2-TSN / L3-DetNet) sub-net
3. DN integrated: end-system is part of the DetNet domain

L3 end-systems may use any of these connection types, however L2 end-systems may use only the first two (directly or indirectly attached). DetNet domain MUST allow communication between any end-systems of the same type (L2-L2, L3-L3), independent of their connection type and DetNet capability. However directly attached and indirectly attached end-systems have no knowledge about the DetNet domain and its encapsulation format at all. See the figure below for L3 end-system scenarios.

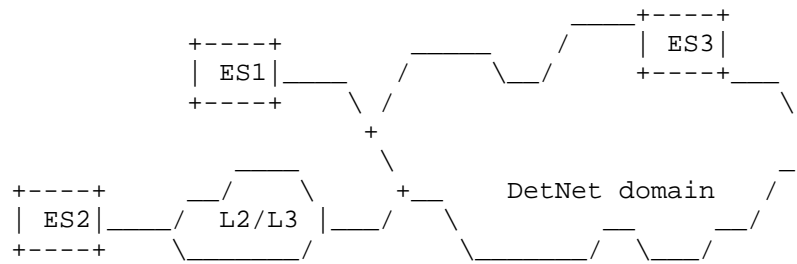
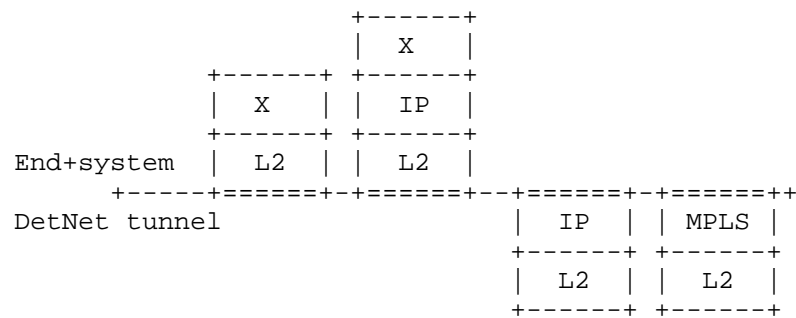


Figure 5: Connection types of L3 end-systems

#### 5.2.1. DetNet Bridging Service

The simplest DetNet service is to provide bridging (i.e., tunneling for L2), where the connected hosts are in the same broadcast (BC) domain. Forwarding over the DetNet domain is based on L2 (MAC) addresses (i.e. dst-MAC), so L2 headers MUST be kept. For both IP and MPLS PSN a DetNet specific tunnel encapsulation MUST be introduced.



## Examples

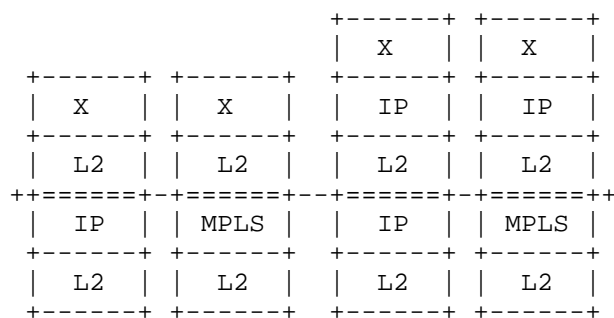


Figure 6: Encapsulation format for DetNet Bridging

As shown on the figure both L2 and L3 end-systems can be served by such a DetNet Bridging service.

### 5.2.2. DetNet Routing Service

DetNet Routing service provides routing, therefore available only for L3 hosts that are in different BC domains. Forwarding over the DetNet domain is based on L3 (IP) addresses (i.e. dst-IP).

#### 5.2.2.1. MPLS PSN

In case of an MPLS PSN at the ingress/egress (i.e., PE nodes of DetNet domain) the IP packets are encapsulated in MPLS. The data-flow IP header MUST be preserved as-is.

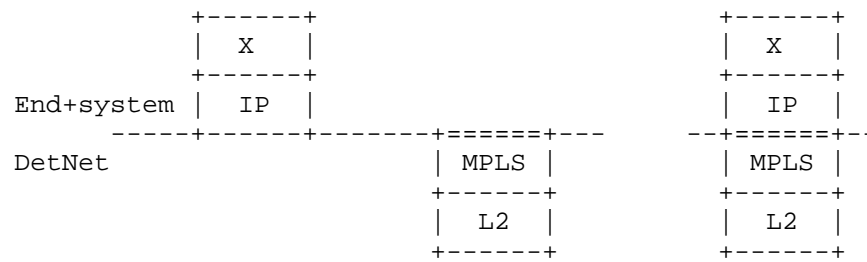


Figure 7: Encapsulation format for DetNet Routing in MPLS PSN for L3 end-systems

#### 5.2.2.2. IP PSN

In case of an IP PSN the same tunneling concept can be used as for an MPLS PSN, but the tunnel is constructed by a new IP header (and possible upper layer fields). The data-flow IP header **MUST** be preserved as-is.

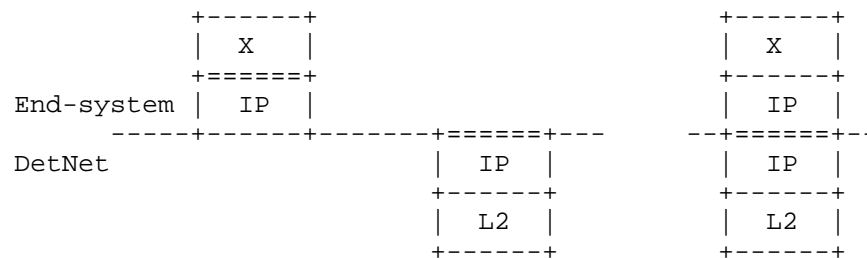


Figure 8: Encapsulation format for DetNet Routing in IP PSN for L3 end-systems

DetNet IP header contains the IP addresses of the ingress/egress PE nodes of DetNet domain. The End-system IP header contains the IP addresses of the end-systems.

Note: In case of IP PSN one may consider avoiding the additional IP encapsulation, however there are many issues with such an approach. First, the DetNet nodes **MUST** be able to extract from the IP header (and maybe upper layers) the attributes required by DetNet functions (i.e. Flow-ID, Seq.Number). The challenge is that encoding of those attributes may be application specific, so DetNet nodes **MUST** be prepared to handle all application specific formats. Second, adding further fields (e.g., explicit path information) to an existing IP header may be impossible (e.g., due to security/encryption).

Furthermore, DetNet domain IP-header format may collide with IP-header format used by the source of a flow. Implementing such an approach requires that source encapsulation is in-line with DetNet domain encapsulation format, however we do not intend to mandate end-systems' encapsulation format (see former text: As a general rule, DetNet domains MUST be capable to forward any Data-flows and the DetNet domain MUST NOT intend to mandate end-system encapsulation format).

Another approach with IP PSN can be based on MPLS over IP [RFC4023] and/or MPLS over UDP/IP [RFC7510]. In this case the encapsulations over the PSNs were the same i.e., basically the DetNet MPLS-based data plane encapsulation as described in Section 6.2 for both IP and MPLS PSNs.

[Editor's note: this approach was actually proposed earlier in draft-dt-detnet-dp-sol-00 in a PseudoWire context for IP PSN]

#### 5.2.2.3. Simplified IP Service

In this case there is no "tunneling" below the DetNet Service, but the DetNet Service flows are mapped to each link / sub net using its technology specific methods. The DetNet IP header contains the IP address destination DetNet end system. The data-flow IP header MUST be preserved as-is.

This solution provides end to end DetNet service consisting of congestion protection and latency control and the rouse allocation (queuing, policing, shaping) done using the underlying link / sub net specific mechanisms. Compared to previously described DetNet routing services, the service protections (packet replication and packet emilination functions) and not provided end to end, but per underlying layer-2 link / sub net.

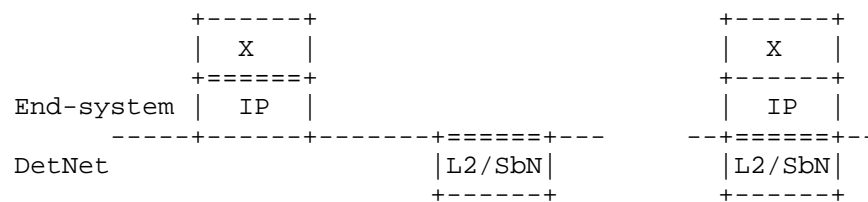


Figure 9: Encapsulation of DetNet Routing in simplified IP service L3 end-systems

Note: the DetNet Service Flow MUST be mapped to the link / sub net specific resources using an underlying system specific means. This



implies each DetNet aware node on path MUST look into the transported DetNet Service Flow packet and utilize e.g., a five tuple to find out the required mapping in a node. As noted earlier, the Service Protection is done within each link / sub net independently using the domain specific mechanisms (due the lack of a unified end to end sequencing information that would be available for intermediate nodes). If end to end service protection is desired that can be implemented, for example, by the DetNet end systems using Layer-4 transport protocols or application protocols. However, these are out of scope of this document.

[Editor's note: the service protection to be clarified further.]

### 5.3. DetNet Inter-Working Function (DN-IWF)

#### 5.3.1. Networks with multiple technology segments

There are network scenarios, where the DetNet domain contains multiple technology segments (IP, MPLS) and all those segments are under the same administrative control (see Figure 10). Furthermore, DetNet nodes may be interconnected via TSN segments.

An important aspect of DetNet network design is placement of DetNet functions across the domain. Designs based on segment-by-segment optimization can provide only suboptimal solutions. In order to achieve global optimum Inter-Working Functions (DN-IWF) can be placed at segment border nodes, which stitch together DetNet flows across connected segments.

DN-IWF may ensure that flow attributes are correlated across segment borders. For example, there are two DetNet functions which require Seq.Numbers: (1) Elimination: removes duplications from flows and (2) IOD: ensures in-order-delivery of packet in a flow. Stitching flows together and correlating attributes means for example that replication of packets can happen in one segment and elimination of duplicates in a different one.

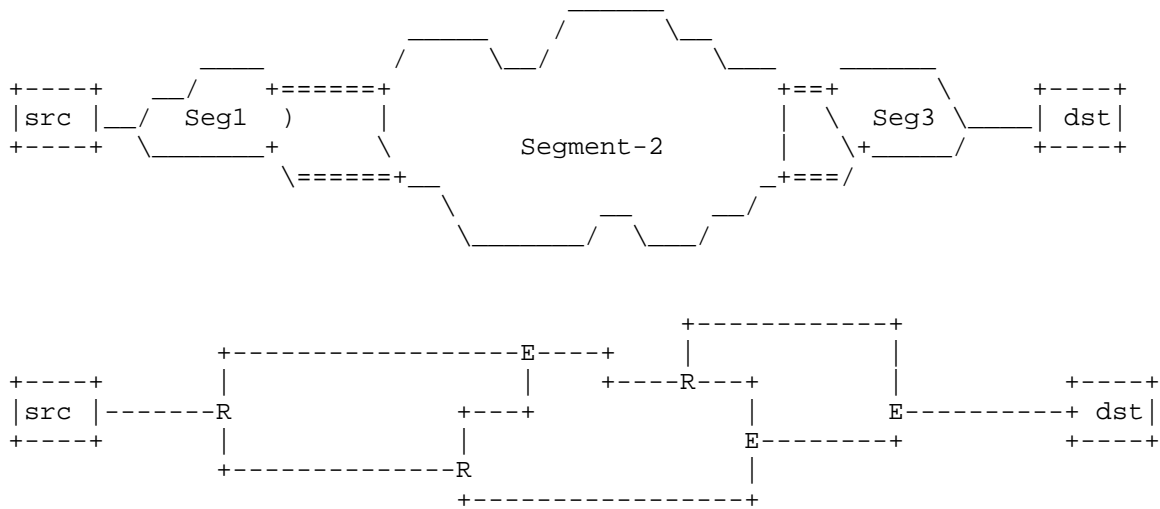


Figure 10: Optimal replication and elimination placement across technology segments example

### 5.3.2. DN-IWF related considerations

The ultimate goal of DN-IWF is to (1) match and (2) translate segment specific flow attributes. The DN-IWF ensures that segment specific attributes comprise per domain unique attributes for the whole DetNet domain. This characteristic can ensure that DetNet functions can be based on per domain attributes and not per segment attributes.

The two DetNet specific attributes have the following characteristics:

- o Flow-ID: it is same in all packets of a flow
- o Seq.Number: it is different packet-by-packet

For the Flow-ID the DN-IWF can implement a static mapping. The situation is more complicated for Seq.Number as it is different packet-by-packet, so it may need more sophisticated translation unless its format is exactly the same in the two technology segments. In this latter case the DN-IWF can simply copy the Seq.Number field between the tunneling encapsulation of the two technology segments.

In case of three technology segments (IP, MPLS and TSN) three DN-IWF functions can be specified. In the rest of this section the focus is on the (1) IP - MPLS network scenario. Note: the use-cases are out-

of-scope for (2) TSN - IP, (3) TSN - MPLS. Note2: incompatible format of Seq.Number with TSN.

Simplest implementation of DN-IWF is provided if the flow attributes have the same format. Such a common denominator of the tunnel encapsulation format is the pseudowire encapsulation over both IP and MPLS.

Placeholder

Figure 11: FIGURE Placeholder PW over X

## 6. MPLS-based DetNet data plane solution

### 6.1. DetNet specific packet fields

The DetNet data plane encapsulation MUST include two DetNet specific information elements in each packet of a DetNet flow: (1) a flow identification and (2) a sequence number.

The DetNet data plane encapsulation may consists further elements used for overlay tunneling, to distinguish between DetNet member flows of the same DetNet compound flow or to support OAM functions.

### 6.2. Data plane encapsulation

Figure 12 illustrates a DetNet data plane MPLS encapsulation. The MPLS-based encapsulation of the DetNet flows is a good fit for the Layer-2 interconnect deployment cases (see Figure 1). Furthermore, end to end DetNet service i.e., native DetNet deployment (see Figure 2) is also possible if DetNet end systems are capable of initiating and termination MPLS encapsulated packets. Transport of IP encapsulated DetNet flows, see Section 7, over MPLS-based DetNet data plane is also possible. Interworking between PW- and IPv6-based encapsulations is discussed further in Section 8.6.

The MPLS-based DetNet data plane encapsulation consists of:

- o DetNet control word (d-CW) containing sequencing information for packet replication and duplicate elimination purposes. There MUST a separate sequence number space for each DetNet flow.
- o DetNet Label that identifies a DetNet flow within a DetNet Edge or a Relay node. The DetNet label MUST be at the bottom of the label stack.

- o An optional DetNet service label (S-Label) that represents DetNet Service LSP used between DetNet Edge and/or Relay nodes. One possible use of an S-Label is to identify DetNet member flows used to provide protection to a DetNet compound flow, perhaps even when both LSPs appear on the same link for some reason.

One or more MPLS transport LSP label(s) (T-label) which may be a hop-by-hop label used between LSR and MUST appear higher in the label stack than S-labels. A top of stack T-label may be PHPed before arriving at a DetNet node. In general T-labels should be considered to be part of the underlying transport network rather than the actual DetNet data plane encapsulation.

DetNet MPLS-based encapsulation

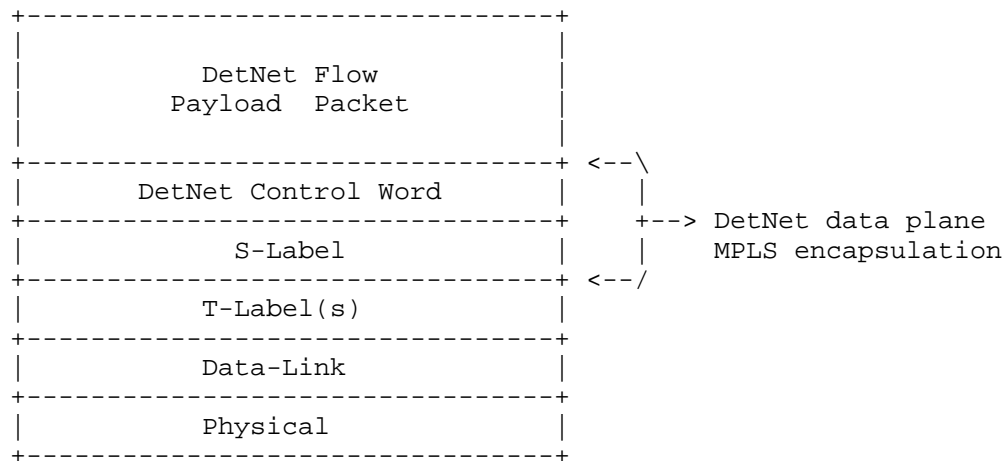


Figure 12: Encapsulation of a DetNet flow in an MPLS(-TP) PSN

### 6.3. DetNet control word

A DetNet control word (d-CW) conforms to the Generic PW MPLS Control Word (PWMCW) defined in [RFC4385] and is illustrated in Figure 13. The upper nibble of the d-CW MUST be set to zero (0). The effective sequence number bit length is between 0 and 28 bits, and configured either by a control plane or manually for each DetNet flow. The sequence number is aligned to the right (least significant bits) and unused bits MUST be set to zero (0). Each DetNet flow MUST have its own sequence number counter. The sequence number is incremented by one for each new packet.

The d-CW MUST always be present in a packet. In a case the sequence number is not used (e.g., for DetNet-t-flows) the control plane or the manual configuration has to define zero (0) bit length sequence number and the value of the sequence number MUST be set to zero (0).

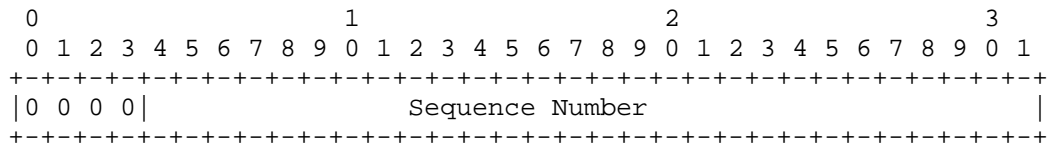


Figure 13: DetNet Control Word

#### 6.4. Flow identification

DetNet flow identification at a DetNet service layer is realized by an S-label. It maps a Detnet flow to a specific d-CW in a DetNet node. The S-label used for flow identification MUST be bottom label of the label stack for a DetNet-s- or DetNet-st-flow and MUST precede the d-CW.

An S-label for a single DetNet flow does not need to be unique DetNet domain wide. As long as two or more different DetNet flows do not erroneously map to a same d-CW in a DetNet node the labels may vary.

#### 6.5. Service layer considerations

[Editor's note: quite a bit of unfinished and old text in the following sections.]

The edge and relay node internal procedures of the PREF are implementation specific. The order of a packet elimination or replication is out of scope in this specification. However, care should be taken that the replication function does not actually loopback packets as "replicas". Looped back packets include artificial delay when the node that originally initiated the packet receives it again. Also, looped back packets may make the network condition to look healthier than it actually is (in some cases link failures are not reflected properly because looped back packets make the situation appear better than it actually is).

Comment #29: SB> There needs to be some text about preventing a node ever receiving its own replicated packets. Indeed that would suggest that the flow id should be changed and replication should only take place on configured flow IDs. I have a feeling that this would all be a lot safer if replication only happened at ingress and we managed the diversity of the paths.

Discussion: Agree on hardening the loopback text considerations.

#### 6.5.1.1. Edge node processing

TBD.

[Editor's note: Since we are not defining the inner workings and implementation of the DetNet Edge node - rather only what goes in and what comes out, and of course the on-wire details, then the figures shown in the coming section would not need to detail the inner architecture of a DetNet Node.]

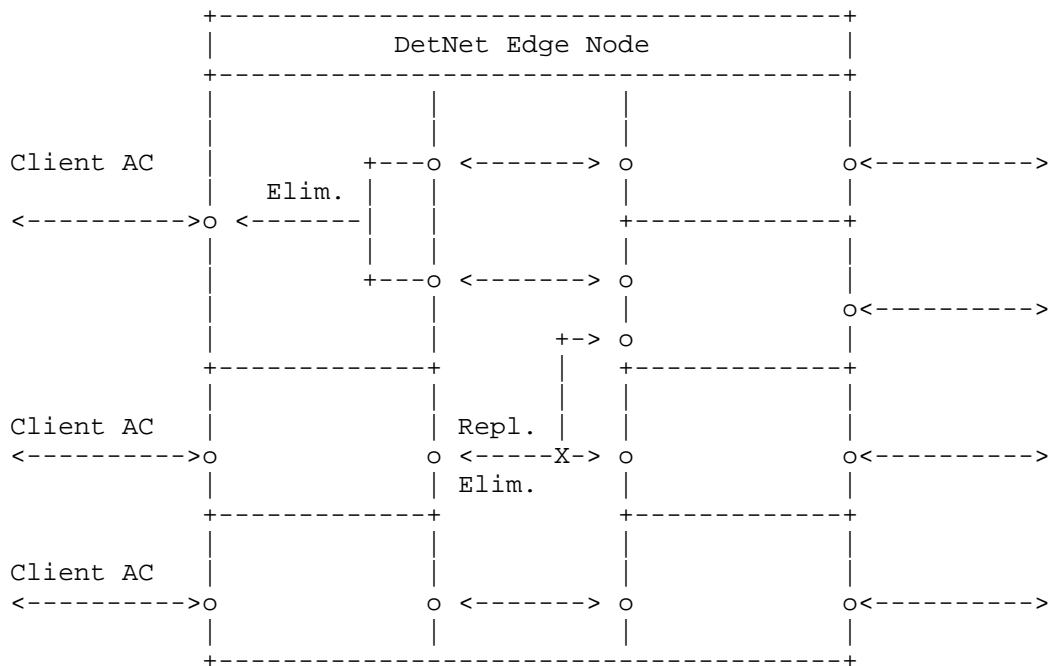


Figure 14: DetNet Edge Node processing

An edge node participates to the packet replication and duplication elimination. Required processing is done within an extended forwarder function. In the case the native service processing (NSP) is IEEE 802.1CB [IEEE8021CB] capable, the packet replication and duplicate elimination MAY entirely be done in the NSP and bypassing the DetNet flow encapsulation and logic entirely, and thus is able to operate over unmodified implementation and deployment. The NSP approach works only between edge nodes and cannot make use of relay nodes (see Section 6.5.2).

Comment #31 SB> This would be a fine way to operate the PW system - edge to edge.

Discussion: When it comes to use of NSPs, agree. Also for "island interconnect" this is a fine. However, when there is a need to do PREF in a middle, plain edge to edge is not enough.

The DetNet-aware extended forwarder selects the egress DetNet member flow based on the DetNet forwarding rules. In both "normal AC" and "Packet AC" cases there may be no DetNet encapsulation header available yet as it is the case with relay nodes (see Section 6.5.2). It is the responsibility of the extended forwarder within the edge node to push the DetNet specific encapsulation (if not already present) to the packet before forwarding it to the appropriate egress DetNet member flow instance(s).

Comment #32 SB> I am not convinced of the wisdom of having a mid-point node convert a flow into a DN flow, which is what you are implying here. This seems like an ingress function.

Discussion: OK. The text here has issues and seems to mix relay and edge.

The extended forwarder MAY copy the sequencing information from the native DetNet packet into the DetNet sequence number field and vice versa. If there is no existing sequencing information available in the native packet or the forwarder chose not to copy it from the native packet, then the extended forwarder MUST maintain a sequence number counter for each DetNet flow (indexed by the DetNet flow identification).

#### 6.5.2. Relay node processing

A DetNet Relay node participates to the packet replication and duplication elimination. This processing is done within an extended forwarder function. Whether an ingress DetNet member flow receives DetNet specific processing depends on how the forwarding is programmed. For some DetNet member flows the relay node can act as a normal relay node and for some apply the DetNet specific processing (i.e., PREF).

Comment #34 SB> Again relay node is not a normal term, so am not sure what it does in the absence of a PREF function.

Discussion: Relay node was a DetNet aware S-PE originally, which is not explicitly stated here anymore, thus slightly confusing text here. The text here needs to clarify the roles of PREF and switching functions. A DetNet relay is described in the

architecture document. However, there is definitely room for terminology and text improvements.

It is also possible to treat the relay node as a transit node, see Section 8.3. Again, this is entirely up to how the forwarding has been programmed.

The DetNet-aware forwarder selects the egress DetNet member flow segment based on the flow identification. The mapping of ingress DetNet member flow segment to egress DetNet member flow segment may be statically or dynamically configured. Additionally the DetNet-aware forwarder does duplicate frame elimination based on the flow identification and the sequence number combination. The packet replication is also done within the DetNet-aware forwarder. During elimination and the replication process the sequence number of the DetNet member flow MUST be preserved and copied to the egress DetNet member flow.

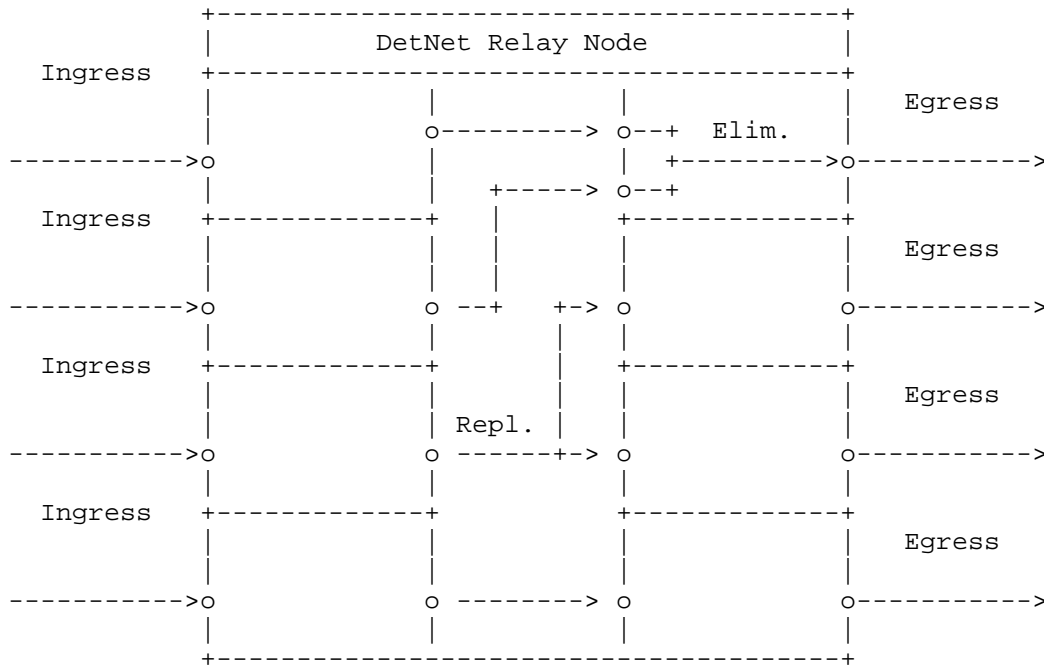


Figure 15: DetNet Relay Node processing

Comment #35 SB> Somewhere in the dp document there needs to be a note of the requirement for interfaces to do fast exchange of counter state, and a note to those planning the network and



designing the control plane that they need to provide support for this.

Discussion: We kind of agree but also think the above exchange or synchronization of counter states is not in our scope to solve.

#### 6.5.3. End system processing

TBD.

#### 6.6. Transport node considerations

##### 6.6.1. Congestion protection

TBD.

##### 6.6.2. Explicit routes

TBD.

#### 7. Simplified IP based DetNet data plane solution

[Editor's note: describe the 6 tuple way of doing DetNet service flows. Also stress that PREF is per network segment as described in Section 5.3.1]

Section 5.2.2.3 illustrated the case for DetNet simplified IP data plane solution.

#### 8. Other DetNet data plane considerations

##### 8.1. Class of Service

Class and quality of service, i.e., CoS and QoS, are terms that are often used interchangeably and confused. In the context of DetNet, CoS is used to refer to mechanisms that provide traffic forwarding treatment based on aggregate group basis and QoS is used to refer to mechanisms that provide traffic forwarding treatment based on a specific DetNet flow basis. Examples of existing network level CoS mechanisms include DiffServ which is enabled by IP header differentiated services code point (DSCP) field [RFC2474] and MPLS label traffic class field [RFC5462], and at Layer-2, by IEEE 802.1p priority code point (PCP).

CoS for DetNet flows carried in PWs and MPLS is provided using the existing MPLS Differentiated Services (DiffServ) architecture [RFC3270]. Both E-LSP and L-LSP MPLS DiffServ modes MAY be used to support DetNet flows. The Traffic Class field (formerly the EXP

field) of an MPLS label follows the definition of [RFC5462] and [RFC3270]. The Uniform, Pipe, and Short Pipe DiffServ tunneling and TTL processing models are described in [RFC3270] and [RFC3443] and MAY be used for MPLS LSPs supporting DetNet flows. MPLS ECN MAY also be used as defined in ECN [RFC5129] and updated by [RFC5462].

CoS for DetNet flows carried in IPv6 is provided using the standard differentiated services code point (DSCP) field [RFC2474] and related mechanisms. The 2-bit explicit congestion notification (ECN) [RFC3168] field MAY also be used.

One additional consideration for DetNet nodes which support CoS services is that they MUST ensure that the CoS service classes do not impact the congestion protection and latency control mechanisms used to provide DetNet QoS. This requirement is similar to requirement for MPLS LSRs to that CoS LSPs do not impact the resources allocated to TE LSPs via [RFC3473].

## 8.2. Quality of Service

Quality of Service (QoS) mechanisms for flow specific traffic treatment typically includes a guarantee/agreement for the service, and allocation of resources to support the service. Example QoS mechanisms include discrete resource allocation, admission control, flow identification and isolation, and sometimes path control, traffic protection, shaping, policing and remarking. Example protocols that support QoS control include Resource ReSerVation Protocol (RSVP) [RFC2205] (RSVP) and RSVP-TE [RFC3209] and [RFC3473]. The existing MPLS mechanisms defined to support CoS [RFC3270] can also be used to reserve resources for specific traffic classes.

In addition to explicit routes, and packet replication and elimination, described in Section 6 above, DetNet provides zero congestion loss and bounded latency and jitter. As described in [I-D.ietf-detnet-architecture], there are different mechanisms that maybe used separately or in combination to deliver a zero congestion loss service. These mechanisms are provided by the either the MPLS or IP layers, and may be combined with the mechanisms defined by the underlying network layer such as 802.1TSN.

A baseline set of QoS capabilities for DetNet flows carried in PWs and MPLS can provided by MPLS with Traffic Engineering (MPLS-TE) [RFC3209] and [RFC3473]. TE LSPs can also support explicit routes (path pinning). Current service definitions for packet TE LSPs can be found in "Specification of the Controlled Load Quality of Service", [RFC2211], "Specification of Guaranteed Quality of Service", [RFC2212], and "Ethernet Traffic Parameters", [RFC6003]. Additional service definitions are expected in future documents to

support the full range of DetNet services. In all cases, the existing label-based marking mechanisms defined for TE-LSPs and even E-LSPs are used to support the identification of flows requiring DetNet QoS.

QoS for DetNet service flows carried in IP MUST be provided locally by the DetNet-aware hosts and routers supporting DetNet flows. Such support will leverage the underlying network layer such as 802.1TSN. The traffic control mechanisms used to deliver QoS for IP encapsulated DetNet flows are expected to be defined in a future document. From an encapsulation perspective, and as defined in Section 7, the combination of the "6 tuple" i.e., the typical 5 tuple enhanced with the DSCP code, uniquely identifies a DetNet service flow.

Packets that are marked with a DetNet Class of Service value, but that have not been the subject of a completed reservation, can disrupt the QoS offered to properly reserved DetNet flows by using resources allocated to the reserved flows. Therefore, the network nodes of a DetNet network MUST:

- o Defend the DetNet QoS by discarding or remarking (to a non-DetNet CoS) packets received that are not the subject of a completed reservation.
- o Not use a DetNet reserved resource, e.g. a queue or shaper reserved for DetNet flows, for any packet that does not carry a DetNet Class of Service marker.

### 8.3. Cross-DetNet flow resource aggregation

The ability to aggregate individual flows, and their associated resource control, into a larger aggregate is an important technique for improving scaling of control in the data, management and control planes. This document identifies the traffic identification related aspects of aggregation of DetNet flows. The resource control and management aspects of aggregation (including the queuing/shaping/policing implications) will be covered in other documents. The data plane implications of aggregation are independent for PW/MPLS and IP encapsulated DetNet flows.

DetNet flows transported via MPLS can leverage MPLS-TE's existing support for hierarchical LSPs (H-LSPs), see [RFC4206]. H-LSPs are typically used to aggregate control and resources, they may also be used to provide OAM or protection for the aggregated LSPs. Arbitrary levels of aggregation naturally fall out of the definition for hierarchy and the MPLS label stack [RFC3032]. DetNet nodes which support aggregation (LSP hierarchy) map one or more LSPs (labels)

into and from an H-LSP. Both carried LSPs and H-LSPs may or may not use the TC field, i.e., L-LSPs or E-LSPs. Such nodes will need to ensure that traffic from aggregated LSPs are placed (shaped/policed/enqueued) onto the H-LSPs in a fashion that ensures the required DetNet service is preserved.

DetNet flows transported via IP have more limited aggregation options, due to the available traffic flow identification fields of the IP solution. One available approach is to manage the resources associated with a DSCP identified traffic class and to map (remark) individually controlled DetNet flows onto that traffic class. This approach also requires that nodes support aggregation ensure that traffic from aggregated LSPs are placed (shaped/policed/enqueued) in a fashion that ensures the required DetNet service is preserved.

Comment #38 SB> I am sure we can do better than this with SR, or the use of routing techniques that map certain addresses to certain paths.

Discussion: --

In both the MPLS and IP cases, additional details of the traffic control capabilities needed at a DetNet-aware node may be covered in the new service descriptions mentioned above or in separate future documents. Management and control plane mechanisms will also need to ensure that the service required on the aggregate flow (H-LSP or DSCP) are provided, which may include the discarding or remarking mentioned in the previous sections.

#### 8.4. Bidirectional traffic

Some DetNet applications generate bidirectional traffic. Using MPLS definitions [RFC5654] there are associated bidirectional flows, and co-routed bidirectional flows. MPLS defines a point-to-point associated bidirectional LSP as consisting of two unidirectional point-to-point LSPs, one from A to B and the other from B to A, which are regarded as providing a single logical bidirectional transport path. This would be analogous of standard IP routing, or PWs running over two reciprocal unidirection LSPs. MPLS defines a point-to-point co-routed bidirectional LSP as an associated bidirectional LSP which satisfies the additional constraint that its two unidirectional component LSPs follow the same path (in terms of both nodes and links) in both directions. An important property of co-routed bidirectional LSPs is that their unidirectional component LSPs share fate. In both types of bidirectional LSPs, resource allocations may differ in each direction. The concepts of associated bidirectional flows and co-routed bidirectional flows can be applied to DetNet flows as well whether IPv6 or MPLS is used.

While the IPv6 and MPLS data planes must support bidirectional DetNet flows, there are no special bidirectional features with respect to the data plane other than need for the two directions take the same paths. Fate sharing and associated vs co-routed bidirectional flows can be managed at the control level. Note, that there is no stated requirement for bidirectional DetNet flows to be supported using the same IPv6 Flow Labels or MPLS Labels in each direction. Control mechanisms will need to support such bidirectional flows for both IPv6 and MPLS, but such mechanisms are out of scope of this document. An example control plane solution for MPLS can be found in [RFC7551].

#### 8.5. Layer 2 addressing and QoS Considerations

The Time-Sensitive Networking (TSN) Task Group of the IEEE 802.1 Working Group have defined (and are defining) a number of amendments to IEEE 802.1Q [IEEE8021Q] that provide zero congestion loss and bounded latency in bridged networks. IEEE 802.1CB [IEEE8021CB] defines packet replication and elimination functions that should prove both compatible with and useful to, DetNet networks.

As is the case for DetNet, a Layer 2 network node such as a bridge may need to identify the specific DetNet flow to which a packet belongs in order to provide the TSN/DetNet QoS for that packet. It also will likely need a CoS marking, such as the priority field of an IEEE Std 802.1Q VLAN tag, to give the packet proper service.

Although the flow identification methods described in IEEE 802.1CB [IEEE8021CB] are flexible, and in fact, include IP 5-tuple identification methods, the baseline TSN standards assume that every Ethernet frame belonging to a TSN stream (i.e. DetNet flow) carries a multicast destination MAC address that is unique to that flow within the bridged network over which it is carried. Furthermore, IEEE 802.1CB [IEEE8021CB] describes three methods by which a packet sequence number can be encoded in an Ethernet frame.

Ensuring that the proper Ethernet VLAN tag priority and destination MAC address are used on a DetNet/TSN packet may require further clarification of the customary L2/L3 transformations carried out by routers and edge label switches. Edge nodes may also have to move sequence number fields among Layer 2, PW, and IPv6 encapsulations.

#### 8.6. Interworking between MPLS- and IPv6-based encapsulations

[Editor's note: add considerations for interworking between MPLS-based and native IPv6-based DetNet encapsuations.]

### 8.7. IPv4 considerations

[Editor's note: The fact is that there are and will be deployments using IPv4. Neglecting it entirely is not feasible.]

### 9. Time synchronization

Comment #39 SB> This section should point the reader to RFC8169 (residence time in MPLS n/w. We need to consider if we need to introduce the same concept in IP.

Discussion: Agree. For IP we could reference to PTPv2 or v3 over UDP/IP, since it measures residence time among other things.

[Editor's note: describe a bit of issues and deployment considerations related to time-synchronization within DetNet. Refer to DT discussion and the slides that summarize different approaches and rough synchronization performance numbers. Finally, scope time-synchronization solution outside data plane.]

When DetNet is used, there is an underlying assumption that the application(s) require clock synchronization such as the Precision Time Protocol (PTP) [IEEE1588]. The relay nodes may or may not utilize clock synchronization in order to provide zero congestion loss and controlled latency delivery. In either case, there are a few possible approaches of how synchronization protocol packets are forwarded and handled by the network:

- o PTP packets can be sent either as DetNet flows or as high-priority best effort packets. Using DetNet for PTP packets requires careful consideration to prevent unwanted interactions between clock-synchronized network nodes and the packets that synchronize the clocks.
- o PTP packets are sent as a normal DetNet flow through network nodes that are not time-synchronized: in this approach PTP traffic is forwarded as a DetNet flow, and as such it is forwarded in a way that allows a low delay variation. However, since intermediate nodes do not take part in the synchronization protocol, this approach provides a relatively low degree of accuracy.
- o PTP with on-path support: in this approach PTP packets are sent as ordinary or as DetNet flows, and intermediate nodes take part in the protocol as Transparent Clocks or Boundary Clocks [IEEE1588]. The on-path PTP support by intermediate nodes provides a higher degree of accuracy than the previous approach. The actual accuracy depends on whether all intermediate nodes are PTP-capable, or only a subset of them.

- o Time-as-a-service: in this approach accurate time is provided as-a-service to the DetNet source and destination, as well as the intermediate nodes. Since traffic between the source and destination is sent over a provider network, if the provider supports time-as-a-service, then accurate time can be provided to both the source and the destination of DetNet traffic. This approach can potentially provide the highest degree of accuracy.

It is expected that the latter approach will be the most common one, as it provides the highest degree of accuracy, and creates a layer separation between the DetNet data and the synchronization service.

It should be noted that in all four approaches it is not recommended to use replication and elimination for synchronization packets; the replication/elimination approach may in some cases reduce the synchronization accuracy, since the observed path delay will be bivalent.

Comment #40 SB> I am not sure why we should not use PREP. We should explain to the reader.

Discussion: Agree that a this can be opened a bit more in detail. The issue is explained briefly in the last sentence but it could be more clear.

#### 10. Management and control considerations

[Editor's note: This section needs to be different for MPLS and IPv6 solutions. Most solutions are technology dependant,]

While management plane and control planes are traditionally considered separately, from the Data Plane perspective there is no practical difference based on the origin of flow provisioning information. This document therefore does not distinguish between information provided by a control plane protocol, e.g., RSVP-TE [RFC3209] and [RFC3473], or by a network management mechanisms, e.g., RestConf [RFC8040] and YANG [RFC7950].

[Editor's note: This section is a work in progress. discuss here what kind of enhancements are needed for DetNet and specifically for PREF and DetNet zero congest loss and latency control. Need to cover both traffic control (queuing) and connection control (control plane).]

## 10.1. MPLS-based data plane

### 10.1.1. S-Label assignment and distribution

[Editor's note: Outdated and MPLS specific.. and needs more work.]

The DetNet S-Label distribution follows the same mechanisms specified for XYZ . The details of the control plane protocol solution required for the label distribution and the management of the label number space are out of scope of this document.

### 10.1.2. Explicit routes

[Editor's note: Outdated.. and needs more work.]

[TBD: based on MPLS TE and possibly IPv6 SR]

## 10.2. IPv6-based data plane

### 10.2.1. Flow Label assignment and distribution

[Editor's note: Outdated and IPv6 Specific.. and needs more work.]

The IPv6 Flow Label distribution and the label number space are out of scope of this document. However, it should be noted that the combination of the IPv6 source address and the IPv6 Flow Label is assumed to be unique within the DetNet-enabled network. Therefore, as long as each node is able to assign unique Flow Labels for the source address(es) it is using the DetNet-enabled network wide flow identification uniqueness is guaranteed.

### 10.2.2. Explicit routes

[Editor's note: Outdated.. and needs more work.]

[TBD: What we have there for IPv6 and explicit routes]

## 10.3. Packet replication and elimination

[Editor's note: Outdated and at the functional level technology independent.. but needs more work.]

The control plane protocol solution required for managing the PREF processing is outside the scope of this document.



#### 10.4. Congestion protection and latency control

[TBD]

#### 10.5. Flow aggregation control

[TBD]

#### 11. Security considerations

The security considerations of DetNet in general are discussed in [I-D.ietf-detnet-architecture] and [I-D.sdt-detnet-security]. Other security considerations will be added in a future version of this draft.

#### 12. IANA considerations

TBD.

#### 13. Acknowledgements

The author(s) ACK and NACK.

The following people were part of the DetNet Data Plane Solution Design Team:

Jouni Korhonen

Janos Farkas

Norman Finn

Balazs Varga

Loa Andersson

Tal Mizrahi

David Mozes

Yuanlong Jiang

Carlos J. Bernardos

The DetNet chairs serving during the DetNet Data Plane Solution Design Team:

Lou Berger

Pat Thaler

## 14. References

### 14.1. Normative references

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2211] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", RFC 2211, DOI 10.17487/RFC2211, September 1997, <<https://www.rfc-editor.org/info/rfc2211>>.
- [RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, DOI 10.17487/RFC2212, September 1997, <<https://www.rfc-editor.org/info/rfc2212>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <<https://www.rfc-editor.org/info/rfc3270>>.

- [RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <<https://www.rfc-editor.org/info/rfc3443>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <<https://www.rfc-editor.org/info/rfc4023>>.
- [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, DOI 10.17487/RFC4206, October 2005, <<https://www.rfc-editor.org/info/rfc4206>>.
- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<https://www.rfc-editor.org/info/rfc4385>>.
- [RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, DOI 10.17487/RFC5129, January 2008, <<https://www.rfc-editor.org/info/rfc5129>>.
- [RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <<https://www.rfc-editor.org/info/rfc5462>>.
- [RFC6003] Papadimitriou, D., "Ethernet Traffic Parameters", RFC 6003, DOI 10.17487/RFC6003, October 2010, <<https://www.rfc-editor.org/info/rfc6003>>.
- [RFC6073] Martini, L., Metz, C., Nadeau, T., Bocci, M., and M. Aissaoui, "Segmented Pseudowire", RFC 6073, DOI 10.17487/RFC6073, January 2011, <<https://www.rfc-editor.org/info/rfc6073>>.

- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black,  
"Encapsulating MPLS in UDP", RFC 7510,  
DOI 10.17487/RFC7510, April 2015,  
<<https://www.rfc-editor.org/info/rfc7510>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6  
(IPv6) Specification", STD 86, RFC 8200,  
DOI 10.17487/RFC8200, July 2017,  
<<https://www.rfc-editor.org/info/rfc8200>>.

#### 14.2. Informative references

- [I-D.ietf-6man-segment-routing-header]  
Previdi, S., Filsfils, C., Raza, K., Dukes, D., Leddy, J.,  
Field, B., daniel.voyer@bell.ca, d.,  
daniel.bernier@bell.ca, d., Matsushima, S., Leung, I.,  
Linkova, J., Aries, E., Kosugi, T., Vyncke, E., Lebrun,  
D., Steinberg, D., and R. Raszuk, "IPv6 Segment Routing  
Header (SRH)", draft-ietf-6man-segment-routing-header-09  
(work in progress), March 2018.
- [I-D.ietf-detnet-architecture]  
Finn, N., Thubert, P., Varga, B., and J. Farkas,  
"Deterministic Networking Architecture", draft-ietf-  
detnet-architecture-04 (work in progress), October 2017.
- [I-D.ietf-detnet-dp-alt]  
Korhonen, J., Farkas, J., Mirsky, G., Thubert, P.,  
Zhuangyan, Z., and L. Berger, "DetNet Data Plane Protocol  
and Solution Alternatives", draft-ietf-detnet-dp-alt-00  
(work in progress), October 2016.
- [I-D.sdt-detnet-security]  
Mizrahi, T., Grossman, E., Hacker, A., Das, S.,  
"Deterministic Networking (DetNet) Security  
Considerations, draft-sdt-detnet-security, work in  
progress", 2017.
- [IEEE1588]  
IEEE, "IEEE 1588 Standard for a Precision Clock  
Synchronization Protocol for Networked Measurement and  
Control Systems Version 2", 2008.

## [IEEE8021CB]

Finn, N., "Draft Standard for Local and metropolitan area networks - Seamless Redundancy", IEEE P802.1CB /D2.1 P802.1CB, December 2015, <<http://www.ieee802.org/1/files/private/cb-drafts/d2/802-1CB-d2-1.pdf>>.

## [IEEE8021Q]

IEEE 802.1, "Standard for Local and metropolitan area networks--Bridges and Bridged Networks (IEEE Std 802.1Q-2014)", 2014, <<http://standards.ieee.org/about/get/>>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.

[RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <<https://www.rfc-editor.org/info/rfc3985>>.

[RFC5654] Niven-Jenkins, B., Ed., Brungard, D., Ed., Betts, M., Ed., Sprecher, N., and S. Ueno, "Requirements of an MPLS Transport Profile", RFC 5654, DOI 10.17487/RFC5654, September 2009, <<https://www.rfc-editor.org/info/rfc5654>>.

[RFC7551] Zhang, F., Ed., Jing, R., and R. Gandhi, Ed., "RSVP-TE Extensions for Associated Bidirectional Label Switched Paths (LSPs)", RFC 7551, DOI 10.17487/RFC7551, May 2015, <<https://www.rfc-editor.org/info/rfc7551>>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

## Appendix A. Example of DetNet data plane operation

[Editor's note: Add a simplified example of DetNet data plane and how labels etc work in the case of MPLS-based PSN and utilizing PREF. The figure is subject to change depending on the further DT decisions on the label handling..]

Appendix B. Example of pinned paths using IPv6

TBD.

Authors' Addresses

Jouni Korhonen (editor)  
Nordic Semiconductor  
  
Email: jouni.nospam@gmail.com

Loa Andersson  
Huawei  
  
Email: loa@pi.nu

Yuanlong Jiang  
Huawei  
  
Email: jiangyuanlong@huawei.com

Norman Finn  
Huawei  
3101 Rio Way  
Spring Valley, CA 91977  
USA  
  
Email: norman.finn@mail01.huawei.com

Balazs Varga  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary  
  
Email: balazs.a.varga@ericsson.com

Janos Farkas  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary

Email: [janos.farkas@ericsson.com](mailto:janos.farkas@ericsson.com)

Carlos J. Bernardos  
Universidad Carlos III de Madrid  
Av. Universidad, 30  
Leganes, Madrid 28911  
Spain

Phone: +34 91624 6236  
Email: [cjbc@it.uc3m.es](mailto:cjbc@it.uc3m.es)  
URI: <http://www.it.uc3m.es/cjbc/>

Tal Mizrahi  
Marvell  
6 Hamada st.  
Yokneam  
Israel

Email: [talmi@marvell.com](mailto:talmi@marvell.com)

Lou Berger  
LabN Consulting, L.L.C.

Email: [lberger@labn.net](mailto:lberger@labn.net)

DetNet  
Internet-Draft  
Intended status: Standards Track  
Expires: September 6, 2018

J. Farkas  
B. Varga  
Ericsson  
R. Cummings  
National Instruments  
Y. Jiang  
Huawei  
Y. Zha  
Tencent  
March 05, 2018

DetNet Flow Information Model  
draft-ietf-detnet-flow-information-model-01

## Abstract

This document describes flow and service information model for Deterministic Networking (DetNet). The DetNet service is provided either for a Layer 3 or a Layer 2 flow. This document provides DetNet flow and service information model both for Layer 3 and Layer 2 flows in an integrated fashion.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of



publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Goals . . . . .	4
1.2. Non Goals . . . . .	5
2. Conventions Used in This Document . . . . .	5
3. Terminology and Definitions . . . . .	5
4. Naming Conventions . . . . .	5
5. Service model . . . . .	6
5.1. Service overview . . . . .	6
5.2. Service parameters . . . . .	6
5.3. Reference Points . . . . .	7
5.4. Service scenarios . . . . .	8
6. End System and DetNet domain . . . . .	8
7. Flow . . . . .	10
7.1. Identification and Specification of Flows . . . . .	11
7.1.1. DetNet L3 Flow Identification and Specification at UNI . . . . .	11
7.1.2. DetNet L2 Flow Identification and Specification at UNI . . . . .	11
7.1.3. DetNetwork Flow Identification and Specification . .	12
7.2. Traffic Specification . . . . .	12
7.3. Flow Rank . . . . .	14
7.4. Service Rank . . . . .	14
8. Source . . . . .	14
9. Destination . . . . .	15
10. Common Attributes of Source and Destination . . . . .	16
10.1. End System Interfaces . . . . .	16
10.2. Interface Capabilities . . . . .	16
10.3. User to Network Requirements . . . . .	17
11. Ingress . . . . .	18
12. Egress . . . . .	18
13. DetNet Domain . . . . .	18
13.1. DetNet Domain Capabilities . . . . .	18
14. Flow-status . . . . .	19
14.1. Status Info . . . . .	20
14.2. Interface Configuration . . . . .	21
14.3. Failed Interfaces . . . . .	21
15. Service-status . . . . .	21
16. Summary . . . . .	21
17. IANA Considerations . . . . .	22

18. Security Considerations . . . . .	22
19. References . . . . .	22
19.1. Normative References . . . . .	22
19.2. Informative References . . . . .	22
Authors' Addresses . . . . .	23

## 1. Introduction

A Deterministic Networking (DetNet) service provides a capability to carry a unicast or a multicast data flow for an application with constrained requirements on network performance, e.g., low packet loss rate and/or latency. The DetNet service is provided either for a Layer 3 (L3) flow or a Layer 2 (L2) flow by an IP/MPLS network, see, e.g., [I-D.ietf-detnet-dp-alt]. Similarly, Time-Sensitive Networking (TSN) [IEEE8021TSN] can be used for L2 flows in a bridged network. DetNet and TSN have common architecture as expressed in [IETFDetNet] and [I-D.ietf-detnet-architecture]. DetNet service can be leveraged both by L3 and L2 flows, i.e., by DetNet L3 flows and DetNet L2 flows. Therefore, the DetNet flow and service information model provided by this document covers both DetNet L3 flows and DetNet L2 flows in an integrated fashion.

In a given network scenario three information models can be distinguished:

- o Flow models describe characteristics of data flows. These models describe in detail all relevant aspects of a flow that are needed to support the flow properly by the network between the source and the destination(s).
- o Service models describe characteristics of services being provided for data flows over a network. These models can be treated as a network operator independent information model.
- o Configuration models describe in detail the settings required on network nodes to serve a data flow properly.

Service and flow information models are used between the user and the network operator. Configuration information models are used between the management/control plane entity of the network and the network nodes. They are shown in Figure 1.

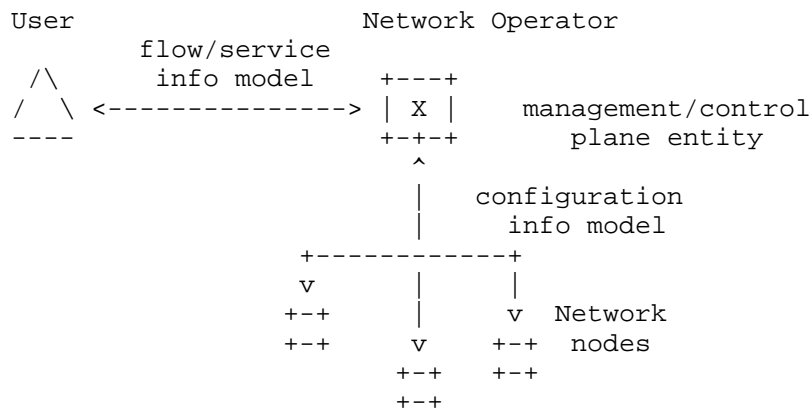


Figure 1: Usage of Information models (flow, service and configuration)

DetNet flow and service information model is based on [I-D.ietf-detnet-architecture] and on the data model specified by [IEEE8021Qcc]. Furthermore, the DetNet flow information model relies on the flow identification possibilities described in [IEEE8021CB], which is used by [IEEE8021Qcc] as well. In addition to TSN data model, [IEEE8021Qcc] also specifies configuration of TSN features (e.g., traffic scheduling specified by [IEEE8021Qbv]). Due to the common architecture and flow model, configuration features can be leveraged in certain deployment scenarios, e.g., when the network that provides the DetNet service includes both L3 and L2 network segments.

Based on the DetNet architecture [I-D.ietf-detnet-architecture] (see Section 4), this document (this revision) only considers the Centralized Network / Distributed User Model out of the models specified by [IEEE8021Qcc]. That is, there is a User-Network Interface (UNI) between an end system and a network. Furthermore, there is a central entity for the control of the network. For instance, the central entity implements a Path Computation Element (PCE) for the calculation and establishment of paths needed for packet replication and elimination, if any.

### 1.1. Goals

As it is expressed in the Charter [IETFDetNet], the DetNet WG collaborates with IEEE 802.1 TSN in order to define a common architecture for both Layer 2 and Layer 3, which is beneficial for various reasons, e.g., in order to simplify implementations. The flow and service information models should be also common along those lines. As the TSN flow information/data model specified by

[IEEE8021Qcc] is mature, the DetNet flow and service information models described in this document are based on [IEEE8021Qcc], which is an amendment to [IEEE8021Q].

This document intends to specify flow and service information models only.

## 1.2. Non Goals

This document (this revision) does not intend to specify either flow data model or DetNet configuration. From these aspects, the goals of this document differ from the goals of [IEEE8021Qcc], which also specifies data model and configuration of certain TSN features.

## 2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The lowercase forms with an initial capital "Must", "Must Not", "Shall", "Shall Not", "Should", "Should Not", "May", and "Optional" in this document are to be interpreted in the sense defined in [RFC2119], but are used where the normative behavior is defined in documents published by SDOs other than the IETF.

## 3. Terminology and Definitions

This document uses the terminology established in Section 2 of the DetNet architecture document [I-D.ietf-detnet-architecture]. The DetNet <=> TSN dictionary of [I-D.ietf-detnet-architecture] is used to perform translation from [IEEE8021Qcc] to this document. Additional terms used in this document:

DetNet L3 Flow: Layer 3 (L3) flow leveraging DetNet service.

DetNet L2 Flow: Layer 2 (L2) flow leveraging DetNet service.

DetNetwork Flow: DetNet data plane specific encapsulated format of a DetNet L2 or L3 flow leveraging DetNet service.

## 4. Naming Conventions

The following naming conventions were used for naming information model components in this document. It is recommended that extensions of the model use the same conventions.

- o Names SHOULD be descriptive.

- o Names MUST start with uppercase letters.
- o Composed names MUST use capital letters for the first letter of each component. All other letters are lowercase, even for acronyms. Exceptions are made for acronyms containing a mixture of lowercase and capital letters, such as IPv6. Examples are SourceMacAddress and DestinationIPv6Address.

## 5. Service model

### 5.1. Service overview

The DetNet service can be defined as a service that provides a capability to carry a unicast or a multicast data flow for an application with constrained requirements on network performance, e.g., low packet loss rate and/or latency.

The simplest DetNet service is to provide bridging over the DN domain (i.e., tunneling for L2), where the connected hosts are in the same broadcast (BC) domain. Forwarding over the DetNet domain is based on L2 (MAC) addresses (i.e. dst-MAC). Somewhat more sophisticated is DetNet Routing service that provides routing, so available only for L3 hosts that are in different BC domains. Forwarding over the DetNet domain is based on L3 (IP) addresses (i.e. dst-IP).

Figure 5. and Figure 8. in [I-D.ietf-detnet-architecture] shows the DetNet service related reference points and main components.

### 5.2. Service parameters

Two forwarding methods are distinguished: (1) Bridging and (2) Routing. The DN service is represented by a DN-PSeudoWire (DN-PW).

Data-flows are received over the UNI. Usually there is a DN service related legacy VPN service. The DN service and the legacy VPN service use a common AC (attachment circuit). Legacy VPN is used by regular traffic of the DetNet end-systems. DN flows are "directed" by a selector to DN-PW(s). (See Figure 8. in [I-D.ietf-detnet-architecture])

Service attributes for DetNet connectivity are:

- o Bandwidth parameter(s),
- o Delay parameter(s),
- o Loss parameter(s),

- o Connectivity type,
- o In order delivery,
- o Service rank.

Time/loss sensitive applications may have somewhat special requirements especially for loss (e.g., no loss in two consecutive communication cycles; very low outage time, etc.).

Two connectivity types are distinguished: point-to-point (p2p) and point-to-multipoint (p2mp). Connectivity type p2mp is created by a transport layer function (e.g., p2mp LSP). (Note: mp2mp connectivity is a superposition of p2mp connections.)

Depending on the application and the end-system capabilities DetNet service may be requested to provide in order delivery.

Service rank provides the rank of a service instance relative to other services in the network. Rank is used by the network in case of network resource limitation scenarios.

### 5.3. Reference Points

From service model design perspective a fundamental question is the location of the service endpoints, i.e., where the service starts and ends.

Note: Further discussion is needed based on data plane encapsulation results what reference points should be defined. Only some possible examples listed here:

- o App-flow endpoint: End system's internal reference point for the native data flow.
- o DetNet-UNI: UNI interface ("U") on a DetNet edge node.
- o DetNet-NNI: NNI interface ("N") between DetNet domains.

[[NOTE: Contributions are welcome whether we should define or distinguish internal reference point(s) for DetNet-aware end-systems as well. ]]

DetNet-UNI and DetNet-NNI are assumed in this document to be packet-based reference points and provide connectivity over the packet network and between domains. A DetNet-UNI adds networking technology specific encapsulation to the data flow in order to transport it over the network.

[[NOTE: Differences between the service over end-systems internal reference points and DetNet-UNI is for further discussions. For example, in-order delivery is expected in end system internal reference points, whereas it is considered optional over the DetNet-UNI. ]]

#### 5.4. Service scenarios

Using the above defined reference points, two major service scenarios can be identified:

- o End-to-End-Service: the service reaches out to final source or destination nodes, so it is an e2e service between application hosting devices (end systems).
- o DetNet-Service: the service connects networking islands, so it is a service between the borders of network domain(s).

[[NOTE: we may consider to define further scenarios based on the result of reference point related discussions. ]]

#### 6. End System and DetNet domain

Deterministic service is required by time/loss sensitive application(s) running on an end system during communication with its peer(s). Such a data exchange has various requirements on delay and/or loss parameters.

The DetNet architecture [I-D.ietf-detnet-architecture] distinguishes two kinds of end systems: Source and Destination. The same distinction is applied for the DetNet flow information model. In addition to the end systems interested in a flow, the status information of the flow is also important. Therefore, the DetNet flow information model relies on three high level groups:

- o Source: an end system capable of sourcing a DetNet flow. The Source information group includes elements that specify the Source for a single flow. This information group is applied from the user to the network.
- o Destination: an end system that is a destination of a DetNet flow. The Destination information group includes elements that specify the Destination for a single flow. This information group is applied from the user to the network.
- o Flow-Status: the status of a DetNet flow. The status information group includes elements that specify the status of the flow in the network. This information group is applied from the network to

the user. This information group informs the user whether or not the flow is ready for use.

From service perspective two kinds of edge nodes can be distinguished: Ingress and Egress. In addition the technology of the DetNet domain and the status of the service are also important. Therefore, the DetNet service information model relies on four high level groups:

- o Ingress: an edge system receiving a DetNet flow from a Source. The Ingress information group includes elements that specify the entry point for a single flow. This information group is applied from the network to the user.
- o Egress: an edge system sending traffic towards a Destination of a DetNet flow. The Egress information group includes elements that specify the egress point for a single flow. This information group is applied from the network to the user.
- o DetNet Domain: an administrative domain providing the DetNet service. The DetNet domain information group includes elements that specify the forwarding capabilities and methods for a single flow. This information group is applied within the network.
- o Service-Status: the status of a DetNet service. The status information group includes elements that specify the status of the service specific state of the network. This information group is applied from the network to the user. This information group informs the user whether or not the service is ready for use.

There are two operations for each flow with respect to a Source or a Destination (and an Ingress or an Egress):

- o Join: Source/Destination request to join the flow.
- o Leave: Source/Destination request to leave the flow.
- o Modify: Source/Destination request to change the flow.

Modify operation can be considered to address cases when a flow is slightly changed, e.g., only MaxPayloadSize (Section 7.2) has been changed. The advantage of having a Modify is that it allows to initiate a change of flow spec while leaving the current flow is operating until the change is accepted. If there is no linkage between the Join and the Leave, then in figuring out whether the new flow spec can be supported, the central entity has to assume that the resources committed to the current flow are in use. Via Modify the central entity knows that the resources supporting the current flow



can be available for supporting the altered flow. Modify is considered to be an optional operation due to possible control-plane limitations.

As the DetNet UNI can provide service for both L3 and L2 flows, end systems may not need to implement the L3 <=> L2 Transfer Function specified by [IEEE8021CB] (see, e.g., subclause 6.3; see also subclause 46.1 in [IEEE8021Qcc]). An edge node may implement a function similar to the Transfer Function, see, e.g., the Svc Proxy in Figure 1 in [I-D.ietf-detnet-dp-alt].

## 7. Flow

The flows leveraging DetNet service can be unicast or multicast data flows for an application with constrained requirements on network performance, e.g., low packet loss rate and/or latency. Therefore, they can require different connectivity types: point-to-point (p2p) or point-to-multipoint (p2mp). The p2mp connectivity is created by a transport layer function (e.g., p2mp LSP) [I-D.ietf-detnet-dp-alt]. (Note that mp2mp connectivity is a superposition of p2mp connections.)

Many flows using DetNet service are periodic with fix packet size (i.e., Constant Bit Rate (CBR) flows), or periodic with variable packet size.

Delay and loss parameters are correlated because the effect of late delivery can result data loss for an application. However, not all applications require hard limits on both parameters (delay and loss). For example, some real-time applications allow graceful degradation if loss happens (e.g., sample-based processing, media distribution). Some others may require high-bandwidth connections that make the usage of techniques like packet replication economically challenging or even impossible. Some applications may not tolerate loss, but are not delay sensitive (e.g., bufferless sensors). Time/loss sensitive applications may have somewhat special requirements especially for loss (e.g., no loss in two consecutive communication cycles; very low outage time, etc.).

Flows have the following attributes:

- a. DataFlowSpecification (Section 7.1)
- b. TrafficSpecification (Section 7.2)
- c. FlowRank (Section 7.3)

Flow attributes are described in the following sections.

## 7.1. Identification and Specification of Flows

Identification options for DetNet flows at the UNI and within the DetNet domain are specified as follows; see Section 7.1.1 for DetNet L3 flows (at UNI), Section 7.1.2 for DetNet L2 flows (at UNI) and Section 7.1.3 for DetNetwork flows (within the network).

### 7.1.1. DetNet L3 Flow Identification and Specification at UNI

DetNet L3 flows can be identified and specified by the following attributes:

- a. SourceIpAddress
- b. DestinationIpAddress
- c. IPv6FlowLabel
- d. Dscp
- e. Protocol
- f. SourcePort
- g. DestinationPort

### 7.1.2. DetNet L2 Flow Identification and Specification at UNI

DetNet L2 flows can be identified and specified by the following attributes:

- a. DestinationMacAddress
- b. SourceMacAddress
- c. Pcp
- d. VlanId
- e. EtherType

Note: The Multiple Stream Registration Protocol (MSRP) [IEEE8021Q] uses StreamID to match Talker registrations with their corresponding Listener registrations, i.e., to identify Streams (L2 TSN flows). The StreamID includes the following subcomponents:

- o A 48-bit MAC Address associated with the Talker sourcing the stream to the bridged network.

- o A 16-bit unsigned integer value, Unique ID, used to distinguish among multiple streams sourced by the same Talker.

#### 7.1.3. DetNetwork Flow Identification and Specification

Identification of DetNet flows within the DetNet domain are used in the service information model. The attributes are specific to the forwarding paradigm within the DetNet domain. DetNetwork flows can be identified and specified by the following attributes:

- a. SourceIpAddress
- b. DestinationIpAddress
- c. IPv6FlowLabel
- d. (Protocol)
- e. (SourcePort)
- f. (DestinationPort)
- g. MplsLabel

[[Note: attributes in brackets are dependant on current dataplane discussions. ]]

#### 7.2. Traffic Specification

TrafficSpecification specifies how the Source transmits packets for the flow. This is effectively the promise/request of the Source to the network. The network uses this traffic specification to allocate resources and adjust queue parameters in network nodes.

TrafficSpecification has the following attributes:

- a. Interval: the period of time in which the traffic specification cannot be exceeded.
- b. MaxPacketsPerInterval: the maximum number of packets that the Source will transmit in one Interval.
- c. MaxPayloadSize: the maximum payload size that the Source will transmit.

[[NOTE (to be removed from a future revision): These attributes can be used to describe any type of traffic (e.g., CBR, VBR, etc.) and can be used during resource allocation to represent worst case

scenarios. Further optional attributes can be considered to achieve more efficient resource allocation. Such optional attributes might be worth for flows with soft requirements (i.e., the flow is only loss sensitive or only delay sensitive, but not both delay-and-loss sensitive). Possible options how to extend TrafficSpecification attributes is for further discussion. Identified options are described in the following notes.]]

[[NOTE1: Based on the already defined attributes the most similar additional attributes for VBR type flows can be defined as follows:

- o AveragePacketsPerInterval: the average number of packets that the Source will transmit in one Interval.
- o AveragePayloadSize: the average payload size that the Source will transmit.

]]

[[NOTE2: another alternative to deal better with various traffic types can rely on [RFC6003], which describes the support of Metro Ethernet Forum (MEF) Ethernet traffic parameters for using for resource reservation purposes. Such a Bandwidth Profile can be also adapted to describe the set of traffic parameters for a Detnet flow. Committed Rate indicates the rate at which traffic commits to be sent by the source (described in terms of the CIR (Committed Information Rate) and CBS (Committed Burst Size) attributes.) Excess Rate indicates the extent by which the traffic sent by the source exceeds the committed rate. The Excess Rate is described in terms of the EIR (Excess Information Rate) and EBS (Excess Burst Size) attributes. ]]

[[NOTE3: a third alternative is to define application based traffic models such as [GPP22885] defines periodic and event-driven traffic model, and 5G PPP work defines traffic model for MTC (Machine Type Communication) use cases. Periodic traffic type is usually for status update between devices or devices transmit status report to a central unit in regular basis. TrafficPeriod, defines the period of the status update message. DataSize, defines the data size of the message which is constant. 3GPP also defines approximately-periodic transmission with variations on period and uncertainty in the time arrival of the packets. Event-triggered traffic type corresponds traffic being triggered by an MTC device event. MinIntervalBetweenEvent, defines the minimum interval between two events. Event-triggered transmission will not happen all the time, whenever an alert is sent, it waits until the issue being solved to be able to send another alert. MaxPacketPerEvent, defines the max number of packets within one message. ]]

### 7.3. Flow Rank

FlowRank provides the rank of this flow relative to other flows in the network. This rank is used to determine success/failure of flow establishment. Rank (boolean) is used by the network to decide which flows can and cannot exist when network resources reach their limit. Rank is used to help to determine which flows can be dropped (i.e., removed from node configuration) if a port of a node becomes oversubscribed (e.g., due to network reconfiguration). The true value is more important than the false value (i.e., flows with false are dropped first).

### 7.4. Service Rank

ServiceRank provides the rank of this service instance relative to other services in the network. This rank is used to determine success/failure of service instance establishment. Rank (boolean) is used by the network to decide which services can and cannot exist when network resources reach their limit. Rank is used to help to determine which services can be dropped (i.e., removed from node configuration) if a port of a node becomes oversubscribed (e.g., due to network reconfiguration). The true value is more important than the false value (i.e., services with false are dropped first).

[[NOTE: relationship between ServiceRank and FlowRank needs further discussions. A 1:N relationship is assumed (a service instance can serv multiple flows). This sub-section is considered to move to the service related sections. ]]

## 8. Source

The Source object specifies:

- o The behavior of the Source for the flow (how/when the Source transmits).
- o The requirements of the Source from the network.
- o The capabilities of the interface(s) of the Source.

The Source object includes the following attributes:

- a. DataFlowSpecification (Section 7.1)
- b. TrafficSpecification (Section 7.2)
- c. FlowRank (Section 7.3)

- d. EndSystemInterfaces (Section 10.1)
- e. InterfaceCapabilities (Section 10.2)
- f. UserToNetworkRequirements (Section 10.3)

For the join operation, the DataFlowSpecification, FlowRank, EndSystemInterfaces, and TrafficSpecification SHALL be included within the Source. For the join operation, the UserToNetworkRequirements and InterfaceCapabilities groups MAY be included within the Source.

For the leave operation, the DataFlowSpecification and EndSystemInterfaces SHALL be included within the Source.

For the modify operation, the same object SHALL and MAY included as for the join operation.

## 9. Destination

The Destination object includes the following attributes:

- a. DataFlowSpecification (Section 7.1)
- b. EndSystemInterfaces (Section 10.1)
- c. InterfaceCapabilities (Section 10.2)
- d. UserToNetworkRequirements (Section 10.3)

For the join operation, the DataFlowSpecification and EndSystemInterfaces SHALL be included within the Destination. For the join operation, the UserToNetworkRequirements and InterfaceCapabilities groups MAY be included within the Destination.

For the leave operation, the DataFlowSpecification and EndSystemInterfaces SHALL be included within the Destination.

For the modify operation, the same object SHALL and MAY included as for the join operation.

[[NOTE (to be removed from a future revision): Should we add DestinationRank? It could distinguish the importance of Destinations if the flow cannot be provided for all Destinations.]]

## 10. Common Attributes of Source and Destination

Source and Destination end systems have the following common attributes in addition to DataFlowSpecification (Section 7.1).

### 10.1. End System Interfaces

EndSystemInterfaces is a list of identifiers, one for each physical interface (port) in the end system acting as a Source or Destination. An interface is identified by an IP or a MAC address.

EndSystemInterfaces can refer also to logical sub-Interfaces if supported by the end system, e.g., based on IfIndex parameter.

### 10.2. Interface Capabilities

InterfaceCapabilities specifies the network capabilities of all interfaces (ports) contained in the EndSystemInterfaces object (Section 10.1). These capabilities may be configured via the InterfaceConfiguration object (Section 14.2) of the Status object (Section 14).

Note that an end system may have multiple interfaces with different network capabilities. In this case, each interface should be specified in a distinct top-level Source or Destination object (i.e., one entry in EndSystemInterfaces (Section 10.1)). Use of multiple entries in EndSystemInterfaces is intended for network capabilities that span multiple interfaces (e.g., packet replication and elimination).";.

InterfaceCapabilities attributes:

- a. SubInterfaceCapable (sub-interface capable)
- b. PREF-Capable (packet replication and elimination capable)

[[NOTE (to be removed from a future revision): InterfaceCapabilities attributes are to be defined. For information, [IEEE8021Qcc] specifies the following attributes:

- o VlanTagCapable (Customer VLAN Tag capable)
- o CB-Capable (frame replication and elimination capable)
- o CB-StreamIdentTypeList (a list of the optional Stream Identification types supported by the interface as specified in [IEEE8021CB].)

- o CB-SequenceTypeList (a list of the optional Sequence Encode/Decode types supported by the interface as specified in [IEEE8021CB].)

]]

### 10.3. User to Network Requirements

UserToNetworkRequirements specifies user requirements for the flow, such as latency and reliability.

The UserToNetworkRequirements object includes the following attributes:

- a. NumReplicationTrees
- b. MaxLatency

NumReplicationTrees specifies the number of maximally disjoint trees that the network should configure to provide packet replication and elimination for the flow. NumReplicationTrees is provided by the Source only. Destinations SHALL set this element to one. Value zero and one indicate no packet replication and elimination for the flow. When NumReplicationTrees is greater than one, packet replication and elimination is to be used for the flow. If the Source sets this element to greater than one, and packet replication and elimination is not possible in the network (e.g., no disjoint paths, or the nodes do not support packet replication and elimination), then the FailureCode of the Status object is non-zero (Section 14.1).

MaxLatency is the maximum latency from Source to Destination(s) for a single packet of the flow. MaxLatency is specified as an integer number of nanoseconds. When this requirement is specified by the Source, it must be satisfied for all Destinations. When this requirement is specified by a Destination, it must be satisfied for that particular Destination only. If the UserToNetworkRequirements group is not provided within the Source or Destination object, then value zero SHALL be used for this element. Value zero represents a special use for the maximum latency requirement. Value zero locks-down the initial latency that the network provides in the AccumulatedLatency parameter of the Status object (Section 14) after the successful configuration of the flow, such that any subsequent increase in the latency beyond that initial value causes the flow to fail.

[[NOTE-1 (to be removed from a future revision): Should we add a parameter to specify the maximum packet loss rate that can be tolerated for the flow?]]



[[NOTE-2 (to be removed from a future revision): TrafficSpecification (Section 7.2) specifies the Peak Information Rate (PIR) of the flow, which is a kind of user requirement to the network. Should we add Committed Information Rate (CIR), i.e., the minimum rate the user requests to be guaranteed for the flow by the network?]]

#### 11. Ingress

Placeholder ...

#### 12. Egress

Placeholder ...

#### 13. DetNet Domain

The DetNet Domain may change the encapsulation of a DetNet L2 or L3 flow at the UNI. That impacts not only how a flow can be recognised inside the DetNet domain but also the resource reservation calculations.

The DetNet Domain object specifies:

- o The behavior of the flow (how/when it is transmitted).
- o The requirements of the flow from the network.
- o The capabilities of the DetNet domain.

The DetNet domain object includes the following attributes:

- a. DataFlowSpecification (Section 7.1)
- b. TrafficSpecification (Section 7.2)
- c. ServiceRank (Section 7.4)
- d. DetnetDomainCapabilities (Section 13.1)
- e. UserToNetworkRequirements (Section 10.3)

##### 13.1. DetNet Domain Capabilities

DetnetDomainCapabilities specifies the network capabilities, which can be used to provide DetNet service. DetNet Edge nodes may change the encapsulation of a flow according to the data plane used inside the DetNet domain.

DetnetDomainCapabilities object includes the following attributes:

- a. EncapsulationFormat (data plane specific encapsulation)
- b. PREF-Capable (packet replication and elimination capable)

#### 14. Flow-status

The FlowStatus object is provided by the network each Source and Destination of the flow. The Status object provides the status of the flow with respect to the establishment of the flow by the network. The Status object is delivered via the corresponding UNI to each Source and Destination end system of the flow. The Status is distinct for each Source or Destination because the AccumulatedLatency and InterfaceConfiguration objects are distinct, see below.

The Status object SHALL include the attributes a), b), c); and MAY include attributes d), e):

- a. DataFlowSpecification (Section 7.1)
- b. StatusInfo (Section 14.1)
- c. AccumulatedLatency (this section below)
- d. InterfaceConfiguration (Section 14.2)
- e. FailedInterfaces (Section 14.3)

DataFlowSpecification identifies the flow for which status is provided. DataFlowSpecification is described in (Section 7.1) If the Status object is provided without a Source or Destination object in a protocol message via a UNI, then the DataFlowSpecification object SHALL be included within the Status object for both join and leave operations. If the Status object immediately follows a Source or Destination object in the protocol message, then the DataFlowSpecification object is obtained from the Source/Destination object, and therefore DataFlowSpecification is not required within the Status object.

AccumulatedLatency provides the worst-case latency that a single packet of the flow can encounter along its current path(s) in the network. When provided to a Source, AccumulatedLatency is the worst-case latency for all Destinations (worst path). AccumulatedLatency is specified as an integer number of nanoseconds. Latency is measured using the time at which the data frame's message timestamp point passes the reference plane marking the boundary between the

network media and PHY. The message timestamp point is specified by IEEE Std 802.1AS [IEEE8021AS] for various media. For a successful Status, the network returns a value less than or equal to the MaxLatency of the UserToNetworkRequirements (Section 10.3). If the NumReplicationTrees of the UserToNetworkRequirements (Section 10.3) is one, then the AccumulatedLatency SHALL provide the worst latency for the current path from the Source to each Destination. If the path is changed (e.g., due to rerouting), then the AccumulatedLatency changes accordingly. If the NumReplicationTrees of the UserToNetworkRequirements (Section 10.3) is greater than one, AccumulatedLatency SHALL provide the worst latency for all paths in use from the Source to each Destination.

#### 14.1. Status Info

StatusInfo provides information regarding the status of a flow's configuration in the network.

The StatusInfo object MAY include the following attributes:

- a. SourceStatus is an enumeration for the status of the flow's Source:
  - \* None: no Source
  - \* Ready: Source is ready
  - \* Failed: Source failed
- b. DestinationStatus is an enumeration for the status of the flow's Destinations:
  - \* None: no Destination
  - \* Ready: all Destinations are ready
  - \* PartialFailed: One or more Destinations ready, and one or more Listeners failed. The flow can be used if the Source is Ready.
  - \* Failed: All Destinations failed.
- c. FailureCode: A non-zero code that specifies the problem if the flow encounters a failure (e.g., packet replication and elimination is requested but not possible, or SourceStatus is Failed, or DestinationStatus is Failed, or DestinationStatus is PartialFailed).

[[NOTE (to be removed from a future revision): FailureCodes to be defined for DetNet. Table 46-1 of [IEEE8021Qcc] describes TSN failure codes.]]

#### 14.2. Interface Configuration

InterfaceConfiguration provides information about of interfaces in the Source/Destination. This configuration related information assists the network in meeting the requirements of the flow. The InterfaceConfiguration object is according to the capabilities of the interface. InterfaceConfiguration can be distinct for each Source or Destination of each flow. If the InterfaceConfiguration object is not provided within the Status object, then the network SHALL assume zero elements as the default (no interface configuration).

The InterfaceConfiguration object MAY include one or more the following attributes:

- a. MAC or IP Address to identify the interface
- b. DataFlowSpecification (Section 7.1)

#### 14.3. Failed Interfaces

FailedInterfaces provides a list of one or more physical interfaces (ports) in the failed node when a failure occurs in the network.

The FailedInterface object includes the following attributes:

- a. MAC or IP Address to identify the interface
- b. InterfaceName

InterfaceName is the name of the interface (port) within the node. This interface name SHALL be persistent, and unique within the node.

#### 15. Service-status

Placeholder ...

#### 16. Summary

This document describes DetNet flow information model both for DetNet L3 flows and DetNet L2 flows based on the TSN data model specified by [IEEE8021Qcc]. This revision is extended with DetNet specific flow information model elements.

## 17. IANA Considerations

N/A.

## 18. Security Considerations

N/A.

## 19. References

## 19.1. Normative References

[I-D.ietf-detnet-architecture]

Finn, N., Thubert, P., Varga, B., and J. Farkas,  
"Deterministic Networking Architecture", draft-ietf-  
detnet-architecture-03 (work in progress), August 2017.

[I-D.ietf-detnet-dp-alt]

Korhonen, J., Farkas, J., Mirsky, G., Thubert, P.,  
Zhuangyan, Z., and L. Berger, "DetNet Data Plane Protocol  
and Solution Alternatives", draft-ietf-detnet-dp-alt-00  
(work in progress), October 2016.

[I-D.ietf-detnet-use-cases]

Grossman, E., Gunther, C., Thubert, P., Wetterwald, P.,  
Raymond, J., Korhonen, J., Kaneko, Y., Das, S., Zha, Y.,  
Varga, B., Farkas, J., Goetz, F., Schmitt, J., Vilajosana,  
X., Mahmoodi, T., Spirou, S., Vizarrata, P., Huang, D.,  
Geng, X., Dujovne, D., and M. Seewald, "Deterministic  
Networking Use Cases", draft-ietf-detnet-use-cases-13  
(work in progress), September 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<https://www.rfc-editor.org/info/rfc2119>>.

[RFC6003] Papadimitriou, D., "Ethernet Traffic Parameters",  
RFC 6003, DOI 10.17487/RFC6003, October 2010,  
<<https://www.rfc-editor.org/info/rfc6003>>.

## 19.2. Informative References

[GPP22885]

3GPP, "Study on LTE support for Vehicle-to-Everything  
(V2X) services",  
<<http://www.3gpp.org/DynaReport/22885.html>>.

## [IEEE8021AS]

IEEE 802.1, "IEEE 802.1AS-2011: IEEE Standard for Local and metropolitan area networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks", 2011, <<http://standards.ieee.org/getieee802/download/802.1AS-2011.pdf>>.

## [IEEE8021CB]

IEEE 802.1, "IEEE P802.1CB: IEEE Draft Standard for Local and metropolitan area networks - Frame Replication and Elimination for Reliability", 2017, <<http://www.ieee802.org/1/pages/802.1cb.html>>.

## [IEEE8021Q]

IEEE 802.1, "IEEE 802.1Q-2014: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks", 2014, <<http://standards.ieee.org/getieee802/download/802-1Q-2014.pdf>>.

## [IEEE8021Qbv]

IEEE 802.1, "IEEE 802.1Qbv-2015: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks -- Amendment 25: Enhancements for Scheduled Traffic", 2015, <<https://standards.ieee.org/findstds/standard/802.1Qbv-2015.html>>.

## [IEEE8021Qcc]

IEEE 802.1, "IEEE P802.1Qcc-2015: IEEE Draft Standard for Local and metropolitan area networks - Bridges and Bridged Networks -- Amendment: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements", 2017, <<http://www.ieee802.org/1/pages/802.1cc.html>>.

## [IEEE8021TSN]

IEEE 802.1, "IEEE 802.1 Time-Sensitive Networking (TSN) Task Group", <<http://www.ieee802.org/1/pages/tsn.html>>.

## [IETFDetNet]

IETF, "IETF Deterministic Networking (DetNet) Working Group", <<https://datatracker.ietf.org/wg/detnet/charter/>>.

Authors' Addresses

Janos Farkas  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary

Email: [janos.farkas@ericsson.com](mailto:janos.farkas@ericsson.com)

Balazs Varga  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary

Email: [balazs.a.varga@ericsson.com](mailto:balazs.a.varga@ericsson.com)

Rodney Cummings  
National Instruments  
11500 N. Mopac Expwy  
Bldg. C  
Austin, TX 78759-3504  
USA

Email: [rodney.cummings@ni.com](mailto:rodney.cummings@ni.com)

Yuanlong Jiang  
Huawei

Email: [jiangyuanlong@huawei.com](mailto:jiangyuanlong@huawei.com)

Yiyong Zha  
Tencent

DetNet  
Internet-Draft  
Intended status: Informational  
Expires: July 28, 2021

B. Varga  
J. Farkas  
Ericsson  
R. Cummings  
National Instruments  
Y. Jiang  
Huawei Technologies Co., Ltd.  
D. Fedyk  
LabN Consulting, L.L.C.  
January 24, 2021

DetNet Flow and Service Information Model  
draft-ietf-detnet-flow-information-model-14

## Abstract

This document describes flow and service information model for Deterministic Networking (DetNet). These models are defined for IP and MPLS DetNet data planes

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 28, 2021.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect



to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Goals . . . . .	5
1.2. Non Goals . . . . .	5
2. Terminology . . . . .	6
2.1. Terms Used in This Document . . . . .	6
2.2. Abbreviations . . . . .	6
2.3. Naming Conventions . . . . .	7
3. DetNet Domain and its Modeling . . . . .	7
3.1. DetNet Service Overview . . . . .	7
3.2. Reference Points Used in Modeling . . . . .	7
3.3. Information Elements . . . . .	8
4. App-flow Related Parameters . . . . .	8
4.1. App-flow Characteristics . . . . .	8
4.2. App-flow Requirements . . . . .	9
5. DetNet Flow Related Parameters . . . . .	9
5.1. Management ID of the DetNet Flow . . . . .	10
5.2. Payload type of the DetNet Flow . . . . .	10
5.3. Format of the DetNet Flow . . . . .	10
5.4. Identification and Specification of DetNet Flows . . . . .	10
5.4.1. DetNet MPLS Flow Identification and Specification . . . . .	11
5.4.2. DetNet IP Flow Identification and Specification . . . . .	11
5.5. Traffic Specification of the DetNet Flow . . . . .	11
5.6. Endpoints of the DetNet Flow . . . . .	12
5.7. Rank of the DetNet Flow . . . . .	13
5.8. Status of the DetNet Flow . . . . .	13
5.9. Requirements of the DetNet Flow . . . . .	14
5.9.1. Minimum Bandwidth of the DetNet Flow . . . . .	14
5.9.2. Maximum Latency of the DetNet Flow . . . . .	14
5.9.3. Maximum Latency Variation of the DetNet Flow . . . . .	14
5.9.4. Maximum Loss of the DetNet Flow . . . . .	14
5.9.5. Maximum Consecutive Loss of the DetNet Flow . . . . .	14
5.9.6. Maximum Misordering Tolerance of the DetNet Flow . . . . .	15
5.10. BiDir requirement of the DetNet Flow . . . . .	15
6. DetNet Service Related Parameters . . . . .	15
6.1. Management ID of the DetNet service . . . . .	15
6.2. Delivery Type of the DetNet service . . . . .	15
6.3. Delivery Profile of the DetNet Service . . . . .	16
6.3.1. Minimum Bandwidth of the DetNet Service . . . . .	16
6.3.2. Maximum Latency of the DetNet Service . . . . .	16
6.3.3. Maximum Latency Variation of the DetNet Service . . . . .	16
6.3.4. Maximum Loss of the DetNet Service . . . . .	16

6.3.5. Maximum Consecutive Loss of the DetNet Service . . .	16
6.3.6. Maximum Misordering Tolerance of the DetNet Service .	17
6.4. Connectivity Type of the DetNet Service . . . . .	17
6.5. BiDir requirement of the DetNet Service . . . . .	17
6.6. Rank of the DetNet Service . . . . .	17
6.7. Status of the DetNet Service . . . . .	17
7. Flow Specific Operations . . . . .	18
7.1. Join Operation . . . . .	19
7.2. Leave Operation . . . . .	19
7.3. Modify Operation . . . . .	19
8. Summary . . . . .	19
9. IANA Considerations . . . . .	19
10. Security Considerations . . . . .	20
11. References . . . . .	20
11.1. Normative References . . . . .	20
11.2. Informative References . . . . .	20
Authors' Addresses . . . . .	21

## 1. Introduction

Deterministic Networking (DetNet) provides a capability to carry specified unicast or multicast data flows for real-time applications with extremely low packet loss rates and assured maximum end-to-end delivery latency. A description of the general background and concepts of DetNet can be found in [RFC8655].

This document describes the Detnet Flow and Service Information Model. For reference [RFC3444] describes the rationale behind Information Models in general. This document describes the Flow and Service information models for operators and users to understand Detnet services, and for implementors as a guide to the functionality required by Detnet services.

The DetNet Architecture treats the DetNet related data plane functions decomposed into two sub-layers: a service sub-layer and a forwarding sub-layer. The service sub-layer is used to provide DetNet service protection and reordering. The forwarding sub-layer provides resource allocation (to ensure low loss, assured latency, and limited out-of-order delivery) and leverages Traffic Engineering mechanisms.

DetNet service utilizes IP or MPLS and DetNet is currently defined for IP and MPLS networks as shown in Figure 1 based on Figure 2 and Figure 3 of [RFC8938]. IEEE 802.1 Time Sensitive Networking (TSN) utilizes Ethernet and is defined over Ethernet networks. A DetNet flow includes one or more App-flow(s) as payload. App-flows can be Ethernet, MPLS, or IP flows, which impacts which header fields are utilized to identify a flow. DetNet flows are identified by the

DetNet encapsulation of App-flow(s) (e.g., MPLS labels, IP 6-tuple etc.). In some scenarios App-flow and DetNet flow look similar on the wire (e.g., L3 App-flow over a DetNet IP network).

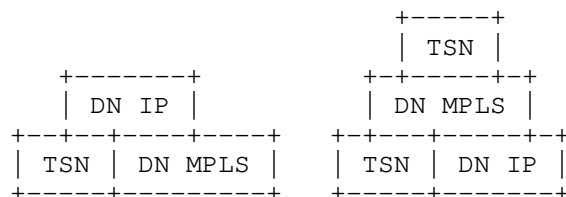


Figure 1: DetNet Service Examples as per Data Plane Framework

As shown in Figure 1 as per [RFC8938] a DetNet flow can be treated as an application level flow (App-flow) e.g., at DetNet flow aggregation or in a sub-network that interconnects DetNet nodes.

The DetNet flow and service information model provided by this document contains both DetNet flow and App-flow specific information in an integrated fashion.

In a given network scenario three information models can be distinguished:

- o Flow models that describe characteristics of data flows. These models describe in detail all relevant aspects of a flow that are needed to support the flow properly by the network between the source and the destination(s).
- o Service models that describe characteristics of services being provided for data flows over a network. These models can be treated as a network operator independent information model.
- o Configuration models that describe in detail the settings required on network nodes to provide a data flow proper service.

Service and flow information models are used between the user and the network operator. Configuration information models are used between the management/control plane entity of the network and the network nodes. They are shown in Figure 2.

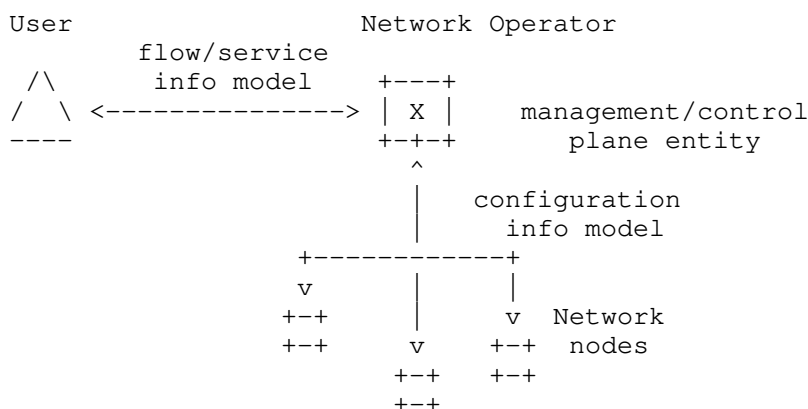


Figure 2: Usage of Information models (flow, service and configuration)

DetNet flow and service information model is based on [RFC8655] and on the concept of data model specified by [IEEE8021Qcc]. In addition to the TSN data model, [IEEE8021Qcc] also specifies configuration of TSN features (e.g., traffic scheduling specified by [IEEE8021Qbv]). The common architecture and flow model, allow configured features to be consistent in certain deployment scenarios, e.g., when the network that provides the DetNet service includes both L3 and L2 network segments.

### 1.1. Goals

As expressed in the [IETFDetNet] Charter, the DetNet WG collaborates with IEEE 802.1 TSN in order to define a common architecture for both Layer 2 and Layer 3. This is beneficial for several reasons, e.g., in order to simplify implementations and maintain consistency across diverse networks. The flow and service information models are also aligned for those reasons. Therefore, the DetNet flow and service information models described in this document are based on [IEEE8021Qcc], which is an amendment to [IEEE8021Q].

This document specifies flow and service information models only.

### 1.2. Non Goals

This document does not specify flow data models or DetNet configuration. Therefore, the goals of this document differ from the goals of [IEEE8021Qcc], which also specifies the TSN data model and configuration of certain TSN features.

The DetNet specific YANG data model is described in [I-D.ietf-detnet-yang].

## 2. Terminology

### 2.1. Terms Used in This Document

This document uses the terminology established in the DetNet architecture [RFC8655] and the DetNet Data Plane Framework [RFC8938]. The reader is assumed to be familiar with these documents and any terminology defined therein. The DetNet  $\Leftrightarrow$  TSN dictionary of [RFC8655] is used to perform translation from [IEEE8021Qcc] to this document.

The following terminology is used in accordance with [RFC8655]:

App-flow	The payload (data) carried over a DetNet service.
DetNet flow	A DetNet flow is a sequence of packets which conform uniquely to a flow identifier, and to which the DetNet service is to be provided. It includes any DetNet headers added to support the DetNet service and forwarding sub-layers.

The following terminology is introduced in this document:

Source	Reference point for an App-flow, where the flow starts.
Destination	Reference point for an App-flow, where the flow terminates.
DN Ingress	Reference point for the start of a DetNet flow. Networking technology specific encapsulation may be added here to the served App-flow(s).
DN Egress	Reference point for the termination of a DetNet flow. Networking technology specific encapsulation may be removed here from the served App-flow(s).

### 2.2. Abbreviations

The following abbreviations are used in this document:

DetNet	Deterministic Networking.
DN	DetNet.
MPLS	Multiprotocol Label Switching.

PSN                    Packet Switched Network.

TSN                    Time-Sensitive Networking.

### 2.3. Naming Conventions

The following naming conventions were used for naming information model components in this document. It is recommended that extensions of the model use the same conventions.

- o Descriptive names are used.
- o Names start with uppercase letters.
- o Composed names use capital letters for the first letter of each component. All other letters are lowercase, even for acronyms. Exceptions are made for acronyms containing a mixture of lowercase and capital letters, such as IPv6. Example composed names are SourceMacAddress and DestinationIPv6Address.

## 3. DetNet Domain and its Modeling

### 3.1. DetNet Service Overview

The DetNet service can be defined as a service that provides a capability to carry a unicast or a multicast data flow for an application with constrained requirements on network performance, e.g., low packet loss rate and/or latency.

Figure 5 and Figure 8 in [RFC8655] show the DetNet service related reference points and main components.

### 3.2. Reference Points Used in Modeling

From service design perspective a fundamental question is the location of the service/flow endpoints, i.e., where the service/flow starts and ends.

App-flow specific reference points are the Source (where it starts) and the Destination (where it terminates). Similarly a DetNet flow has reference points termed DN Ingress (where a DetNet flow starts) and DN Egress (where a DetNet flow ends). These reference points may coexist in the same node (e.g., in a DetNet IP end system). DN Ingress and DN Egress reference points are intermediate reference points for a served App-flow.

All reference points are assumed in this document to be packet-based reference points. A DN Ingress may add and a DN Egress may remove

networking technology specific encapsulation to/from the served App-flow(s) (e.g., MPLS label(s), UDP and IP headers).

### 3.3. Information Elements

The DetNet flow information model and the service model relies on three groups of information elements:

- o App-flow related parameters: these describe the App-flow characteristics (e.g., identification, encapsulation, traffic specification, endpoints, status, etc.) and the App-flow service expectations (e.g., delay, loss, etc.).
- o DetNet flow related parameters: these describe the DetNet flow characteristics (e.g., identification, format, traffic specification, endpoints, rank, etc.).
- o DetNet service related parameters: these describe the expected service characteristics (e.g., delivery type, connectivity delay/loss, status, rank, etc.).

In the information model a DetNet flow contains one or more (aggregated) App-flows (N:1 mapping). During DetNet aggregation the aggregated DetNet flows are treated simply as App-flows and the aggregate is the DetNet flow, which provides N:1 mapping. Similarly, there is an aggregated many to one relationship for the DetNet flow(s) to the DetNet Service.

## 4. App-flow Related Parameters

When Deterministic service is required by time/loss sensitive application(s) running on an end system during communication with its peer(s), the resulting data exchange has various requirements on delay and/or loss parameters.

### 4.1. App-flow Characteristics

App-flow characteristics are described by the following parameters:

- o FlowID: a unique (management) identifier of the App-flow. It can be used to define the N:1 mapping of App-flows to a DetNet flow.
- o FlowType: set by the encapsulation format of the flow. It can be Ethernet (TSN), MPLS, or IP.
- o DataFlowSpecification: a flow descriptor, defining which packets belongs to a flow, using specific packet header fields such as src-addr, dst-addr, label, VLAN-ID, etc.

- o **TrafficSpecification:** a flow descriptor, defining traffic parameters such as packet size, transmission time interval, and maximum packets per time interval.
- o **FlowEndpoints:** delineate the start and termination reference points of the App-flow by pointing to the source interface/node and destination interface(s)/node(s).
- o **FlowStatus:** indicates the status of the App-flow with respect to the establishment of the flow by the connected network, e.g., ready, failed, etc.
- o **FlowRank:** indicates the rank of this flow relative to other flows in the connected network.

Note: When defining the N:1 mapping of App-flows to a DetNet flow, the App-flows must have the same FlowType and different DataFlowSpecification parameters

#### 4.2. App-flow Requirements

App-flow requirements are described by the following parameters:

- o **FlowRequirements:** defines the attributes of the App-flow regarding bandwidth, latency, latency variation, loss, and misordering tolerance.
- o **FlowBiDir:** defines the data path requirement of the App-flow whether it must share the same data path and physical path for both directions through the network, e.g., to provide congruent paths in the two directions.

#### 5. DetNet Flow Related Parameters

The Data model specified by [IEEE8021Qcc] describes data flows using TSN service as periodic flows with fixed packet size (i.e., Constant Bit Rate (CBR) flows) or with variable packet size. The same concept is applied for flows using DetNet service.

Latency and loss parameters are correlated because the effect of late delivery can result in data loss for an application. However, not all applications require hard limits on both latency and loss. For example, some real-time applications allow graceful degradation if loss happens (e.g., sample-based data processing, media distribution). Some other applications may require high-bandwidth connections that make use of packet replication techniques which are economically challenging or even impossible. Some applications may not tolerate loss, but are not delay sensitive (e.g., bufferless



sensors). Time or loss sensitive applications may have somewhat special requirements especially for loss (e.g., no loss over two consecutive communication cycles; very low outage time, etc.).

DetNet flows have the following attributes:

- a. DnFlowID (Section 5.1)
- b. DnPayloadType (Section 5.2)
- c. DnFlowFormat (Section 5.3)
- d. DnFlowSpecification (Section 5.4)
- e. DnTrafficSpecification (Section 5.5)
- f. DnFlowEndpoints (Section 5.6)
- g. DnFlowRank (Section 5.7)
- h. DnFlowStatus (Section 5.8)

DetNet flows have the following requirement attributes:

- o DnFlowRequirements (Section 5.9)
- o DnFlowBiDir (Section 5.10)

Flow attributes are described in the following sections.

#### 5.1. Management ID of the DetNet Flow

A unique (management) identifier is needed for each DetNet flow within the DetNet domain. It is specified by DnFlowID. It can be used to define the N:1 mapping of DetNet flows to a DetNet service.

#### 5.2. Payload type of the DetNet Flow

The DnPayloadType attribute is set according to the encapsulated App-flow format. The attribute can be Ethernet, MPLS, or IP.

#### 5.3. Format of the DetNet Flow

The DnFlowFormat attribute is set according to the DetNet PSN technology. The attribute can be MPLS or IP.

#### 5.4. Identification and Specification of DetNet Flows

Identification options for DetNet flows at the Ingress/Egress and within the DetNet domain are specified as follows; see Section 5.4.1 for DetNet MPLS flows and Section 5.4.2 for DetNetw IP flows.

#### 5.4.1. DetNet MPLS Flow Identification and Specification

The identification of DetNet MPLS flows within the DetNet domain is based on the MPLS context in the service information model. The attributes are specific to the MPLS forwarding paradigm within the DetNet domain [I-D.ietf-detnet-mpls]. DetNet MPLS flows can be identified and specified by the following attributes:

- a. SLabel
- b. FLabelStack

#### 5.4.2. DetNet IP Flow Identification and Specification

DetNet IP flows can be identified and specified by the following attributes [RFC8939]:

- a. SourceIpAddress
- b. DestinationIpAddress
- c. IPv6FlowLabel
- d. Dscp (attribute)
- e. Protocol
- f. SourcePort
- g. DestinationPort
- h. IPSecSpi

The IP 6-tuple that is used for DetNet IP flow identification consists of items a, b, d, e, f, and g. Items c and h are additional attributes that can be used for DetNet flow identification in addition to the 6-tuple. 6-tuple and use of wild cards for these attributes are specified in [RFC8939].

#### 5.5. Traffic Specification of the DetNet Flow

DnTrafficSpecification attributes specify how the DN Ingress transmits packets for the DetNet flow. This is effectively the promise/request of the DN Ingress to the network. The network uses this traffic specification to allocate resources and adjust queue parameters in network nodes.

TrafficSpecification has the following attributes:

- a. Interval: the period of time in which the traffic specification is specified.
- b. MaxPacketsPerInterval: the maximum number of packets that the Ingress will transmit in one Interval.

- c. `MaxPayloadSize`: the maximum payload size that the Ingress will transmit.
- d. `MinPayloadSize`: the minimum payload size that the Ingress will transmit.
- e. `MinPacketsPerInterval`: the minimum number of packets that the Ingress will transmit in one Interval.

These attributes can be used to describe any type of traffic (e.g., CBR, VBR, etc.) and can be used during resource allocation to represent worst case scenarios. Intervals are specified as an integer number of nanoseconds. PayloadSizes are specified in octets.

Flows exceeding the traffic specification (i.e., having more traffic than defined by the maximum attributes) may receive a different network behavior than the DetNet network has been engineered for. Excess traffic due to malicious or malfunctioning devices can be prevented or mitigated (e.g., through the use of existing mechanisms such as policing and shaping).

When `MinPayloadSize` and `MinPacketsPerInterval` parameters are used, then all packets less than the `MinPayloadSize` will be counted as being of the size `MinPayloadSize` during packet processing when packet size matters, e.g., when policing; and all flows having less than `MinPacketsPerInterval` will be counted as having `MinPacketsPerInterval` when the number of packets per interval matters, e.g., during resource reservation. However, flows having less than `MinPacketsPerInterval` may result in a different network behavior than the DetNet network has been engineered for. `MinPayloadSize` and `MinPacketsPerInterval` parameters, for example, may be used when engineering the latency bounds of a DetNet flow when POF is applied to the given DetNet flow.

Further optional attributes can be considered to achieve more efficient resource allocation. Such optional attributes might be worth for flows with soft requirements (i.e., the flow is only loss sensitive or only delay sensitive, but not both delay-and-loss sensitive). Possible options how to extend `DnTrafficSpecification` attributes is for further discussion.

## 5.6. Endpoints of the DetNet Flow

The `DnFlowEndpoints` attribute defines the starting and termination reference points of the DetNet flow by pointing to the ingress interface/node and egress interface(s)/node(s). Depending on the network scenario it defines an interface or a node. Interface can be defined for example if the App-flow is a TSN Stream and it is

received over a well defined UNI (User-to-Network Interface). For example, for App-flows with MPLS encapsulation defining an ingress node is more common when per platform label space is used.

#### 5.7. Rank of the DetNet Flow

The DnFlowRank attribute provides the rank of this flow relative to other flows in the DetNet domain. Rank (range: 0-255) is used by the DetNet domain to decide which flows can and cannot exist when network resources reach their limit. Rank is used to help to determine which flows can be bumped (i.e., removed from node configuration thereby releasing its resources) if for example a port of a node becomes oversubscribed (e.g., due to network re-configuration). DnFlowRank value 0 is the highest priority.

#### 5.8. Status of the DetNet Flow

DnFlowStatus provides the status of the DetNet flow with respect to the establishment of the flow by the DetNet domain.

The DnFlowStatus includes the following attributes:

- a. DnIngressStatus is an enumeration for the status of the flow's Ingress reference point:
  - \* None: no Ingress.
  - \* Ready: Ingress is ready.
  - \* Failed: Ingress failed.
  - \* OutOfService: Administratively blocked.
- b. DnEgressStatus is an enumeration for the status of the flow's Egress reference points:
  - \* None: no Egress.
  - \* Ready: all Egresses are ready.
  - \* PartialFailed: One or more Egress ready, and one or more Egress failed. The DetNet flow can be used if the Ingress is Ready.
  - \* Failed: All Egresses failed.
  - \* OutOfService: All Egresses are administratively blocked.
- c. FailureCode: A non-zero code that specifies the error if the DetNet flow encounters a failure (e.g., packet replication and elimination is requested but not possible, or DnIngressStatus is Failed, or DnEgressStatus is Failed, or DnEgressStatus is PartialFailed).

Defining FailureCodes for DetNet is out-of-scope in this document. Table 46-1 of [IEEE8021Qcc] describes TSN failure codes.

#### 5.9. Requirements of the DetNet Flow

DnFlowRequirements specifies requirements to ensure the service level desired for the DetNet flow.

The DnFlowRequirements includes the following attributes:

- a. MinBandwidth(Section 5.9.1)
- b. MaxLatency(Section 5.9.2)
- c. MaxLatencyVariation(Section 5.9.3)
- d. MaxLoss(Section 5.9.4)
- e. MaxConsecutiveLossTolerance(Section 5.9.5)
- f. MaxMisordering(Section 5.9.6)

##### 5.9.1. Minimum Bandwidth of the DetNet Flow

MinBandwidth is the minimum bandwidth that has to be guaranteed for the DetNet flow. MinBandwidth is specified in octets per second.

##### 5.9.2. Maximum Latency of the DetNet Flow

MaxLatency is the maximum latency from Ingress to Egress(es) for a single packet of the DetNet flow. MaxLatency is specified as an integer number of nanoseconds.

##### 5.9.3. Maximum Latency Variation of the DetNet Flow

MaxLatencyVariation is the difference between the minimum and the maximum end-to-end one-way latency. MaxLatencyVariation is specified as an integer number of nanoseconds.

##### 5.9.4. Maximum Loss of the DetNet Flow

MaxLoss defines the maximum Packet Loss Ratio (PLR) requirement for the DetNet flow between the Ingress and Egress(es) and the loss measurement interval.

##### 5.9.5. Maximum Consecutive Loss of the DetNet Flow

Some applications have special loss requirement, such as MaxConsecutiveLossTolerance. The maximum consecutive loss tolerance parameter describes the maximum number of consecutive packets whose loss can be tolerated. The maximum consecutive loss tolerance can be measured for example based on sequence number.

#### 5.9.6. Maximum Misordering Tolerance of the DetNet Flow

MaxMisordering describes the tolerable maximum number of packets that can be received out of order. The value zero for the maximum allowed misordering indicates that in order delivery is required, misordering cannot be tolerated.

The maximum allowed misordering can be measured for example based on sequence numbers. When a packet arrives at the egress after a packet with a higher sequence number, the difference between the sequence number values cannot be bigger than "MaxMisordering + 1".

#### 5.10. BiDir requirement of the DetNet Flow

DnFlowBiDir attribute defines the requirement that the flow and the corresponding reverse direction flow must share the same path (links and nodes) through the routed or switch network in the DetNet domain, e.g., to provide congruent paths in the two directions that share fate and path characteristics.

### 6. DetNet Service Related Parameters

DetNet service have the following attributes:

- a. DnServiceID (Section 6.1)
- b. DnServiceDeliveryType (Section 6.2)
- c. DnServiceDeliveryProfile (Section 6.3)
- d. DnServiceConnectivity (Section 6.4)
- e. DnServiceBiDir (Section 6.5)
- f. DnServiceRank (Section 6.6)
- g. DnServiceStatus (Section 6.7)

Service attributes are described in the following sections.

#### 6.1. Management ID of the DetNet service

A unique (management) identifier for each DetNet service within the DetNet domain. It can be used to define the many to one mapping of DetNet flows to a DetNet service.

#### 6.2. Delivery Type of the DetNet service

The DnServiceDeliveryType attribute is set according to the payload of the served DetNet flow (i.e., the encapsulated App-flow format). The attribute can be Ethernet, MPLS, or IP.

### 6.3. Delivery Profile of the DetNet Service

DnServiceDeliveryProfile specifies delivery profile to ensure proper serving of the DetNet flow.

The DnServiceDeliveryProfile includes the following attributes:

- a. MinBandwidth(Section 6.3.1)
- b. MaxLatency(Section 6.3.2)
- c. MaxLatencyVariation(Section 6.3.3)
- d. MaxLoss(Section 6.3.4)
- e. MaxConsecutiveLossTolerance(Section 6.3.5)
- f. MaxMisordering(Section 6.3.6)

#### 6.3.1. Minimum Bandwidth of the DetNet Service

MinBandwidth is the minimum bandwidth that has to be guaranteed for the DetNet service. MinBandwidth is specified in octets per second and excludes additional DetNet header (if any).

#### 6.3.2. Maximum Latency of the DetNet Service

MaxLatency is the maximum latency from Ingress to Egress(es) for a single packet of the DetNet flow. MaxLatency is specified as an integer number of nanoseconds.

#### 6.3.3. Maximum Latency Variation of the DetNet Service

MaxLatencyVariation is the difference between the minimum and the maximum end-to-end one-way latency. MaxLatencyVariation is specified as an integer number of nanoseconds.

#### 6.3.4. Maximum Loss of the DetNet Service

MaxLoss defines the maximum Packet Loss Ratio (PLR) parameter for the DetNet service between the Ingress and Egress(es) of the DetNet domain.

#### 6.3.5. Maximum Consecutive Loss of the DetNet Service

Some applications have special loss requirement, such as MaxConsecutiveLossTolerance. The maximum consecutive loss tolerance parameter describes the maximum number of consecutive packets whose loss can be tolerated. The maximum consecutive loss tolerance can be measured for example based on sequence number.

#### 6.3.6. Maximum Misordering Tolerance of the DetNet Service

MaxMisordering describes the tolerable maximum number of packets that can be received out of order. The maximum allowed misordering can be measured for example based on sequence number. The value zero for the maximum allowed misordering indicates that in order delivery is required, misordering cannot be tolerated.

#### 6.4. Connectivity Type of the DetNet Service

Two connectivity types are distinguished: point-to-point (p2p) and point-to-multipoint (p2mp). Connectivity type p2mp may be created by a forwarding function (e.g., p2mp LSP). (Note: from service perspective mp2mp connectivity can be treated as a superposition of p2mp connections.)

#### 6.5. BiDir requirement of the DetNet Service

The DnServiceBiDir attribute defines the requirement that the flow and the corresponding reverse direction flow must share the same path (links and nodes) through the routed or switch network in the DetNet domain, e.g., to provide congruent paths in the two directions that share fate and path characteristics.

#### 6.6. Rank of the DetNet Service

The DnServiceRank attribute provides the rank of a service instance relative to other services in the DetNet domain. DnServiceRank (range: 0-255) is used by the network in case of network resource limitation scenarios. DnServiceRank value 0 is the highest priority.

#### 6.7. Status of the DetNet Service

DnServiceStatus information group includes elements that specify the status of the service specific state of the DetNet domain. This information group informs the user whether or not the service is ready for use.

The DnServiceStatus includes the following attributes:

- a. DnServiceIngressStatus is an enumeration for the status of the service's Ingress:

- \* None: no Ingress.
- \* Ready: Ingress is ready.
- \* Failed: Ingress failed.
- \* OutOfService: Administratively blocked.



- b. DnServiceEgressStatus is an enumeration for the status of the service's Egress:
- \* None: no Egress.
  - \* Ready: all Egresses are ready.
  - \* PartialFailed: One or more Egress ready, and one or more Egress failed. The DetNet flow can be used if the Ingress is Ready.
  - \* Failed: All Egresses failed.
  - \* OutOfService: Administratively blocked.
- c. DnServiceFailureCode: A non-zero code that specifies the error if the DetNet service encounters a failure (e.g., packet replication and elimination is requested but not possible, or DnServiceIngressStatus is Failed, or DnServiceEgressStatus is Failed, or DnServiceEgressStatus is PartialFailed).

Defining DnServiceFailureCodes for DetNet service is out-of-scope in this document. Table 46-1 of [IEEE8021Qcc] describes TSN failure codes.

## 7. Flow Specific Operations

The DetNet flow information model relies on three high level information groups:

- o DnIngress: The DnIngress information group includes elements that specify the source for a single DetNet flow. This information group is applied from the user of the DetNet service to the network.
- o DnEgress: The DnEgress information group includes elements that specify the destination for a single DetNet flow. This information group is applied from the user of the DetNet service to the network.
- o DnFlowStatus: The status information group includes elements that specify the status of the flow in the network. This information group is applied from the network to the user of the DetNet service. This information group informs the user whether or not the DetNet flow is ready for use.

There are three possible operations for each DetNet flow with respect to its DetNet service at a DN Ingress or a DN Egress (similarly to App-flows at a Source or a Destination):

- o Join: DN Ingress/DN Egress intends to join the flow.
- o Leave: DN Ingress/DN Egress intends to leave the flow.

- o Modify: DN Ingress/DN Egress intends to change the flow.

#### 7.1. Join Operation

For the join operation, the DnFlowSpecification, DnFlowRank, DnFlowEndpoint, and DnTrafficSpecification are included within the DnIngress or DnEgress information group. For the join operation, the DnServiceRequirements groups can be included.

#### 7.2. Leave Operation

For the leave operation, the DnFlowSpecification and DnFlowEndpoint are included within the DnIngress or DnEgress information group.

#### 7.3. Modify Operation

For the modify operation, the DnFlowSpecification, DnFlowRank, DnFlowEndpoint, and DnTrafficSpecification are included within the DnIngress or DnEgress information group. For the join operation, the DnServiceRequirements groups can be included.

The Modify operation can be considered to address cases when a flow is slightly changed, e.g., only MaxPayloadSize (Section 5.5) has been changed. The advantage of having a Modify is that it allows initiation of a change of flow spec while leaving the current flow is operating until the change is accepted. If there is no linkage between the Join and the Leave, then while figuring out whether the new flow spec can be supported, the controller entity has to assume that the resources committed to the current flow are in use. By using Modify the controller entity knows that the resources supporting the current flow can be available for supporting the altered flow. Modify is considered to be an optional operation due to possible controller plane limitations.

### 8. Summary

This document describes the DetNet flow information model and the service information model for DetNet IP networks and DetNet MPLS networks. These models are used as input for creating the DetNet specific YANG model.

### 9. IANA Considerations

N/A.

## 10. Security Considerations

The external interfaces of the DetNet domain need to be subject to appropriate confidentiality. Additionally, knowledge of which flows/services are provided to a customer or delivered by a network operator may supply information that can be used in a variety of security attacks. Security considerations for DetNet are described in detail in [I-D.ietf-detnet-security]. General security considerations are described in [RFC8655]. This document discusses modeling the information, not how it is exchanged.

## 11. References

### 11.1. Normative References

[I-D.ietf-detnet-mpls]

Varga, B., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS", draft-ietf-detnet-mpls-13 (work in progress), October 2020.

[IEEE8021Qcc]

IEEE Standards Association, "IEEE Std 802.1Qcc-2018: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks -- Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements", 2018,  
<<https://ieeexplore.ieee.org/document/8514112/>>.

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019,  
<<https://www.rfc-editor.org/info/rfc8655>>.

[RFC8939] Varga, B., Ed., Farkas, J., Berger, L., Fedyk, D., and S. Bryant, "Deterministic Networking (DetNet) Data Plane: IP", RFC 8939, DOI 10.17487/RFC8939, November 2020,  
<<https://www.rfc-editor.org/info/rfc8939>>.

### 11.2. Informative References

[I-D.ietf-detnet-security]

Grossman, E., Mizrahi, T., and A. Hacker, "Deterministic Networking (DetNet) Security Considerations", draft-ietf-detnet-security-13 (work in progress), December 2020.

[I-D.ietf-detnet-yang]

Geng, X., Chen, M., Ryoo, Y., Fedyk, D., Rahman, R., and Z. Li, "Deterministic Networking (DetNet) Configuration YANG Model", draft-ietf-detnet-yang-09 (work in progress), November 2020.

[IEEE8021Q]

IEEE Standards Association, "IEEE Std 802.1Q-2018 IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks", 2018,  
<<https://ieeexplore.ieee.org/document/8403927>>.

[IEEE8021Qbv]

IEEE Standards Association, "IEEE Std 802.1Qbv-2015 IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", 2015,  
<<https://ieeexplore.ieee.org/document/7572858/>>.

[IETFDetNet]

IETF, "IETF Deterministic Networking (DetNet) Working Group", <<https://datatracker.ietf.org/wg/detnet/charter/>>.

[RFC3444]

Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003,  
<<https://www.rfc-editor.org/info/rfc3444>>.

[RFC8938]

Varga, B., Ed., Farkas, J., Berger, L., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC 8938, DOI 10.17487/RFC8938, November 2020,  
<<https://www.rfc-editor.org/info/rfc8938>>.

#### Authors' Addresses

Balazs Varga  
Ericsson  
Magyar tudosok korutja 11  
Budapest 1117  
Hungary  
  
Email: [balazs.a.varga@ericsson.com](mailto:balazs.a.varga@ericsson.com)

Janos Farkas  
Ericsson  
Magyar tudosok korutja 11  
Budapest 1117  
Hungary

Email: [janos.farkas@ericsson.com](mailto:janos.farkas@ericsson.com)

Rodney Cummings  
National Instruments  
11500 N. Mopac Expwy  
Bldg. C  
Austin, TX 78759-3504  
USA

Email: [rodney.cummings@ni.com](mailto:rodney.cummings@ni.com)

Yuanlong Jiang  
Huawei Technologies Co., Ltd.  
Bantian, Longgang district

Shenzhen 518129  
China

Email: [jiangyuanlong@huawei.com](mailto:jiangyuanlong@huawei.com)

Don Fedyk  
LabN Consulting, L.L.C.

Email: [dfedyk@labn.net](mailto:dfedyk@labn.net)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: May 3, 2018

T. Mizrahi  
MARVELL  
E. Grossman, Ed.  
DOLBY  
A. Hacker  
MISTIQ  
S. Das  
Applied Communication Sciences  
J. Dowdell  
Airbus Defence and Space  
H. Austad  
Cisco Systems  
K. Stanton  
INTEL  
N. Finn  
HUAWEI  
October 30, 2017

Deterministic Networking (DetNet) Security Considerations  
draft-ietf-detnet-security-01

Abstract

A deterministic network is one that can carry data flows for real-time applications with extremely low data loss rates and bounded latency. Deterministic networks have been successfully deployed in real-time operational technology (OT) applications for some years (for example [ARINC664P7]). However, such networks are typically isolated from external access, and thus the security threat from external attackers is low. IETF Deterministic Networking (DetNet) specifies a set of technologies that enable creation of deterministic networks on IP-based networks of potentially wide area (on the scale of a corporate network) potentially bringing the OT network into contact with Information Technology (IT) traffic and security threats that lie outside of a tightly controlled and bounded area (such as the internals of an aircraft). These DetNet technologies have not previously been deployed together on a wide area IP-based network, and thus can present security considerations that may be new to IP-based wide area network designers. This draft, intended for use by DetNet network designers, provides insight into these security considerations. In addition, this draft collects all security-related statements from the various DetNet drafts (Architecture, Use Cases, etc) into a single location Section 7.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2018.

## Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
2. Abbreviations . . . . .	5
3. Security Threats . . . . .	6
3.1. Threat Model . . . . .	6
3.2. Threat Analysis . . . . .	7
3.2.1. Delay . . . . .	7
3.2.1.1. Delay Attack . . . . .	7
3.2.2. DetNet Flow Modification or Spoofing . . . . .	7
3.2.3. Resource Segmentation or Slicing . . . . .	7
3.2.3.1. Inter-segment Attack . . . . .	7
3.2.4. Packet Replication and Elimination . . . . .	8
3.2.4.1. Replication: Increased Attack Surface . . . . .	8
3.2.4.2. Replication-related Header Manipulation . . . . .	8
3.2.5. Path Choice . . . . .	8

3.2.5.1.	Path Manipulation . . . . .	8
3.2.5.2.	Path Choice: Increased Attack Surface . . . . .	9
3.2.6.	Control Plane . . . . .	9
3.2.6.1.	Control or Signaling Packet Modification . . . . .	9
3.2.6.2.	Control or Signaling Packet Injection . . . . .	9
3.2.7.	Scheduling or Shaping . . . . .	9
3.2.7.1.	Reconnaissance . . . . .	9
3.2.8.	Time Synchronization Mechanisms . . . . .	9
3.3.	Threat Summary . . . . .	9
4.	Security Threat Impacts . . . . .	10
4.1.	Delay-Attacks . . . . .	13
4.1.1.	Data Plane Delay Attacks . . . . .	13
4.1.2.	Control Plane Delay Attacks . . . . .	13
4.2.	Flow Modification and Spoofing . . . . .	14
4.2.1.	Flow Modification . . . . .	14
4.2.2.	Spoofing . . . . .	14
4.2.2.1.	Dataplane Spoofing . . . . .	14
4.2.2.2.	Control Plane Spoofing . . . . .	14
4.3.	Segmentation attacks (injection) . . . . .	15
4.3.1.	Data Plane Segmentation . . . . .	15
4.3.2.	Control Plane segmentation . . . . .	15
4.4.	Replication and Elimination . . . . .	15
4.4.1.	Increased Attack Surface . . . . .	15
4.4.2.	Header Manipulation at Elimination Bridges . . . . .	15
4.5.	Control or Signaling Packet Modification . . . . .	16
4.6.	Control or Signaling Packet Injection . . . . .	16
4.7.	Reconnaissance . . . . .	16
4.8.	Attacks on Time Sync Mechanisms . . . . .	16
4.9.	Attacks on Path Choice . . . . .	16
5.	Security Threat Mitigation . . . . .	16
5.1.	Path Redundancy . . . . .	16
5.2.	Integrity Protection . . . . .	17
5.3.	DetNet Node Authentication . . . . .	17
5.4.	Encryption . . . . .	17
5.5.	Control and Signaling Message Protection . . . . .	18
5.6.	Dynamic Performance Analytics . . . . .	18
5.7.	Mitigation Summary . . . . .	18
6.	Association of Attacks to Use Cases . . . . .	20
6.1.	Use Cases by Common Themes . . . . .	20
6.1.1.	Network Layer - AVB/TSN Ethernet . . . . .	20
6.1.2.	Central Administration . . . . .	21
6.1.3.	Hot Swap . . . . .	21
6.1.4.	Data Flow Information Models . . . . .	22
6.1.5.	L2 and L3 Integration . . . . .	22
6.1.6.	End-to-End Delivery . . . . .	22
6.1.7.	Proprietary Deterministic Ethernet Networks . . . . .	23
6.1.8.	Replacement for Proprietary Fieldbuses . . . . .	23
6.1.9.	Deterministic vs Best-Effort Traffic . . . . .	23



6.1.10. Deterministic Flows . . . . .	24
6.1.11. Unused Reserved Bandwidth . . . . .	24
6.1.12. Interoperability . . . . .	24
6.1.13. Cost Reductions . . . . .	25
6.1.14. Insufficiently Secure Devices . . . . .	25
6.1.15. DetNet Network Size . . . . .	25
6.1.16. Multiple Hops . . . . .	26
6.1.17. Level of Service . . . . .	26
6.1.18. Bounded Latency . . . . .	27
6.1.19. Low Latency . . . . .	27
6.1.20. Symmetrical Path Delays . . . . .	27
6.1.21. Reliability and Availability . . . . .	27
6.1.22. Redundant Paths . . . . .	28
6.1.23. Security Measures . . . . .	28
6.2. Attack Types by Use Case Common Theme . . . . .	28
7. Appendix A: DetNet Draft Security-Related Statements . . . . .	30
7.1. Architecture (draft 8) . . . . .	31
7.1.1. Fault Mitigation (sec 4.5) . . . . .	31
7.1.2. Security Considerations (sec 7) . . . . .	31
7.2. Data Plane Alternatives (draft 4) . . . . .	32
7.2.1. Security Considerations (sec 7) . . . . .	32
7.3. Problem Statement (draft 5) . . . . .	32
7.3.1. Security Considerations (sec 5) . . . . .	32
7.4. Use Cases (draft 11) . . . . .	33
7.4.1. (Utility Networks) Security Current Practices and Limitations (sec 3.2.1) . . . . .	33
7.4.2. (Utility Networks) Security Trends in Utility Networks (sec 3.3.3) . . . . .	34
7.4.3. (BAS) Security Considerations (sec 4.2.4) . . . . .	36
7.4.4. (6TiSCH) Security Considerations (sec 5.3.3) . . . . .	36
7.4.5. (Cellular radio) Security Considerations (sec 6.1.5) . . . . .	36
7.4.6. (Industrial M2M) Communication Today (sec 7.2) . . . . .	37
8. IANA Considerations . . . . .	37
9. Security Considerations . . . . .	37
10. Informative References . . . . .	37
Authors' Addresses . . . . .	38

## 1. Introduction

Security is of particularly high importance in DetNet networks because many of the use cases which are enabled by DetNet [I-D.ietf-detnet-use-cases] include control of physical devices (power grid components, industrial controls, building controls) which can have high operational costs for failure, and present potentially attractive targets for cyber-attackers.

This situation is even more acute given that one of the goals of DetNet is to provide a "converged network", i.e. one that includes

both IT traffic and OT traffic, thus exposing potentially sensitive OT devices to attack in ways that were not previously common (usually because they were under a separate control system or otherwise isolated from the IT network). Security considerations for OT networks is not a new area, and there are many OT networks today that are connected to wide area networks or the Internet; this draft focuses on the issues that are specific to the DetNet technologies and use cases.

The DetNet technologies include ways to:

- o Reserve data plane resources for DetNet flows in some or all of the intermediate nodes (e.g. bridges or routers) along the path of the flow
- o Provide explicit routes for DetNet flows that do not rapidly change with the network topology
- o Distribute data from DetNet flow packets over time and/or space to ensure delivery of each packet's data in spite of the loss of a path

This draft includes sections on threat modeling and analysis, threat impact and mitigation, and the association of attacks with use cases based on the Use Case Common Themes section of the DetNet Use Cases draft [I-D.ietf-detnet-use-cases].

This draft also provides context for the DetNet security considerations by collecting into one place Section 7 the various remarks about security from the various DetNet drafts (Use Cases, Architecture, etc). This text is duplicated here primarily because the DetNet working group has elected not to produce a Requirements draft and thus collectively these statements are as close as we have to "DetNet Security Requirements".

## 2. Abbreviations

IT            Information technology (the application of computers to store, study, retrieve, transmit, and manipulate data or information, often in the context of a business or other enterprise - Wikipedia).

OT            Operational Technology (the hardware and software dedicated to detecting or causing changes in physical processes through direct monitoring and/or control of physical devices such as valves, pumps, etc. - Wikipedia)

MITM        Man in the Middle

SN                    Sequence Number

STRIDE                Addresses risk and severity associated with threat categories: Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, Elevation of privilege.

DREAD                Compares and prioritizes risk represented by these threat categories: Damage potential, Reproducibility, Exploitability, how many Affected users, Discoverability.

PTP                   Precision Time Protocol [IEEE1588]

### 3. Security Threats

This section presents a threat model, and analyzes the possible threats in a DetNet-enabled network.

We distinguish control plane threats from data plane threats. The attack surface may be the same, but the types of attacks as well as the motivation behind them, are different. For example, a delay attack is more relevant to data plane than to control plane. There is also a difference in terms of security solutions: the way you secure the data plane is often different than the way you secure the control plane.

#### 3.1. Threat Model

The threat model used in this memo is based on the threat model of Section 3.1 of [RFC7384]. This model classifies attackers based on two criteria:

- o Internal vs. external: internal attackers either have access to a trusted segment of the network or possess the encryption or authentication keys. External attackers, on the other hand, do not have the keys and have access only to the encrypted or authenticated traffic.
- o Man in the Middle (MITM) vs. packet injector: MITM attackers are located in a position that allows interception and modification of in-flight protocol packets, whereas a traffic injector can only attack by generating protocol packets.

Care has also been taken to adhere to Section 5 of [RFC3552], both with respect to what attacks are considered out-of-scope for this document, but also what is considered to be the most common threats (explored further in Section 3.2. Most of the direct threats to DetNet are Active attacks, but it is highly suggested that DetNet

application developers take appropriate measures to protect the content of the streams from passive attacks.

DetNet-Service, one of the service scenarios described in [I-D.varga-detnet-service-model], is the case where a service connects DetNet networking islands, i.e. two or more otherwise independent DetNet network domains are connected via a link that is not intrinsically part of either network. This implies that there could be DetNet traffic flowing over a non-DetNet link, which may provide an attacker with an advantageous opportunity to tamper with DetNet traffic. The security properties of non-DetNet links are outside of the scope of DetNet Security, but it should be noted that use of non-DetNet services to interconnect DetNet networks merits security analysis to ensure the integrity of the DetNet networks involved.

### 3.2. Threat Analysis

#### 3.2.1. Delay

##### 3.2.1.1. Delay Attack

An attacker can maliciously delay DetNet data flow traffic. By delaying the traffic, the attacker can compromise the service of applications that are sensitive to high delays or to high delay variation.

##### 3.2.2. DetNet Flow Modification or Spoofing

An attacker can modify some header fields of en route packets in a way that causes the DetNet flow identification mechanisms to misclassify the flow. Alternatively, the attacker can inject traffic that is tailored to appear as if it belongs to a legitimate DetNet flow. The potential consequence is that the DetNet flow resource allocation cannot guarantee the performance that is expected when the flow identification works correctly.

##### 3.2.3. Resource Segmentation or Slicing

###### 3.2.3.1. Inter-segment Attack

An attacker can inject traffic, consuming network device resources, thereby affecting DetNet flows. This can be performed using non-DetNet traffic that affects DetNet traffic, or by using DetNet traffic from one DetNet flow that affects traffic from different DetNet flows.

### 3.2.4. Packet Replication and Elimination

#### 3.2.4.1. Replication: Increased Attack Surface

Redundancy is intended to increase the robustness and survivability of DetNet flows, and replication over multiple paths can potentially mitigate an attack that is limited to a single path. However, the fact that packets are replicated over multiple paths increases the attack surface of the network, i.e., there are more points in the network that may be subject to attacks.

#### 3.2.4.2. Replication-related Header Manipulation

An attacker can manipulate the replication-related header fields (R-TAG). This capability opens the door for various types of attacks. For example:

- o Forward both replicas - malicious change of a packet SN (Sequence Number) can cause both replicas of the packet to be forwarded. Note that this attack has a similar outcome to a replay attack.
- o Eliminate both replicas - SN manipulation can be used to cause both replicas to be eliminated. In this case an attacker that has access to a single path can cause packets from other paths to be dropped, thus compromising some of the advantage of path redundancy.
- o Flow hijacking - an attacker can hijack a DetNet flow with access to a single path by systematically replacing the SNs on the given path with higher SN values. For example, an attacker can replace every SN value  $S$  with a higher value  $S+C$ , where  $C$  is a constant integer. Thus, the attacker creates a false illusion that the attacked path has the lowest delay, causing all packets from other paths to be eliminated. Once the flow is hijacked the attacker can either replace en route packets with malicious packets, or simply injecting errors, causing the packets to be dropped at their destination.

### 3.2.5. Path Choice

#### 3.2.5.1. Path Manipulation

An attacker can maliciously change, add, or remove a path, thereby affecting the corresponding DetNet flows that use the path.

#### 3.2.5.2. Path Choice: Increased Attack Surface

One of the possible consequences of a path manipulation attack is an increased attack surface. Thus, when the attack described in the previous subsection is implemented, it may increase the potential of other attacks to be performed.

#### 3.2.6. Control Plane

##### 3.2.6.1. Control or Signaling Packet Modification

An attacker can maliciously modify en route control packets in order to disrupt or manipulate the DetNet path/resource allocation.

##### 3.2.6.2. Control or Signaling Packet Injection

An attacker can maliciously inject control packets in order to disrupt or manipulate the DetNet path/resource allocation.

#### 3.2.7. Scheduling or Shaping

##### 3.2.7.1. Reconnaissance

A passive eavesdropper can identify DetNet flows and then gather information about en route DetNet flows, e.g., the number of DetNet flows, their bandwidths, and their schedules. The gathered information can later be used to invoke other attacks on some or all of the flows.

Note that in some cases DetNet flows may be identified based on an explicit DetNet header, but in some cases the flow identification may be based on fields from the L3/L4 headers. If L3/L4 headers are involved, for purposes of this draft we assume they are encrypted and/or integrity-protected from external attackers.

##### 3.2.8. Time Synchronization Mechanisms

An attacker can use any of the attacks described in [RFC7384] to attack the synchronization protocol, thus affecting the DetNet service.

#### 3.3. Threat Summary

A summary of the attacks that were discussed in this section is presented in Figure 1. For each attack, the table specifies the type of attackers that may invoke the attack. In the context of this summary, the distinction between internal and external attacks is under the assumption that a corresponding security mechanism is being

used, and that the corresponding network equipment takes part in this mechanism.

Attack	Attacker Type			
	Internal MITM	Internal Inj.	External MITM	External Inj.
Delay attack	+		+	
DetNet Flow Modification or Spoofing	+	+		
Inter-segment Attack	+	+		
Replication: Increased Attack Surface	+	+	+	+
Replication-related Header Manipulation	+			
Path Manipulation	+	+		
Path Choice: Increased Attack Surface	+	+	+	+
Control or Signaling Packet Modification	+			
Control or Signaling Packet Injection		+		
Reconnaissance	+		+	
Attacks on Time Sync Mechanisms	+	+	+	+

Figure 1: Threat Analysis Summary

#### 4. Security Threat Impacts

This section describes and rates the impact of the attacks described in Section 3. In this section, the impacts as described assume that the associated mitigation is not present or has failed. Mitigations are discussed in Section 5.

In computer security, the impact (or consequence) of an incident can be measured in loss of confidentiality, integrity or availability of information.

DetNet raises these stakes significantly for OT applications, particularly those which may have been designed to run in an OT-only

environment and thus may not have been designed for security in an IT environment with its associated devices, services and protocols.

The severity of various components of the impact of a successful vulnerability exploit to use cases by industry is available in more detail in [I-D.ietf-detnet-use-cases]. Each of the use cases in the DetNet Use Cases draft is represented in the table below, including Pro Audio, Electrical Utilities, Industrial M2M (split into two areas, M2M Data Gathering and M2M Control Loop), and others.

Components of Impact (left column) include Criticality of Failure, Effects of Failure, Recovery, and DetNet Functional Dependence. Criticality of failure summarizes the seriousness of the impact. The impact of a resulting failure can affect many different metrics that vary greatly in scope and severity. In order to reduce the number of variables, only the following were included: Financial, Health and Safety, People well being, Affect on a single organization, and affect on multiple organizations. Recovery outlines how long it would take for an affected use case to get back to its pre-failure state (Recovery time objective, RTO), and how much of the original service would be lost in between the time of service failure and recovery to original state (Recovery Point Objective, RPO). DetNet dependence maps how much the following DetNet service objectives contribute to impact of failure: Time dependency, data integrity, source node integrity, availability, latency/jitter.

The scale of the Impact mappings is low, medium, and high. In some use cases there may be a multitude of specific applications in which DetNet is used. For simplicity this section attempts to average the varied impacts of different applications. This section does not address the overall risk of a certain impact which would require the likelihood of a failure happening.

In practice any such ratings will vary from case to case; the ratings shown here are given as examples.

Table, Part One (of Two)

	Pro A	Util	Bldg	Wire- less	Cell	M2M Data	M2M Ctrl
Criticality	Med	Hi	Low	Med	Med	Med	Med
Effects							
Financial	Med	Hi	Med	Med	Low	Med	Med



Health/Safety	Med	Hi	Hi	Med	Med	Med	Med
People WB	Med	Hi	Hi	Low	Hi	Low	Low
Effect 1 org	Hi	Hi	Med	Hi	Med	Med	Med
Effect >1 org	Med	Hi	Low	Med	Med	Med	Med
Recovery							
Recov Time Obj	Med	Hi	Med	Hi	Hi	Hi	Hi
Recov Point Obj	Med	Hi	Low	Med	Low	Hi	Hi
DetNet Dependence							
Time Dependency	Hi	Hi	Low	Hi	Med	Low	Hi
Latency/Jitter	Hi	Hi	Med	Med	Low	Low	Hi
Data Integrity	Hi	Hi	Med	Hi	Low	Hi	Low
Src Node Integ	Hi	Hi	Med	Hi	Med	Hi	Hi
Availability	Hi	Hi	Med	Hi	Low	Hi	Hi

Table, Part Two (of Two)

	Mining	Block Chain	Network Slicing
Criticality	Hi	Med	Hi
Effects			
Financial	Hi	Hi	Hi
Health/Safety	Hi	Low	Med
People WB	Hi	Low	Med
Effect 1 org	Hi	Hi	Hi
Effect >1 org	Hi	Low	Hi
Recovery			

Recov Time Obj	Hi	Low	Hi	
+-----+	+-----+	+-----+	+-----+	+-----+
Recov Point Obj	Hi	Low	Hi	
+-----+	+-----+	+-----+	+-----+	+-----+
DetNet Dependence				
+-----+	+-----+	+-----+	+-----+	+-----+
Time Dependency	Hi	Low	Hi	
+-----+	+-----+	+-----+	+-----+	+-----+
Latency/Jitter	Hi	Low	Hi	
+-----+	+-----+	+-----+	+-----+	+-----+
Data Integrity	Hi	Hi	Hi	
+-----+	+-----+	+-----+	+-----+	+-----+
Src Node Integ	Hi	Hi	Hi	
+-----+	+-----+	+-----+	+-----+	+-----+
Availability	Hi	Hi	Hi	
+-----+	+-----+	+-----+	+-----+	+-----+

Figure 2: Impact of Attacks by Use Case Industry

The rest of this section will cover impact of the different groups in more detail.

#### 4.1. Delay-Attacks

##### 4.1.1. Data Plane Delay Attacks

Severely delayed messages in a DetNet link can result in the same behavior as dropped messages in ordinary networks as the services attached to the stream has strict deterministic requirements.

For a single path scenario, disruption is a real possibility, whereas in a multipath scenario, large delays or instabilities in one stream can lead to increased buffer and CPU resources on the elimination bridge.

##### 4.1.2. Control Plane Delay Attacks

In and of itself, this is not directly a threat to the DetNet service, but the effects of delaying control messages can have quite adverse effects later.

- o Delayed tear-down can lead to resource leakage, which in turn can result in failure to allocate new streams finally giving rise to a denial of service attack.

- o Failure to deliver, or severely delaying, signalling messages adding an end-point to a multicast-group will prevent the new EP from receiving expected frames thus disrupting expected behavior.
- o Delaying messages removing an EP from a group can lead to loss of privacy as the EP will continue to receive messages even after it is supposedly removed.

## 4.2. Flow Modification and Spoofing

### 4.2.1. Flow Modification

ToDo.

### 4.2.2. Spoofing

#### 4.2.2.1. Dataplane Spoofing

Spoofing dataplane messages can result in increased resource consumptions on the bridges throughout the network as it will increase buffer usage and CPU utilization. This can lead to resource exhaustion and/or increased delay.

If the attacker manages to create valid headers, the false messages can be forwarded through the network, using part of the allocated bandwidth. This in turn can cause legitimate messages to be dropped when the budget has been exhausted.

Finally, the endpoint will have to deal with invalid messages being delivered to the endpoint instead of (or in addition to) a valid message.

#### 4.2.2.2. Control Plane Spoofing

A successful control plane spoofing-attack will potentially have adverse effects. It can do virtually anything from:

- o modifying existing streams by changing the available bandwidth
- o add or remove endpoints from a stream
- o drop streams completely
- o falsely create new streams (exhaust the systems resources, or to enable streams outside the Network engineer's control)

#### 4.3. Segmentation attacks (injection)

##### 4.3.1. Data Plane Segmentation

Injection of false messages in a DetNet stream could lead to exhaustion of the available bandwidth for a stream if the bridges accounts false messages to the stream's budget.

In a multipath scenario, injected messages will cause increased CPU utilization in elimination bridges. If enough paths are subject to malicious injection, the legitimate messages can be dropped. Likewise it can cause an increase in buffer usage. In total, it will consume more resources in the bridges than normal, giving rise to a resource exhaustion attack on the bridges.

If a stream is interrupted, the end application will be affected by what is now a non-deterministic stream.

##### 4.3.2. Control Plane segmentation

A successful Control Plane segmentation attack control messages to be interpreted by nodes in the network, unbeknownst to the central controller or the network engineer. This has the potential to create

- o new streams (exhausting resources)
- o drop existing (denial of service)
- o add/remove end-stations to a multicast group (loss of privacy)
- o modify the stream attributes (affecting available bandwidth)

#### 4.4. Replication and Elimination

The Replication and Elimination is relevant only to Data Plane messages as Signalling is not subject to multipath routing.

##### 4.4.1. Increased Attack Surface

Covered briefly in Section 4.3

##### 4.4.2. Header Manipulation at Elimination Bridges

Covered briefly in Section 4.3

#### 4.5. Control or Signaling Packet Modification

ToDo.

#### 4.6. Control or Signaling Packet Injection

ToDo.

#### 4.7. Reconnaissance

Of all the attacks, this is one of the most difficult to detect and counter. Often, an attacker will start out by observing the traffic going through the network and use the knowledge gathered in this phase to mount future attacks.

The attacker can, at their leisure, observe over time all aspects of the messaging and signalling, learning the intent and purpose of all traffic flows. At some later date, possibly at an important time in an operational context, the attacker can launch a multi-faceted attack, possibly in conjunction with some demand for ransom.

The flow-id in the header of the data plane-messages gives an attacker a very reliable identifier for DetNet traffic, and this traffic has a high probability of going to lucrative targets.

#### 4.8. Attacks on Time Sync Mechanisms

ToDo.

#### 4.9. Attacks on Path Choice

This is covered in part in Section 4.3, and as with Replication and Elimination (Section 4.4, this is relevant for DataPlane messages.

### 5. Security Threat Mitigation

This section describes a set of measures that can be taken to mitigate the attacks described in Section 3. These mitigations should be viewed as a toolset that includes several different and diverse tools. Each application or system will typically use a subset of these tools, based on a system-specific threat analysis.

#### 5.1. Path Redundancy

##### Description

A DetNet flow that can be forwarded simultaneously over multiple paths. Path replication and elimination

[I-D.ietf-detnet-architecture] provides resiliency to dropped or delayed packets. This redundancy improves the robustness to failures and to man-in-the-middle attacks.

#### Related attacks

Path redundancy can be used to mitigate various man-in-the-middle attacks, including attacks described in Section 3.2.1, Section 3.2.2, Section 3.2.3, and Section 3.2.8.

### 5.2. Integrity Protection

#### Description

An integrity protection mechanism, such as a Hash-based Message Authentication Code (HMAC) can be used to mitigate modification attacks. Integrity protection can be used on the data plane header, to prevent its modification and tampering. Integrity protection in the control plane is discussed in Section 5.5.

#### Related attacks

Integrity protection mitigates attacks related to modification and tampering, including the attacks described in Section 3.2.2 and Section 3.2.4.

### 5.3. DetNet Node Authentication

#### Description

Source authentication verifies the authenticity of DetNet sources, allowing to mitigate spoofing attacks. Note that while integrity protection (Section 5.2) prevents intermediate nodes from modifying information, authentication verifies the source of the information.

#### Related attacks

DetNet node authentication is used to mitigate attacks related to spoofing, including the attacks of Section 3.2.2, and Section 3.2.4.

### 5.4. Encryption

#### Description

DetNet flows can be forwarded in encrypted form.

#### Related attacks

While confidentiality is not considered an important goal with respect to DetNet, encryption can be used to mitigate recon attacks (Section 3.2.7).

### 5.5. Control and Signaling Message Protection

#### Description

Control and signaling messages can be protected using authentication and integrity protection mechanisms.

#### Related attacks

These mechanisms can be used to mitigate various attacks on the control plane, as described in Section 3.2.6, Section 3.2.8 and Section 3.2.5.

### 5.6. Dynamic Performance Analytics

#### Description

Information about the network performance can be gathered in real-time in order to detect anomalies and unusual behavior that may be the symptom of a security attack. The gathered information can be based, for example, on per-flow counters, bandwidth measurement, and monitoring of packet arrival times. Unusual behavior or potentially malicious nodes can be reported to a management system, or can be used as a trigger for taking corrective actions. The information can be tracked by DetNet end systems and transit nodes, and exported to a management system, for example using NETCONF.

#### Related attacks

Performance analytics can be used to mitigate various attacks, including the ones described in Section 3.2.1, Section 3.2.3, and Section 3.2.8.

### 5.7. Mitigation Summary

The following table maps the attacks of Section 3 to the impacts of Section 4, and to the mitigations of the current section. Each row specifies an attack, the impact of this attack if it is successfully implemented, and possible mitigation methods.

Attack	Impact	Mitigations
Delay Attack	-Non-deterministic delay -Data disruption -Increased resource consumption	-Path redundancy -Performance analytics
Reconnaissance	-Enabler for other attacks	-Encryption
DetNet Flow Modification or Spoofing	-Increased resource consumption -Data disruption	-Path redundancy -Integrity protection -DetNet Node authentication
Inter-Segment Attack	-Increased resource consumption -Data disruption	-Path redundancy -Performance analytics
Replication: Increased attack surface	-All impacts of other attacks	-Integrity protection -DetNet Node authentication
Replication-related Header Manipulation	-Non-deterministic delay -Data disruption	-Integrity protection -DetNet Node authentication
Path Manipulation	-Enabler for other attacks	-Control message protection
Path Choice: Increased Attack Surface	-All impacts of other attacks	-Control message protection
Control or Signaling Packet Modification	-Increased resource consumption -Non-deterministic delay -Data disruption	-Control message protection
Control or Signaling Packet Injection	-Increased resource consumption -Non-deterministic delay -Data disruption	-Control message protection
Attacks on Time Sync	-Non-deterministic	-Path redundancy



Mechanisms	delay	-Control message
	-Increased resource	protection
	consumption	-Performance
	-Data disruption	analytics

Figure 3: Mapping Attacks to Impact and Mitigations

## 6. Association of Attacks to Use Cases

Different attacks can have different impact and/or mitigation depending on the use case, so we would like to make this association in our analysis. However since there is a potentially unbounded list of use cases, we categorize the attacks with respect to the common themes of the use cases as identified in the Use Case Common Themes section of the DetNet Use Cases draft [I-D.ietf-detnet-use-cases].

See also Figure 2 for a mapping of the impact of attacks per use case by industry.

### 6.1. Use Cases by Common Themes

In this section we review each theme and discuss the attacks that are applicable to that theme, as well as anything specific about the impact and mitigations for that attack with respect to that theme. The table Figure 5 then provides a summary of the attacks that are applicable to each theme.

#### 6.1.1. Network Layer - AVB/TSN Ethernet

DetNet is expected to run over various transmission mediums, with Ethernet being explicitly supported. Attacks such as Delay or Reconnaissance might be implemented differently on a different transmission medium, however the impact on the DetNet as a whole would be essentially the same. We thus conclude that all attacks and impacts that would be applicable to DetNet over Ethernet (i.e. all those named in this draft) would also be applicable to DetNet over other transmission mediums.

With respect to mitigations, some methods are specific to the Ethernet medium, for example time-aware scheduling using 802.1Qbv can protect against excessive use of bandwidth at the ingress - for other mediums, other mitigations would have to be implemented to provide analogous protection.

### 6.1.2. Central Administration

A DetNet network is expected to be controlled by a centralized network configuration and control system (CNC). Such a system may be in a single central location, or it may be distributed across multiple control entities that function together as a unified control system for the network.

In this draft we distinguish between attacks on the DetNet Control plane vs. Data plane. But is an attack affecting control plane packets synonymous with an attack on the CNC itself? For purposes of this draft let us consider an attack on the CNC itself to be out of scope, and consider all attacks named in this draft which are relevant to control plane packets to be relevant to this theme, including Path Manipulation, Path Choice, Control Packet Modification or Injection, Reconnaissance and Attacks on Time Sync Mechanisms.

### 6.1.3. Hot Swap

A DetNet network is not expected to be "plug and play" - it is expected that there is some centralized network configuration and control system. However, the ability to "hot swap" components (e.g. due to malfunction) is similar enough to "plug and play" that this kind of behavior may be expected in DetNet networks, depending on the implementation.

An attack surface related to Hot Swap is that the DetNet network must at least consider input at runtime from devices that were not part of the initial configuration of the network. Even a "perfect" (or "hitless") replacement of a device at runtime would not necessarily be ideal, since presumably one would want to distinguish it from the original for OAM purposes (e.g. to report hot swap of a failed device).

This implies that an attack such as Flow Modification, Spoofing or Inter-segment (which could introduce packets from a "new" device (i.e. one heretofore unknown on the network) could be used to exploit the need to consider such packets (as opposed to rejecting them out of hand as one would do if one did not have to consider introduction of a new device).

Similarly if the network was designed to support runtime replacement of a clock device, then presence (or apparent presence) and thus consideration of packets from a new such device could affect the network, or the time sync of the network, for example by initiating a new Best Master Clock selection process. Thus attacks on time sync should be considered when designing hot swap type functionality.

#### 6.1.4. Data Flow Information Models

Data Flow Information Models specific to DetNet networks are to be specified by DetNet. Thus they are "new" and thus potentially present a new attack surface. Does the threat take advantage of any aspect of our new Data Flow Info Models?

This is TBD, thus there are no specific entries in our table, however that does not imply that there could be no relevant attacks.

#### 6.1.5. L2 and L3 Integration

A DetNet network integrates Layer 2 (bridged) networks (e.g. AVB/TSN LAN) and Layer 3 (routed) networks via the use of well-known protocols such as IPv6, MPLS-PW, and Ethernet. Presumably security considerations applicable directly to those individual protocols is not specific to DetNet, and thus out of scope for this draft. However enabling DetNet to coordinate Layer 2 and Layer 3 behavior will require some additions to existing protocols (see draft-dt-detnet-dp-alt) and any such new work can introduce new attack surfaces.

This is TBD, thus there are no specific entries in our table, however that does not imply that there could be no relevant attacks.

#### 6.1.6. End-to-End Delivery

Packets sent over DetNet are guaranteed not to be dropped by the network due to congestion. (Packets may however be dropped for intended reasons, e.g. per security measures).

A Data plane attack may force packets to be dropped, for example a "long" Delay or Replication/Elimination or Flow Modification attack.

The same result might be obtained by a Control plane attack, e.g. Path Manipulation or Signaling Packet Modification.

It may be that such attacks are limited to Internal MITM attackers, but other possibilities should be considered.

An attack may also cause packets that should not be delivered to be delivered, such as by forcing packets from one (e.g. replicated) path to be preferred over another path when they should not be (Replication attack), or by Flow Modification, or by Path Choice or Packet Injection. A Time Sync attack could cause a system that was expecting certain packets at certain times to accept unintended packets based on compromised system time or time windowing in the scheduler.

#### 6.1.1.7. Proprietary Deterministic Ethernet Networks

There are many proprietary non-interoperable deterministic Ethernet-based networks currently available; DetNet is intended to provide an open-standards-based alternative to such networks. In cases where a DetNet intersects with remnants of such networks or their protocols, such as by protocol emulation or access to such a network via a gateway, new attack surfaces can be opened.

For example an Inter-Segment or Control plane attack such as Path Manipulation, Path Choice or Control Packet Modification/Injection could be used to exploit commands specific to such a protocol, or that are interpreted differently by the different protocols or gateway.

#### 6.1.1.8. Replacement for Proprietary Fieldbuses

There are many proprietary "field buses" used in today's industrial and other industries; DetNet is intended to provide an open-standards-based alternative to such buses. In cases where a DetNet intersects with such fieldbuses or their protocols, such as by protocol emulation or access via a gateway, new attack surfaces can be opened.

For example an Inter-Segment or Control plane attack such as Path Manipulation, Path Choice or Control Packet Modification/Injection could be used to exploit commands specific to such a protocol, or that are interpreted differently by the different protocols or gateway.

#### 6.1.1.9. Deterministic vs Best-Effort Traffic

DetNet is intended to support coexistence of time-sensitive operational (OT, deterministic) traffic and information (IT, "best effort") traffic on the same ("unified") network.

The presence of IT traffic on a network carrying OT traffic has long been considered insecure design [reference needed here]. With DetNet, this coexistence will become more common, and mitigations will need to be established. The fact that the IT traffic on a DetNet is limited to a corporate controlled network makes this a less difficult problem compared to being exposed to the open Internet, however this aspect of DetNet security should not be underestimated.

Most of the themes described in this draft address OT (reserved) streams - this item is intended to address issues related to IT traffic on a DetNet.

An Inter-segment attack can flood the network with IT-type traffic with the intent of disrupting handling of IT traffic, and/or the goal of interfering with OT traffic. Presumably if the stream reservation and isolation of the DetNet is well-designed (better-designed than the attack) then interference with OT traffic should not result from an attack that floods the network with IT traffic.

However the DetNet's handling of IT traffic may not (by design) be as resilient to DOS attack, and thus designers must be otherwise prepared to mitigate DOS attacks on IT traffic in a DetNet.

#### 6.1.10. Deterministic Flows

Reserved bandwidth data flows (deterministic flows) must provide the allocated bandwidth, and must be isolated from each other.

A Spoofing or Inter-segment attack which adds packet traffic to a bandwidth-reserved stream could cause that stream to occupy more bandwidth than it is allocated, resulting in interference with other deterministic flows.

A Flow Modification or Spoofing or Header Manipulation or Control Packet Modification attack could cause packets from one flow to be directed to another flow, thus breaching isolation between the flows.

#### 6.1.11. Unused Reserved Bandwidth

If bandwidth reservations are made for a stream but the associated bandwidth is not used at any point in time, that bandwidth is made available on the network for best-effort traffic. If the owner of the reserved stream then starts transmitting again, the bandwidth is no longer available for best-effort traffic, on a moment-to-moment basis. (Such "temporarily available" bandwidth is not available for time-sensitive traffic, which must have its own reservation).

An Inter-segment attack could flood the network with IT traffic, interfering with the intended IT traffic.

A Flow Modification or Spoofing or Control Packet Modification or Injection attack could cause extra bandwidth to be reserved by a new or existing stream, thus making it unavailable for use by best-effort traffic.

#### 6.1.12. Interoperability

The DetNet network specifications are intended to enable an ecosystem in which multiple vendors can create interoperable products, thus promoting device diversity and potentially higher numbers of each

device manufactured. Does the threat take advantage of differences in implementation of "interoperable" products made by different vendors?

This is TBD, thus there are no specific entries in our table, however that does not imply that there could be no relevant attacks.

#### 6.1.13. Cost Reductions

The DetNet network specifications are intended to enable an ecosystem in which multiple vendors can create interoperable products, thus promoting higher numbers of each device manufactured, promoting cost reduction and cost competition among vendors. Does the threat take advantage of "low cost" HW or SW components or other "cost-related shortcuts" that might be present in devices?

This is TBD, thus there are no specific entries in our table, however that does not imply that there could be no relevant attacks.

#### 6.1.14. Insufficiently Secure Devices

The DetNet network specifications are intended to enable an ecosystem in which multiple vendors can create interoperable products, thus promoting device diversity and potentially higher numbers of each device manufactured. Does the threat attack "naivete" of SW, for example SW that was not designed to be sufficiently secure (or secure at all) but is deployed on a DetNet network that is intended to be highly secure? (For example IoT exploits like the Mirai video-camera botnet ([MIRAI])).

This is TBD, thus there are no specific entries in our table, however that does not imply that there could be no relevant attacks.

#### 6.1.15. DetNet Network Size

DetNet networks range in size from very small, e.g. inside a single industrial machine, to very large, for example a Utility Grid network spanning a whole country.

The size of the network might be related to how the attack is introduced into the network, for example if the entire network is local, there is a threat that power can be cut to the entire network. If the network is large, perhaps only a part of the network is attacked.

A Delay attack might be as relevant to a small network as to a large network, although the amount of delay might be different.

Attacks sourced from IT traffic might be more likely in large networks, since more people might have access to the network. Similarly Path Manipulation, Path Choice and Time Sync attacks seem more likely relevant to large networks.

#### 6.1.16. Multiple Hops

Large DetNet networks (e.g. a Utility Grid network) may involve many "hops" over various kinds of links for example radio repeaters, microwave links, fiber optic links, etc..

An attack that takes advantage of flaws (or even normal operation) in the device drivers for the various links (through internal knowledge of how the individual driver or firmware operates, perhaps like the Stuxnet attack) could take proportionately greater advantage of this topology. We don't currently have an attack like this defined; we have only "protocol" (time or packet) based attacks. Perhaps we need to define an attack like this? Or is that out of scope for DetNet?

It is also possible that this DetNet topology will not be in as common use as other more homogeneous topologies so there may be more opportunity for attackers to exploit software and/or protocol flaws in the implementations which have not been wrung out by extensive use, particularly in the case of early adopters.

Of the attacks we have defined, the ones identified above as relevant to "large" networks seem to be most relevant.

#### 6.1.17. Level of Service

A DetNet is expected to provide means to configure the network that include querying network path latency, requesting bounded latency for a given stream, requesting worst case maximum and/or minimum latency for a given path or stream, and so on. It is an expected case that the network cannot provide a given requested service level. In such cases the network control system should reply that the requested service level is not available (as opposed to accepting the parameter but then not delivering the desired behavior).

Control plane attacks such as Signaling Packet Modification and Injection could be used to modify or create control traffic that could interfere with the process of a user requesting a level of service and/or the network's reply.

Reconnaissance could be used to characterize flows and perhaps target specific flows for attack via the Control plane as noted above.

#### 6.1.18. Bounded Latency

DetNet provides the expectation of guaranteed bounded latency.

Delay attacks can cause packets to miss their agreed-upon latency boundaries.

Time Sync attacks can corrupt the system's time reference, resulting in missed latency deadlines (with respect to the "correct" time reference).

#### 6.1.19. Low Latency

Applications may require "extremely low latency" however depending on the application these may mean very different latency values; for example "low latency" across a Utility grid network is on a different time scale than "low latency" in a motor control loop in a small machine. The intent is that the mechanisms for specifying desired latency include wide ranges, and that architecturally there is nothing to prevent arbitrarily low latencies from being implemented in a given network.

Attacks on the Control plane (as described in the Level of Service theme) and Delay and Time attacks (as described in the Bounded Latency theme) both apply here.

#### 6.1.20. Symmetrical Path Delays

Some applications would like to specify that the transit delay time values be equal for both the transmit and return paths.

Delay attacks can cause path delays to differ.

Time Sync attacks can corrupt the system's time reference, resulting in differing path delays (with respect to the "correct" time reference).

#### 6.1.21. Reliability and Availability

DetNet based systems are expected to be implemented with essentially arbitrarily high availability (for example 99.9999% up time, or even 12 nines). The intent is that the DetNet designs should not make any assumptions about the level of reliability and availability that may be required of a given system, and should define parameters for communicating these kinds of metrics within the network.

Any attack on the system, of any type, can affect its overall reliability and availability, thus in our table we have marked every



attack. Since every DetNet depends to a greater or lesser degree on reliability and availability, this essentially means that all networks have to mitigate all attacks, which to a greater or lesser degree defeats the purpose of associating attacks with use cases. It also underscores the difficulty of designing "extremely high reliability" networks. I hope that in future drafts we can say something more useful here.

#### 6.1.22. Redundant Paths

DetNet based systems are expected to be implemented with essentially arbitrarily high reliability/availability. A strategy used by DetNet for providing such extraordinarily high levels of reliability is to provide redundant paths that can be seamlessly switched between, all the while maintaining the required performance of that system.

Replication-related attacks are by definition applicable here. Control plane attacks can also interfere with the configuration of redundant paths.

#### 6.1.23. Security Measures

A DetNet network must be made secure against devices failures, attackers, misbehaving devices, and so on. Does the threat affect such security measures themselves, e.g. by attacking SW designed to protect against device failure?

This is TBD, thus there are no specific entries in our table, however that does not imply that there could be no relevant attacks.

#### 6.2. Attack Types by Use Case Common Theme

The following table lists the attacks of Section 3, assigning a number to each type of attack. That number is then used as a short form identifier for the attack in Figure 5.

Attack	Section
1 Delay Attack	Section 3.2.1
2 DetNet Flow Modification or Spoofing	Section 3.2.2
3 Inter-Segment Attack	Section 3.2.3
4 Replication: Increased attack surface	Section 3.2.4.1
5 Replication-related Header Manipulation	Section 3.2.4.2
6 Path Manipulation	Section 3.2.5.1
7 Path Choice: Increased Attack Surface	Section 3.2.5.2
8 Control or Signaling Packet Modification	Section 3.2.6.1
9 Control or Signaling Packet Injection	Section 3.2.6.2
10 Reconnaissance	Section 3.2.7
11 Attacks on Time Sync Mechanisms	Section 3.2.8

Figure 4: List of Attacks

The following table maps the use case themes presented in this memo to the attacks of Figure 4. Each row specifies a theme, and the attacks relevant to this theme are marked with a '+'.

Theme	Attack										
	1	2	3	4	5	6	7	8	9	10	11
Network Layer - AVB/TSN Eth.	+	+	+	+	+	+	+	+	+	+	+
Central Administration						+	+	+	+	+	+
Hot Swap		+	+								+
Data Flow Information Models											
L2 and L3 Integration											

[illegible]

Figure 5: Mapping Between Themes and Attacks

## 7. Appendix A: DetNet Draft Security-Related Statements

This section collects the various statements in the currently existing DetNet Working Group drafts. For each draft, the section name and number of the quoted section is shown. The text shown here

is the work of the original draft authors, quoted verbatim from the drafts. The intention is to explicitly quote all relevant text, not to summarize it.

## 7.1. Architecture (draft 8)

### 7.1.1. Fault Mitigation (sec 4.5)

One key to building robust real-time systems is to reduce the infinite variety of possible failures to a number that can be analyzed with reasonable confidence. DetNet aids in the process by providing filters and policers to detect DetNet packets received on the wrong interface, or at the wrong time, or in too great a volume, and to then take actions such as discarding the offending packet, shutting down the offending DetNet flow, or shutting down the offending interface.

It is also essential that filters and service remarking be employed at the network edge to prevent non-DetNet packets from being mistaken for DetNet packets, and thus impinging on the resources allocated to DetNet packets.

There exist techniques, at present and/or in various stages of standardization, that can perform these fault mitigation tasks that deliver a high probability that misbehaving systems will have zero impact on well-behaved DetNet flows, except of course, for the receiving interface(s) immediately downstream of the misbehaving device. Examples of such techniques include traffic policing functions (e.g. [RFC2475]) and separating flows into per-flow rate-limited queues.

### 7.1.2. Security Considerations (sec 7)

Security in the context of Deterministic Networking has an added dimension; the time of delivery of a packet can be just as important as the contents of the packet, itself. A man-in-the-middle attack, for example, can impose, and then systematically adjust, additional delays into a link, and thus disrupt or subvert a real-time application without having to crack any encryption methods employed. See [RFC7384] for an exploration of this issue in a related context.

Furthermore, in a control system where millions of dollars of equipment, or even human lives, can be lost if the DetNet QoS is not delivered, one must consider not only simple equipment failures, where the box or wire instantly becomes perfectly silent, but bizarre errors such as can be caused by software failures. Because there is essential no limit to the kinds of failures that can occur, protecting against realistic equipment failures is indistinguishable,

in most cases, from protecting against malicious behavior, whether accidental or intentional.

Security must cover:

- o Protection of the signaling protocol
- o Authentication and authorization of the controlling nodes
- o Identification and shaping of the flows

## 7.2. Data Plane Alternatives (draft 4)

### 7.2.1. Security Considerations (sec 7)

This document does not add any new security considerations beyond what the referenced technologies already have.

## 7.3. Problem Statement (draft 5)

### 7.3.1. Security Considerations (sec 5)

Security in the context of Deterministic Networking has an added dimension; the time of delivery of a packet can be just as important as the contents of the packet, itself. A man-in-the-middle attack, for example, can impose, and then systematically adjust, additional delays into a link, and thus disrupt or subvert a real-time application without having to crack any encryption methods employed. See [RFC7384] for an exploration of this issue in a related context.

Typical control networks today rely on complete physical isolation to prevent rogue access to network resources. DetNet enables the virtualization of those networks over a converged IT/OT infrastructure. Doing so, DetNet introduces an additional risk that flows interact and interfere with one another as they share physical resources such as Ethernet trunks and radio spectrum. The requirement is that there is no possible data leak from and into a deterministic flow, and in a more general fashion there is no possible influence whatsoever from the outside on a deterministic flow. The expectation is that physical resources are effectively associated with a given flow at a given point of time. In that model, Time Sharing of physical resources becomes transparent to the individual flows which have no clue whether the resources are used by other flows at other times.

Security must cover:

- o Protection of the signaling protocol

- o Authentication and authorization of the controlling nodes
- o Identification and shaping of the flows
- o Isolation of flows from leakage and other influences from any activity sharing physical resources

#### 7.4. Use Cases (draft 11)

##### 7.4.1. (Utility Networks) Security Current Practices and Limitations (sec 3.2.1)

Grid monitoring and control devices are already targets for cyber attacks, and legacy telecommunications protocols have many intrinsic network-related vulnerabilities. For example, DNP3, Modbus, PROFIBUS/PROFINET, and other protocols are designed around a common paradigm of request and respond. Each protocol is designed for a master device such as an HMI (Human Machine Interface) system to send commands to subordinate slave devices to retrieve data (reading inputs) or control (writing to outputs). Because many of these protocols lack authentication, encryption, or other basic security measures, they are prone to network-based attacks, allowing a malicious actor or attacker to utilize the request-and-respond system as a mechanism for command-and-control like functionality. Specific security concerns common to most industrial control, including utility telecommunication protocols include the following:

- o Network or transport errors (e.g. malformed packets or excessive latency) can cause protocol failure.
- o Protocol commands may be available that are capable of forcing slave devices into inoperable states, including powering-off devices, forcing them into a listen-only state, disabling alarming.
- o Protocol commands may be available that are capable of restarting communications and otherwise interrupting processes.
- o Protocol commands may be available that are capable of clearing, erasing, or resetting diagnostic information such as counters and diagnostic registers.
- o Protocol commands may be available that are capable of requesting sensitive information about the controllers, their configurations, or other need-to-know information.

- o Most protocols are application layer protocols transported over TCP; therefore it is easy to transport commands over non-standard ports or inject commands into authorized traffic flows.
- o Protocol commands may be available that are capable of broadcasting messages to many devices at once (i.e. a potential DoS).
- o Protocol commands may be available to query the device network to obtain defined points and their values (i.e. a configuration scan).
- o Protocol commands may be available that will list all available function codes (i.e. a function scan).
- o These inherent vulnerabilities, along with increasing connectivity between IT and OT networks, make network-based attacks very feasible.
- o Simple injection of malicious protocol commands provides control over the target process. Altering legitimate protocol traffic can also alter information about a process and disrupt the legitimate controls that are in place over that process. A man-in-the-middle attack could provide both control over a process and misrepresentation of data back to operator consoles.

#### 7.4.2. (Utility Networks) Security Trends in Utility Networks (sec 3.3.3)

Although advanced telecommunications networks can assist in transforming the energy industry by playing a critical role in maintaining high levels of reliability, performance, and manageability, they also introduce the need for an integrated security infrastructure. Many of the technologies being deployed to support smart grid projects such as smart meters and sensors can increase the vulnerability of the grid to attack. Top security concerns for utilities migrating to an intelligent smart grid telecommunications platform center on the following trends:

- o Integration of distributed energy resources
- o Proliferation of digital devices to enable management, automation, protection, and control
- o Regulatory mandates to comply with standards for critical infrastructure protection

- o Migration to new systems for outage management, distribution automation, condition-based maintenance, load forecasting, and smart metering
- o Demand for new levels of customer service and energy management

This development of a diverse set of networks to support the integration of microgrids, open-access energy competition, and the use of network-controlled devices is driving the need for a converged security infrastructure for all participants in the smart grid, including utilities, energy service providers, large commercial and industrial, as well as residential customers. Securing the assets of electric power delivery systems (from the control center to the substation, to the feeders and down to customer meters) requires an end-to-end security infrastructure that protects the myriad of telecommunications assets used to operate, monitor, and control power flow and measurement.

"Cyber security" refers to all the security issues in automation and telecommunications that affect any functions related to the operation of the electric power systems. Specifically, it involves the concepts of:

- o Integrity : data cannot be altered undetectably
- o Authenticity : the telecommunications parties involved must be validated as genuine
- o Authorization : only requests and commands from the authorized users can be accepted by the system
- o Confidentiality : data must not be accessible to any unauthenticated users

When designing and deploying new smart grid devices and telecommunications systems, it is imperative to understand the various impacts of these new components under a variety of attack situations on the power grid. Consequences of a cyber attack on the grid telecommunications network can be catastrophic. This is why security for smart grid is not just an ad hoc feature or product, it's a complete framework integrating both physical and Cyber security requirements and covering the entire smart grid networks from generation to distribution. Security has therefore become one of the main foundations of the utility telecom network architecture and must be considered at every layer with a defense-in-depth approach. Migrating to IP based protocols is key to address these challenges for two reasons:



- o IP enables a rich set of features and capabilities to enhance the security posture
- o IP is based on open standards, which allows interoperability between different vendors and products, driving down the costs associated with implementing security solutions in OT networks.

Securing OT (Operation technology) telecommunications over packet-switched IP networks follow the same principles that are foundational for securing the IT infrastructure, i.e., consideration must be given to enforcing electronic access control for both person-to-machine and machine-to-machine communications, and providing the appropriate levels of data privacy, device and platform integrity, and threat detection and mitigation.

#### 7.4.3. (BAS) Security Considerations (sec 4.2.4)

When BAS field networks were developed it was assumed that the field networks would always be physically isolated from external networks and therefore security was not a concern. In today's world many BASs are managed remotely and are thus connected to shared IP networks and so security is definitely a concern, yet security features are not available in the majority of BAS field network deployments .

The management network, being an IP-based network, has the protocols available to enable network security, but in practice many BAS systems do not implement even the available security features such as device authentication or encryption for data in transit.

#### 7.4.4. (6TiSCH) Security Considerations (sec 5.3.3)

On top of the classical requirements for protection of control signaling, it must be noted that 6TiSCH networks operate on limited resources that can be depleted rapidly in a DoS attack on the system, for instance by placing a rogue device in the network, or by obtaining management control and setting up unexpected additional paths.

#### 7.4.5. (Cellular radio) Security Considerations (sec 6.1.5)

Establishing time-sensitive streams in the network entails reserving networking resources for long periods of time. It is important that these reservation requests be authenticated to prevent malicious reservation attempts from hostile nodes (or accidental misconfiguration). This is particularly important in the case where the reservation requests span administrative domains. Furthermore, the reservation information itself should be digitally signed to

reduce the risk of a legitimate node pushing a stale or hostile configuration into another networking node.

Note: This is considered important for the security policy of the network, but does not affect the core DetNet architecture and design.

#### 7.4.6. (Industrial M2M) Communication Today (sec 7.2)

Industrial network scenarios require advanced security solutions. Many of the current industrial production networks are physically separated. Preventing critical flows from be leaked outside a domain is handled today by filtering policies that are typically enforced in firewalls.

### 8. IANA Considerations

This memo includes no requests from IANA.

### 9. Security Considerations

The security considerations of DetNet networks are presented throughout this document.

### 10. Informative References

[ARINC664P7]

ARINC, "ARINC 664 Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network", 2009.

[I-D.ietf-detnet-architecture]

Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-03 (work in progress), August 2017.

[I-D.ietf-detnet-use-cases]

Grossman, E., Gunther, C., Thubert, P., Wetterwald, P., Raymond, J., Korhonen, J., Kaneko, Y., Das, S., Zha, Y., Varga, B., Farkas, J., Goetz, F., Schmitt, J., Vilajosana, X., Mahmoodi, T., Spirou, S., Vizarrata, P., Huang, D., Geng, X., Dujovne, D., and M. Seewald, "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-13 (work in progress), September 2017.

[I-D.varga-detnet-service-model]

Varga, B. and J. Farkas, "DetNet Service Model", draft-varga-detnet-service-model-02 (work in progress), May 2017.

- [IEEE1588] IEEE, "IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", 2008.
- [MIRAI] krebsonsecurity.com, "<https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/>", 2016.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.
- [RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <<https://www.rfc-editor.org/info/rfc7384>>.

## Authors' Addresses

Tal Mizrahi  
Marvell

Email: [talmi@marvell.com](mailto:talmi@marvell.com)

Ethan Grossman (editor)  
Dolby Laboratories, Inc.  
1275 Market Street  
San Francisco, CA 94103  
USA

Phone: +1 415 645 4726  
Email: [ethan.grossman@dolby.com](mailto:ethan.grossman@dolby.com)  
URI: <http://www.dolby.com>

Andrew J. Hacker  
MistIQ Technologies, Inc  
Harrisburg, PA  
USA

Email: [ajhacker@mistiqttech.com](mailto:ajhacker@mistiqttech.com)  
URI: <http://www.mistiqttech.com>

Subir Das  
Applied Communication Sciences  
150 Mount Airy Road, Basking Ridge  
New Jersey, 07920  
USA

Email: [sdas@appcomsci.com](mailto:sdas@appcomsci.com)

John Dowdell  
Airbus Defence and Space  
Celtic Springs  
Newport NP10 8FZ  
United Kingdom

Email: [john.dowdell.ietf@gmail.com](mailto:john.dowdell.ietf@gmail.com)

Henrik Austad  
Cisco Systems  
Philip Pedersens vei 1  
Lysaker 1366  
Norway

Email: [henrik@austad.us](mailto:henrik@austad.us)

Kevin Stanton  
Intel

Email: [kevin.b.stanton@intel.com](mailto:kevin.b.stanton@intel.com)

Norman Finn  
Huawei

Email: [norman.finn@mail01.huawei.com](mailto:norman.finn@mail01.huawei.com)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: September 3, 2021

E. Grossman, Ed.  
DOLBY  
T. Mizrahi  
HUAWEI  
A. Hacker  
MISTIQ  
March 2, 2021

Deterministic Networking (DetNet) Security Considerations  
draft-ietf-detnet-security-16

Abstract

A DetNet (deterministic network) provides specific performance guarantees to its data flows, such as extremely low data loss rates and bounded latency (including bounded latency variation, i.e. "jitter"). As a result, securing a DetNet requires that in addition to the best practice security measures taken for any mission-critical network, additional security measures may be needed to secure the intended operation of these novel service properties.

This document addresses DetNet-specific security considerations from the perspectives of both the DetNet system-level designer and component designer. System considerations include a taxonomy of relevant threats and attacks, and associations of threats versus use cases and service properties. Component-level considerations include ingress filtering and packet arrival time violation detection.

This document also addresses security considerations specific to the IP and MPLS data plane technologies, thereby complementing the Security Considerations sections of those documents.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 3, 2021.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
2. Abbreviations and Terminology . . . . .	7
3. Security Considerations for DetNet Component Design . . . . .	8
3.1. Resource Allocation . . . . .	8
3.1.1. Inviolable Flows . . . . .	8
3.1.2. Design Trade-Off Considerations in the Use Cases Continuum . . . . .	9
3.1.3. Documenting the Security Properties of a Component . . . . .	10
3.1.4. Fail-Safe Component Behavior . . . . .	10
3.1.5. Flow Aggregation Example . . . . .	10
3.2. Explicit Routes . . . . .	11
3.3. Redundant Path Support . . . . .	11
3.4. Timing (or other) Violation Reporting . . . . .	12
4. DetNet Security Considerations Compared With DiffServ Security Considerations . . . . .	13
5. Security Threats . . . . .	14
5.1. Threat Taxonomy . . . . .	15
5.2. Threat Analysis . . . . .	16
5.2.1. Delay . . . . .	16
5.2.2. DetNet Flow Modification or Spoofing . . . . .	16
5.2.3. Resource Segmentation (Inter-segment Attack) Vulnerability . . . . .	16
5.2.4. Packet Replication and Elimination . . . . .	17
5.2.4.1. Replication: Increased Attack Surface . . . . .	17
5.2.4.2. Replication-related Header Manipulation . . . . .	17
5.2.5. Controller Plane . . . . .	18
5.2.5.1. Path Choice Manipulation . . . . .	18
5.2.5.2. Compromised Controller . . . . .	18
5.2.6. Reconnaissance . . . . .	19

5.2.7. Time Synchronization Mechanisms . . . . .	19
5.3. Threat Summary . . . . .	19
6. Security Threat Impacts . . . . .	20
6.1. Delay-Attacks . . . . .	23
6.1.1. Data Plane Delay Attacks . . . . .	23
6.1.2. Controller Plane Delay Attacks . . . . .	23
6.2. Flow Modification and Spoofing . . . . .	23
6.2.1. Flow Modification . . . . .	24
6.2.2. Spoofing . . . . .	24
6.2.2.1. Dataplane Spoofing . . . . .	24
6.2.2.2. Controller Plane Spoofing . . . . .	24
6.3. Segmentation Attacks (injection) . . . . .	24
6.3.1. Data Plane Segmentation . . . . .	25
6.3.2. Controller Plane Segmentation . . . . .	25
6.4. Replication and Elimination . . . . .	25
6.4.1. Increased Attack Surface . . . . .	26
6.4.2. Header Manipulation at Elimination Routers . . . . .	26
6.5. Control or Signaling Packet Modification . . . . .	26
6.6. Control or Signaling Packet Injection . . . . .	26
6.7. Reconnaissance . . . . .	26
6.8. Attacks on Time Synchronization Mechanisms . . . . .	27
6.9. Attacks on Path Choice . . . . .	27
7. Security Threat Mitigation . . . . .	27
7.1. Path Redundancy . . . . .	27
7.2. Integrity Protection . . . . .	28
7.3. DetNet Node Authentication . . . . .	29
7.4. Dummy Traffic Insertion . . . . .	30
7.5. Encryption . . . . .	31
7.5.1. Encryption Considerations for DetNet . . . . .	32
7.6. Control and Signaling Message Protection . . . . .	33
7.7. Dynamic Performance Analytics . . . . .	33
7.8. Mitigation Summary . . . . .	36
8. Association of Attacks to Use Cases . . . . .	37
8.1. Association of Attacks to Use Case Common Themes . . . . .	38
8.1.1. Sub-Network Layer . . . . .	38
8.1.2. Central Administration . . . . .	38
8.1.3. Hot Swap . . . . .	38
8.1.4. Data Flow Information Models . . . . .	39
8.1.5. L2 and L3 Integration . . . . .	39
8.1.6. End-to-End Delivery . . . . .	40
8.1.7. Replacement for Proprietary Fieldbuses and Ethernet- based Networks . . . . .	40
8.1.8. Deterministic vs Best-Effort Traffic . . . . .	41
8.1.9. Deterministic Flows . . . . .	42
8.1.10. Unused Reserved Bandwidth . . . . .	42
8.1.11. Interoperability . . . . .	42
8.1.12. Cost Reductions . . . . .	43
8.1.13. Insufficiently Secure Components . . . . .	43

8.1.14. DetNet Network Size . . . . .	43
8.1.15. Multiple Hops . . . . .	44
8.1.16. Level of Service . . . . .	44
8.1.17. Bounded Latency . . . . .	45
8.1.18. Low Latency . . . . .	45
8.1.19. Bounded Jitter (Latency Variation) . . . . .	45
8.1.20. Symmetrical Path Delays . . . . .	45
8.1.21. Reliability and Availability . . . . .	46
8.1.22. Redundant Paths . . . . .	46
8.1.23. Security Measures . . . . .	46
8.2. Summary of Attack Types per Use Case Common Theme . . . . .	47
9. Security Considerations for OAM Traffic . . . . .	49
10. DetNet Technology-Specific Threats . . . . .	49
10.1. IP . . . . .	50
10.2. MPLS . . . . .	51
11. IANA Considerations . . . . .	52
12. Security Considerations . . . . .	52
13. Privacy Considerations . . . . .	52
14. Contributors . . . . .	53
15. References . . . . .	53
15.1. Normative References . . . . .	53
15.2. Informative References . . . . .	54
Authors' Addresses . . . . .	59

## 1. Introduction

A deterministic IP network (IETF DetNet, [RFC8655]) can carry data flows for real-time applications with extremely low data loss rates and bounded latency. The bounds on latency defined by DetNet (as described in [I-D.ietf-detnet-flow-information-model]) include both worst case latency (Maximum Latency, Section 5.9.2) and worst case jitter (Maximum Latency Variation, Section 5.9.3). Data flows with deterministic properties are well-established for Ethernet networks (see TSN, [IEEE802.1BA]); DetNet brings these capabilities to the IP network.

Deterministic IP networks have been successfully deployed in real-time Operational Technology (OT) applications for some years, however such networks are typically isolated from external access, and thus the security threat from external attackers is low. An example of such an isolated network is a network deployed within an aircraft, which is "air gapped" from the outside world. DetNet specifies a set of technologies that enable creation of deterministic flows on IP-based networks of potentially wide area (on the scale of a corporate network), potentially merging OT traffic with best-effort (Information Technology, IT) traffic, and placing OT network components into contact with IT network components, thereby exposing



the OT traffic and components to security threats that were not present in an isolated OT network.

These DetNet (OT-type) technologies may not have previously been deployed on a wide area IP-based network that also carries IT traffic, and thus can present security considerations that may be new to IP-based wide area network designers; this document provides insight into such system-level security considerations. In addition, designers of DetNet components (such as routers) face new security-related challenges in providing DetNet services, for example maintaining reliable isolation between traffic flows in an environment where IT traffic co-mingles with critical reserved-bandwidth OT traffic; this document also examines security implications internal to DetNet components.

Security is of particularly high importance in DetNet because many of the use cases which are enabled by DetNet [RFC8578] include control of physical devices (power grid devices, industrial controls, building controls) which can have high operational costs for failure, and present potentially attractive targets for cyber-attackers.

This situation is even more acute given that one of the goals of DetNet is to provide a "converged network", i.e. one that includes both IT traffic and OT traffic, thus exposing potentially sensitive OT devices to attack in ways that were not previously common (usually because they were under a separate control system or otherwise isolated from the IT network, for example [ARINC664P7]). Security considerations for OT networks are not a new area, and there are many OT networks today that are connected to wide area networks or the Internet; this document focuses on the issues that are specific to the DetNet technologies and use cases.

Given the above considerations, securing a DetNet starts with a scrupulously well-designed and well-managed engineered network following industry best practices for security at both the data plane and controller plane, as well as for any OAM implementation; this is the assumed starting point for the considerations discussed herein. Such assumptions also depend on the network components themselves upholding the security-related properties that are to be assumed by DetNet system-level designers; for example, the assumption that network traffic associated with a given flow can never affect traffic associated with a different flow is only true if the underlying components make it so. Such properties, which may represent new challenges to component designers, are also considered herein.

Starting with a "well-managed network" as noted above enables us to exclude some of the more powerful adversary capabilities from the Internet Threat Model of BCP 72 ([RFC3552]), such as the ability to

arbitrarily drop or delay any or all traffic. Given this reduced attacker capability, we can present security considerations based on attacker capabilities that are more directly relevant to a DetNet.

In this context we view the "traditional" (i.e. non-time-sensitive) network design and management aspects of network security as being primarily concerned with denial-of service prevention, i.e. they must ensure that DetNet traffic goes where it's supposed to and that an external attacker can't inject traffic that disrupts the delivery timing assurance of the DetNet. The time-specific aspects of DetNet security presented here take up where those "traditional" design and management aspects leave off.

However note that "traditional" methods for mitigating (among all the others) denial-of service attack (such as throttling) can only be effectively used in a DetNet when their use does not compromise the required time-sensitive or behavioral properties required for the OT flows on the network. For example, a "retry" protocol is typically not going to be compatible with a low-latency (worst-case maximum latency) requirement, however if in a specific use case and implementation such a retry protocol is able to meet the timing constraints, then it may well be used in that context. Similarly if common security protocols such as TLS/DTLS or IPsec are to be used, it must be verified that their implementations are able to meet the timing and behavioral requirements of the time-sensitive network as implemented for the given use case. An example of "behavioral properties" might be that dropping of more than a specific number of packets in a row is not acceptable according to the service level agreement.

The exact security requirements for any given DetNet are necessarily specific to the use cases handled by that network. Thus the reader is assumed to be familiar with the specific security requirements of their use cases, for example those outlined in the DetNet Use Cases [RFC8578] and the Security Considerations sections of the DetNet documents applicable to the network technologies in use, for example [RFC8939] for an IP data plane and [RFC8964] for an MPLS data plane. Readers can find a general introduction to the DetNet Architecture in [RFC8655], the DetNet Data Plane in [RFC8938], and the Flow Information Model in [I-D.ietf-detnet-flow-information-model].

The DetNet technologies include ways to:

- o Assign data plane resources for DetNet flows in some or all of the intermediate nodes (routers) along the path of the flow

- o Provide explicit routes for DetNet flows that do not dynamically change with the network topology in ways that affect the quality of service received by the affected flow(s)
- o Distribute data from DetNet flow packets over time and/or space to ensure delivery of the data in each packet in spite of the loss of a path.

This document includes sections considering DetNet component design as well as system design. The latter includes a taxonomy and analysis of threats, threat impacts and mitigations, and an association of attacks with use cases (based on the Use Case Common Themes section of the DetNet Use Cases [RFC8578]).

This document is based on the premise that there will be a very broad range of DetNet applications and use cases, ranging in size and scope from individual industrial machines to networks that span an entire country ([RFC8578]). Thus no single set of prescriptions (such as exactly which mitigation should be applied to which segment of a DetNet) can be applicable to all of them, and indeed any single one that we might prescribe would inevitably prove impractical for some use case, perhaps one that does not even exist at the time of this writing. Thus we are not prescriptive here - we are stating the desired end result, with the understanding that most DetNet use cases will necessarily differ from each other, and there is no "one size fits all".

## 2. Abbreviations and Terminology

IT: Information Technology (the application of computers to store, study, retrieve, transmit, and manipulate data or information, often in the context of a business or other enterprise - [IT\_DEF]).

OT: Operational Technology (the hardware and software dedicated to detecting or causing changes in physical processes through direct monitoring and/or control of physical devices such as valves, pumps, etc. - [OT\_DEF])

Component: A component of a DetNet system - used here to refer to any hardware or software element of a DetNet which implements DetNet-specific functionality, for example all or part of a router, switch, or end system.

Device: Used here to refer to a physical entity controlled by the DetNet, for example a motor.

Resource Segmentation: Used as a more general form for Network Segmentation (the act or practice of splitting a computer network into subnetworks, each being a network segment - [RS\_DEF])

Controller Plane: In DetNet the Controller Plane corresponds to the aggregation of the Control and Management Planes (see [RFC8655] section 4.4.2).

### 3. Security Considerations for DetNet Component Design

This section provides guidance for implementers of components to be used in a DetNet.

As noted above, DetNet provides resource allocation, explicit routes and redundant path support. Each of these has associated security implications, which are discussed in this section, in the context of component design. Detection, reporting and appropriate action in the case of packet arrival time violations are also discussed.

#### 3.1. Resource Allocation

##### 3.1.1. Inviolable Flows

A DetNet system security designer relies on the premise that any resources allocated to a resource-reserved (OT-type) flow are inviolable; in other words there is no physical possibility within a DetNet component that resources allocated to a given DetNet flow can be compromised by any type of traffic in the network; this includes malicious traffic as well as inadvertent traffic such as might be produced by a malfunctioning component, or due to interactions between components that were not sufficiently tested for interoperability. From a security standpoint this is a critical assumption, for example when designing against DOS attacks. In other words, with correctly designed components and security mechanisms, one can prevent malicious activities from impacting other resources.

However, achieving the goal of absolutely inviolable flows may not be technically or economically feasible for any given use case, given the broad range of possible use cases (e.g. [reference to DetNet Use Cases RFC8578]) and their associated security considerations as outlined in this document. It can be viewed as a continuum of security requirements, from isolated ultra-low latency systems that may have little security vulnerability (such as an industrial machine) to broadly distributed systems with many possible attack vectors and OT security concerns (such as a utility network). Given this continuum, the design principle employed in this document is to specify the desired end results, without being overly prescriptive in how the results are achieved, reflecting the understanding that no

individual implementation is likely to be appropriate for every DetNet use case.

### 3.1.2. Design Trade-Off Considerations in the Use Cases Continuum

It is important for the DetNet system designer to understand, for any given DetNet use case and its associated security requirements, the interaction and design trade-offs that inevitably need to be reconciled between the desired end results and the DetNet protocols, as well as the DetNet system and component design.

For any given component, as designed for any given use case (or scope of use cases), it is the responsibility of the component designer to ensure that the premise of inviolable flows is supported, to the extent that they deem necessary to support their target use cases.

For example, the component may include traffic shaping and policing at the ingress, to prevent corrupted or malicious or excessive packets from entering the network, thereby decreasing the likelihood that any traffic will interfere with any DetNet OT flow. The component may include integrity protection for some or all of the header fields such as those used for flow ID, thereby decreasing the likelihood that a packet whose flow ID has been compromised might be directed into a different flow path. The component may verify every single packet header at every forwarding location, or only at certain points. In any of these cases the component may use dynamic performance analytics (Section 7.7) to cause action to be initiated to address the situation in an appropriate and timely manner, either at the data plane or controller plane, or both in concert. The component's software and hardware may include measures to ensure the integrity of the resource allocation/deallocation process. Other design aspects of the component may help ensure that the adverse effects of malicious traffic are more limited, for example by protecting network control interfaces, or minimizing cascade failures. The component may include features specific to a given use case, such as configuration of the response to a given sequential packet loss count.

Ultimately, due to cost and complexity factors, the security properties of a component designed for low-cost systems may be (by design) far inferior to a component with similar intended functionality, but designed for highly secure or otherwise critical applications, perhaps at substantially higher cost. Any given component is designed for some set of use cases and accordingly will have certain limitations on its security properties and vulnerabilities. It is thus the responsibility of the system designer to assure themselves that the components they use in their

design are capable of satisfying their overall system security requirements.

#### 3.1.3. Documenting the Security Properties of a Component

In order for the system designer to adequately understand the security related behavior of a given component, the designer of any component intended for use with DetNet needs to clearly document the security properties of that component. For example, to address the case where a corrupted packet in which the flow identification information is compromised and thus may incidentally match the flow ID of another ("victim") DetNet flow, resulting in additional unauthorized traffic on the victim, the documentation might state that the component employs integrity protection on the flow identification fields.

#### 3.1.4. Fail-Safe Component Behavior

Even when the security properties of a component are understood and well specified, if the component malfunctions, for example due to physical circumstances unpredicted by the component designer, it may be difficult or impossible to fully prevent malfunction of the network. The degree to which a component is hardened against various types of failures is a distinguishing feature of the component and its design, and the overall system design can only be as strong as its weakest link.

However, all networks are subject to this level of uncertainty; it is not unique to DetNet. Having said that, DetNet raises the bar by changing many added latency scenarios from tolerable annoyances to unacceptable service violations. That in turn underscores the importance of system integrity, as well as correct and stable configuration of the network and its nodes, as discussed in Section 1.

#### 3.1.5. Flow Aggregation Example

As another example regarding resource allocation implementation, consider the implementation of Flow Aggregation for DetNet flows (as discussed in [RFC8938]). In this example say there are N flows that are to be aggregated, thus the bandwidth resources of the aggregate flow must be sufficient to contain the sum of the bandwidth reservation for the N flows. However if one of those flows were to consume more than its individually allocated bandwidth, this could cause starvation of the other flows. Thus simply providing and enforcing the calculated aggregate bandwidth may not be a complete solution - the bandwidth for each individual flow must still be guaranteed, for example via ingress policing of each flow (i.e.

before it is aggregated). Alternatively, if by some other means each flow to be aggregated can be trusted not to exceed its allocated bandwidth, the same goal can be achieved.

### 3.2. Explicit Routes

The DetNet-specific purpose for constraining the ability of the DetNet to re-route OT traffic is to maintain the specified service parameters (such as upper and lower latency boundaries) for a given flow. For example if the network were to re-route a flow (or some part of a flow) based exclusively on statistical path usage metrics, or due to malicious activity, it is possible that the new path would have a latency that is outside the required latency bounds which were designed into the original TE-designed path, thereby violating the quality of service for the affected flow (or part of that flow).

However, it is acceptable for the network to re-route OT traffic in such a way as to maintain the specified latency bounds (and any other specified service properties) for any reason, for example in response to a runtime component or path failure.

So from a DetNet security standpoint, the DetNet system designer can expect that any component designed for use in a DetNet will deliver the packets within the agreed-upon service parameters. For the component designer, this means that in order for a component to achieve that expectation, any component that is involved in controlling or implementing any change of the initially TE-configured flow routes must prevent re-routing of OT flows (whether malicious or accidental) which might adversely affect delivering the traffic within the specified service parameters.

### 3.3. Redundant Path Support

The DetNet provision for redundant paths (PREOF) (as defined in the DetNet Architecture [RFC8655]) provides the foundation for high reliability of a DetNet, by virtually eliminating packet loss (i.e. to a degree which is implementation-dependent) through hitless redundant packet delivery. Note: At the time of this writing, PREOF is not defined for the IP data plane.

It is the responsibility of the system designer to determine the level of reliability required by their use case, and to specify redundant paths sufficient to provide the desired level of reliability (in as much as that reliability can be provided through the use of redundant paths). It is the responsibility of the component designer to ensure that the relevant PREOF operations are executed reliably and securely, to avoid potentially catastrophic situations for the operational technology relying on them.

However, note that not all PREOF operations are necessarily implemented in every network; for example a packet re-ordering function may not be necessary if the packets are either not required to be in order, or if the ordering is performed in some other part of the network.

Ideally a redundant path for a flow could be specified from end to end, however given that this is not always possible (as described in [RFC8655]) the system designer will need to consider the resulting end-to-end reliability and security resulting from any given arrangement of network segments along the path, each of which provides its individual PREOF implementation and thus its individual level of reliability and security.

At the data plane the implementation of PREOF depends on the correct assignment and interpretation of packet sequence numbers, as well as the actions taken based on them, such as elimination (including elimination of packets with spurious sequence numbers). Thus the integrity of these values must be maintained by the component as they are assigned by the DetNet Data Plane Service sub-layer, and transported by the Forwarding sub-layer. This is no different than the integrity of the values in any header used by the DetNet (or any other) data plane, and is not unique to redundant paths. The integrity protection of header values is technology-dependent; for example, in Layer 2 networks the integrity of the header fields can be protected by using MACsec [IEEE802.1AE-2018]. Similarly, from the sequence number injection perspective, it is no different from any other protocols that use sequence numbers. In particular IPSec Authentication Header ([RFC4302], Sec. 3 Authentication Header (AH) Processing) provides useful insights.

### 3.4. Timing (or other) Violation Reporting

A task of the DetNet system designer is to create a network such that for any incoming packet which arrives with any timing or bandwidth violation, an appropriate action can be taken in order to prevent damage to the system. The reporting step may be accomplished through dynamic performance analysis (see Section 7.7) or by any other means as implemented in one or more components. The action to be taken for any given circumstance within any given application will depend on the use case. The action may involve intervention from the controller plane, or it may be taken "immediately" by an individual component, for example if very fast response is required.

The definitions and selections of the actions that can be taken are properties of the components. The component designer implements these options according to their expected use cases, which may vary widely from component to component. Clearly selecting an



inappropriate response to a given condition may cause more problems than it is intending to mitigate; for example, a naive approach might be to have the component shut down the link if a packet arrives outside of its prescribed time window; however such a simplistic action may serve the attacker better than it serves the network. Similarly, simple logging of such issues may not be adequate, since a delay in response could result in material damage, for example to mechanical devices controlled by the network. Thus a breadth of possible and effective security-related actions and their configuration is a positive attribute for a DetNet component.

Some possible violations that warrant detection include cases where a packet arrives:

- o Outside of its prescribed time window
- o Within its time window but with a compromised time stamp that makes it appear that it is not within its window
- o Exceeding the reserved flow bandwidth

Some possible direct actions that may be taken at the data plane include traffic policing and shaping functions (e.g., those described in [RFC2475]), separating flows into per-flow rate-limited queues, and potentially applying active queue management [RFC7567]. However if those (or any other) actions are to be taken, the system designer must ensure that the results of such actions do not compromise the continued safe operation of the system. For example, the network (i.e. the controller plane and data plane working together) must mitigate in a timely fashion any potential adverse effect on mechanical devices controlled by the network.

#### 4. DetNet Security Considerations Compared With DiffServ Security Considerations

DetNet is designed to be compatible with DiffServ [RFC2474] as applied to IT traffic in the DetNet. DetNet also incorporates the use of the 6-bit value of the DSCP field of the Type of Service (IPv4) and Traffic Class (IPv6) bytes for flow identification. However, the DetNet interpretation of the DSCP value for OT traffic is not equivalent to the PHB selection behavior as defined by DiffServ.

Thus security consideration for DetNet have some aspects in common with DiffServ, in fact overlapping 100% with respect to IP IT traffic. Security considerations for these aspects are part of the existing literature on IP network security, specifically the Security Considerations sections of [RFC2474] and [RFC2475]. However, DetNet

also introduces timing and other considerations which are not present in DiffServ, so the DiffServ security considerations are a subset of the DetNet security considerations.

In the case of DetNet OT traffic, the DSCP value is interpreted differently than in DiffServ and contribute to determination of the service provided to the packet. In DetNet, there are similar consequences to DiffServ for lack of detection of, or incorrect handling of, packets with mismarked DSCP values, and many of the points made in the DiffServ Security discussions ([RFC2475] Sec. 6.1, [RFC2474] Sec. 7 and [RFC6274] Sec 3.3.2.1) are also relevant to DetNet OT traffic, though perhaps in modified form. For example, in DetNet the effect of an undetected or incorrectly handled maliciously mismarked DSCP field in an OT packet is not identical to affecting the PHB of that packet, since DetNet does not use the PHB concept for OT traffic; but nonetheless the service provided to the packet could be affected, so mitigation measures analogous to those prescribed by DiffServ would be appropriate for DetNet. For example, mismarked DSCP values should not cause failure of network nodes. The remarks in [RFC2474] regarding IPsec and Tunnelling Interactions are also relevant (though this is not to say that other sections are less relevant).

In this discussion, interpretation (and any possible intentional re-marking) of the DSCP values of packets destined for DetNet OT flows is expected to occur at the ingress to the DetNet domain; once inside the domain, maintaining the integrity of the DSCP values is subject to the same handling considerations as any other field in the packet.

## 5. Security Threats

This section presents a taxonomy of threats, and analyzes the possible threats in a DetNet-enabled network. The threats considered in this section are independent of any specific technologies used to implement the DetNet; Section 10 considers attacks that are associated with the DetNet technologies encompassed by [RFC8938].

We distinguish controller plane threats from data plane threats. The attack surface may be the same, but the types of attacks as well as the motivation behind them, are different. For example, a delay attack is more relevant to data plane than to controller plane. There is also a difference in terms of security solutions: the way you secure the data plane is often different than the way you secure the controller plane.

### 5.1. Threat Taxonomy

This document employs organizational elements of the threat models of [RFC7384] and [RFC7835]. This model classifies attackers based on two criteria:

- o Internal vs. external: internal attackers either have access to a trusted segment of the network or possess the encryption or authentication keys. External attackers, on the other hand, do not have the keys and have access only to the encrypted or authenticated traffic.
- o On-path vs. off-path: on-path attackers are located in a position that allows interception, modification, or dropping of in-flight protocol packets, whereas off-path attackers can only attack by generating protocol packets.

Regarding the boundary between internal vs. external attackers as defined above, please note that in this document we do not make concrete recommendations regarding which specific segments of the network are to be protected in any specific way, for example via encryption or authentication. As a result, the boundary as defined above is not unequivocally specified here. Given that constraint, the reader can view an internal attacker as one who can operate within the perimeter defined by the DetNet Edge Nodes (as defined in the DetNet Architecture [RFC8655]), allowing that the specifics of what is encrypted or authenticated within this perimeter will vary depending on the implementation.

Care has also been taken to adhere to Section 5 of [RFC3552], both with respect to which attacks are considered out-of-scope for this document, but also which are considered to be the most common threats (explored further in Section 5.2, Threat Analysis). Most of the direct threats to DetNet are active attacks (i.e. attacks that modify DetNet traffic), but it is highly suggested that DetNet application developers take appropriate measures to protect the content of the DetNet flows from passive attacks (i.e. attacks that observe but do not modify DetNet traffic) for example through the use of TLS or DTLS.

DetNet-Service, one of the service scenarios described in [I-D.varga-detnet-service-model], is the case where a service connects DetNet islands, i.e. two or more otherwise independent DetNets are connected via a link that is not intrinsically part of either network. This implies that there could be DetNet traffic flowing over a non-DetNet link, which may provide an attacker with an advantageous opportunity to tamper with DetNet traffic. The security properties of non-DetNet links are outside of the scope of DetNet

Security, but it should be noted that use of non-DetNet services to interconnect DetNets merits security analysis to ensure the integrity of the networks involved.

## 5.2. Threat Analysis

### 5.2.1. Delay

An attacker can maliciously delay DetNet data flow traffic. By delaying the traffic, the attacker can compromise the service of applications that are sensitive to high delays or to high delay variation. The delay may be constant or modulated.

### 5.2.2. DetNet Flow Modification or Spoofing

An attacker can modify some header fields of en route packets in a way that causes the DetNet flow identification mechanisms to misclassify the flow. Alternatively, the attacker can inject traffic that is tailored to appear as if it belongs to a legitimate DetNet flow. The potential consequence is that the DetNet flow resource allocation cannot guarantee the performance that is expected when the flow identification works correctly.

### 5.2.3. Resource Segmentation (Inter-segment Attack) Vulnerability

DetNet components are expected to split their resources between DetNet flows in a way that prevents traffic from one DetNet flow from affecting the performance of other DetNet flows, and also prevents non-DetNet traffic from affecting DetNet flows. However, perhaps due to implementation constraints, some resources may be partially shared, and an attacker may try to exploit this property. For example, an attacker can inject traffic in order to exhaust network resources such that DetNet packets which share resources with the injected traffic may be dropped or delayed. Such injected traffic may be part of DetNet flows or non-DetNet traffic.

Another example of a resource segmentation attack is the case in which an attacker is able to overload the exception path queue on the router, i.e. a "slow path" typically taken by control or OAM packets which are diverted from the data plane because they require processing by a CPU. DetNet OT flows are typically configured to take the "fast path" through the data plane, to minimize latency. However if there is only one queue from the forwarding ASIC to the exception path, and for some reason the system is configured such that any DetNet packets must be handled on this exception path, then saturating the exception path could result in delaying or dropping of DetNet packets.

#### 5.2.4. Packet Replication and Elimination

##### 5.2.4.1. Replication: Increased Attack Surface

Redundancy is intended to increase the robustness and survivability of DetNet flows, and replication over multiple paths can potentially mitigate an attack that is limited to a single path. However, the fact that packets are replicated over multiple paths increases the attack surface of the network, i.e., there are more points in the network that may be subject to attacks.

##### 5.2.4.2. Replication-related Header Manipulation

An attacker can manipulate the replication-related header fields. This capability opens the door for various types of attacks. For example:

- o Forward both replicas - malicious change of a packet SN (Sequence Number) can cause both replicas of the packet to be forwarded. Note that this attack has a similar outcome to a replay attack.
- o Eliminate both replicas - SN manipulation can be used to cause both replicas to be eliminated. In this case an attacker that has access to a single path can cause packets from other paths to be dropped, thus compromising some of the advantage of path redundancy.
- o Flow hijacking - an attacker can hijack a DetNet flow with access to a single path by systematically replacing the SNs on the given path with higher SN values. For example, an attacker can replace every SN value  $S$  with a higher value  $S+C$ , where  $C$  is a constant integer. Thus, the attacker creates a false illusion that the attacked path has the lowest delay, causing all packets from other paths to be eliminated in favor of the attacked path. Once the flow from the compromised path is favored by the eliminating bridge, the flow has effectively been hijacked by the attacker. It is now possible for the attacker to either replace en route packets with malicious packets, or to simply inject errors into the packets, causing the packets to be dropped at their destination.
- o Amplification - an attacker who injects packets into a flow that is to be replicated will have their attack amplified through the replication process. This is no different than any attacker who injects packets that are delivered through multicast, broadcast, or other point-to-multi-point mechanisms.

## 5.2.5. Controller Plane

### 5.2.5.1. Path Choice Manipulation

#### 5.2.5.1.1. Control or Signaling Packet Modification

An attacker can maliciously modify en route control packets in order to disrupt or manipulate the DetNet path/resource allocation.

#### 5.2.5.1.2. Control or Signaling Packet Injection

An attacker can maliciously inject control packets in order to disrupt or manipulate the DetNet path/resource allocation.

#### 5.2.5.1.3. Increased Attack Surface

One of the possible consequences of a path manipulation attack is an increased attack surface. Thus, when the attack described in the previous subsection is implemented, it may increase the potential of other attacks to be performed.

### 5.2.5.2. Compromised Controller

An attacker can subvert a legitimate controller (or subvert another component such that it represents itself as a legitimate controller) with the result that the network nodes incorrectly believe it is authorized to instruct them.

The presence of a compromised node or controller in a DetNet is not a threat that arises as a result of determinism or time sensitivity; the same techniques used to prevent or mitigate against compromised nodes in any network are equally applicable in the DetNet case. The act of compromising a controller may not even be within the capabilities of our defined attacker types - in other words it may not be achievable via packet traffic at all, whether internal or external, on-path or off-path. It might be accomplished for example by a human with physical access to the component, who could upload bogus firmware to it via a USB stick. All of this underscores the requirement for careful overall system security design in a DetNet, given that the effects of even one bad actor on the network can be potentially catastrophic.

Security concerns specific to any given controller plane technology used in DetNet will be addressed by the DetNet documents associated with that technology.

#### 5.2.6. Reconnaissance

A passive eavesdropper can identify DetNet flows and then gather information about en route DetNet flows, e.g., the number of DetNet flows, their bandwidths, their schedules, or other temporal or statistical properties. The gathered information can later be used to invoke other attacks on some or all of the flows.

DetNet flows are typically uniquely identified by their 6-tuple, i.e. fields within the L3 or L4 header, however in some implementations the flow ID may also be augmented by additional per-flow attributes known to the system, e.g. above L4. For the purpose of this document we assume any such additional fields used for flow ID are encrypted and/or integrity-protected from external attackers. Note however that existing OT protocols designed for use on dedicated secure networks may not intrinsically provide such protection, in which case IPsec or transport layer security mechanisms may be needed.

#### 5.2.7. Time Synchronization Mechanisms

An attacker can use any of the attacks described in [RFC7384] to attack the synchronization protocol, thus affecting the DetNet service.

#### 5.3. Threat Summary

A summary of the attacks that were discussed in this section is presented in Figure 1. For each attack, the table specifies the type of attackers that may invoke the attack. In the context of this summary, the distinction between internal and external attacks is under the assumption that a corresponding security mechanism is being used, and that the corresponding network equipment takes part in this mechanism.

Attack	Attacker Type			
	Internal On-P	Off-P	External On-P	Off-P
Delay attack	+		+	
DetNet Flow Modification or Spoofing	+	+		
Inter-segment Attack	+	+	+	+
Replication: Increased Attack Surface	+	+	+	+
Replication-related Header Manipulation	+			
Path Manipulation	+	+		
Path Choice: Increased Attack Surface	+	+	+	+
Control or Signaling Packet Modification	+			
Control or Signaling Packet Injection	+	+		
Reconnaissance	+		+	
Attacks on Time Synchronization Mechanisms	+	+	+	+

Figure 1: Threat Analysis Summary

## 6. Security Threat Impacts

When designing security for a DetNet, as with any network, it may be prohibitively expensive or technically infeasible to thoroughly protect against every possible threat. Thus the security designer must be informed (for example by an application domain expert such as a product manager) regarding the relative significance of the various threats and their impact if a successful attack is carried out. In this section we present an example of a possible template for such a communication, culminating in a table (Figure 2) which lists a set of threats under consideration, and some values characterizing their relative impact in the context of a given industry. The specific threats, industries, and impact values in the table are provided only as an example of this kind of assessment and its communication; they are not intended to be taken literally.



This section considers assessment of the relative impacts of the attacks described in Section 5, Security Threats. In this section, the impacts as described assume that the associated mitigation is not present or has failed. Mitigations are discussed in Section 7, Security Threat Mitigation.

In computer security, the impact (or consequence) of an incident can be measured in loss of confidentiality, integrity or availability of information. In the case of time sensitive or OT networks (though not to the exclusion of IT or non-time-sensitive networks) the impact of an exploit can also include failure or malfunction of mechanical and/or other physical systems.

DetNet raises these stakes significantly for OT applications, particularly those which may have been designed to run in an OT-only environment and thus may not have been designed for security in an IT environment with its associated components, services and protocols.

The extent of impact of a successful vulnerability exploit varies considerably by use case and by industry; additional insights regarding the individual use cases is available from [RFC8578], DetNet Use Cases. Each of those use cases is represented in Figure 2, including Pro Audio, Electrical Utilities, Industrial M2M (split into two areas, M2M Data Gathering and M2M Control Loop), and others.

Aspects of Impact (left column) include Criticality of Failure, Effects of Failure, Recovery, and DetNet Functional Dependence. Criticality of failure summarizes the seriousness of the impact. The impact of a resulting failure can affect many different metrics that vary greatly in scope and severity. In order to reduce the number of variables, only the following were included: Financial, Health and Safety, Effect on a Single Organization, and Effect on Multiple Organizations. Recovery outlines how long it would take for an affected use case to get back to its pre-failure state (Recovery time objective, RTO), and how much of the original service would be lost in between the time of service failure and recovery to original state (Recovery Point Objective, RPO). DetNet dependence maps how much the following DetNet service objectives contribute to impact of failure: Time dependency, data integrity, source node integrity, availability, latency/jitter.

The scale of the Impact mappings is low, medium, and high. In some use cases there may be a multitude of specific applications in which DetNet is used. For simplicity this section attempts to average the varied impacts of different applications. This section does not address the overall risk of a certain impact which would require the likelihood of a failure happening.

In practice any such ratings will vary from case to case; the ratings shown here are given as examples.

Table

	Pro A	Util	Bldg	Wire-less	Cell	M2M Data	M2M Ctrl
Criticality	Med	Hi	Low	Med	Med	Med	Med
Effects							
Financial	Med	Hi	Med	Med	Low	Med	Med
Health/Safety	Med	Hi	Hi	Med	Med	Med	Med
Affects 1 org	Hi	Hi	Med	Hi	Med	Med	Med
Affects >1 org	Med	Hi	Low	Med	Med	Med	Med
Recovery							
Recov Time Obj	Med	Hi	Med	Hi	Hi	Hi	Hi
Recov Point Obj	Med	Hi	Low	Med	Low	Hi	Hi
DetNet Dependence							
Time Dependency	Hi	Hi	Low	Hi	Med	Low	Hi
Latency/Jitter	Hi	Hi	Med	Med	Low	Low	Hi
Data Integrity	Hi	Hi	Med	Hi	Low	Hi	Hi
Src Node Integ	Hi	Hi	Med	Hi	Med	Hi	Hi
Availability	Hi	Hi	Med	Hi	Low	Hi	Hi

Figure 2: Impact of Attacks by Use Case Industry

The rest of this section will cover impact of the different groups in more detail.

## 6.1. Delay-Attacks

### 6.1.1. Data Plane Delay Attacks

Note that 'delay attack' also includes the possibility of a 'negative delay' or early arrival of a packet, or possibly adversely changing the timestamp value.

Delayed messages in a DetNet link can result in the same behavior as dropped messages in ordinary networks, since the services attached to the DetNet flow are likely to have strict delivery time requirements.

For a single path scenario, disruption within the single flow is a real possibility. In a multipath scenario, large delays or instabilities in one DetNet flow can also lead to increased buffer and processor resource consumption at the eliminating router.

A data-plane delay attack on a system controlling substantial moving devices, for example in industrial automation, can cause physical damage. For example, if the network promises a bounded latency of 2ms for a flow, yet the machine receives it with 5ms latency, the control loop of the machine may become unstable.

### 6.1.2. Controller Plane Delay Attacks

In and of itself, this is not directly a threat to the DetNet service, but the effects of delaying control messages can have quite adverse effects later.

- o Delayed tear-down can lead to resource leakage, which in turn can result in failure to allocate new DetNet flows, finally giving rise to a denial of service attack.
- o Failure to deliver, or severely delaying, controller plane messages adding an endpoint to a multicast-group will prevent the new endpoint from receiving expected frames thus disrupting expected behavior.
- o Delaying messages removing an endpoint from a group can lead to loss of privacy as the endpoint will continue to receive messages even after it is supposedly removed.

## 6.2. Flow Modification and Spoofing

### 6.2.1. Flow Modification

If the contents of a packet header or body can be modified by the attacker, this can cause the packet to be routed incorrectly or dropped, or the payload to be corrupted or subtly modified. Thus, the potential impact of a modification attack includes disrupting the application as well as the network equipment.

### 6.2.2. Spoofing

#### 6.2.2.1. Dataplane Spoofing

Spoofing dataplane messages can result in increased resource consumptions on the routers throughout the network as it will increase buffer usage and processor utilization. This can lead to resource exhaustion and/or increased delay.

If the attacker manages to create valid headers, the false messages can be forwarded through the network, using part of the allocated bandwidth. This in turn can cause legitimate messages to be dropped when the resource budget has been exhausted.

Finally, the endpoint will have to deal with invalid messages being delivered to the endpoint instead of (or in addition to) a valid message.

#### 6.2.2.2. Controller Plane Spoofing

A successful controller plane spoofing-attack will potentially have adverse effects. It can do virtually anything from:

- o modifying existing DetNet flows by changing the available bandwidth
- o add or remove endpoints from a DetNet flow
- o drop DetNet flows completely
- o falsely create new DetNet flows (exhaust the systems resources, or to enable DetNet flows that are outside the control of the Network Engineer)

### 6.3. Segmentation Attacks (injection)

#### 6.3.1. Data Plane Segmentation

Injection of false messages in a DetNet flow could lead to exhaustion of the available bandwidth for that flow if the routers attribute these false messages to the resource budget of that flow.

In a multipath scenario, injected messages will cause increased processor utilization in elimination routers. If enough paths are subject to malicious injection, the legitimate messages can be dropped. Likewise it can cause an increase in buffer usage. In total, it will consume more resources in the routers than normal, giving rise to a resource exhaustion attack on the routers.

If a DetNet flow is interrupted, the end application will be affected by what is now a non-deterministic flow. Note that there are many possible sources of flow interruptions, for example, but not limited to, such physical layer conditions as a broken wire or a radio link which is compromised by interference.

#### 6.3.2. Controller Plane Segmentation

In a successful controller plane segmentation attack, control messages are acted on by nodes in the network, unbeknownst to the central controller or the network engineer. This has the potential to:

- o create new DetNet flows (exhausting resources)
- o drop existing DetNet flows (denial of service)
- o add end-stations to a multicast group (loss of privacy)
- o remove end-stations from a multicast group (reduction of service)
- o modify the DetNet flow attributes (affecting available bandwidth)

If an attacker can inject control messages without the central controller knowing, then one or more components in the network may get into a state that is not expected by the controller. At that point, if the controller initiates a command, the effect of that command may not be as expected, since the target of the command may have started from a different initial state.

#### 6.4. Replication and Elimination

The Replication and Elimination is relevant only to data plane messages as controller plane messages are not subject to multipath routing.

#### 6.4.1. Increased Attack Surface

The impact of an increased attack surface is that it increases the probability that the network can be exposed to an attacker. This can facilitate a wide range of specific attacks, and their respective impacts are discussed in other subsections of this section.

#### 6.4.2. Header Manipulation at Elimination Routers

This attack can potentially cause DoS to the application that uses the attacked DetNet flows or to the network equipment that forwards them. Furthermore, it can allow an attacker to manipulate the network paths and the behavior of the network layer.

#### 6.5. Control or Signaling Packet Modification

If control packets are subject to manipulation undetected, the network can be severely compromised.

#### 6.6. Control or Signaling Packet Injection

If an attacker can inject control packets undetected, the network can be severely compromised.

#### 6.7. Reconnaissance

Of all the attacks, this is one of the most difficult to detect and counter.

An attacker can, at their leisure, observe over time various aspects of the messaging and signalling, learning the intent and purpose of the traffic flows. Then at some later date, possibly at an important time in the operational context, they might launch an attack based on that knowledge.

The flow-id in the header of the data plane messages gives an attacker a very reliable identifier for DetNet traffic, and this traffic has a high probability of going to lucrative targets.

Applications which are ported from a private OT network to the higher visibility DetNet environment may need to be adapted to limit distinctive flow properties that could make them susceptible to reconnaissance.

## 6.8. Attacks on Time Synchronization Mechanisms

DetNet relies on an underlying time synchronization mechanism, and therefore a compromised synchronization mechanism may cause DetNet nodes to malfunction. Specifically, DetNet flows may fail to meet their latency requirements and deterministic behavior, thus causing DoS to DetNet applications.

## 6.9. Attacks on Path Choice

This is covered in part in Section 6.3, Segmentation Attacks, and as with Replication and Elimination (Section 6.4), this is relevant for DataPlane messages.

## 7. Security Threat Mitigation

This section describes a set of measures that can be taken to mitigate the attacks described in Section 5, Security Threats. These mitigations should be viewed as a set of tools, any of which can be used individually or in concert. The DetNet component and/or system and/or application designer can apply these tools, as necessary based on a system-specific threat analysis.

Some of the technology-specific security considerations and mitigation approaches are further discussed in the DetNet data plane solution documents, such as [RFC8938], [RFC8939], [RFC8964], [I-D.ietf-detnet-mpls-over-udp-ip], and [I-D.ietf-detnet-ip-over-mpls].

### 7.1. Path Redundancy

#### Description

A DetNet flow that can be forwarded simultaneously over multiple paths. Packet replication and elimination [RFC8655] provides resiliency to dropped or delayed packets. This redundancy improves the robustness to failures and to on-path attacks. Note: At the time of this writing, PREOF is not defined for the IP data plane.

#### Related attacks

Path redundancy can be used to mitigate various on-path attacks, including attacks described in Section 5.2.1, Section 5.2.2, Section 5.2.3, and Section 5.2.7. However it is also possible that multiple paths may make it more difficult to locate the source of an on-path attacker.

A delay modulation attack could result in extensively exercising parts of the code that wouldn't normally be extensively exercised and thus might expose flaws in the system that might otherwise not be exposed.

## 7.2. Integrity Protection

### Description

Integrity Protection in the scope of DetNet is the ability to detect if a packet header has been modified (maliciously or otherwise) and if so, take some appropriate action (as discussed in Section 7.7). The decision on where in the network to apply integrity protection is part of the DetNet system design, and the implementation of the protection method itself is a part of a DetNet component design.

The most common technique for detecting header modification is the use of a Message Authentication Code (MAC) (for examples see Section 10). The MAC can be distributed either in-line (included in the same packet) or via a side channel. Of these, the in-line method is generally preferred due to the low latency that may be required on DetNet flows and the relative complexity and computational overhead of a sideband approach.

There are different levels of security available for integrity protection, ranging from the basic ability to detect if a header has been corrupted in transit (no malicious attack) to stopping a skilled and determined attacker capable of both subtly modifying fields in the headers as well as updating an unkeyed checksum. Common for all are the 2 steps that need to be performed in both ends. The first is computing the checksum or MAC. The corresponding verification step must perform the same steps before comparing the provided with the computed value. Only then can the receiver be reasonably sure that the header is authentic.

The most basic protection mechanism consists of computing a simple checksum of the header fields and provide it to the next entity in the packets path for verification. Using a MAC combined with a secret key provides the best protection against modification and replication attacks (see Section 5.2.2 and Section 5.2.4). This MAC usage needs to be part of a security association that is established and managed by a security association protocol (such as IKEv2 for IPsec security associations). Integrity protection in the controller plane is discussed in Section 7.6. The secret key, regardless of MAC used, must be protected from falling into the hands of unauthorized users. Once key management becomes a topic, it is important to understand that this is a delicate



process and should not be undertaken lightly. BCP 107 [RFC4107] provides best practices in this regard.

DetNet system and/or component designers need to be aware of these distinctions and enforce appropriate integrity protection mechanisms as needed based on a threat analysis. Note that adding integrity protection mechanisms may introduce latency, thus many of the same considerations in Section 7.5.1 also apply here.

#### Packet Sequence Number Integrity Considerations

The use of PREOF in a DetNet implementation implies the use of a sequence number for each packet. There is a trust relationship between the component that adds the sequence number and the component that removes the sequence number. The sequence number may be end-to-end source to destination, or may be added/deleted by network edge components. The adder and remover(s) have the trust relationship because they are the ones that ensure that the sequence numbers are not modifiable. Thus, sequence numbers can be protected by using authenticated encryption, or by a MAC without using encryption. Between the adder and remover there may or may not be replication and elimination functions. The elimination functions must be able to see the sequence numbers. Therefore, if encryption is done between adders and removers it must not obscure the sequence number. If the sequence removers and the eliminators are in the same physical component, it may be possible to obscure the sequence number, however that is a layer violation, and is not recommended practice. Note: At the time of this writing, PREOF is not defined for the IP data plane.

#### Related attacks

Integrity protection mitigates attacks related to modification and tampering, including the attacks described in Section 5.2.2 and Section 5.2.4.

### 7.3. DetNet Node Authentication

#### Description

Authentication verifies the identity of DetNet nodes (including DetNet Controller Plane nodes), and this enables mitigation of spoofing attacks. While integrity protection (Section 7.2) prevents intermediate nodes from modifying information, authentication can provide traffic origin verification, i.e. to verify that each packet in a DetNet flow is from a known source. Although node authentication and integrity protection are two different goals of a security protocol, in most cases a common

protocol (such as IPsec [RFC4301] or MACsec [IEEE802.1AE-2018]) is used for achieving both purposes.

#### Related attacks

DetNet node authentication is used to mitigate attacks related to spoofing, including the attacks of Section 5.2.2, and Section 5.2.4.

### 7.4. Dummy Traffic Insertion

#### Description

With some queueing methods such as [IEEE802.1Qch-2017] it is possible to introduce dummy traffic in order to regularize the timing of packet transmission. This will subsequently reduce the value of passive monitoring from internal threats (see Section 5) as it will be much more difficult to associate discrete events with particular network packets.

#### Related attacks

Removing distinctive temporal properties of individual packets or flows can be used to mitigate against reconnaissance attacks Section 5.2.6. For example, dummy traffic can be used to synthetically maintain constant traffic rate even when no user data is transmitted, thus making it difficult to collect information about the times at which users are active, and the times at which DetNet flows are added or removed.

#### Traffic Insertion Challenges

Once an attacker is able to monitor the frames traversing a network to such a degree that they can differentiate between best-effort traffic and traffic belonging to a specific DetNet flow, it becomes difficult to not reveal to the attacker whether a given frame is valid traffic or an inserted frame. Thus, having the DetNet components generate and remove the dummy traffic may or may not be a viable option, unless certain challenges are solved; for example, but not limited to:

- o Inserted traffic must be indistinguishable from valid stream traffic from the viewpoint of the attacker.
- o DetNet components must be able to safely identify and remove all inserted traffic (and only inserted traffic).

- o The controller plane must manage where to insert and remove dummy traffic, but this information must not be revealed to an attacker.

An alternative design is to have the insertion and removal of dummy traffic be performed at the application layer, rather than by the DetNet itself. Further discussions and reading about how sRTP handles this can be found in [RFC6562]

## 7.5. Encryption

### Description

Reconnaissance attacks (Section 5.2.6) can be mitigated to some extent through the use of encryption, thereby preventing the attacker from accessing the packet header or contents. Specific encryption protocols will depend on the lower layers that DetNet is forwarded over. For example, IP flows may be forwarded over IPsec [RFC4301], and Ethernet flows may be secured using MACsec [IEEE802.1AE-2018].

However, despite the use of encryption, a reconnaissance attack can provide the attacker with insight into the network, even without visibility into the packet. For example, an attacker can observe which nodes are communicating with which other nodes, including when, how often, and with how much data. In addition, the timing of packets may be correlated in time with external events such as action of an external device. Such information may be used by the attacker, for example in mapping out specific targets for a different type of attack at a different time.

DetNet nodes do not have any need to inspect the payload of any DetNet packets, making them data-agnostic. This means that end-to-end encryption at the application layer is an acceptable way to protect user data.

Note that reconnaissance is a threat that is not specific to DetNet flows, and therefore reconnaissance mitigation will typically be analyzed and provided by a network operator regardless of whether DetNet flows are deployed. Thus, encryption requirements will typically not be defined in DetNet technology-specific specifications, but considerations of using DetNet in encrypted environments will be discussed in these specifications. For example, Section 5.1.2.3. of [RFC8939] discusses flow identification of DetNet flows running over IPsec.

### Related attacks

As noted above, encryption can be used to mitigate reconnaissance attacks (Section 5.2.6). However, for a DetNet to provide differentiated quality of service on a flow-by-flow basis, the network must be able to identify the flows individually. This implies that in a reconnaissance attack the attacker may also be able to track individual flows to learn more about the system.

#### 7.5.1. Encryption Considerations for DetNet

Any compute time which is required for encryption and decryption processing ('crypto') must be included in the flow latency calculations. Thus, crypto algorithms used in a DetNet must have bounded worst-case execution times, and these values must be used in the latency calculations. Fortunately, encryption and decryption operations typically are designed to have constant execution times, in order to avoid side channel leakage.

Some crypto algorithms are symmetric in encode/decode time (such as AES) and others are asymmetric (such as public key algorithms). There are advantages and disadvantages to the use of either type in a given DetNet context. The discussion in this document relates to the timing implications of crypto for DetNet; it is assumed that integrity considerations are covered elsewhere in the literature.

Asymmetrical crypto is typically not used in networks on a packet-by-packet basis due to its computational cost. For example, if only endpoint checks or checks at a small number of intermediate points are required, asymmetric crypto can be used to authenticate distribution or exchange of a secret symmetric crypto key; a successful check based on that key will provide traffic origin verification, as long as the key is kept secret by the participants. TLS (v1.3 [RFC8446], in particular section 4.1 "Key exchange") and IKEv2 [RFC6071]) are examples of this for endpoint checks.

However, if secret symmetric keys are used for this purpose the key must be given to all relays, which increases the probability of a secret key being leaked. Also, if any relay is compromised or faulty then it may inject traffic into the flow. Group key management protocols can be used to automate management of such symmetric keys; for an example in the context of IPsec, see [I-D.ietf-ipsecme-g-ikev2].

Alternatively, asymmetric crypto can provide traffic origin verification at every intermediate node. For example, a DetNet flow can be associated with an (asymmetric) keypair, such that the private key is available to the source of the flow and the public key is distributed with the flow information, allowing verification at every

node for every packet. However, this is more computationally expensive.

In either case, origin verification also requires replay detection as part of the security protocol to prevent an attacker from recording and resending traffic, e.g., as a denial of service attack on flow forwarding resources.

In the general case, cryptographic hygiene requires the generation of new keys during the lifetime of an encrypted flow (e.g. see [RFC4253] section 9), and any such key generation (or key exchange) requires additional computing time which must be accounted for in the latency calculations for that flow. For modern ECDH (Elliptical Curve Diffie-Hellman) key-exchange operations (such as x25519, see [RFC7748]) these operations can be performed in constant (predictable) time, however this is not universally true (for example for legacy RSA key exchange, [RFC4432]). Thus implementers should be aware of the time properties of these algorithms and avoid algorithms that make constant-time implementation difficult or impossible.

## 7.6. Control and Signaling Message Protection

### Description

Control and signaling messages can be protected through the use of any or all of encryption, authentication, and integrity protection mechanisms. Compared with data-flows, the timing constraints for controller and signaling messages may be less strict, and the number of such packets may be fewer. If that is the case in a given application, then it may enable the use of asymmetric cryptography for signing of both payload and headers for such messages, as well as encrypting the payload. Given that a DetNet is managed by a central controller, the use of a shared public key approach for these processes is well-proven. This is further discussed in Section 7.5.1.

### Related attacks

These mechanisms can be used to mitigate various attacks on the controller plane, as described in Section 5.2.5, Section 5.2.7 and Section 5.2.5.1.

## 7.7. Dynamic Performance Analytics

### Description

Incorporating Dynamic Performance Analytics ("DPA") implies that the DetNet design includes a performance monitoring system to

validate that timing guarantees are being met and to detect timing violations or other anomalies that may be the symptom of a security attack or system malfunction. If this monitoring system detects unexpected behavior, it must then cause action to be initiated to address the situation in an appropriate and timely manner, either at the data plane or controller plane, or both in concert.

The overall DPA system can thus be decomposed into the "detection" and "notification" functions. Although the time-specific DPA performance indicators and their implementation will likely be specific to a given DetNet, and as such are nascent technology at the time of this writing, DPA is commonly used in existing networks so we can make some observations on how such a system might be implemented for a DetNet, given that it would need to be adapted to address the time-specific performance indicators.

#### Detection Mechanisms

Measurement of timing performance can be done via "passive" or "active" monitoring, as discussed below.

Examples of passive monitoring strategies include

- \* Monitoring of queue and buffer levels, e.g. via Active Queue Management (e.g. [RFC7567])
- \* Monitoring of per-flow counters
- \* Measurement of link statistics such as traffic volume, bandwidth, and QoS
- \* Detection of dropped packets
- \* Use of commercially available Network Monitoring tools

Examples of active monitoring include

- \* In-band timing measurements (such as packet arrival times) e.g. by timestamping and packet inspection
- \* Use of OAM. For DetNet-specific OAM considerations see [I-D.ietf-detnet-ip-oam], [I-D.ietf-detnet-mpls-oam]. Note: At the time of this writing, specifics of DPA have not been

developed for the DetNet OAM, but could be a subject for future investigation

- \* For OAM for Ethernet specifically, see also Connectivity Fault Management (CFM, [IEEE802.1Q]) which defines protocols and practices for OAM for paths through 802.1 bridges and LANs
- \* Out-of-band detection. following the data path or parts of a data path, for example Bidirectional Forwarding Detection (BFD, e.g. [RFC5880])

Note that for some measurements (e.g. packet delay) it may be necessary to make and reconcile measurements from more than one physical location (e.g. a source and destination), possibly in both directions, in order to arrive at a given performance indicator value.

#### Notification Mechanisms

Making DPA measurement results available at the right place(s) and time(s) to effect timely response can be challenging. Two notification mechanisms that are in general use are Netconf/YANG Notifications (e.g. [RFC5880]) and the proprietary local telemetry interfaces provided with components from some vendors. The CoAP Observe Option ([RFC7641]) could also be relevant to such scenarios.

At the time of this writing YANG Notifications are not addressed by the DetNet YANG drafts, however this may be a topic for future work. It is possible that some of the passive mechanisms could be covered by notifications from non-DetNet-specific YANG modules; for example if there is OAM or other performance monitoring that can monitor delay bounds then that could have its own associated YANG model which could be relevant to DetNet, for example some "threshold" values for timing measurement notifications.

At the time of this writing there is an IETF Working Group for network/performance monitoring (IP Performance Measurement, ippm). See also previous work by the completed Remote Network Monitoring Working Group (rmonmib). See also [RFC6632], An Overview of the IETF Network Management Standards.

Vendor-specific local telemetry may be available on some commercially available systems, whereby the system can be programmed (via a proprietary dedicated port and API) to monitor and report on specific conditions, based on both passive and active measurements.

## Related attacks

Performance analytics can be used to detect various attacks, including the ones described in Section 5.2.1 (Delay Attack), Section 5.2.3 (Resource Segmentation Attack), and Section 5.2.7 (Time Synchronization Attack). Once detection and notification have occurred, the appropriate action can be taken to mitigate the threat.

For example, in the case of data plane delay attacks, one possible mitigation is to timestamp the data at the source, and timestamp it again at the destination, and if the resulting latency does not meet the service agreement, take appropriate action. Note that DetNet specifies packet sequence numbering, however it does not specify use of packet timestamps, although they may be used by the underlying transport (for example TSN, [IEEE802.1BA]) to provide the service.

## 7.8. Mitigation Summary

The following table maps the attacks of Section 5, Security Threats, to the impacts of Section 6, Security Threat Impacts, and to the mitigations of the current section. Each row specifies an attack, the impact of this attack if it is successfully implemented, and possible mitigation methods.

Attack	Impact	Mitigations
Delay Attack	-Non-deterministic delay -Data disruption -Increased resource consumption	-Path redundancy -Performance analytics
Reconnaissance	-Enabler for other attacks	-Encryption -Dummy traffic insertion
DetNet Flow Modification or Spoofing	-Increased resource consumption -Data disruption	-Path redundancy -Integrity protection -DetNet Node authentication
Inter-Segment Attack	-Increased resource consumption -Data disruption	-Path redundancy -Performance analytics



Replication: Increased attack surface	-All impacts of other attacks	-Integrity protection -DetNet Node authentication -Encryption
Replication-related Header Manipulation	-Non-deterministic delay -Data disruption	-Integrity protection -DetNet Node authentication
Path Manipulation	-Enabler for other attacks	-Control and signaling message protection
Path Choice: Increased Attack Surface	-All impacts of other attacks	-Control and signaling message protection
Control or Signaling Packet Modification	-Increased resource consumption -Non-deterministic delay -Data disruption	-Control and signaling message protection
Control or Signaling Packet Injection	-Increased resource consumption -Non-deterministic delay -Data disruption	-Control and signaling message protection
Attacks on Time Synchronization Mechanisms	-Non-deterministic delay -Increased resource consumption -Data disruption	-Path redundancy -Control and signaling message protection -Performance analytics

Figure 3: Mapping Attacks to Impact and Mitigations

## 8. Association of Attacks to Use Cases

Different attacks can have different impact and/or mitigation depending on the use case, so we would like to make this association in our analysis. However since there is a potentially unbounded list of use cases, we categorize the attacks with respect to the common themes of the use cases as identified in the Use Case Common Themes section of the DetNet Use Cases [RFC8578].

See also Figure 2 for a mapping of the impact of attacks per use case by industry.

#### 8.1. Association of Attacks to Use Case Common Themes

In this section we review each theme and discuss the attacks that are applicable to that theme, as well as anything specific about the impact and mitigations for that attack with respect to that theme. The table Figure 5, Mapping Between Themes and Attacks, then provides a summary of the attacks that are applicable to each theme.

##### 8.1.1. Sub-Network Layer

DetNet is expected to run over various transmission mediums, with Ethernet being the first identified. Attacks such as Delay or Reconnaissance might be implemented differently on a different transmission medium, however the impact on the DetNet as a whole would be essentially the same. We thus conclude that all attacks and impacts that would be applicable to DetNet over Ethernet (i.e. all those named in this document) would also be applicable to DetNet over other transmission mediums.

With respect to mitigations, some methods are specific to the Ethernet medium, for example time-aware scheduling using 802.1Qbv [IEEE802.1Qbv-2015] can protect against excessive use of bandwidth at the ingress - for other mediums, other mitigations would have to be implemented to provide analogous protection.

##### 8.1.2. Central Administration

A DetNet network can be controlled by a centralized network configuration and control system. Such a system may be in a single central location, or it may be distributed across multiple control entities that function together as a unified control system for the network.

All attacks named in this document which are relevant to controller plane packets (and the controller itself) are relevant to this theme, including Path Manipulation, Path Choice, Control Packet Modification or Injection, Reconnaissance and Attacks on Time Synchronization Mechanisms.

##### 8.1.3. Hot Swap

A DetNet network is not expected to be "plug and play" - it is expected that there is some centralized network configuration and control system. However, the ability to "hot swap" components (e.g. due to malfunction) is similar enough to "plug and play" that this

kind of behavior may be expected in DetNet networks, depending on the implementation.

An attack surface related to Hot Swap is that the DetNet network must at least consider input at runtime from components that were not part of the initial configuration of the network. Even a "perfect" (or "hitless") replacement of a component at runtime would not necessarily be ideal, since presumably one would want to distinguish it from the original for OAM purposes (e.g. to report hot swap of a failed component).

This implies that an attack such as Flow Modification, Spoofing or Inter-segment (which could introduce packets from a "new" component, i.e. one heretofore unknown on the network) could be used to exploit the need to consider such packets (as opposed to rejecting them out of hand as one would do if one did not have to consider introduction of a new component).

To mitigate this situation, deployments should provide a method for dynamic and secure registration of new components, and (possibly manual) deregistration and re-keying of retired components. This would avoid the situation in which the network must accommodate potentially insecure packet flows from unknown components.

Similarly if the network was designed to support runtime replacement of a clock component, then presence (or apparent presence) and thus consideration of packets from a new such component could affect the network, or the time synchronization of the network, for example by initiating a new Best Master Clock selection process. These types of attacks should therefore be considered when designing hot swap type functionality (see [RFC7384]).

#### 8.1.4. Data Flow Information Models

DetNet specifies new YANG models ([I-D.ietf-detnet-yang]) which may present new attack surfaces. Per IETF guidelines, security considerations for any YANG model are expected to be part of the YANG model specification, as described in [IETF\_YANG\_SEC].

#### 8.1.5. L2 and L3 Integration

A DetNet network integrates Layer 2 (bridged) networks (e.g. AVB/TSN LAN) and Layer 3 (routed) networks (e.g. IP) via the use of well-known protocols such as IP, MPLS Pseudowire, and Ethernet. Various DetNet drafts address many specific aspects of Layer 2 and Layer 3 integration within a DetNet, and these are not individually referenced here; security considerations for those aspects are

covered within those drafts or within the related subsections of the present document.

Please note that although there are no entries in the L2 and L3 Integration line of the Mapping Between Themes and Attacks table Figure 4, this does not imply that there could be no relevant attacks related to L2-L3 integration.

#### 8.1.6. End-to-End Delivery

Packets that are part of a resource-reserved DetNet flow are not to be dropped by the DetNet due to congestion. Packets may however be dropped for intended reasons, for example security measures. For example, consider the case in which a packet becomes corrupted (whether incidentally or maliciously) such that the resulting flow ID incidentally matches the flow ID of another DetNet flow, potentially resulting in additional unauthorized traffic on the latter. In such a case it may be a security requirement that the system report and/or take some defined action, perhaps when a packet drop count threshold has been reached (see also Section 7.7).

A data plane attack may force packets to be dropped, for example as a result of a Delay attack, Replication/Elimination attack, or Flow Modification attack.

The same result might be obtained by a controller plane attack, e.g. Path Manipulation or Signaling Packet Modification.

An attack may also cause packets that should not be delivered to be delivered, such as by forcing packets from one (e.g. replicated) path to be preferred over another path when they should not be (Replication attack), or by Flow Modification, or by Path Choice or Packet Injection. A Time Synchronization attack could cause a system that was expecting certain packets at certain times to accept unintended packets based on compromised system time or time windowing in the scheduler.

#### 8.1.7. Replacement for Proprietary Fieldbuses and Ethernet-based Networks

There are many proprietary "field buses" used in Industrial and other industries, as well as proprietary non-interoperable deterministic Ethernet-based networks. DetNet is intended to provide an open-standards-based alternative to such buses/networks. In cases where a DetNet intersects with such fieldbuses/networks or their protocols, such as by protocol emulation or access via a gateway, new attack surfaces can be opened.

For example an Inter-Segment or Controller plane attack such as Path Manipulation, Path Choice or Control Packet Modification/Injection could be used to exploit commands specific to such a protocol, or that are interpreted differently by the different protocols or gateway.

#### 8.1.8. Deterministic vs Best-Effort Traffic

Most of the themes described in this document address OT (reserved) DetNet flows - this item is intended to address issues related to IT traffic on a DetNet.

DetNet is intended to support coexistence of time-sensitive operational (OT, deterministic) traffic and information (IT, "best effort") traffic on the same ("unified") network.

With DetNet, this coexistence will become more common, and mitigations will need to be established. The fact that the IT traffic on a DetNet is limited to a corporate controlled network makes this a less difficult problem compared to being exposed to the open Internet, however this aspect of DetNet security should not be underestimated.

An Inter-segment attack can flood the network with IT-type traffic with the intent of disrupting handling of IT traffic, and/or the goal of interfering with OT traffic. Presumably if the DetNet flow reservation and isolation of the DetNet is well-designed (better-designed than the attack) then interference with OT traffic should not result from an attack that floods the network with IT traffic.

The handling of IT traffic (i.e. traffic which by definition is not guaranteed any given deterministic service properties) by the DetNet will by definition not be given the DetNet-specific protections provided to DetNet (resource-reserved) flows. The implication is that the IT traffic on the DetNet network will necessarily have its own specific set of product (component or system) requirements for protection against attacks such as DOS; presumably they will be less stringent than those for OT flows, but nonetheless component and system designers must employ whatever mitigations will meet the specified security requirements for IT traffic for the given component or DetNet.

The network design as a whole also needs to consider possible application-level dependencies of "OT"-type applications on services provided by the "IT part" of the network; for example, does the OT application depend on IT network services such as DNS or OAM? If such dependencies exist, how are malicious packet flows handled? Such considerations are typically outside the scope of DetNet proper,

but nonetheless need to be addressed in the overall DetNet network design for a given use case.

#### 8.1.9. Deterministic Flows

Reserved bandwidth data flows (deterministic flows) must provide the allocated bandwidth, and must be isolated from each other.

A Spoofing or Inter-segment attack which adds packet traffic to a bandwidth-reserved DetNet flow could cause that flow to occupy more bandwidth than it was allocated, resulting in interference with other DetNet flows.

A Flow Modification or Spoofing or Header Manipulation or Control Packet Modification attack could cause packets from one flow to be directed to another flow, thus breaching isolation between the flows.

#### 8.1.10. Unused Reserved Bandwidth

If bandwidth reservations are made for a DetNet flow but the associated bandwidth is not used at any point in time, that bandwidth is made available on the network for best-effort traffic. However, note that security considerations for best-effort traffic on a DetNet network is out of scope of the present document, provided that any such attacks on best-effort traffic do not affect performance for DetNet OT traffic.

#### 8.1.11. Interoperability

The DetNet specifications as a whole are intended to enable an ecosystem in which multiple vendors can create interoperable products, thus promoting component diversity and potentially higher numbers of each component manufactured. Toward that end, the security measures and protocols discussed in this document are intended to encourage interoperability.

Given that the DetNet specifications are unambiguously written and that the implementations are accurate, the property of interoperability should not in and of itself cause security concerns; however, flaws in interoperability between components could result in security weaknesses. The network operator as well as system and component designer can all contribute to reducing such weaknesses through interoperability testing.

#### 8.1.12. Cost Reductions

The DetNet network specifications are intended to enable an ecosystem in which multiple vendors can create interoperable products, thus promoting higher numbers of each component manufactured, promoting cost reduction and cost competition among vendors.

This envisioned breadth of DetNet-enabled products is in general a positive factor, however implementation flaws in any individual component can present an attack surface. In addition, implementation differences between components from different vendors can result in attack surfaces (resulting from their interaction) which may not exist in any individual component.

Network operators can mitigate such concerns through sufficient product and interoperability testing.

#### 8.1.13. Insufficiently Secure Components

The DetNet network specifications are intended to enable an ecosystem in which multiple vendors can create interoperable products, thus promoting component diversity and potentially higher numbers of each component manufactured. However this raises the possibility that a vendor might repurpose for DetNet applications a hardware or software component that was originally designed for operation in an isolated OT network, and thus may not have been designed to be sufficiently secure, or secure at all, against the sorts of attacks described in this document. Deployment of such a component on a DetNet network that is intended to be highly secure may present an attack surface; thus the DetNet network operator may need to take specific actions to protect such components, for example by implementing a secure interface (such as a firewall) to isolate the component from the threats that may be present in the greater network.

#### 8.1.14. DetNet Network Size

DetNet networks range in size from very small, e.g. inside a single industrial machine, to very large, for example a Utility Grid network spanning a whole country.

The size of the network might be related to how the attack is introduced into the network, for example if the entire network is local, there is a threat that power can be cut to the entire network. If the network is large, perhaps only a part of the network is attacked.

A Delay attack might be as relevant to a small network as to a large network, although the amount of delay might be different.

Attacks sourced from IT traffic might be more likely in large networks, since more people might have access to the network, presenting a larger attack surface. Similarly Path Manipulation, Path Choice and Time Synchronization attacks seem more likely relevant to large networks.

#### 8.1.15. Multiple Hops

Large DetNet networks (e.g. a Utility Grid network) may involve many "hops" over various kinds of links for example radio repeaters, microwave links, fiber optic links, etc.

An attacker who has knowledge of the operation of a component or device's internal software (such as "device drivers") may be able to take advantage of this knowledge to design an attack that could exploit flaws (or even the specifics of normal operation) in the communication between the various links.

It is also possible that a large scale DetNet topology containing various kinds of links may not be in as common use as other more homogeneous topologies. This situation may present more opportunity for attackers to exploit software and/or protocol flaws in or between these components, because these components or configurations may not have been sufficiently tested for interoperability (in the way they would be as a result of broad usage). This may be of particular concern to early adopters of new DetNet components or technologies.

Of the attacks we have defined, the ones identified in Section 8.1.14 as germane to large networks are the most relevant.

#### 8.1.16. Level of Service

A DetNet is expected to provide means to configure the network that include querying network path latency, requesting bounded latency for a given DetNet flow, requesting worst case maximum and/or minimum latency for a given path or DetNet flow, and so on. It is an expected case that the network cannot provide a given requested service level. In such cases the network control system should reply that the requested service level is not available (as opposed to accepting the parameter but then not delivering the desired behavior).

Controller plane attacks such as Signaling Packet Modification and Injection could be used to modify or create control traffic that could interfere with the process of a user requesting a level of service and/or the reply from the network.



Reconnaissance could be used to characterize flows and perhaps target specific flows for attack via the controller plane as noted in Section 6.7.

#### 8.1.17. Bounded Latency

DetNet provides the expectation of guaranteed bounded latency.

Delay attacks can cause packets to miss their agreed-upon latency boundaries.

Time Synchronization attacks can corrupt the time reference of the system, resulting in missed latency deadlines (with respect to the "correct" time reference).

#### 8.1.18. Low Latency

Applications may require "extremely low latency" however depending on the application these may mean very different latency values; for example "low latency" across a Utility grid network is on a different time scale than "low latency" in a motor control loop in a small machine. The intent is that the mechanisms for specifying desired latency include wide ranges, and that architecturally there is nothing to prevent arbitrarily low latencies from being implemented in a given network.

Attacks on the controller plane (as described in the Level of Service theme Section 8.1.16) and Delay and Time attacks (as described in the Bounded Latency theme Section 8.1.17) both apply here.

#### 8.1.19. Bounded Jitter (Latency Variation)

DetNet is expected to provide bounded jitter (packet to packet latency variation).

Delay attacks can cause packets to vary in their arrival times, resulting in packet to packet latency variation, thereby violating the jitter specification.

#### 8.1.20. Symmetrical Path Delays

Some applications would like to specify that the transit delay time values be equal for both the transmit and return paths.

Delay attacks can cause path delays to materially differ between paths.

Time Synchronization attacks can corrupt the time reference of the system, resulting in path delays that may be perceived to be different (with respect to the "correct" time reference) even if they are not materially different.

#### 8.1.21. Reliability and Availability

DetNet based systems are expected to be implemented with essentially arbitrarily high availability (for example 99.9999% up time, or even 12 nines). The intent is that the DetNet designs should not make any assumptions about the level of reliability and availability that may be required of a given system, and should define parameters for communicating these kinds of metrics within the network.

Any attack on the system, of any type, can affect its overall reliability and availability, thus in the mapping table Figure 4 we have marked every attack. Since every DetNet depends to a greater or lesser degree on reliability and availability, this essentially means that all networks have to mitigate all attacks, which to a greater or lesser degree defeats the purpose of associating attacks with use cases. It also underscores the difficulty of designing "extremely high reliability" networks.

In practice, network designers can adopt a risk-based approach, in which only those attacks are mitigated whose potential cost is higher than the cost of mitigation.

#### 8.1.22. Redundant Paths

This document expects that each DetNet system will be implemented to some essentially arbitrary level of reliability and/or availability, depending on the use case. A strategy used by DetNet for providing extraordinarily high levels of reliability when justified is to provide redundant paths between which traffic can be seamlessly switched, all the while maintaining the required performance of that system.

Replication-related attacks are by definition applicable here. Controller plane attacks can also interfere with the configuration of redundant paths.

#### 8.1.23. Security Measures

If any of the security mechanisms which protect the DetNet are attacked or subverted, this can result in malfunction of the network. Thus the security systems themselves needs to be robust against attacks.

The general topic of protection of security mechanisms is not unique to DetNet; it is identical to the case of securing any security mechanism for any network. This document addresses these concerns only to the extent that they are unique to DetNet.

## 8.2. Summary of Attack Types per Use Case Common Theme

The List of Attacks table Figure 4 lists the attacks of Section 5, Security Threats, assigning a number to each type of attack. That number is then used as a short form identifier for the attack in Figure 5, Mapping Between Themes and Attacks.

	Attack
1	Delay Attack
2	DetNet Flow Modification or Spoofing
3	Inter-Segment Attack
4	Replication: Increased attack surface
5	Replication-related Header Manipulation
6	Path Manipulation
7	Path Choice: Increased Attack Surface
8	Control or Signaling Packet Modification
9	Control or Signaling Packet Injection
10	Reconnaissance
11	Attacks on Time Synchronization Mechanisms

Figure 4: List of Attacks

The Mapping Between Themes and Attacks table Figure 5 maps the use case themes of [RFC8578] (as also enumerated in this document) to the attacks of Figure 4. Each row specifies a theme, and the attacks relevant to this theme are marked with a '+'. The row items which have no threats associated with them are included in the table for completeness of the list of Use Case Common Themes, and do not have DetNet-specific threats associated with them.

Theme	Attack										
	1	2	3	4	5	6	7	8	9	10	11
Network Layer - AVB/TSN Eth.	+	+	+	+	+	+	+	+	+	+	+
Central Administration						+	+	+	+	+	+
Hot Swap		+	+								+
Data Flow Information Models											
L2 and L3 Integration											
End-to-end Delivery	+	+	+	+	+	+	+	+	+		+
Proprietary Deterministic Ethernet Networks			+			+	+	+	+		
Replacement for Proprietary Fieldbuses			+			+	+	+	+		
Deterministic vs. Best-Effort Traffic			+								
Deterministic Flows	+	+	+		+	+		+			
Unused Reserved Bandwidth		+	+					+	+		
Interoperability											
Cost Reductions											
Insufficiently Secure Components											
DetNet Network Size	+					+	+				+
Multiple Hops	+	+				+	+				+
Level of Service								+	+	+	
Bounded Latency	+										+
Low Latency	+							+	+		+
Bounded Jitter	+										

Symmetric Path Delays	+											+
Reliability and Availability	+	+	+	+	+	+	+	+	+	+	+	+
Redundant Paths				+	+				+	+		
Security Measures												

Figure 5: Mapping Between Themes and Attacks

## 9. Security Considerations for OAM Traffic

This section considers DetNet-specific security considerations for packet traffic that is generated and transmitted over a DetNet as part of OAM (Operations, Administration, and Maintenance). For the purposes of this discussion, OAM traffic falls into one of two basic types:

- o OAM traffic generated by the network itself. The additional bandwidth required for such packets is added by the network administration, presumably transparent to the customer. Security considerations for such traffic are not DetNet-specific (apart from such traffic being subject to the same DetNet-specific security considerations as any other DetNet data flow) and are thus not covered in this document.
- o OAM traffic generated by the customer. From a DetNet security point of view, DetNet security considerations for such traffic are exactly the same as for any other customer data flows.

From the perspective of an attack, OAM traffic is indistinguishable from DetNet traffic and the network needs to be secure against injection, removal, or modification of traffic of any kind, including OAM traffic. A DetNet is sensitive to any form of packet injection, removal or manipulation and in this respect DetNet OAM traffic is no different. Techniques for securing a DetNet against these threats have been discussed elsewhere in this document.

## 10. DetNet Technology-Specific Threats

Section 5, Security Threats, described threats which are independent of a DetNet implementation. This section considers threats specifically related to the IP- and MPLS-specific aspects of DetNet implementations.

The primary security considerations for the data plane specifically are to maintain the integrity of the data and the delivery of the associated DetNet service traversing the DetNet network.

The primary relevant differences between IP and MPLS implementations are in flow identification and OAM methodologies.

As noted in [RFC8655], DetNet operates at the IP layer ( [RFC8939]) and delivers service over sub-layer technologies such as MPLS ([RFC8964]) and IEEE 802.1 Time-Sensitive Networking (TSN) ([I-D.ietf-detnet-ip-over-tsn]). Application flows can be protected through whatever means are provided by the layer and sub-layer technologies. For example, technology-specific encryption may be used, for example for IP flows, IPsec [RFC4301]. For IP over Ethernet (Layer 2) flows using an underlying sub-net, MACSec [IEEE802.1AE-2018] may be appropriate. For some use cases packet integrity protection without encryption may be sufficient.

However, if the DetNet nodes cannot decrypt IPsec traffic, then DetNet flow identification for encrypted IP traffic flows must be performed in a different way than it would be for unencrypted IP DetNet flows. The DetNet IP Data Plane identifies unencrypted flows via a 6-tuple that consists of two IP addresses, the transport protocol ID, two transport protocol port numbers and the DSCP in the IP header. When IPsec is used, the transport header is encrypted and the next protocol ID is an IPsec protocol, usually ESP, and not a transport protocol, leaving only three components of the 6-tuple, which are the two IP addresses and the DSCP. If the IPsec sessions are established by a controller, then this controller could also transmit (in the clear) the Security Parameter Index (SPI) and thus the SPI could be used (in addition to the pair of IP addresses) for flow identification. Identification of DetNet flows over IPsec is further discussed in Section 5.1.2.3. of [RFC8939].

Sections below discuss threats specific to IP and MPLS in more detail.

#### 10.1. IP

The IP protocol has a long history of security considerations and architectural protection mechanisms. From a data plane perspective DetNet does not add or modify any IP header information, so the carriage of DetNet traffic over an IP data plane does not introduce any new security issues that were not there before, apart from those already described in the data-plane-independent threats section Section 5, Security Threats.

Thus the security considerations for a DetNet based on an IP data plane are purely inherited from the rich IP Security literature and code/application base, and the data-plane-independent section of this document.

Maintaining security for IP segments of a DetNet may be more challenging than for the MPLS segments of the network, given that the IP segments of the network may reach the edges of the network, which are more likely to involve interaction with potentially malevolent outside actors. Conversely MPLS is inherently more secure than IP since it is internal to routers and it is well-known how to protect it from outside influence.

Another way to look at DetNet IP security is to consider it in the light of VPN security; as an industry we have a lot of experience with VPNs running through networks with other VPNs, it is well known how to secure the network for that. However for a DetNet we have the additional subtlety that any possible interaction of one packet with another can have a potentially deleterious effect on the time properties of the flows. So the network must provide sufficient isolation between flows, for example by protecting the forwarding bandwidth and related resources so that they are available to detnet traffic, by whatever means are appropriate for the data plane of that network, for example through the use of queueing mechanisms.

In a VPN, bandwidth is generally guaranteed over a period of time, whereas in DetNet it is not aggregated over time. This implies that any VPN-type protection mechanism must also maintain the DetNet timing constraints.

## 10.2. MPLS

An MPLS network carrying DetNet traffic is expected to be a "well-managed" network. Given that this is the case, it is difficult for an attacker to pass a raw MPLS encoded packet into a network because operators have considerable experience at excluding such packets at the network boundaries, as well as excluding MPLS packets being inserted through the use of a tunnel.

MPLS security is discussed extensively in [RFC5920] ("Security Framework for MPLS and GMPLS Networks") to which the reader is referred.

[RFC6941] builds on [RFC5920] by providing additional security considerations that are applicable to the MPLS-TP extensions appropriate to the MPLS Transport Profile [RFC5921], and thus to the operation of DetNet over some types of MPLS network.

[RFC5921] introduces to MPLS new Operations, Administration, and Maintenance (OAM) capabilities, a transport-oriented path protection mechanism, and strong emphasis on static provisioning supported by network management systems.

The operation of DetNet over an MPLS network builds on MPLS and pseudowire encapsulation. Thus for guidance on securing the DetNet elements of DetNet over MPLS the reader is also referred to the security considerations of [RFC4385], [RFC5586], [RFC3985], [RFC6073], and [RFC6478].

Having attended to the conventional aspects of network security it is necessary to attend to the dynamic aspects. The closest experience that the IETF has with securing protocols that are sensitive to manipulation of delay are the two way time transfer protocols (TWTT), which are NTP [RFC5905] and Precision Time Protocol [IEEE1588]. The security requirements for these are described in [RFC7384].

One particular problem that has been observed in operational tests of TWTT protocols is the ability for two closely but not completely synchronized flows to beat and cause a sudden phase hit to one of the flows. This can be mitigated by the careful use of a scheduling system in the underlying packet transport.

Some investigations into protection of MPLS systems against dynamic attacks exist, such as [I-D.ietf-mpls-opportunistic-encrypt]; perhaps deployment of DetNets will encourage additional such investigations.

## 11. IANA Considerations

This document includes no requests from IANA.

## 12. Security Considerations

The security considerations of DetNet networks are presented throughout this document.

## 13. Privacy Considerations

Privacy in the context of DetNet is maintained by the base technologies specific to the DetNet and user traffic. For example TSN can use MACsec, IP can use IPsec, applications can use IP transport protocol-provided methods e.g. TLS and DTLS. MPLS typically uses L2/L3 VPNs combined with the previously mentioned privacy methods.



However, note that reconnaissance threats such as traffic analysis and monitoring of electrical side channels can still cause there to be privacy considerations even when traffic is encrypted.

#### 14. Contributors

The Editor would like to recognize the contributions of the following individuals to this draft.

Subir Das (Applied Communication Sciences)  
150 Mount Airy Road, Basking Ridge, New Jersey, 07920, USA  
email sdas@appcomsci.com

John Dowdell (Airbus Defence and Space)  
Celtic Springs, Newport, NP10 8FZ, United Kingdom  
email john.dowdell.ietf@gmail.com

Henrik Austad (SINTEF Digital)  
Klaebuveien 153, Trondheim, 7037, Norway  
email henrik@austad.us

Norman Finn (Huawei)  
3101 Rio Way, Spring Valley, California 91977, USA  
email nfinn@nfinnconsulting.com

Stewart Bryant (Futurewei Technologies)  
email: stewart.bryant@gmail.com

David Black (Dell EMC)  
176 South Street, Hopkinton, MA 01748, USA  
email: david.black@dell.com

Carsten Bormann (Universitat Bremen TZI)  
Postfach 330440, D-28359 Bremen, Germany  
email: cabo@tzi.org

#### 15. References

##### 15.1. Normative References

- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas,  
"Deterministic Networking Architecture", RFC 8655,  
DOI 10.17487/RFC8655, October 2019,  
<<https://www.rfc-editor.org/info/rfc8655>>.

- [RFC8938] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC 8938, DOI 10.17487/RFC8938, November 2020, <<https://www.rfc-editor.org/info/rfc8938>>.
- [RFC8939] Varga, B., Ed., Farkas, J., Berger, L., Fedyk, D., and S. Bryant, "Deterministic Networking (DetNet) Data Plane: IP", RFC 8939, DOI 10.17487/RFC8939, November 2020, <<https://www.rfc-editor.org/info/rfc8939>>.
- [RFC8964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "Deterministic Networking (DetNet) Data Plane: MPLS", RFC 8964, DOI 10.17487/RFC8964, January 2021, <<https://www.rfc-editor.org/info/rfc8964>>.

## 15.2. Informative References

- [ARINC664P7] ARINC, "ARINC 664 Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network", 2009.
- [I-D.ietf-detnet-flow-information-model] Varga, B., Farkas, J., Cummings, R., Jiang, Y., and D. Fedyk, "DetNet Flow and Service Information Model", draft-ietf-detnet-flow-information-model-14 (work in progress), January 2021.
- [I-D.ietf-detnet-ip-oam] Mirsky, G., Chen, M., and D. Black, "Operations, Administration and Maintenance (OAM) for Deterministic Networks (DetNet) with IP Data Plane", draft-ietf-detnet-ip-oam-01 (work in progress), January 2021.
- [I-D.ietf-detnet-ip-over-mpls] Varga, B., Berger, L., Fedyk, D., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP over MPLS", draft-ietf-detnet-ip-over-mpls-09 (work in progress), October 2020.
- [I-D.ietf-detnet-ip-over-tsn] Varga, B., Farkas, J., Malis, A., and S. Bryant, "DetNet Data Plane: IP over IEEE 802.1 Time Sensitive Networking (TSN)", draft-ietf-detnet-ip-over-tsn-05 (work in progress), December 2020.

- [I-D.ietf-detnet-mpls-oam]  
Mirsky, G. and M. Chen, "Operations, Administration and Maintenance (OAM) for Deterministic Networks (DetNet) with MPLS Data Plane", draft-ietf-detnet-mpls-oam-02 (work in progress), January 2021.
- [I-D.ietf-detnet-mpls-over-udp-ip]  
Varga, B., Farkas, J., Berger, L., Malis, A., and S. Bryant, "DetNet Data Plane: MPLS over UDP/IP", draft-ietf-detnet-mpls-over-udp-ip-08 (work in progress), December 2020.
- [I-D.ietf-detnet-yang]  
Geng, X., Chen, M., Ryoo, Y., Fedyk, D., Rahman, R., and Z. Li, "Deterministic Networking (DetNet) Configuration YANG Model", draft-ietf-detnet-yang-09 (work in progress), November 2020.
- [I-D.ietf-ipsecme-g-ikev2]  
Smyslov, V. and B. Weis, "Group Key Management using IKEv2", draft-ietf-ipsecme-g-ikev2-02 (work in progress), January 2021.
- [I-D.ietf-mpls-opportunistic-encrypt]  
Farrel, A. and S. Farrell, "Opportunistic Security in MPLS Networks", draft-ietf-mpls-opportunistic-encrypt-03 (work in progress), March 2017.
- [I-D.varga-detnet-service-model]  
Varga, B. and J. Farkas, "DetNet Service Model", draft-varga-detnet-service-model-02 (work in progress), May 2017.
- [IEEE1588]  
IEEE, "IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", 2008.
- [IEEE802.1AE-2018]  
IEEE Standards Association, "IEEE Std 802.1AE-2018 MAC Security (MACsec)", 2018,  
<<https://ieeexplore.ieee.org/document/8585421>>.
- [IEEE802.1BA]  
IEEE Standards Association, "IEEE Standard for Local and Metropolitan Area Networks -- Audio Video Bridging (AVB) Systems", 2011,  
<<https://ieeexplore.ieee.org/document/6032690>>.

- [IEEE802.1Q]  
IEEE Standards Association, "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks - Annex J - Connectivity Fault Management", 2014,  
<<https://ieeexplore.ieee.org/document/6991462>>.
- [IEEE802.1Qbv-2015]  
IEEE Standards Association, "IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", 2015,  
<<https://ieeexplore.ieee.org/document/8613095>>.
- [IEEE802.1Qch-2017]  
IEEE Standards Association, "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks--Amendment 29: Cyclic Queuing and Forwarding", 2017,  
<<https://ieeexplore.ieee.org/document/7961303>>.
- [IETF\_YANG\_SEC]  
IETF, "YANG Module Security Considerations", 2018,  
<<https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines>>.
- [IT\_DEF] Wikipedia, "IT Definition", 2020,  
<[https://en.wikiquote.org/wiki/Information\\_technology](https://en.wikiquote.org/wiki/Information_technology)>.
- [OT\_DEF] Wikipedia, "OT Definition", 2020,  
<[https://en.wikipedia.org/wiki/Operational\\_technology](https://en.wikipedia.org/wiki/Operational_technology)>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black,  
"Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474,  
DOI 10.17487/RFC2474, December 1998,  
<<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z.,  
and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998,  
<<https://www.rfc-editor.org/info/rfc2475>>.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552,  
DOI 10.17487/RFC3552, July 2003,  
<<https://www.rfc-editor.org/info/rfc3552>>.

- [RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <<https://www.rfc-editor.org/info/rfc3985>>.
- [RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, DOI 10.17487/RFC4107, June 2005, <<https://www.rfc-editor.org/info/rfc4107>>.
- [RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <<https://www.rfc-editor.org/info/rfc4253>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<https://www.rfc-editor.org/info/rfc4385>>.
- [RFC4432] Harris, B., "RSA Key Exchange for the Secure Shell (SSH) Transport Layer Protocol", RFC 4432, DOI 10.17487/RFC4432, March 2006, <<https://www.rfc-editor.org/info/rfc4432>>.
- [RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <<https://www.rfc-editor.org/info/rfc5586>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <<https://www.rfc-editor.org/info/rfc5920>>.

- [RFC5921] Bocci, M., Ed., Bryant, S., Ed., Frost, D., Ed., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, DOI 10.17487/RFC5921, July 2010, <<https://www.rfc-editor.org/info/rfc5921>>.
- [RFC6071] Frankel, S. and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, DOI 10.17487/RFC6071, February 2011, <<https://www.rfc-editor.org/info/rfc6071>>.
- [RFC6073] Martini, L., Metz, C., Nadeau, T., Bocci, M., and M. Aissaoui, "Segmented Pseudowire", RFC 6073, DOI 10.17487/RFC6073, January 2011, <<https://www.rfc-editor.org/info/rfc6073>>.
- [RFC6274] Gont, F., "Security Assessment of the Internet Protocol Version 4", RFC 6274, DOI 10.17487/RFC6274, July 2011, <<https://www.rfc-editor.org/info/rfc6274>>.
- [RFC6478] Martini, L., Swallow, G., Heron, G., and M. Bocci, "Pseudowire Status for Static Pseudowires", RFC 6478, DOI 10.17487/RFC6478, May 2012, <<https://www.rfc-editor.org/info/rfc6478>>.
- [RFC6562] Perkins, C. and JM. Valin, "Guidelines for the Use of Variable Bit Rate Audio with Secure RTP", RFC 6562, DOI 10.17487/RFC6562, March 2012, <<https://www.rfc-editor.org/info/rfc6562>>.
- [RFC6632] Ersue, M., Ed. and B. Claise, "An Overview of the IETF Network Management Standards", RFC 6632, DOI 10.17487/RFC6632, June 2012, <<https://www.rfc-editor.org/info/rfc6632>>.
- [RFC6941] Fang, L., Ed., Niven-Jenkins, B., Ed., Mansfield, S., Ed., and R. Graveman, Ed., "MPLS Transport Profile (MPLS-TP) Security Framework", RFC 6941, DOI 10.17487/RFC6941, April 2013, <<https://www.rfc-editor.org/info/rfc6941>>.
- [RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <<https://www.rfc-editor.org/info/rfc7384>>.
- [RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <<https://www.rfc-editor.org/info/rfc7567>>.

- [RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641, DOI 10.17487/RFC7641, September 2015, <<https://www.rfc-editor.org/info/rfc7641>>.
- [RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <<https://www.rfc-editor.org/info/rfc7748>>.
- [RFC7835] Saucez, D., Iannone, L., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Threat Analysis", RFC 7835, DOI 10.17487/RFC7835, April 2016, <<https://www.rfc-editor.org/info/rfc7835>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <<https://www.rfc-editor.org/info/rfc8578>>.
- [RS\_DEF] Wikipedia, "RS Definition", 2020, <[https://en.wikipedia.org/wiki/Network\\_segmentation](https://en.wikipedia.org/wiki/Network_segmentation)>.

#### Authors' Addresses

Ethan Grossman (editor)  
Dolby Laboratories, Inc.  
1275 Market Street  
San Francisco, CA 94103  
USA

Phone: +1 415 465 4339  
Email: [ethan@ieee.org](mailto:ethan@ieee.org)  
URI: <http://www.dolby.com>

Tal Mizrahi  
Huawei Network.IO Innovation Lab  
  
Email: [tal.mizrahi.phd@gmail.com](mailto:tal.mizrahi.phd@gmail.com)

Andrew J. Hacker  
MistIQ Technologies, Inc  
Harrisburg, PA  
USA

Email: [ajhacker@mistiqttech.com](mailto:ajhacker@mistiqttech.com)



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: August 27, 2018

E. Grossman, Ed.  
DOLBY  
February 23, 2018

Deterministic Networking Use Cases  
draft-ietf-detnet-use-cases-14

Abstract

This draft documents requirements in several diverse industries to establish multi-hop paths for characterized flows with deterministic properties. In this context deterministic implies that streams can be established which provide guaranteed bandwidth and latency which can be established from either a Layer 2 or Layer 3 (IP) interface, and which can co-exist on an IP network with best-effort traffic.

Additional requirements include optional redundant paths, very high reliability paths, time synchronization, and clock distribution. Industries considered include professional audio, electrical utilities, building automation systems, wireless for industrial, cellular radio, industrial machine-to-machine, mining, private blockchain, and network slicing.

For each case, this document will identify the application, identify representative solutions used today, and the improvements IETF DetNet solutions may enable.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 27, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	5
2. Pro Audio and Video . . . . .	6
2.1. Use Case Description . . . . .	6
2.1.1. Uninterrupted Stream Playback . . . . .	7
2.1.2. Synchronized Stream Playback . . . . .	7
2.1.3. Sound Reinforcement . . . . .	8
2.1.4. Deterministic Time to Establish Streaming . . . . .	8
2.1.5. Secure Transmission . . . . .	8
2.1.5.1. Safety . . . . .	8
2.2. Pro Audio Today . . . . .	9
2.3. Pro Audio Future . . . . .	9
2.3.1. Layer 3 Interconnecting Layer 2 Islands . . . . .	9
2.3.2. High Reliability Stream Paths . . . . .	9
2.3.3. Integration of Reserved Streams into IT Networks . . . . .	9
2.3.4. Use of Unused Reservations by Best-Effort Traffic . . . . .	10
2.3.5. Traffic Segregation . . . . .	10
2.3.5.1. Packet Forwarding Rules, VLANs and Subnets . . . . .	10
2.3.5.2. Multicast Addressing (IPv4 and IPv6) . . . . .	11
2.3.6. Latency Optimization by a Central Controller . . . . .	11
2.3.7. Reduced Device Cost Due To Reduced Buffer Memory . . . . .	11
2.4. Pro Audio Asks . . . . .	12
3. Electrical Utilities . . . . .	12
3.1. Use Case Description . . . . .	12
3.1.1. Transmission Use Cases . . . . .	12
3.1.1.1. Protection . . . . .	12
3.1.1.2. Intra-Substation Process Bus Communications . . . . .	18
3.1.1.3. Wide Area Monitoring and Control Systems . . . . .	19
3.1.1.4. IEC 61850 WAN engineering guidelines requirement classification . . . . .	20
3.1.2. Generation Use Case . . . . .	21
3.1.2.1. Control of the Generated Power . . . . .	21

3.1.2.2.	Control of the Generation Infrastructure . . . . .	22
3.1.3.	Distribution use case . . . . .	27
3.1.3.1.	Fault Location Isolation and Service Restoration (FLISR) . . . . .	27
3.2.	Electrical Utilities Today . . . . .	28
3.2.1.	Security Current Practices and Limitations . . . . .	28
3.3.	Electrical Utilities Future . . . . .	30
3.3.1.	Migration to Packet-Switched Network . . . . .	31
3.3.2.	Telecommunications Trends . . . . .	31
3.3.2.1.	General Telecommunications Requirements . . . . .	31
3.3.2.2.	Specific Network topologies of Smart Grid Applications . . . . .	32
3.3.2.3.	Precision Time Protocol . . . . .	33
3.3.3.	Security Trends in Utility Networks . . . . .	34
3.4.	Electrical Utilities Asks . . . . .	36
4.	Building Automation Systems . . . . .	36
4.1.	Use Case Description . . . . .	36
4.2.	Building Automation Systems Today . . . . .	37
4.2.1.	BAS Architecture . . . . .	37
4.2.2.	BAS Deployment Model . . . . .	38
4.2.3.	Use Cases for Field Networks . . . . .	40
4.2.3.1.	Environmental Monitoring . . . . .	40
4.2.3.2.	Fire Detection . . . . .	40
4.2.3.3.	Feedback Control . . . . .	41
4.2.4.	Security Considerations . . . . .	41
4.3.	BAS Future . . . . .	41
4.4.	BAS Asks . . . . .	42
5.	Wireless for Industrial . . . . .	42
5.1.	Use Case Description . . . . .	42
5.1.1.	Network Convergence using 6TiSCH . . . . .	43
5.1.2.	Common Protocol Development for 6TiSCH . . . . .	43
5.2.	Wireless Industrial Today . . . . .	44
5.3.	Wireless Industrial Future . . . . .	44
5.3.1.	Unified Wireless Network and Management . . . . .	44
5.3.1.1.	PCE and 6TiSCH ARQ Retries . . . . .	46
5.3.2.	Schedule Management by a PCE . . . . .	47
5.3.2.1.	PCE Commands and 6TiSCH CoAP Requests . . . . .	47
5.3.2.2.	6TiSCH IP Interface . . . . .	48
5.3.3.	6TiSCH Security Considerations . . . . .	49
5.4.	Wireless Industrial Asks . . . . .	49
6.	Cellular Radio . . . . .	49
6.1.	Use Case Description . . . . .	49
6.1.1.	Network Architecture . . . . .	49
6.1.2.	Delay Constraints . . . . .	50
6.1.3.	Time Synchronization Constraints . . . . .	52
6.1.4.	Transport Loss Constraints . . . . .	54
6.1.5.	Security Considerations . . . . .	54
6.2.	Cellular Radio Networks Today . . . . .	55

6.2.1.	Fronthaul . . . . .	55
6.2.2.	Midhaul and Backhaul . . . . .	55
6.3.	Cellular Radio Networks Future . . . . .	56
6.4.	Cellular Radio Networks Asks . . . . .	58
7.	Industrial M2M . . . . .	58
7.1.	Use Case Description . . . . .	58
7.2.	Industrial M2M Communication Today . . . . .	59
7.2.1.	Transport Parameters . . . . .	60
7.2.2.	Stream Creation and Destruction . . . . .	61
7.3.	Industrial M2M Future . . . . .	61
7.4.	Industrial M2M Asks . . . . .	61
8.	Mining Industry . . . . .	62
8.1.	Use Case Description . . . . .	62
8.2.	Mining Industry Today . . . . .	62
8.3.	Mining Industry Future . . . . .	63
8.4.	Mining Industry Asks . . . . .	64
9.	Private Blockchain . . . . .	64
9.1.	Use Case Description . . . . .	64
9.1.1.	Blockchain Operation . . . . .	64
9.1.2.	Blockchain Network Architecture . . . . .	65
9.1.3.	Security Considerations . . . . .	65
9.2.	Private Blockchain Today . . . . .	65
9.3.	Private Blockchain Future . . . . .	66
9.4.	Private Blockchain Asks . . . . .	66
10.	Network Slicing . . . . .	66
10.1.	Use Case Description . . . . .	66
10.2.	Network Slicing Use Cases . . . . .	67
10.2.1.	Enhanced Mobile Broadband (eMBB) . . . . .	67
10.2.2.	Ultra-Reliable and Low Latency Communications (URLLC) . . . . .	67
10.2.3.	massive Machine Type Communications (mMTC) . . . . .	67
10.3.	Using DetNet in Network Slicing . . . . .	67
10.4.	Network Slicing Today and Future . . . . .	68
10.5.	Network Slicing Asks . . . . .	68
11.	Use Case Common Themes . . . . .	68
11.1.	Unified, standards-based network . . . . .	68
11.1.1.	Extensions to Ethernet . . . . .	68
11.1.2.	Centrally Administered . . . . .	68
11.1.3.	Standardized Data Flow Information Models . . . . .	69
11.1.4.	L2 and L3 Integration . . . . .	69
11.1.5.	Guaranteed End-to-End Delivery . . . . .	69
11.1.6.	Replacement for Multiple Proprietary Deterministic Networks . . . . .	69
11.1.7.	Mix of Deterministic and Best-Effort Traffic . . . . .	69
11.1.8.	Unused Reserved BW to be Available to Best Effort Traffic . . . . .	69
11.1.9.	Lower Cost, Multi-Vendor Solutions . . . . .	70
11.2.	Scalable Size . . . . .	70

11.3.	Scalable Timing Parameters and Accuracy . . . . .	70
11.3.1.	Bounded Latency . . . . .	70
11.3.2.	Low Latency . . . . .	70
11.3.3.	Symmetrical Path Delays . . . . .	71
11.4.	High Reliability and Availability . . . . .	71
11.5.	Security . . . . .	71
11.6.	Deterministic Flows . . . . .	71
12.	Use Cases Explicitly Out of Scope for DetNet . . . . .	71
12.1.	DetNet Scope Limitations . . . . .	72
12.2.	Internet-based Applications . . . . .	72
12.2.1.	Use Case Description . . . . .	72
12.2.1.1.	Media Content Delivery . . . . .	73
12.2.1.2.	Online Gaming . . . . .	73
12.2.1.3.	Virtual Reality . . . . .	73
12.2.2.	Internet-Based Applications Today . . . . .	73
12.2.3.	Internet-Based Applications Future . . . . .	73
12.2.4.	Internet-Based Applications Asks . . . . .	73
12.3.	Pro Audio and Video - Digital Rights Management (DRM) . . . . .	74
12.4.	Pro Audio and Video - Link Aggregation . . . . .	74
13.	Contributors . . . . .	75
14.	Acknowledgments . . . . .	76
14.1.	Pro Audio . . . . .	76
14.2.	Utility Telecom . . . . .	77
14.3.	Building Automation Systems . . . . .	77
14.4.	Wireless for Industrial . . . . .	77
14.5.	Cellular Radio . . . . .	77
14.6.	Industrial M2M . . . . .	77
14.7.	Internet Applications and CoMP . . . . .	78
14.8.	Electrical Utilities . . . . .	78
14.9.	Network Slicing . . . . .	78
14.10.	Mining . . . . .	78
14.11.	Private Blockchain . . . . .	78
15.	Informative References . . . . .	78
	Author's Address . . . . .	88

## 1. Introduction

This draft presents use cases from diverse industries which have in common a need for deterministic streams, but which also differ notably in their network topologies and specific desired behavior. Together, they provide broad industry context for DetNet and a yardstick against which proposed DetNet designs can be measured (to what extent does a proposed design satisfy these various use cases?)

For DetNet, use cases explicitly do not define requirements; The DetNet WG will consider the use cases, decide which elements are in scope for DetNet, and the results will be incorporated into future drafts. Similarly, the DetNet use case draft explicitly does not

suggest any specific design, architecture or protocols, which will be topics of future drafts.

We present for each use case the answers to the following questions:

- o What is the use case?
- o How is it addressed today?
- o How would you like it to be addressed in the future?
- o What do you want the IETF to deliver?

The level of detail in each use case should be sufficient to express the relevant elements of the use case, but not more.

At the end we consider the use cases collectively, and examine the most significant goals they have in common.

## 2. Pro Audio and Video

### 2.1. Use Case Description

The professional audio and video industry ("ProAV") includes:

- o Music and film content creation
- o Broadcast
- o Cinema
- o Live sound
- o Public address, media and emergency systems at large venues (airports, stadiums, churches, theme parks).

These industries have already transitioned audio and video signals from analog to digital. However, the digital interconnect systems remain primarily point-to-point with a single (or small number of) signals per link, interconnected with purpose-built hardware.

These industries are now transitioning to packet-based infrastructure to reduce cost, increase routing flexibility, and integrate with existing IT infrastructure.

Today ProAV applications have no way to establish deterministic streams from a standards-based Layer 3 (IP) interface, which is a fundamental limitation to the use cases described here. Today

deterministic streams can be created within standards-based layer 2 LANs (e.g. using IEEE 802.1 AVB) however these are not routable via IP and thus are not effective for distribution over wider areas (for example broadcast events that span wide geographical areas).

It would be highly desirable if such streams could be routed over the open Internet, however solutions with more limited scope (e.g. enterprise networks) would still provide a substantial improvement.

The following sections describe specific ProAV use cases.

#### 2.1.1.1. Uninterrupted Stream Playback

Transmitting audio and video streams for live playback is unlike common file transfer because uninterrupted stream playback in the presence of network errors cannot be achieved by re-trying the transmission; by the time the missing or corrupt packet has been identified it is too late to execute a re-try operation. Buffering can be used to provide enough delay to allow time for one or more retries, however this is not an effective solution in applications where large delays (latencies) are not acceptable (as discussed below).

Streams with guaranteed bandwidth can eliminate congestion on the network as a cause of transmission errors that would lead to playback interruption. Use of redundant paths can further mitigate transmission errors to provide greater stream reliability.

#### 2.1.1.2. Synchronized Stream Playback

Latency in this context is the time between when a signal is initially sent over a stream and when it is received. A common example in ProAV is time-synchronizing audio and video when they take separate paths through the playback system. In this case the latency of both the audio and video streams must be bounded and consistent if the sound is to remain matched to the movement in the video. A common tolerance for audio/video sync is one NTSC video frame (about 33ms) and to maintain the audience perception of correct lip sync the latency needs to be consistent within some reasonable tolerance, for example 10%.

A common architecture for synchronizing multiple streams that have different paths through the network (and thus potentially different latencies) is to enable measurement of the latency of each path, and have the data sinks (for example speakers) delay (buffer) all packets on all but the slowest path. Each packet of each stream is assigned a presentation time which is based on the longest required delay. This implies that all sinks must maintain a common time reference of

sufficient accuracy, which can be achieved by any of various techniques.

This type of architecture is commonly implemented using a central controller that determines path delays and arbitrates buffering delays.

#### 2.1.3. Sound Reinforcement

Consider the latency (delay) from when a person speaks into a microphone to when their voice emerges from the speaker. If this delay is longer than about 10-15 milliseconds it is noticeable and can make a sound reinforcement system unusable (see slide 6 of [SRP\_LATENCY]). (If you have ever tried to speak in the presence of a delayed echo of your voice you may know this experience).

Note that the 15ms latency bound includes all parts of the signal path, not just the network, so the network latency must be significantly less than 15ms.

In some cases local performers must perform in synchrony with a remote broadcast. In such cases the latencies of the broadcast stream and the local performer must be adjusted to match each other, with a worst case of one video frame (33ms for NTSC video).

In cases where audio phase is a consideration, for example beam-forming using multiple speakers, latency requirements can be in the 10 microsecond range (1 audio sample at 96kHz).

#### 2.1.4. Deterministic Time to Establish Streaming

Note: The WG has decided that guidelines for deterministic time to establish stream startup is not within scope of DetNet. If bounded timing of establishing or re-establish streams is required in a given use case, it is up to the application/system to achieve this. (The supporting text from this section has been removed as of draft 12).

#### 2.1.5. Secure Transmission

##### 2.1.5.1. Safety

Professional audio systems can include amplifiers that are capable of generating hundreds or thousands of watts of audio power which if used incorrectly can cause hearing damage to those in the vicinity. Apart from the usual care required by the systems operators to prevent such incidents, the network traffic that controls these devices must be secured (as with any sensitive application traffic).



## 2.2. Pro Audio Today

Some proprietary systems have been created which enable deterministic streams at Layer 3 however they are "engineered networks" which require careful configuration to operate, often require that the system be over-provisioned, and it is implied that all devices on the network voluntarily play by the rules of that network. To enable these industries to successfully transition to an interoperable multi-vendor packet-based infrastructure requires effective open standards, and we believe that establishing relevant IETF standards is a crucial factor.

## 2.3. Pro Audio Future

### 2.3.1. Layer 3 Interconnecting Layer 2 Islands

It would be valuable to enable IP to connect multiple Layer 2 LANs.

As an example, ESPN recently constructed a state-of-the-art 194,000 sq ft, \$125 million broadcast studio called DC2. The DC2 network is capable of handling 46 Tbps of throughput with 60,000 simultaneous signals. Inside the facility are 1,100 miles of fiber feeding four audio control rooms (see [ESPN\_DC2] ).

In designing DC2 they replaced as much point-to-point technology as they could with packet-based technology. They constructed seven individual studios using layer 2 LANS (using IEEE 802.1 AVB) that were entirely effective at routing audio within the LANs. However to interconnect these layer 2 LAN islands together they ended up using dedicated paths in a custom SDN (Software Defined Networking) router because there is no standards-based routing solution available.

### 2.3.2. High Reliability Stream Paths

On-air and other live media streams are often backed up with redundant links that seamlessly act to deliver the content when the primary link fails for any reason. In point-to-point systems this is provided by an additional point-to-point link; the analogous requirement in a packet-based system is to provide an alternate path through the network such that no individual link can bring down the system.

### 2.3.3. Integration of Reserved Streams into IT Networks

A commonly cited goal of moving to a packet based media infrastructure is that costs can be reduced by using off the shelf, commodity network hardware. In addition, economy of scale can be realized by combining media infrastructure with IT infrastructure.

In keeping with these goals, stream reservation technology should be compatible with existing protocols, and not compromise use of the network for best effort (non-time-sensitive) traffic.

#### 2.3.4. Use of Unused Reservations by Best-Effort Traffic

In cases where stream bandwidth is reserved but not currently used (or is under-utilized) that bandwidth must be available to best-effort (i.e. non-time-sensitive) traffic. For example a single stream may be nailed up (reserved) for specific media content that needs to be presented at different times of the day, ensuring timely delivery of that content, yet in between those times the full bandwidth of the network can be utilized for best-effort tasks such as file transfers.

This also addresses a concern of IT network administrators that are considering adding reserved bandwidth traffic to their networks that ("users will reserve large quantities of bandwidth and then never un-reserve it even though they are not using it, and soon the network will have no bandwidth left").

#### 2.3.5. Traffic Segregation

Note: It is still under WG discussion whether this topic will be addressed by DetNet.

Sink devices may be low cost devices with limited processing power. In order to not overwhelm the CPUs in these devices it is important to limit the amount of traffic that these devices must process.

As an example, consider the use of individual seat speakers in a cinema. These speakers are typically required to be cost reduced since the quantities in a single theater can reach hundreds of seats. Discovery protocols alone in a one thousand seat theater can generate enough broadcast traffic to overwhelm a low powered CPU. Thus an installation like this will benefit greatly from some type of traffic segregation that can define groups of seats to reduce traffic within each group. All seats in the theater must still be able to communicate with a central controller.

There are many techniques that can be used to support this requirement including (but not limited to) the following examples.

##### 2.3.5.1. Packet Forwarding Rules, VLANs and Subnets

Packet forwarding rules can be used to eliminate some extraneous streaming traffic from reaching potentially low powered sink devices,

however there may be other types of broadcast traffic that should be eliminated using other means for example VLANs or IP subnets.

#### 2.3.5.2. Multicast Addressing (IPv4 and IPv6)

Multicast addressing is commonly used to keep bandwidth utilization of shared links to a minimum.

Because of the MAC Address forwarding nature of Layer 2 bridges it is important that a multicast MAC address is only associated with one stream. This will prevent reservations from forwarding packets from one stream down a path that has no interested sinks simply because there is another stream on that same path that shares the same multicast MAC address.

Since each multicast MAC Address can represent 32 different IPv4 multicast addresses there must be a process put in place to make sure this does not occur. Requiring use of IPv6 address can achieve this, however due to their continued prevalence, solutions that are effective for IPv4 installations are also required.

#### 2.3.6. Latency Optimization by a Central Controller

A central network controller might also perform optimizations based on the individual path delays, for example sinks that are closer to the source can inform the controller that they can accept greater latency since they will be buffering packets to match presentation times of farther away sinks. The controller might then move a stream reservation on a short path to a longer path in order to free up bandwidth for other critical streams on that short path. See slides 3-5 of [SRP\_LATENCY].

Additional optimization can be achieved in cases where sinks have differing latency requirements, for example in a live outdoor concert the speaker sinks have stricter latency requirements than the recording hardware sinks. See slide 7 of [SRP\_LATENCY].

#### 2.3.7. Reduced Device Cost Due To Reduced Buffer Memory

Device cost can be reduced in a system with guaranteed reservations with a small bounded latency due to the reduced requirements for buffering (i.e. memory) on sink devices. For example, a theme park might broadcast a live event across the globe via a layer 3 protocol; in such cases the size of the buffers required is proportional to the latency bounds and jitter caused by delivery, which depends on the worst case segment of the end-to-end network path. For example on today's open internet the latency is typically unacceptable for audio and video streaming without many seconds of buffering. In such

scenarios a single gateway device at the local network that receives the feed from the remote site would provide the expensive buffering required to mask the latency and jitter issues associated with long distance delivery. Sink devices in the local location would have no additional buffering requirements, and thus no additional costs, beyond those required for delivery of local content. The sink device would be receiving the identical packets as those sent by the source and would be unaware that there were any latency or jitter issues along the path.

#### 2.4. Pro Audio Asks

- o Layer 3 routing on top of AVB (and/or other high QoS networks)
- o Content delivery with bounded, lowest possible latency
- o IntServ and DiffServ integration with AVB (where practical)
- o Single network for A/V and IT traffic
- o Standards-based, interoperable, multi-vendor
- o IT department friendly
- o Enterprise-wide networks (e.g. size of San Francisco but not the whole Internet (yet...))

### 3. Electrical Utilities

#### 3.1. Use Case Description

Many systems that an electrical utility deploys today rely on high availability and deterministic behavior of the underlying networks. Here we present use cases in Transmission, Generation and Distribution, including key timing and reliability metrics. We also discuss security issues and industry trends which affect the architecture of next generation utility networks

##### 3.1.1. Transmission Use Cases

###### 3.1.1.1. Protection

Protection means not only the protection of human operators but also the protection of the electrical equipment and the preservation of the stability and frequency of the grid. If a fault occurs in the transmission or distribution of electricity then severe damage can occur to human operators, electrical equipment and the grid itself, leading to blackouts.

Communication links in conjunction with protection relays are used to selectively isolate faults on high voltage lines, transformers, reactors and other important electrical equipment. The role of the teleprotection system is to selectively disconnect a faulty part by transferring command signals within the shortest possible time.

#### 3.1.1.1.1. Key Criteria

The key criteria for measuring teleprotection performance are command transmission time, dependability and security. These criteria are defined by the IEC standard 60834 as follows:

- o Transmission time (Speed): The time between the moment where state changes at the transmitter input and the moment of the corresponding change at the receiver output, including propagation delay. Overall operating time for a teleprotection system includes the time for initiating the command at the transmitting end, the propagation delay over the network (including equipments) and the selection and decision time at the receiving end, including any additional delay due to a noisy environment.
- o Dependability: The ability to issue and receive valid commands in the presence of interference and/or noise, by minimizing the probability of missing command (PMC). Dependability targets are typically set for a specific bit error rate (BER) level.
- o Security: The ability to prevent false tripping due to a noisy environment, by minimizing the probability of unwanted commands (PUC). Security targets are also set for a specific bit error rate (BER) level.

Additional elements of the the teleprotection system that impact its performance include:

- o Network bandwidth
- o Failure recovery capacity (aka resiliency)

#### 3.1.1.1.2. Fault Detection and Clearance Timing

Most power line equipment can tolerate short circuits or faults for up to approximately five power cycles before sustaining irreversible damage or affecting other segments in the network. This translates to total fault clearance time of 100ms. As a safety precaution, however, actual operation time of protection systems is limited to 70- 80 percent of this period, including fault recognition time, command transmission time and line breaker switching time.

Some system components, such as large electromechanical switches, require particularly long time to operate and take up the majority of the total clearance time, leaving only a 10ms window for the telecommunications part of the protection scheme, independent of the distance to travel. Given the sensitivity of the issue, new networks impose requirements that are even more stringent: IEC standard 61850 limits the transfer time for protection messages to  $1/4 - 1/2$  cycle or 4 - 8ms (for 60Hz lines) for the most critical messages.

#### 3.1.1.1.3. Symmetric Channel Delay

Note: It is currently under WG discussion whether symmetric path delays are to be guaranteed by DetNet.

Teleprotection channels which are differential must be synchronous, which means that any delays on the transmit and receive paths must match each other. Teleprotection systems ideally support zero asymmetric delay; typical legacy relays can tolerate delay discrepancies of up to 750us.

Some tools available for lowering delay variation below this threshold are:

- o For legacy systems using Time Division Multiplexing (TDM), jitter buffers at the multiplexers on each end of the line can be used to offset delay variation by queuing sent and received packets. The length of the queues must balance the need to regulate the rate of transmission with the need to limit overall delay, as larger buffers result in increased latency.
- o For jitter-prone IP packet networks, traffic management tools can ensure that the teleprotection signals receive the highest transmission priority to minimize jitter.
- o Standard packet-based synchronization technologies, such as 1588-2008 Precision Time Protocol (PTP) and Synchronous Ethernet (Sync-E), can help keep networks stable by maintaining a highly accurate clock source on the various network devices.

#### 3.1.1.1.4. Teleprotection Network Requirements (IEC 61850)

The following table captures the main network metrics as based on the IEC 61850 standard.

Teleprotection Requirement	Attribute
One way maximum delay	4-10 ms
Asymmetric delay required	Yes
Maximum jitter	less than 250 us (750 us for legacy IED)
Topology	Point to point, point to Multi-point
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1% to 1%

Table 1: Teleprotection network requirements

## 3.1.1.1.5. Inter-Trip Protection scheme

"Inter-tripping" is the signal-controlled tripping of a circuit breaker to complete the isolation of a circuit or piece of apparatus in concert with the tripping of other circuit breakers.

Inter-Trip protection Requirement	Attribute
One way maximum delay	5 ms
Asymmetric delay required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 2: Inter-Trip protection network requirements

## 3.1.1.1.6. Current Differential Protection Scheme

Current differential protection is commonly used for line protection, and is typical for protecting parallel circuits. At both end of the lines the current is measured by the differential relays, and both relays will trip the circuit breaker if the current going into the line does not equal the current going out of the line. This type of protection scheme assumes some form of communications being present between the relays at both end of the line, to allow both relays to compare measured current values. Line differential protection schemes assume a very low telecommunications delay between both relays, often as low as 5ms. Moreover, as those systems are often not time-synchronized, they also assume symmetric telecommunications paths with constant delay, which allows comparing current measurement values taken at the exact same time.

Current Differential protection Requirement	Attribute
One way maximum delay	5 ms
Asymmetric delay Required	Yes
Maximum jitter	less than 250 us (750us for legacy IED)
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 3: Current Differential Protection metrics

## 3.1.1.1.7. Distance Protection Scheme

Distance (Impedance Relay) protection scheme is based on voltage and current measurements. The network metrics are similar (but not identical to) Current Differential protection.



Distance protection Requirement	Attribute
One way maximum delay	5 ms
Asymmetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 4: Distance Protection requirements

#### 3.1.1.1.8. Inter-Substation Protection Signaling

This use case describes the exchange of Sampled Value and/or GOOSE (Generic Object Oriented Substation Events) message between Intelligent Electronic Devices (IED) in two substations for protection and tripping coordination. The two IEDs are in a master-slave mode.

The Current Transformer or Voltage Transformer (CT/VT) in one substation sends the sampled analog voltage or current value to the Merging Unit (MU) over hard wire. The MU sends the time-synchronized 61850-9-2 sampled values to the slave IED. The slave IED forwards the information to the Master IED in the other substation. The master IED makes the determination (for example based on sampled value differentials) to send a trip command to the originating IED. Once the slave IED/Relay receives the GOOSE trip for breaker tripping, it opens the breaker. It then sends a confirmation message back to the master. All data exchanges between IEDs are either through Sampled Value and/or GOOSE messages.

Inter-Substation protection Requirement	Attribute
One way maximum delay	5 ms
Asymmetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	1%

Table 5: Inter-Substation Protection requirements

#### 3.1.1.2. Intra-Substation Process Bus Communications

This use case describes the data flow from the CT/VT to the IEDs in the substation via the MU. The CT/VT in the substation send the analog voltage or current values to the MU over hard wire. The MU converts the analog values into digital format (typically time-synchronized Sampled Values as specified by IEC 61850-9-2) and sends them to the IEDs in the substation. The GPS Master Clock can send 1PPS or IRIG-B format to the MU through a serial port or IEEE 1588 protocol via a network. Process bus communication using 61850 simplifies connectivity within the substation and removes the requirement for multiple serial connections and removes the slow serial bus architectures that are typically used. This also ensures increased flexibility and increased speed with the use of multicast messaging between multiple devices.

Intra-Substation protection Requirement	Attribute
One way maximum delay	5 ms
Asymetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on Node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes - No
Packet loss	0.1%

Table 6: Intra-Substation Protection requirements

#### 3.1.1.1.3. Wide Area Monitoring and Control Systems

The application of synchrophasor measurement data from Phasor Measurement Units (PMU) to Wide Area Monitoring and Control Systems promises to provide important new capabilities for improving system stability. Access to PMU data enables more timely situational awareness over larger portions of the grid than what has been possible historically with normal SCADA (Supervisory Control and Data Acquisition) data. Handling the volume and real-time nature of synchrophasor data presents unique challenges for existing application architectures. Wide Area management System (WAMS) makes it possible for the condition of the bulk power system to be observed and understood in real-time so that protective, preventative, or corrective action can be taken. Because of the very high sampling rate of measurements and the strict requirement for time synchronization of the samples, WAMS has stringent telecommunications requirements in an IP network that are captured in the following table:

WAMS Requirement	Attribute
One way maximum delay	50 ms
Asymetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point, Multi-point to Multi-point
Bandwidth	100 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on Node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	1%
Consecutive Packet Loss	At least 1 packet per application cycle must be received.

Table 7: WAMS Special Communication Requirements

#### 3.1.1.4. IEC 61850 WAN engineering guidelines requirement classification

The IEC (International Electrotechnical Commission) has recently published a Technical Report which offers guidelines on how to define and deploy Wide Area Networks for the interconnections of electric substations, generation plants and SCADA operation centers. The IEC 61850-90-12 is providing a classification of WAN communication requirements into 4 classes. Table 8 summarizes these requirements:

WAN Requirement	Class WA	Class WB	Class WC	Class WD
Application field	EHV (Extra High Voltage)	HV (High Voltage)	MV (Medium Voltage)	General purpose
Latency	5 ms	10 ms	100 ms	> 100 ms
Jitter	10 us	100 us	1 ms	10 ms
Latency Asymetry	100 us	1 ms	10 ms	100 ms
Time Accuracy	1 us	10 us	100 us	10 to 100 ms
Bit Error rate	10 <sup>-7</sup> to 10 <sup>-6</sup>	10 <sup>-5</sup> to 10 <sup>-4</sup>	10 <sup>-3</sup>	
Unavailability	10 <sup>-7</sup> to 10 <sup>-6</sup>	10 <sup>-5</sup> to 10 <sup>-4</sup>	10 <sup>-3</sup>	
Recovery delay	Zero	50 ms	5 s	50 s
Cyber security	extremely high	High	Medium	Medium

Table 8: 61850-90-12 Communication Requirements; Courtesy of IEC

### 3.1.2. Generation Use Case

Energy generation systems are complex infrastructures that require control of both the generated power and the generation infrastructure.

#### 3.1.2.1. Control of the Generated Power

The electrical power generation frequency must be maintained within a very narrow band. Deviations from the acceptable frequency range are detected and the required signals are sent to the power plants for frequency regulation.

Automatic Generation Control (AGC) is a system for adjusting the power output of generators at different power plants, in response to changes in the load.

FCAG (Frequency Control Automatic Generation) Requirement	Attribute
One way maximum delay	500 ms
Asymmetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point
Bandwidth	20 Kbps
Availability	99.999
precise timing required	Yes
Recovery time on Node failure	N/A
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	1%

Table 9: FCAG Communication Requirements

#### 3.1.2.2. Control of the Generation Infrastructure

The control of the generation infrastructure combines requirements from industrial automation systems and energy generation systems. In this section we present the use case of the control of the generation infrastructure of a wind turbine.

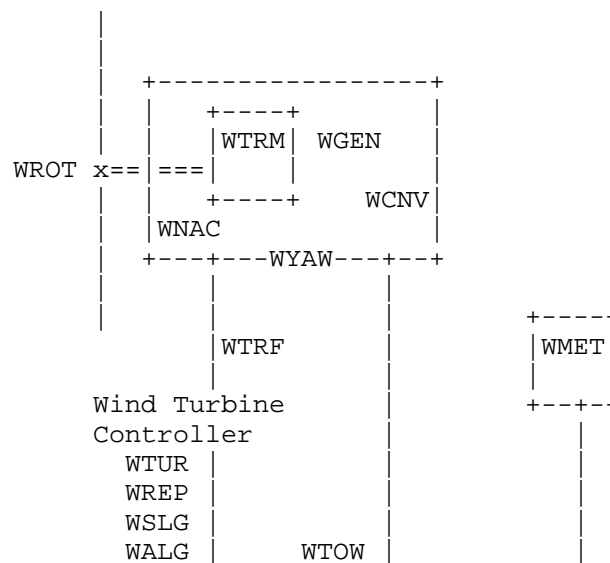


Figure 1: Wind Turbine Control Network

Figure 1 presents the subsystems that operate a wind turbine. These subsystems include

- o WROT (Rotor Control)
- o WNAC (Nacelle Control) (nacelle: housing containing the generator)
- o WTRM (Transmission Control)
- o WGEN (Generator)
- o WYAW (Yaw Controller) (of the tower head)
- o WCVN (In-Turbine Power Converter)
- o WMET (External Meteorological Station providing real time information to the controllers of the tower)

Traffic characteristics relevant for the network planning and dimensioning process in a wind turbine scenario are listed below. The values in this section are based mainly on the relevant references [Ahm14] and [Spe09]. Each logical node (Figure 1) is a part of the metering network and produces analog measurements and status information which must comply with their respective data rate constraints.

Subsystem	Sensor Count	Analog Sample Count	Data Rate (bytes/sec)	Status Sample Count	Data rate (bytes/sec)
WROT	14	9	642	5	10
WTRM	18	10	2828	8	16
WGEN	14	12	73764	2	4
WCNV	14	12	74060	2	4
WTRF	12	5	73740	2	4
WNAC	12	9	112	3	6
WYAW	7	8	220	4	8
WTOW	4	1	8	3	6
WMET	7	7	228	-	-

Table 10: Wind Turbine Data Rate Constraints

Quality of Service (QoS) constraints for different services are presented in Table 11. These constraints are defined by IEEE 1646 standard [IEEE1646] and IEC 61400 standard [IEC61400].

Service	Latency	Reliability	Packet Loss Rate
Analogue measure	16 ms	99.99%	< 10 <sup>-6</sup>
Status information	16 ms	99.99%	< 10 <sup>-6</sup>
Protection traffic	4 ms	100.00%	< 10 <sup>-9</sup>
Reporting and logging	1 s	99.99%	< 10 <sup>-6</sup>
Video surveillance	1 s	99.00%	No specific requirement
Internet connection	60 min	99.00%	No specific requirement
Control traffic	16 ms	100.00%	< 10 <sup>-9</sup>
Data polling	16 ms	99.99%	< 10 <sup>-6</sup>

Table 11: Wind Turbine Reliability and Latency Constraints

### 3.1.2.2.1. Intra-Domain Network Considerations

A wind turbine is composed of a large set of subsystems including sensors and actuators which require time-critical operation. The reliability and latency constraints of these different subsystems is shown in Table 11. These subsystems are connected to an intra-domain network which is used to monitor and control the operation of the turbine and connect it to the SCADA subsystems. The different



components are interconnected using fiber optics, industrial buses, industrial Ethernet, EtherCat, or a combination of them. Industrial signaling and control protocols such as Modbus, Profibus, Profinet and EtherCat are used directly on top of the Layer 2 transport or encapsulated over TCP/IP.

The Data collected from the sensors and condition monitoring systems is multiplexed onto fiber cables for transmission to the base of the tower, and to remote control centers. The turbine controller continuously monitors the condition of the wind turbine and collects statistics on its operation. This controller also manages a large number of switches, hydraulic pumps, valves, and motors within the wind turbine.

There is usually a controller both at the bottom of the tower and in the nacelle. The communication between these two controllers usually takes place using fiber optics instead of copper links. Sometimes, a third controller is installed in the hub of the rotor and manages the pitch of the blades. That unit usually communicates with the nacelle unit using serial communications.

#### 3.1.2.2.2. Inter-Domain network considerations

A remote control center belonging to a grid operator regulates the power output, enables remote actuation, and monitors the health of one or more wind parks in tandem. It connects to the local control center in a wind park over the Internet (Figure 2) via firewalls at both ends. The AS path between the local control center and the Wind Park typically involves several ISPs at different tiers. For example, a remote control center in Denmark can regulate a wind park in Greece over the normal public AS path between the two locations.

The remote control center is part of the SCADA system, setting the desired power output to the wind park and reading back the result once the new power output level has been set. Traffic between the remote control center and the wind park typically consists of protocols like IEC 60870-5-104 [IEC-60870-5-104], OPC XML-DA [OPCXML], Modbus [MODBUS], and SNMP [RFC3411]. Currently, traffic flows between the wind farm and the remote control center are best effort. QoS requirements are not strict, so no SLAs or service provisioning mechanisms (e.g., VPN) are employed. In case of events like equipment failure, tolerance for alarm delay is on the order of minutes, due to redundant systems already in place.

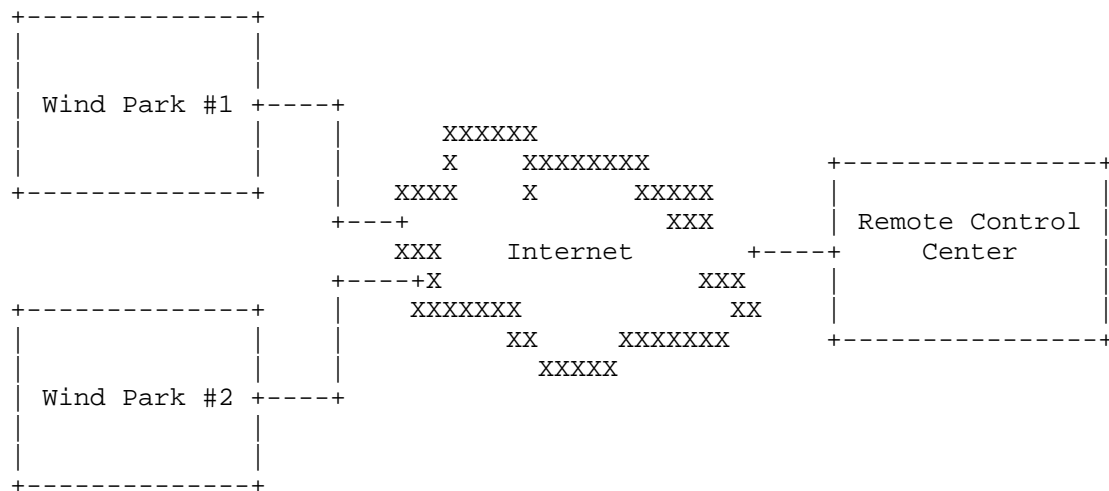


Figure 2: Wind Turbine Control via Internet

We expect future use cases which require bounded latency, bounded jitter and extraordinary low packet loss for inter-domain traffic flows due to the softwarization and virtualization of core wind farm equipment (e.g. switches, firewalls and SCADA server components). These factors will create opportunities for service providers to install new services and dynamically manage them from remote locations. For example, to enable fail-over of a local SCADA server, a SCADA server in another wind farm site (under the administrative control of the same operator) could be utilized temporarily (Figure 3). In that case local traffic would be forwarded to the remote SCADA server and existing intra-domain QoS and timing parameters would have to be met for inter-domain traffic flows.

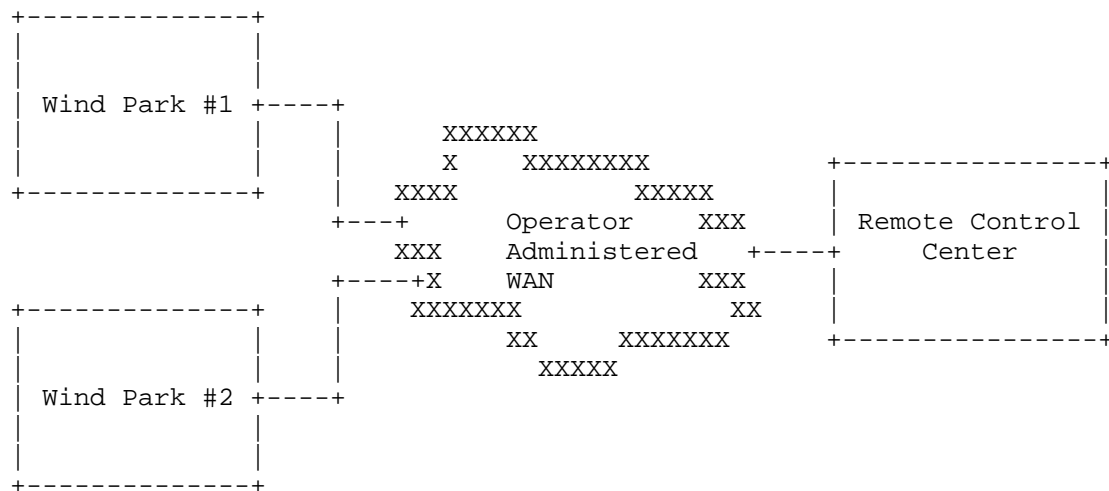


Figure 3: Wind Turbine Control via Operator Administered WAN

### 3.1.3. Distribution use case

#### 3.1.3.1. Fault Location Isolation and Service Restoration (FLISR)

Fault Location, Isolation, and Service Restoration (FLISR) refers to the ability to automatically locate the fault, isolate the fault, and restore service in the distribution network. This will likely be the first widespread application of distributed intelligence in the grid.

Static power switch status (open/closed) in the network dictates the power flow to secondary substations. Reconfiguring the network in the event of a fault is typically done manually on site to energize/de-energize alternate paths. Automating the operation of substation switchgear allows the flow of power to be altered automatically under fault conditions.

FLISR can be managed centrally from a Distribution Management System (DMS) or executed locally through distributed control via intelligent switches and fault sensors.

FLISR Requirement	Attribute
One way maximum delay	80 ms
Asymmetric delay Required	No
Maximum jitter	40 ms
Topology	Point to point, point to Multi-point, Multi-point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on Node failure	Depends on customer impact
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 12: FLISR Communication Requirements

### 3.2. Electrical Utilities Today

Many utilities still rely on complex environments formed of multiple application-specific proprietary networks, including TDM networks.

In this kind of environment there is no mixing of OT and IT applications on the same network, and information is siloed between operational areas.

Specific calibration of the full chain is required, which is costly.

This kind of environment prevents utility operations from realizing the operational efficiency benefits, visibility, and functional integration of operational information across grid applications and data networks.

In addition, there are many security-related issues as discussed in the following section.

#### 3.2.1. Security Current Practices and Limitations

Grid monitoring and control devices are already targets for cyber attacks, and legacy telecommunications protocols have many intrinsic network-related vulnerabilities. For example, DNP3, Modbus,

PROFIBUS/PROFINET, and other protocols are designed around a common paradigm of request and respond. Each protocol is designed for a master device such as an HMI (Human Machine Interface) system to send commands to subordinate slave devices to retrieve data (reading inputs) or control (writing to outputs). Because many of these protocols lack authentication, encryption, or other basic security measures, they are prone to network-based attacks, allowing a malicious actor or attacker to utilize the request-and-respond system as a mechanism for command-and-control like functionality. Specific security concerns common to most industrial control, including utility telecommunication protocols include the following:

- o Network or transport errors (e.g. malformed packets or excessive latency) can cause protocol failure.
- o Protocol commands may be available that are capable of forcing slave devices into inoperable states, including powering-off devices, forcing them into a listen-only state, disabling alarming.
- o Protocol commands may be available that are capable of restarting communications and otherwise interrupting processes.
- o Protocol commands may be available that are capable of clearing, erasing, or resetting diagnostic information such as counters and diagnostic registers.
- o Protocol commands may be available that are capable of requesting sensitive information about the controllers, their configurations, or other need-to-know information.
- o Most protocols are application layer protocols transported over TCP; therefore it is easy to transport commands over non-standard ports or inject commands into authorized traffic flows.
- o Protocol commands may be available that are capable of broadcasting messages to many devices at once (i.e. a potential DoS).
- o Protocol commands may be available to query the device network to obtain defined points and their values (i.e. a configuration scan).
- o Protocol commands may be available that will list all available function codes (i.e. a function scan).

These inherent vulnerabilities, along with increasing connectivity between IT and OT networks, make network-based attacks very feasible.

Simple injection of malicious protocol commands provides control over the target process. Altering legitimate protocol traffic can also alter information about a process and disrupt the legitimate controls that are in place over that process. A man-in-the-middle attack could provide both control over a process and misrepresentation of data back to operator consoles.

### 3.3. Electrical Utilities Future

The business and technology trends that are sweeping the utility industry will drastically transform the utility business from the way it has been for many decades. At the core of many of these changes is a drive to modernize the electrical grid with an integrated telecommunications infrastructure. However, interoperability concerns, legacy networks, disparate tools, and stringent security requirements all add complexity to the grid transformation. Given the range and diversity of the requirements that should be addressed by the next generation telecommunications infrastructure, utilities need to adopt a holistic architectural approach to integrate the electrical grid with digital telecommunications across the entire power delivery chain.

The key to modernizing grid telecommunications is to provide a common, adaptable, multi-service network infrastructure for the entire utility organization. Such a network serves as the platform for current capabilities while enabling future expansion of the network to accommodate new applications and services.

To meet this diverse set of requirements, both today and in the future, the next generation utility telecommunications network will be based on open-standards-based IP architecture. An end-to-end IP architecture takes advantage of nearly three decades of IP technology development, facilitating interoperability and device management across disparate networks and devices, as it has been already demonstrated in many mission-critical and highly secure networks.

IPv6 is seen as a future telecommunications technology for the Smart Grid; the IEC (International Electrotechnical Commission) and different National Committees have mandated a specific adhoc group (AHG8) to define the migration strategy to IPv6 for all the IEC TC57 power automation standards. The AHG8 has recently finalised the work on the migration strategy and the following Technical Report has been issued: IEC TR 62357-200:2015: Guidelines for migration from Internet Protocol version 4 (IPv4) to Internet Protocol version 6 (IPv6).

We expect cloud-based SCADA systems to control and monitor the critical and non-critical subsystems of generation systems, for example wind farms.

### 3.3.1. Migration to Packet-Switched Network

Throughout the world, utilities are increasingly planning for a future based on smart grid applications requiring advanced telecommunications systems. Many of these applications utilize packet connectivity for communicating information and control signals across the utility's Wide Area Network (WAN), made possible by technologies such as multiprotocol label switching (MPLS). The data that traverses the utility WAN includes:

- o Grid monitoring, control, and protection data
- o Non-control grid data (e.g. asset data for condition-based monitoring)
- o Physical safety and security data (e.g. voice and video)
- o Remote worker access to corporate applications (voice, maps, schematics, etc.)
- o Field area network backhaul for smart metering, and distribution grid management
- o Enterprise traffic (email, collaboration tools, business applications)

WANs support this wide variety of traffic to and from substations, the transmission and distribution grid, generation sites, between control centers, and between work locations and data centers. To maintain this rapidly expanding set of applications, many utilities are taking steps to evolve present time-division multiplexing (TDM) based and frame relay infrastructures to packet systems. Packet-based networks are designed to provide greater functionalities and higher levels of service for applications, while continuing to deliver reliability and deterministic (real-time) traffic support.

### 3.3.2. Telecommunications Trends

These general telecommunications topics are in addition to the use cases that have been addressed so far. These include both current and future telecommunications related topics that should be factored into the network architecture and design.

#### 3.3.2.1. General Telecommunications Requirements

- o IP Connectivity everywhere
- o Monitoring services everywhere and from different remote centers

- o Move services to a virtual data center
- o Unify access to applications / information from the corporate network
- o Unify services
- o Unified Communications Solutions
- o Mix of fiber and microwave technologies - obsolescence of SONET/SDH or TDM
- o Standardize grid telecommunications protocol to opened standard to ensure interoperability
- o Reliable Telecommunications for Transmission and Distribution Substations
- o IEEE 1588 time synchronization Client / Server Capabilities
- o Integration of Multicast Design
- o QoS Requirements Mapping
- o Enable Future Network Expansion
- o Substation Network Resilience
- o Fast Convergence Design
- o Scalable Headend Design
- o Define Service Level Agreements (SLA) and Enable SLA Monitoring
- o Integration of 3G/4G Technologies and future technologies
- o Ethernet Connectivity for Station Bus Architecture
- o Ethernet Connectivity for Process Bus Architecture
- o Protection, teleprotection and PMU (Phaser Measurement Unit) on IP

#### 3.3.2.2. Specific Network topologies of Smart Grid Applications

Utilities often have very large private telecommunications networks. It covers an entire territory / country. The main purpose of the network, until now, has been to support transmission network monitoring, control, and automation, remote control of generation



sites, and providing FCAPS (Fault, Configuration, Accounting, Performance, Security) services from centralized network operation centers.

Going forward, one network will support operation and maintenance of electrical networks (generation, transmission, and distribution), voice and data services for ten of thousands of employees and for exchange with neighboring interconnections, and administrative services. To meet those requirements, utility may deploy several physical networks leveraging different technologies across the country: an optical network and a microwave network for instance. Each protection and automatism system between two points has two telecommunications circuits, one on each network. Path diversity between two substations is key. Regardless of the event type (hurricane, ice storm, etc.), one path shall stay available so the system can still operate.

In the optical network, signals are transmitted over more than tens of thousands of circuits using fiber optic links, microwave and telephone cables. This network is the nervous system of the utility's power transmission operations. The optical network represents ten of thousands of km of cable deployed along the power lines, with individual runs as long as 280 km.

#### 3.3.2.3. Precision Time Protocol

Some utilities do not use GPS clocks in generation substations. One of the main reasons is that some of the generation plants are 30 to 50 meters deep under ground and the GPS signal can be weak and unreliable. Instead, atomic clocks are used. Clocks are synchronized amongst each other. Rubidium clocks provide clock and 1ms timestamps for IRIG-B.

Some companies plan to transition to the Precision Time Protocol (PTP, [IEEE1588]), distributing the synchronization signal over the IP/MPLS network. PTP provides a mechanism for synchronizing the clocks of participating nodes to a high degree of accuracy and precision.

PTP operates based on the following assumptions:

It is assumed that the network eliminates cyclic forwarding of PTP messages within each communication path (e.g. by using a spanning tree protocol).

PTP is tolerant of an occasional missed message, duplicated message, or message that arrived out of order. However, PTP assumes that such impairments are relatively rare.

PTP was designed assuming a multicast communication model, however PTP also supports a unicast communication model as long as the behavior of the protocol is preserved.

Like all message-based time transfer protocols, PTP time accuracy is degraded by delay asymmetry in the paths taken by event messages. Asymmetry is not detectable by PTP, however, if such delays are known a priori, PTP can correct for asymmetry.

IEC 61850 defines the use of IEC/IEEE 61850-9-3:2016. The title is: Precision time protocol profile for power utility automation. It is based on Annex B/IEC 62439 which offers the support of redundant attachment of clocks to Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR) networks.

### 3.3.3. Security Trends in Utility Networks

Although advanced telecommunications networks can assist in transforming the energy industry by playing a critical role in maintaining high levels of reliability, performance, and manageability, they also introduce the need for an integrated security infrastructure. Many of the technologies being deployed to support smart grid projects such as smart meters and sensors can increase the vulnerability of the grid to attack. Top security concerns for utilities migrating to an intelligent smart grid telecommunications platform center on the following trends:

- o Integration of distributed energy resources
- o Proliferation of digital devices to enable management, automation, protection, and control
- o Regulatory mandates to comply with standards for critical infrastructure protection
- o Migration to new systems for outage management, distribution automation, condition-based maintenance, load forecasting, and smart metering
- o Demand for new levels of customer service and energy management

This development of a diverse set of networks to support the integration of microgrids, open-access energy competition, and the use of network-controlled devices is driving the need for a converged security infrastructure for all participants in the smart grid, including utilities, energy service providers, large commercial and industrial, as well as residential customers. Securing the assets of electric power delivery systems (from the control center to the

substation, to the feeders and down to customer meters) requires an end-to-end security infrastructure that protects the myriad of telecommunications assets used to operate, monitor, and control power flow and measurement.

"Cyber security" refers to all the security issues in automation and telecommunications that affect any functions related to the operation of the electric power systems. Specifically, it involves the concepts of:

- o Integrity : data cannot be altered undetectably
- o Authenticity : the telecommunications parties involved must be validated as genuine
- o Authorization : only requests and commands from the authorized users can be accepted by the system
- o Confidentiality : data must not be accessible to any unauthenticated users

When designing and deploying new smart grid devices and telecommunications systems, it is imperative to understand the various impacts of these new components under a variety of attack situations on the power grid. Consequences of a cyber attack on the grid telecommunications network can be catastrophic. This is why security for smart grid is not just an ad hoc feature or product, it's a complete framework integrating both physical and Cyber security requirements and covering the entire smart grid networks from generation to distribution. Security has therefore become one of the main foundations of the utility telecom network architecture and must be considered at every layer with a defense-in-depth approach. Migrating to IP based protocols is key to address these challenges for two reasons:

- o IP enables a rich set of features and capabilities to enhance the security posture
- o IP is based on open standards, which allows interoperability between different vendors and products, driving down the costs associated with implementing security solutions in OT networks.

Securing OT (Operation technology) telecommunications over packet-switched IP networks follow the same principles that are foundational for securing the IT infrastructure, i.e., consideration must be given to enforcing electronic access control for both person-to-machine and machine-to-machine communications, and providing the appropriate

levels of data privacy, device and platform integrity, and threat detection and mitigation.

### 3.4. Electrical Utilities Asks

- o Mixed L2 and L3 topologies
- o Deterministic behavior
- o Bounded latency and jitter
- o Tight feedback intervals
- o High availability, low recovery time
- o Redundancy, low packet loss
- o Precise timing
- o Centralized computing of deterministic paths
- o Distributed configuration may also be useful

## 4. Building Automation Systems

### 4.1. Use Case Description

A Building Automation System (BAS) manages equipment and sensors in a building for improving residents' comfort, reducing energy consumption, and responding to failures and emergencies. For example, the BAS measures the temperature of a room using sensors and then controls the HVAC (heating, ventilating, and air conditioning) to maintain a set temperature and minimize energy consumption.

A BAS primarily performs the following functions:

- o Periodically measures states of devices, for example humidity and illuminance of rooms, open/close state of doors, FAN speed, etc.
- o Stores the measured data.
- o Provides the measured data to BAS systems and operators.
- o Generates alarms for abnormal state of devices.
- o Controls devices (e.g. turn off room lights at 10:00 PM).

## 4.2. Building Automation Systems Today

### 4.2.1. BAS Architecture

A typical BAS architecture of today is shown in Figure 4.

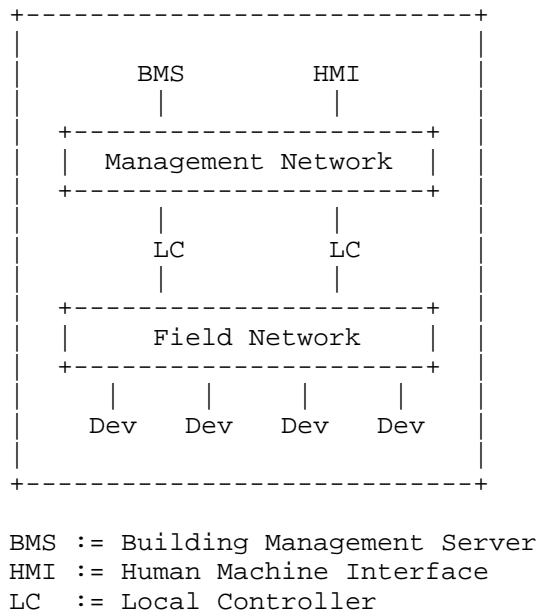


Figure 4: BAS architecture

There are typically two layers of network in a BAS. The upper one is called the Management Network and the lower one is called the Field Network. In management networks an IP-based communication protocol is used, while in field networks non-IP based communication protocols ("field protocols") are mainly used. Field networks have specific timing requirements, whereas management networks can be best-effort.

A Human Machine Interface (HMI) is typically a desktop PC used by operators to monitor and display device states, send device control commands to Local Controllers (LCs), and configure building schedules (for example "turn off all room lights in the building at 10:00 PM").

A Building Management Server (BMS) performs the following operations.

- o Collect and store device states from LCs at regular intervals.
- o Send control values to LCs according to a building schedule.

- o Send an alarm signal to operators if it detects abnormal devices states.

The BMS and HMI communicate with LCs via IP-based "management protocols" (see standards [bacnetip], [knx]).

A LC is typically a Programmable Logic Controller (PLC) which is connected to several tens or hundreds of devices using "field protocols". An LC performs the following kinds of operations:

- o Measure device states and provide the information to BMS or HMI.
- o Send control values to devices, unilaterally or as part of a feedback control loop.

There are many field protocols used today; some are standards-based and others are proprietary (see standards [lontalk], [modbus], [profibus] and [flnet]). The result is that BASs have multiple MAC/PHY modules and interfaces. This makes BASs more expensive, slower to develop, and can result in "vendor lock-in" with multiple types of management applications.

#### 4.2.2. BAS Deployment Model

An example BAS for medium or large buildings is shown in Figure 5. The physical layout spans multiple floors, and there is a monitoring room where the BAS management entities are located. Each floor will have one or more LCs depending upon the number of devices connected to the field network.

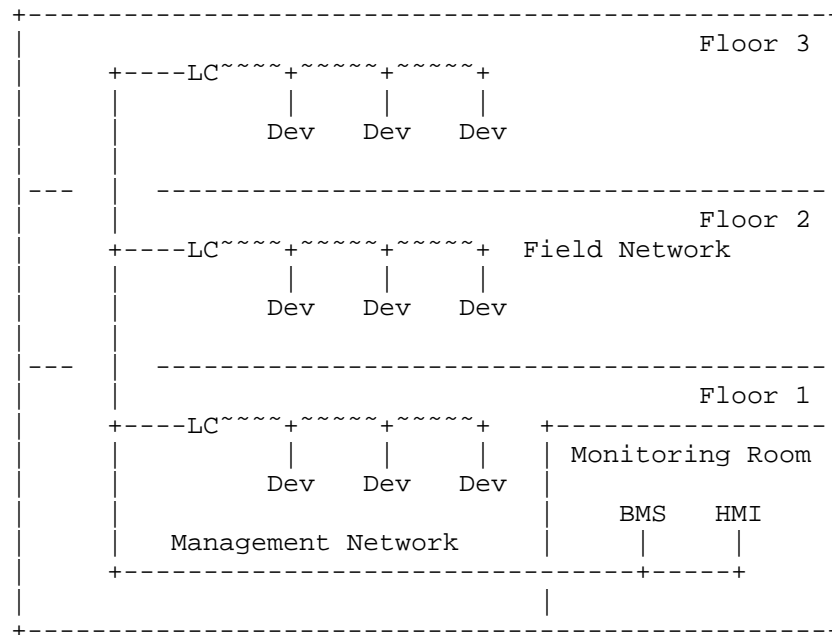


Figure 5: BAS Deployment model for Medium/Large Buildings

Each LC is connected to the monitoring room via the Management network, and the management functions are performed within the building. In most cases, fast Ethernet (e.g. 100BASE-T) is used for the management network. Since the management network is non-realtime, use of Ethernet without quality of service is sufficient for today's deployment.

In the field network a variety of physical interfaces such as RS232C and RS485 are used, which have specific timing requirements. Thus if a field network is to be replaced with an Ethernet or wireless network, such networks must support time-critical deterministic flows.

In Figure 6, another deployment model is presented in which the management system is hosted remotely. This is becoming popular for small office and residential buildings in which a standalone monitoring system is not cost-effective.

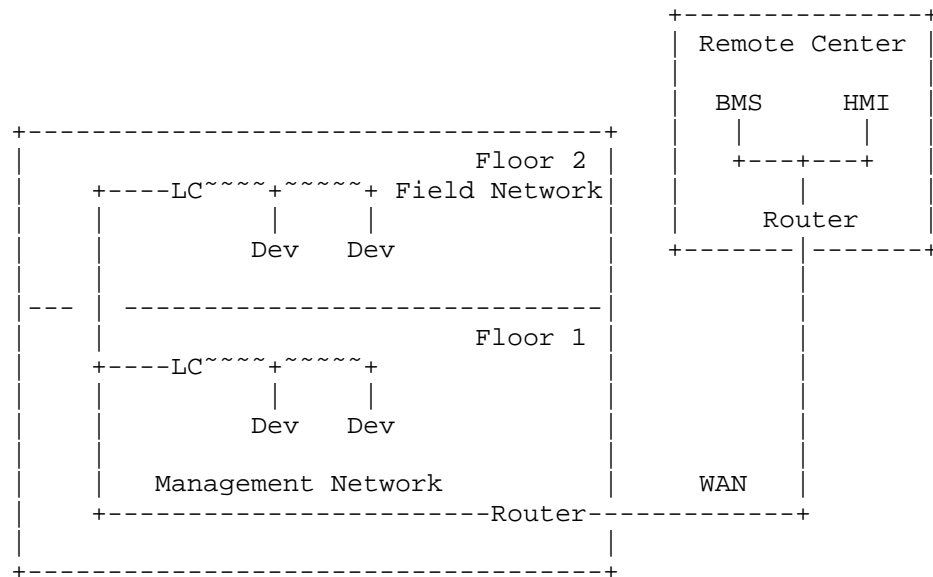


Figure 6: Deployment model for Small Buildings

Some interoperability is possible today in the Management Network, but not in today's field networks due to their non-IP-based design.

#### 4.2.3. Use Cases for Field Networks

Below are use cases for Environmental Monitoring, Fire Detection, and Feedback Control, and their implications for field network performance.

##### 4.2.3.1. Environmental Monitoring

The BMS polls each LC at a maximum measurement interval of 100ms (for example to draw a historical chart of 1 second granularity with a 10x sampling interval) and then performs the operations as specified by the operator. Each LC needs to measure each of its several hundred sensors once per measurement interval. Latency is not critical in this scenario as long as all sensor values are completed in the measurement interval. Availability is expected to be 99.999 %.

##### 4.2.3.2. Fire Detection

On detection of a fire, the BMS must stop the HVAC, close the fire shutters, turn on the fire sprinklers, send an alarm, etc. There are typically ~10s of sensors per LC that BMS needs to manage. In this



scenario the measurement interval is 10-50ms, the communication delay is 10ms, and the availability must be 99.9999 %.

#### 4.2.3.3. Feedback Control

BAS systems utilize feedback control in various ways; the most time-critical is control of DC motors, which require a short feedback interval (1-5ms) with low communication delay (10ms) and jitter (1ms). The feedback interval depends on the characteristics of the device and a target quality of control value. There are typically ~10s of such devices per LC.

Communication delay is expected to be less than 10 ms, jitter less than 1 sec while the availability must be 99.9999% .

#### 4.2.4. Security Considerations

When BAS field networks were developed it was assumed that the field networks would always be physically isolated from external networks and therefore security was not a concern. In today's world many BASs are managed remotely and are thus connected to shared IP networks and so security is definitely a concern, yet security features are not available in the majority of BAS field network deployments .

The management network, being an IP-based network, has the protocols available to enable network security, but in practice many BAS systems do not implement even the available security features such as device authentication or encryption for data in transit.

#### 4.3. BAS Future

In the future we expect more fine-grained environmental monitoring and lower energy consumption, which will require more sensors and devices, thus requiring larger and more complex building networks.

We expect building networks to be connected to or converged with other networks (Enterprise network, Home network, and Internet).

Therefore better facilities for network management, control, reliability and security are critical in order to improve resident and operator convenience and comfort. For example the ability to monitor and control building devices via the internet would enable (for example) control of room lights or HVAC from a resident's desktop PC or phone application.

#### 4.4. BAS Asks

The community would like to see an interoperable protocol specification that can satisfy the timing, security, availability and QoS constraints described above, such that the resulting converged network can replace the disparate field networks. Ideally this connectivity could extend to the open Internet.

This would imply an architecture that can guarantee

- o Low communication delays (from <10ms to 100ms in a network of several hundred devices)
- o Low jitter (< 1 ms)
- o Tight feedback intervals (1ms - 10ms)
- o High network availability (up to 99.9999% )
- o Availability of network data in disaster scenario
- o Authentication between management and field devices (both local and remote)
- o Integrity and data origin authentication of communication data between field and management devices
- o Confidentiality of data when communicated to a remote device

### 5. Wireless for Industrial

#### 5.1. Use Case Description

Wireless networks are useful for industrial applications, for example when portable, fast-moving or rotating objects are involved, and for the resource-constrained devices found in the Internet of Things (IoT).

Such network-connected sensors, actuators, control loops (etc.) typically require that the underlying network support real-time quality of service (QoS), as well as specific classes of other network properties such as reliability, redundancy, and security.

These networks may also contain very large numbers of devices, for example for factories, "big data" acquisition, and the IoT. Given the large numbers of devices installed, and the potential pervasiveness of the IoT, this is a huge and very cost-sensitive

market. For example, a 1% cost reduction in some areas could save \$100B

#### 5.1.1.1. Network Convergence using 6TiSCH

Some wireless network technologies support real-time QoS, and are thus useful for these kinds of networks, but others do not. For example WiFi is pervasive but does not provide guaranteed timing or delivery of packets, and thus is not useful in this context.

In this use case we focus on one specific wireless network technology which does provide the required deterministic QoS, which is "IPv6 over the TSCH mode of IEEE 802.15.4e" (6TiSCH, where TSCH stands for "Time-Slotted Channel Hopping", see [I-D.ietf-6tisch-architecture], [IEEE802154], [IEEE802154e], and [RFC7554]).

There are other deterministic wireless busses and networks available today, however they are incompatible with each other, and incompatible with IP traffic (for example [ISA100], [WirelessHART]).

Thus the primary goal of this use case is to apply 6TiSCH as a converged IP- and standards-based wireless network for industrial applications, i.e. to replace multiple proprietary and/or incompatible wireless networking and wireless network management standards.

#### 5.1.1.2. Common Protocol Development for 6TiSCH

Today there are a number of protocols required by 6TiSCH which are still in development, and a second intent of this use case is to highlight the ways in which these "missing" protocols share goals in common with DetNet. Thus it is possible that some of the protocol technology developed for DetNet will also be applicable to 6TiSCH.

These protocol goals are identified here, along with their relationship to DetNet. It is likely that ultimately the resulting protocols will not be identical, but will share design principles which contribute to the efficiency of enabling both DetNet and 6TiSCH.

One such commonality is that although at a different time scale, in both TSN [IEEE802.1TSNTG] and TSCH a packet crosses the network from node to node follows a precise schedule, as a train that leaves intermediate stations at precise times along its path. This kind of operation reduces collisions, saves energy, and enables engineering the network for deterministic properties.

Another commonality is remote monitoring and scheduling management of a TSCH network by a Path Computation Element (PCE) and Network

Management Entity (NME). The PCE/NME manage timeslots and device resources in a manner that minimizes the interaction with and the load placed on resource-constrained devices. For example, a tiny IoT device may have just enough buffers to store one or a few IPv6 packets, and will have limited bandwidth between peers such that it can maintain only a small amount of peer information, and will not be able to store many packets waiting to be forwarded. It is advantageous then for it to only be required to carry out the specific behavior assigned to it by the PCE/NME (as opposed to maintaining its own IP stack, for example).

Note: Current WG discussion indicates that some peer-to-peer communication must be assumed, i.e. the PCE may communicate only indirectly with any given device, enabling hierarchical configuration of the system.

6TiSCH depends on [PCE] and [I-D.finn-detnet-architecture].

6TiSCH also depends on the fact that DetNet will maintain consistency with [IEEE802.1TSNTG].

## 5.2. Wireless Industrial Today

Today industrial wireless is accomplished using multiple deterministic wireless networks which are incompatible with each other and with IP traffic.

6TiSCH is not yet fully specified, so it cannot be used in today's applications.

## 5.3. Wireless Industrial Future

### 5.3.1. Unified Wireless Network and Management

We expect DetNet and 6TiSCH together to enable converged transport of deterministic and best-effort traffic flows between real-time industrial devices and wide area networks via IP routing. A high level view of a basic such network is shown in Figure 7.

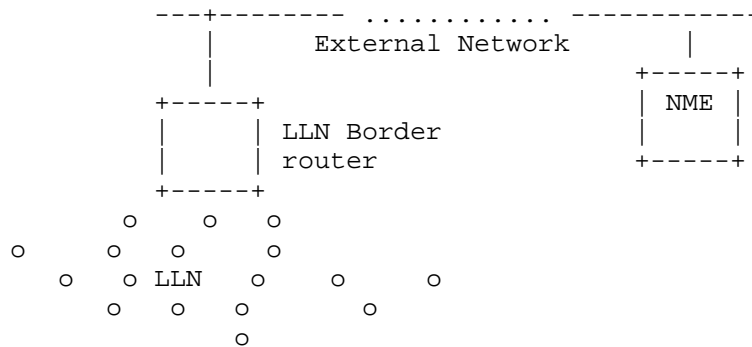


Figure 7: Basic 6TiSCH Network

Figure 8 shows a backbone router federating multiple synchronized 6TiSCH subnets into a single subnet connected to the external network.

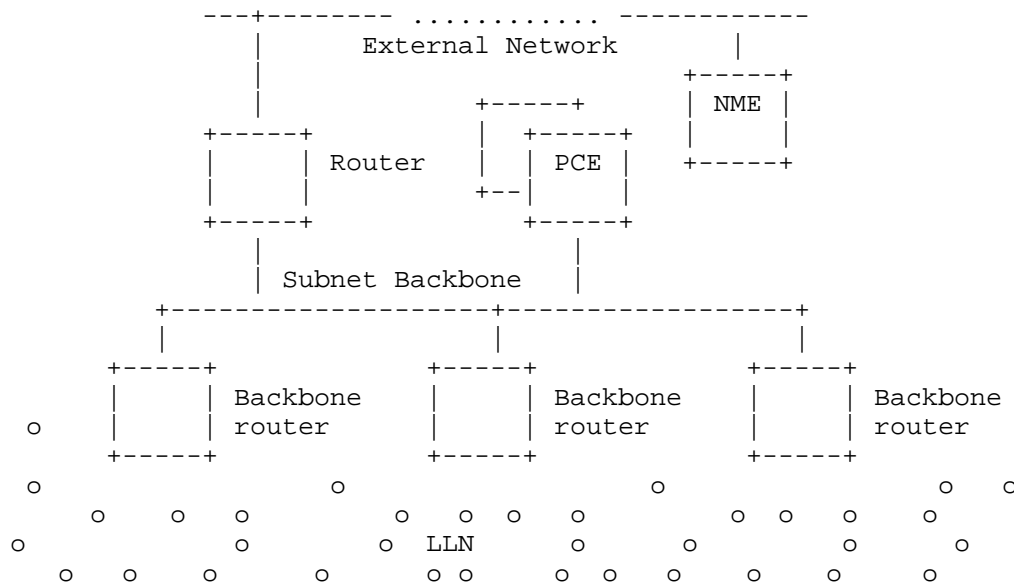


Figure 8: Extended 6TiSCH Network

The backbone router must ensure end-to-end deterministic behavior between the LLN and the backbone. We would like to see this accomplished in conformance with the work done in [I-D.finn-detnet-architecture] with respect to Layer-3 aspects of deterministic networks that span multiple Layer-2 domains.

The PCE must compute a deterministic path end-to-end across the TSCH network and IEEE802.1 TSN Ethernet backbone, and DetNet protocols are expected to enable end-to-end deterministic forwarding.

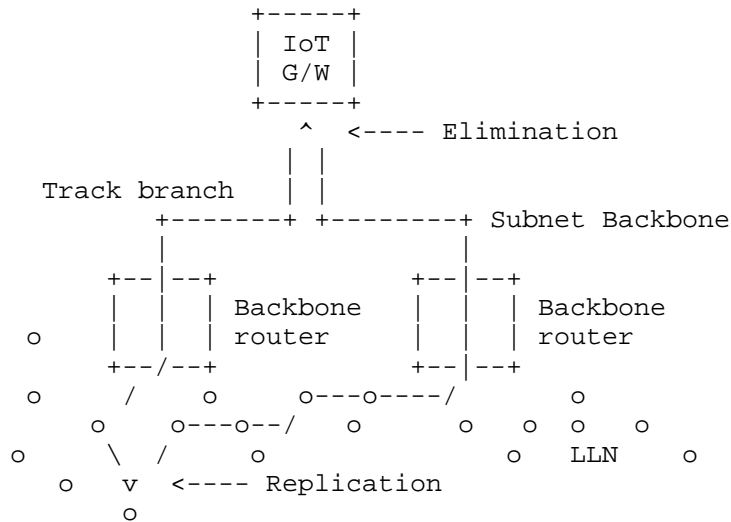


Figure 9: 6TiSCH Network with PRE

#### 5.3.1.1. PCE and 6TiSCH ARQ Retries

Note: The possible use of ARQ techniques in DetNet is currently considered a possible design alternative.

6TiSCH uses the IEEE802.15.4 Automatic Repeat-reQuest (ARQ) mechanism to provide higher reliability of packet delivery. ARQ is related to packet replication and elimination because there are two independent paths for packets to arrive at the destination, and if an expected packet does not arrive on one path then it checks for the packet on the second path.

Although to date this mechanism is only used by wireless networks, this may be a technique that would be appropriate for DetNet and so aspects of the enabling protocol could be co-developed.

For example, in Figure 9, a Track is laid out from a field device in a 6TiSCH network to an IoT gateway that is located on a IEEE802.1 TSN backbone.

In ARQ the Replication function in the field device sends a copy of each packet over two different branches, and the PCE schedules each hop of both branches so that the two copies arrive in due time at the gateway. In case of a loss on one branch, hopefully the other copy of the packet still arrives within the allocated time. If two copies make it to the IoT gateway, the Elimination function in the gateway ignores the extra packet and presents only one copy to upper layers.

At each 6TiSCH hop along the Track, the PCE may schedule more than one timeSlot for a packet, so as to support Layer-2 retries (ARQ).

In current deployments, a TSCH Track does not necessarily support PRE but is systematically multi-path. This means that a Track is scheduled so as to ensure that each hop has at least two forwarding solutions, and the forwarding decision is to try the preferred one and use the other in case of Layer-2 transmission failure as detected by ARQ.

#### 5.3.2. Schedule Management by a PCE

A common feature of 6TiSCH and DetNet is the action of a PCE to configure paths through the network. Specifically, what is needed is a protocol and data model that the PCE will use to get/set the relevant configuration from/to the devices, as well as perform operations on the devices. We expect that this protocol will be developed by DetNet with consideration for its reuse by 6TiSCH. The remainder of this section provides a bit more context from the 6TiSCH side.

##### 5.3.2.1. PCE Commands and 6TiSCH CoAP Requests

The 6TiSCH device does not expect to place the request for bandwidth between itself and another device in the network. Rather, an operation control system invoked through a human interface specifies the required traffic specification and the end nodes (in terms of latency and reliability). Based on this information, the PCE must compute a path between the end nodes and provision the network with per-flow state that describes the per-hop operation for a given packet, the corresponding timeslots, and the flow identification that enables recognizing that a certain packet belongs to a certain path, etc.

For a static configuration that serves a certain purpose for a long period of time, it is expected that a node will be provisioned in one shot with a full schedule, which incorporates the aggregation of its behavior for multiple paths. 6TiSCH expects that the programming of the schedule will be done over COAP as discussed in [I-D.ietf-6tisch-coap].

6TiSCH expects that the PCE commands will be mapped back and forth into CoAP by a gateway function at the edge of the 6TiSCH network. For instance, it is possible that a mapping entity on the backbone transforms a non-CoAP protocol such as PCEP into the RESTful interfaces that the 6TiSCH devices support. This architecture will be refined to comply with DetNet [I-D.finn-detnet-architecture] when the work is formalized. Related information about 6TiSCH can be found at [I-D.ietf-6tisch-6top-interface] and RPL [RFC6550].

A protocol may be used to update the state in the devices during runtime, for example if it appears that a path through the network has ceased to perform as expected, but in 6TiSCH that flow was not designed and no protocol was selected. We would like to see DetNet define the appropriate end-to-end protocols to be used in that case. The implication is that these state updates take place once the system is configured and running, i.e. they are not limited to the initial communication of the configuration of the system.

A "slotFrame" is the base object that a PCE would manipulate to program a schedule into an LLN node ([I-D.ietf-6tisch-architecture]).

We would like to see the PCE read energy data from devices, and compute paths that will implement policies on how energy in devices is consumed, for instance to ensure that the spent energy does not exceed the available energy over a period of time. Note: this statement implies that an extensible protocol for communicating device info to the PCE and enabling the PCE to act on it will be part of the DetNet architecture, however for subnets with specific protocols (e.g. CoAP) a gateway may be required.

6TiSCH devices can discover their neighbors over the radio using a mechanism such as beacons, but even though the neighbor information is available in the 6TiSCH interface data model, 6TiSCH does not describe a protocol to proactively push the neighborhood information to a PCE. We would like to see DetNet define such a protocol; one possible design alternative is that it could operate over CoAP, alternatively it could be converted to/from CoAP by a gateway. We would like to see such a protocol carry multiple metrics, for example similar to those used for RPL operations [RFC6551]

#### 5.3.2.2. 6TiSCH IP Interface

"6top" ([I-D.wang-6tisch-6top-sublayer]) is a logical link control sitting between the IP layer and the TSCH MAC layer which provides the link abstraction that is required for IP operations. The 6top data model and management interfaces are further discussed in [I-D.ietf-6tisch-6top-interface] and [I-D.ietf-6tisch-coap].



An IP packet that is sent along a 6TiSCH path uses the Differentiated Services Per-Hop-Behavior Group called Deterministic Forwarding, as described in [I-D.svshah-tsvwg-deterministic-forwarding].

#### 5.3.3. 6TiSCH Security Considerations

On top of the classical requirements for protection of control signaling, it must be noted that 6TiSCH networks operate on limited resources that can be depleted rapidly in a DoS attack on the system, for instance by placing a rogue device in the network, or by obtaining management control and setting up unexpected additional paths.

#### 5.4. Wireless Industrial Asks

6TiSCH depends on DetNet to define:

- o Configuration (state) and operations for deterministic paths
- o End-to-end protocols for deterministic forwarding (tagging, IP)
- o Protocol for packet replication and elimination

### 6. Cellular Radio

#### 6.1. Use Case Description

This use case describes the application of deterministic networking in the context of cellular telecom transport networks. Important elements include time synchronization, clock distribution, and ways of establishing time-sensitive streams for both Layer-2 and Layer-3 user plane traffic.

##### 6.1.1. Network Architecture

Figure 10 illustrates a typical 3GPP-defined cellular network architecture, which includes "Fronthaul", "Midhaul" and "Backhaul" network segments. The "Fronthaul" is the network connecting base stations (baseband processing units) to the remote radio heads (antennas). The "Midhaul" is the network inter-connecting base stations (or small cell sites). The "Backhaul" is the network or links connecting the radio base station sites to the network controller/gateway sites (i.e. the core of the 3GPP cellular network).

In Figure 10 "eNB" ("E-UTRAN Node B") is the hardware that is connected to the mobile phone network which communicates directly with mobile handsets ([TS36300]).

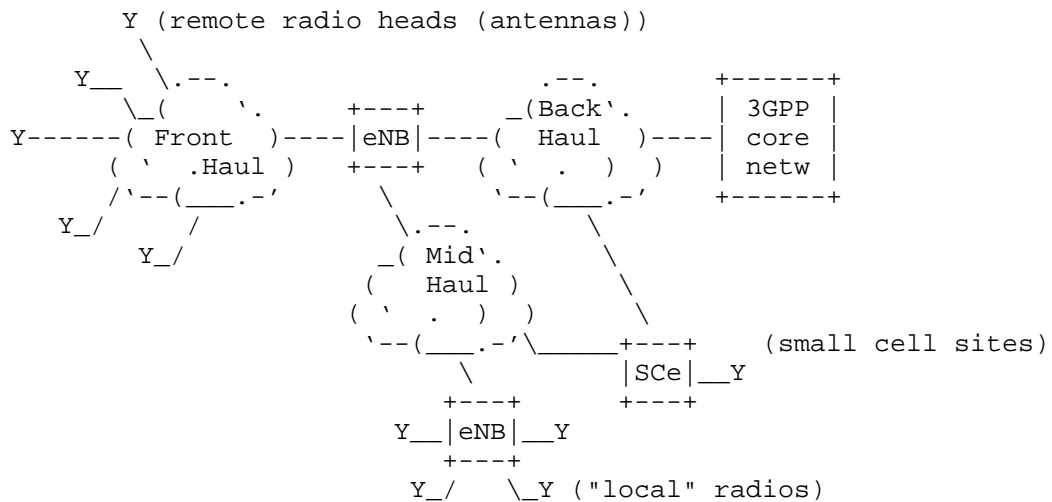


Figure 10: Generic 3GPP-based Cellular Network Architecture

#### 6.1.2. Delay Constraints

The available processing time for Fronthaul networking overhead is limited to the available time after the baseband processing of the radio frame has completed. For example in Long Term Evolution (LTE) radio, processing of a radio frame is allocated 3ms but typically the processing uses most of it, allowing only a small fraction to be used by the Fronthaul network (e.g. up to 250us one-way delay, though the existing spec ([NGMN-fronth]) supports delay only up to 100us). This ultimately determines the distance the remote radio heads can be located from the base stations (e.g., 100us equals roughly 20 km of optical fiber-based transport). Allocation options of the available time budget between processing and transport are under heavy discussions in the mobile industry.

For packet-based transport the allocated transport time (e.g. CPRI would allow for 100us delay [CPRI]) is consumed by all nodes and buffering between the remote radio head and the baseband processing unit, plus the distance-incurred delay.

The baseband processing time and the available "delay budget" for the fronthaul is likely to change in the forthcoming "5G" due to reduced radio round trip times and other architectural and service requirements [NGMN].

The transport time budget, as noted above, places limitations on the distance that remote radio heads can be located from base stations (i.e. the link length). In the above analysis, the entire transport

time budget is assumed to be available for link propagation delay. However the transport time budget can be broken down into three components: scheduling /queueing delay, transmission delay, and link propagation delay. Using today's Fronthaul networking technology, the queuing, scheduling and transmission components might become the dominant factors in the total transport time rather than the link propagation delay. This is especially true in cases where the Fronthaul link is relatively short and it is shared among multiple Fronthaul flows, for example in indoor and small cell networks, massive MIMO antenna networks, and split Fronthaul architectures.

DetNet technology can improve this application by controlling and reducing the time required for the queuing, scheduling and transmission operations by properly assigning the network resources, thus leaving more of the transport time budget available for link propagation, and thus enabling longer link lengths. However, link length is usually a given parameter and is not a controllable network parameter, since RRH and BBU sites are usually located in predetermined locations. However, the number of antennas in an RRH site might increase for example by adding more antennas, increasing the MIMO capability of the network or support of massive MIMO. This means increasing the number of the fronthaul flows sharing the same fronthaul link. DetNet can now control the bandwidth assignment of the fronthaul link and the scheduling of fronthaul packets over this link and provide adequate buffer provisioning for each flow to reduce the packet loss rate.

Another way in which DetNet technology can aid Fronthaul networks is by providing effective isolation from best-effort (and other classes of) traffic, which can arise as a result of network slicing in 5G networks where Fronthaul traffic generated in different network slices might have differing performance requirements. DetNet technology can also dynamically control the bandwidth assignment, scheduling and packet forwarding decisions and the buffer provisioning of the Fronthaul flows to guarantee the end-to-end delay of the Fronthaul packets and minimize the packet loss rate.

[METIS] documents the fundamental challenges as well as overall technical goals of the future 5G mobile and wireless system as the starting point. These future systems should support much higher data volumes and rates and significantly lower end-to-end latency for 100x more connected devices (at similar cost and energy consumption levels as today's system).

For Midhaul connections, delay constraints are driven by Inter-Site radio functions like Coordinated Multipoint Processing (CoMP, see [CoMP]). CoMP reception and transmission is a framework in which multiple geographically distributed antenna nodes cooperate to

improve the performance of the users served in the common cooperation area. The design principal of CoMP is to extend the current single-cell to multi-UE (User Equipment) transmission to a multi-cell-to-multi-UEs transmission by base station cooperation.

CoMP has delay-sensitive performance parameters, which are "midhaul latency" and "CSI (Channel State Information) reporting and accuracy". The essential feature of CoMP is signaling between eNBs, so Midhaul latency is the dominating limitation of CoMP performance. Generally, CoMP can benefit from coordinated scheduling (either distributed or centralized) of different cells if the signaling delay between eNBs is within 1-10ms. This delay requirement is both rigid and absolute because any uncertainty in delay will degrade the performance significantly.

Inter-site CoMP is one of the key requirements for 5G and is also a near-term goal for the current 4.5G network architecture.

#### 6.1.3. Time Synchronization Constraints

Fronthaul time synchronization requirements are given by [TS25104], [TS36104], [TS36211], and [TS36133]. These can be summarized for the current 3GPP LTE-based networks as:

##### Delay Accuracy:

$\pm 8\text{ns}$  (i.e.  $\pm 1/32 T_c$ , where  $T_c$  is the UMTS Chip time of  $1/3.84\text{ MHz}$ ) resulting in a round trip accuracy of  $\pm 16\text{ns}$ . The value is this low to meet the 3GPP Timing Alignment Error (TAE) measurement requirements. Note: performance guarantees of low nanosecond values such as these are considered to be below the DetNet layer - it is assumed that the underlying implementation, e.g. the hardware, will provide sufficient support (e.g. buffering) to enable this level of accuracy. These values are maintained in the use case to give an indication of the overall application.

##### Timing Alignment Error:

Timing Alignment Error (TAE) is problematic to Fronthaul networks and must be minimized. If the transport network cannot guarantee low enough TAE then additional buffering has to be introduced at the edges of the network to buffer out the jitter. Buffering is not desirable as it reduces the total available delay budget. Packet Delay Variation (PDV) requirements can be derived from TAE for packet based Fronthaul networks.

- \* For multiple input multiple output (MIMO) or TX diversity transmissions, at each carrier frequency, TAE shall not exceed 65 ns (i.e.  $1/4 T_c$ ).
- \* For intra-band contiguous carrier aggregation, with or without MIMO or TX diversity, TAE shall not exceed 130 ns (i.e.  $1/2 T_c$ ).
- \* For intra-band non-contiguous carrier aggregation, with or without MIMO or TX diversity, TAE shall not exceed 260 ns (i.e. one  $T_c$ ).
- \* For inter-band carrier aggregation, with or without MIMO or TX diversity, TAE shall not exceed 260 ns.

Transport link contribution to radio frequency error:

+/-2 PPB. This value is considered to be "available" for the Fronthaul link out of the total 50 PPB budget reserved for the radio interface. Note: the reason that the transport link contributes to radio frequency error is as follows. The current way of doing Fronthaul is from the radio unit to remote radio head directly. The remote radio head is essentially a passive device (without buffering etc.) The transport drives the antenna directly by feeding it with samples and everything the transport adds will be introduced to radio as-is. So if the transport causes additional frequency error that shows immediately on the radio as well. Note: performance guarantees of low nanosecond values such as these are considered to be below the DetNet layer - it is assumed that the underlying implementation, e.g. the hardware, will provide sufficient support to enable this level of performance. These values are maintained in the use case to give an indication of the overall application.

The above listed time synchronization requirements are difficult to meet with point-to-point connected networks, and more difficult when the network includes multiple hops. It is expected that networks must include buffering at the ends of the connections as imposed by the jitter requirements, since trying to meet the jitter requirements in every intermediate node is likely to be too costly. However, every measure to reduce jitter and delay on the path makes it easier to meet the end-to-end requirements.

In order to meet the timing requirements both senders and receivers must remain time synchronized, demanding very accurate clock distribution, for example support for IEEE 1588 transparent clocks or boundary clocks in every intermediate node.

In cellular networks from the LTE radio era onward, phase synchronization is needed in addition to frequency synchronization ([TS36300], [TS23401]). Time constraints are also important due to their impact on packet loss. If a packet is delivered too late, then the packet may be dropped by the host.

#### 6.1.4. Transport Loss Constraints

Fronthaul and Midhaul networks assume almost error-free transport. Errors can result in a reset of the radio interfaces, which can cause reduced throughput or broken radio connectivity for mobile customers.

For packetized Fronthaul and Midhaul connections packet loss may be caused by BER, congestion, or network failure scenarios. Different fronthaul functional splits are being considered by 3GPP, requiring strict frame loss ratio (FLR) guarantees. As one example (referring to the legacy CPRI split which is option 8 in 3GPP) lower layers splits may imply an FLR of less than  $10E-7$  for data traffic and less than  $10E-6$  for control and management traffic. Current tools for eliminating packet loss for Fronthaul and Midhaul networks have serious challenges, for example retransmitting lost packets and/or using forward error correction (FEC) to circumvent bit errors is practically impossible due to the additional delay incurred. Using redundant streams for better guarantees for delivery is also practically impossible in many cases due to high bandwidth requirements of Fronthaul and Midhaul networks. Protection switching is also a candidate but current technologies for the path switch are too slow to avoid reset of mobile interfaces.

Fronthaul links are assumed to be symmetric, and all Fronthaul streams (i.e. those carrying radio data) have equal priority and cannot delay or pre-empt each other. This implies that the network must guarantee that each time-sensitive flow meets their schedule.

#### 6.1.5. Security Considerations

Establishing time-sensitive streams in the network entails reserving networking resources for long periods of time. It is important that these reservation requests be authenticated to prevent malicious reservation attempts from hostile nodes (or accidental misconfiguration). This is particularly important in the case where the reservation requests span administrative domains. Furthermore, the reservation information itself should be digitally signed to reduce the risk of a legitimate node pushing a stale or hostile configuration into another networking node.

Note: This is considered important for the security policy of the network, but does not affect the core DetNet architecture and design.

## 6.2. Cellular Radio Networks Today

### 6.2.1. Fronthaul

Today's Fronthaul networks typically consist of:

- o Dedicated point-to-point fiber connection is common
- o Proprietary protocols and framings
- o Custom equipment and no real networking

Current solutions for Fronthaul are direct optical cables or Wavelength-Division Multiplexing (WDM) connections.

### 6.2.2. Midhaul and Backhaul

Today's Midhaul and Backhaul networks typically consist of:

- o Mostly normal IP networks, MPLS-TP, etc.
- o Clock distribution and sync using 1588 and SyncE

Telecommunication networks in the Mid- and Backhaul are already heading towards transport networks where precise time synchronization support is one of the basic building blocks. While the transport networks themselves have practically transitioned to all-IP packet-based networks to meet the bandwidth and cost requirements, highly accurate clock distribution has become a challenge.

In the past, Mid- and Backhaul connections were typically based on Time Division Multiplexing (TDM-based) and provided frequency synchronization capabilities as a part of the transport media. Alternatively other technologies such as Global Positioning System (GPS) or Synchronous Ethernet (SyncE) are used [SyncE].

Both Ethernet and IP/MPLS [RFC3031] (and PseudoWires (PWE) [RFC3985] for legacy transport support) have become popular tools to build and manage new all-IP Radio Access Networks (RANs) [I-D.kh-spring-ip-ran-use-case]. Although various timing and synchronization optimizations have already been proposed and implemented including 1588 PTP enhancements [I-D.ietf-tictoc-1588overmpls] and [I-D.ietf-mpls-residence-time], these solution are not necessarily sufficient for the forthcoming RAN architectures nor do they guarantee the more stringent time-synchronization requirements such as [CPRI].

There are also existing solutions for TDM over IP such as [RFC5087] and [RFC4553], as well as TDM over Ethernet transports such as [RFC5086].

### 6.3. Cellular Radio Networks Future

Future Cellular Radio Networks will be based on a mix of different xHaul networks (xHaul = front-, mid- and backhaul), and future transport networks should be able to support all of them simultaneously. It is already envisioned today that:

- o Not all "cellular radio network" traffic will be IP, for example some will remain at Layer 2 (e.g. Ethernet based). DetNet solutions must address all traffic types (Layer 2, Layer 3) with the same tools and allow their transport simultaneously.
- o All forms of xHaul networks will need some form of DetNet solutions. For example with the advent of 5G some Backhaul traffic will also have DetNet requirements, for example traffic belonging to time-critical 5G applications.
- o Different splits of the functionality run on the base stations and the on-site units could co-exist on the same Fronthaul and Backhaul network.

We would like to see the following in future Cellular Radio networks:

- o Unified standards-based transport protocols and standard networking equipment that can make use of underlying deterministic link-layer services
- o Unified and standards-based network management systems and protocols in all parts of the network (including Fronthaul)

New radio access network deployment models and architectures may require time- sensitive networking services with strict requirements on other parts of the network that previously were not considered to be packetized at all. Time and synchronization support are already topical for Backhaul and Midhaul packet networks [MEF] and are becoming a real issue for Fronthaul networks also. Specifically in Fronthaul networks the timing and synchronization requirements can be extreme for packet based technologies, for example, on the order of sub +-20 ns packet delay variation (PDV) and frequency accuracy of +0.002 PPM [Fronthaul].

The actual transport protocols and/or solutions to establish required transport "circuits" (pinned-down paths) for Fronthaul traffic are still undefined. Those are likely to include (but are not limited



to) solutions directly over Ethernet, over IP, and using MPLS/PseudoWire transport.

Even the current time-sensitive networking features may not be sufficient for Fronthaul traffic. Therefore, having specific profiles that take the requirements of Fronthaul into account is desirable [IEEE8021CM].

Interesting and important work for time-sensitive networking has been done for Ethernet [TSNTG], which specifies the use of IEEE 1588 time precision protocol (PTP) [IEEE1588] in the context of IEEE 802.1D and IEEE 802.1Q. [IEEE8021AS] specifies a Layer 2 time synchronizing service, and other specifications such as IEEE 1722 [IEEE1722] specify Ethernet-based Layer-2 transport for time-sensitive streams.

New promising work seeks to enable the transport of time-sensitive fronthaul streams in Ethernet bridged networks [IEEE8021CM]. Analogous to IEEE 1722 there is an ongoing standardization effort to define the Layer-2 transport encapsulation format for transporting radio over Ethernet (RoE) in the IEEE 1904.3 Task Force [IEEE19043].

All-IP RANs and xHaul networks would benefit from time synchronization and time-sensitive transport services. Although Ethernet appears to be the unifying technology for the transport, there is still a disconnect providing Layer 3 services. The protocol stack typically has a number of layers below the Ethernet Layer 2 that shows up to the Layer 3 IP transport. It is not uncommon that on top of the lowest layer (optical) transport there is the first layer of Ethernet followed one or more layers of MPLS, PseudoWires and/or other tunneling protocols finally carrying the Ethernet layer visible to the user plane IP traffic.

While there are existing technologies to establish circuits through the routed and switched networks (especially in MPLS/PWE space), there is still no way to signal the time synchronization and time-sensitive stream requirements/reservations for Layer-3 flows in a way that addresses the entire transport stack, including the Ethernet layers that need to be configured.

Furthermore, not all "user plane" traffic will be IP. Therefore, the same solution also must address the use cases where the user plane traffic is a different layer, for example Ethernet frames.

There is existing work describing the problem statement [I-D.finn-detnet-problem-statement] and the architecture [I-D.finn-detnet-architecture] for deterministic networking (DetNet) that targets solutions for time-sensitive (IP/transport) streams with deterministic properties over Ethernet-based switched networks.

#### 6.4. Cellular Radio Networks Asks

A standard for data plane transport specification which is:

- o Unified among all xHauls (meaning that different flows with diverse DetNet requirements can coexist in the same network and traverse the same nodes without interfering with each other)
- o Deployed in a highly deterministic network environment
- o Capable of supporting multiple functional splits simultaneously, including existing Backhaul and CPRI Fronthaul and potentially new modes as defined for example in 3GPP; these goals can be supported by the existing DetNet Use Case Common Themes, notably "Mix of Deterministic and Best-Effort Traffic", "Bounded Latency", "Low Latency", "Symmetrical Path Delays", and "Deterministic Flows".
- o Capable of supporting Network Slicing and Multi-tenancy; these goals can be supported by the same DetNet themes noted above.
- o Capable of transporting both in-band and out-band control traffic (OAM info, ...).
- o Deployable over multiple data link technologies (e.g., IEEE 802.3, mmWave, etc.).

A standard for data flow information models that are:

- o Aware of the time sensitivity and constraints of the target networking environment
- o Aware of underlying deterministic networking services (e.g., on the Ethernet layer)

### 7. Industrial M2M

#### 7.1. Use Case Description

Industrial Automation in general refers to automation of manufacturing, quality control and material processing. In this "machine to machine" (M2M) use case we consider machine units in a plant floor which periodically exchange data with upstream or downstream machine modules and/or a supervisory controller within a local area network.

The actors of M2M communication are Programmable Logic Controllers (PLCs). Communication between PLCs and between PLCs and the

supervisory PLC (S-PLC) is achieved via critical control/data streams Figure 11.

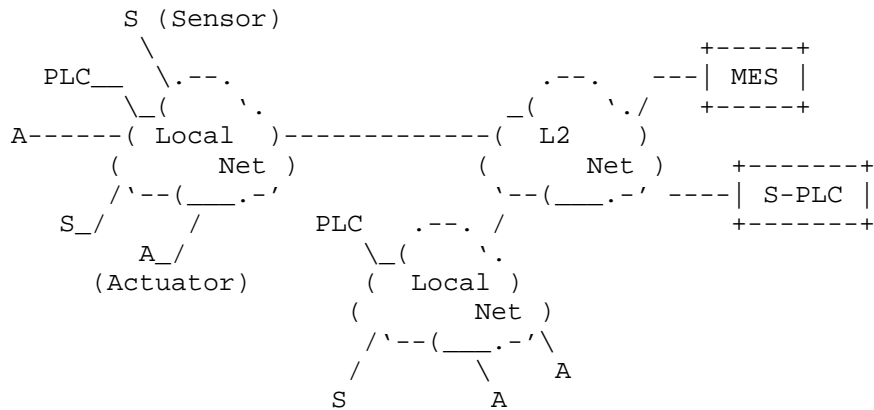


Figure 11: Current Generic Industrial M2M Network Architecture

This use case focuses on PLC-related communications; communication to Manufacturing-Execution-Systems (MESs) are not addressed.

This use case covers only critical control/data streams; non-critical traffic between industrial automation applications (such as communication of state, configuration, set-up, and database communication) are adequately served by currently available prioritizing techniques. Such traffic can use up to 80% of the total bandwidth required. There is also a subset of non-time-critical traffic that must be reliable even though it is not time sensitive.

In this use case the primary need for deterministic networking is to provide end-to-end delivery of M2M messages within specific timing constraints, for example in closed loop automation control. Today this level of determinism is provided by proprietary networking technologies. In addition, standard networking technologies are used to connect the local network to remote industrial automation sites, e.g. over an enterprise or metro network which also carries other types of traffic. Therefore, flows that should be forwarded with deterministic guarantees need to be sustained regardless of the amount of other flows in those networks.

## 7.2. Industrial M2M Communication Today

Today, proprietary networks fulfill the needed timing and availability for M2M networks.

The network topologies used today by industrial automation are similar to those used by telecom networks: Daisy Chain, Ring, Hub and Spoke, and Comb (a subset of Daisy Chain).

PLC-related control/data streams are transmitted periodically and carry either a pre-configured payload or a payload configured during runtime.

Some industrial applications require time synchronization at the end nodes. For such time-coordinated PLCs, accuracy of 1 microsecond is required. Even in the case of "non-time-coordinated" PLCs time sync may be needed e.g. for timestamping of sensor data.

Industrial network scenarios require advanced security solutions. Many of the current industrial production networks are physically separated. Preventing critical flows from be leaked outside a domain is handled today by filtering policies that are typically enforced in firewalls.

#### 7.2.1. Transport Parameters

The Cycle Time defines the frequency of message(s) between industrial actors. The Cycle Time is application dependent, in the range of 1ms - 100ms for critical control/data streams.

Because industrial applications assume deterministic transport for critical Control-Data-Stream parameters (instead of defining latency and delay variation parameters) it is sufficient to fulfill the upper bound of latency (maximum latency). The underlying networking infrastructure must ensure a maximum end-to-end delivery time of messages in the range of 100 microseconds to 50 milliseconds depending on the control loop application.

The bandwidth requirements of control/data streams are usually calculated directly from the bytes-per-cycle parameter of the control loop. For PLC-to-PLC communication one can expect 2 - 32 streams with packet size in the range of 100 - 700 bytes. For S-PLC to PLCs the number of streams is higher - up to 256 streams. Usually no more than 20% of available bandwidth is used for critical control/data streams. In today's networks 1Gbps links are commonly used.

Most PLC control loops are rather tolerant of packet loss, however critical control/data streams accept no more than 1 packet loss per consecutive communication cycle (i.e. if a packet gets lost in cycle "n", then the next cycle ("n+1") must be lossless). After two or more consecutive packet losses the network may be considered to be "down" by the Application.

As network downtime may impact the whole production system the required network availability is rather high (99,999%).

Based on the above parameters we expect that some form of redundancy will be required for M2M communications, however any individual solution depends on several parameters including cycle time, delivery time, etc.

#### 7.2.2. Stream Creation and Destruction

In an industrial environment, critical control/data streams are created rather infrequently, on the order of ~10 times per day / week / month. Most of these critical control/data streams get created at machine startup, however flexibility is also needed during runtime, for example when adding or removing a machine. Going forward as production systems become more flexible, we expect a significant increase in the rate at which streams are created, changed and destroyed.

#### 7.3. Industrial M2M Future

We would like to see a converged IP-standards-based network with deterministic properties that can satisfy the timing, security and reliability constraints described above. Today's proprietary networks could then be interfaced to such a network via gateways or, in the case of new installations, devices could be connected directly to the converged network.

For this use case we expect time synchronization accuracy on the order of 1us.

#### 7.4. Industrial M2M Asks

- o Converged IP-based network
- o Deterministic behavior (bounded latency and jitter )
- o High availability (presumably through redundancy) (99.999 %)
- o Low message delivery time (100us - 50ms)
- o Low packet loss (burstless, 0.1-1 %)
- o Security (e.g. prevent critical flows from being leaked between physically separated networks)

## 8. Mining Industry

### 8.1. Use Case Description

The mining industry is highly dependent on networks to monitor and control their systems both in open-pit and underground extraction, transport and refining processes. In order to reduce risks and increase operational efficiency in mining operations, a number of processes have migrated the operators from the extraction site to remote control and monitoring.

In the case of open pit mining, autonomous trucks are used to transport the raw materials from the open pit to the refining factory where the final product (e.g. Copper) is obtained. Although the operation is autonomous, the trucks are remotely monitored from a central facility.

In pit mines, the monitoring of the tailings or mine dumps is critical in order to avoid any environmental pollution. In the past, monitoring has been conducted through manual inspection of pre-installed dataloggers. Cabling is not usually exploited in such scenarios due to the cost and complex deployment requirements. Currently, wireless technologies are being employed to monitor these cases permanently. Slopes are also monitored in order to anticipate possible mine collapse. Due to the unstable terrain, cable maintenance is costly and complex and hence wireless technologies are employed.

In the underground monitoring case, autonomous vehicles with extraction tools travel autonomously through the tunnels, but their operational tasks (such as excavation, stone breaking and transport) are controlled remotely from a central facility. This generates video and feedback upstream traffic plus downstream actuator control traffic.

### 8.2. Mining Industry Today

Currently the mining industry uses a packet switched architecture supported by high speed ethernet. However in order to achieve the delay and packet loss requirements the network bandwidth is overestimated, thus providing very low efficiency in terms of resource usage.

QoS is implemented at the Routers to separate video, management, monitoring and process control traffic for each stream.

Since mobility is involved in this process, the connection between the backbone and the mobile devices (e.g. trucks, trains and

excavators) is solved using a wireless link. These links are based on 802.11 for open-pit mining and leaky feeder for underground mining.

Lately in pit mines the use of LPWAN technologies has been extended: Tailings, slopes and mine dumps are monitored by battery-powered dataloggers that make use of robust long range radio technologies. Reliability is usually ensured through retransmissions at L2. Gateways or concentrators act as bridges forwarding the data to the backbone ethernet network. Deterministic requirements are biased towards reliability rather than latency as events are slowly triggered or can be anticipated in advance.

At the mineral processing stage, conveyor belts and refining processes are controlled by a SCADA system, which provides the in-factory delay-constrained networking requirements.

Voice communications are currently served by a redundant trunking infrastructure, independent from current data networks.

### 8.3. Mining Industry Future

Mining operations and management are currently converging towards a combination of autonomous operation and teleoperation of transport and extraction machines. This means that video, audio, monitoring and process control traffic will increase dramatically. Ideally, all activities on the mine will rely on network infrastructure.

Wireless for open-pit mining is already a reality with LPWAN technologies and it is expected to evolve to more advanced LPWAN technologies such as those based on LTE to increase last hop reliability or novel LPWAN flavours with deterministic access.

One area in which DetNet can improve this use case is in the wired networks that make up the "backbone network" of the system, which connect together many wireless access points (APs). The mobile machines (which are connected to the network via wireless) transition from one AP to the next as they move about. A deterministic, reliable, low latency backbone can enable these transitions to be more reliable.

Connections which extend all the way from the base stations to the machinery via a mix of wired and wireless hops would also be beneficial, for example to improve remote control responsiveness of digging machines. However to guarantee deterministic performance of a DetNet, the end-to-end underlying network must be deterministic. Thus for this use case if a deterministic wireless transport is

integrated with a wire-based DetNet network, it could create the desired wired plus wireless end-to-end deterministic network.

#### 8.4. Mining Industry Asks

- o Improved bandwidth efficiency
- o Very low delay to enable machine teleoperation
- o Dedicated bandwidth usage for high resolution video streams
- o Predictable delay to enable realtime monitoring
- o Potential to construct a unified DetNet network over a combination of wired and deterministic wireless links

### 9. Private Blockchain

#### 9.1. Use Case Description

Blockchain was created with bitcoin, as a 'public' blockchain on the open Internet, however blockchain has also spread far beyond its original host into various industries such as smart manufacturing, logistics, security, legal rights and others. In these industries blockchain runs in designated and carefully managed network in which deterministic networking requirements could be addressed by Detnet. Such implementations are referred to as 'private' blockchain.

The sole distinction between public and private blockchain is related to who is allowed to participate in the network, execute the consensus protocol and maintain the shared ledger.

Today's networks treat the traffic from blockchain on a best-effort basis, but blockchain operation could be made much more efficient if deterministic networking service were available to minimize latency and packet loss in the network.

##### 9.1.1. Blockchain Operation

A 'block' runs as a container of a batch of primary items such as transactions, property records etc. The blocks are chained in such a way that the hash of the previous block works as the pointer header of the new block, where confirmation of each block requires a consensus mechanism. When an item arrives at a blockchain node, the latter broadcasts this item to the rest of nodes which receive and verify it and put it in the ongoing block. Block confirmation process begins as the amount of items reaches the predefined block



capacity, and the node broadcasts its proved block to the rest of nodes to be verified and chained.

#### 9.1.2. Blockchain Network Architecture

Blockchain node communication and coordination is achieved mainly through frequent point to multi-point communication, however persistent point-to-point connections are used to transport both the items and the blocks to the other nodes.

When a node initiates, it first requests the other nodes' address from a specific entity such as DNS, then it creates persistent connections each of with other nodes. If node A confirms an item, it sends the item to the other nodes via the persistent connections.

As a new block in a node completes and gets proved among the nodes, it starts propagating this block towards its neighbor nodes. Assume node A receives a block, it sends invite message after verification to its neighbor B, B checks if the designated block is available, it responds get message to A if it is unavailable, and A send the complete block to B. B repeats the process as A to start the next round of block propagation.

The challenge of blockchain network operation is not overall data rates, since the volume from both block and item stays between hundreds of bytes to a couple of mega bytes per second, but is in transporting the blocks with minimum latency to maximize efficiency of the blockchain consensus process.

#### 9.1.3. Security Considerations

Security is crucial to blockchain applications, and todayt blockchain addresses its security issues mainly at the application level, where cryptography as well as hash-based consensus play a leading role preventing both double-spending and malicious service attack. However, there is concern that in the proposed use case of a private blockchain network which is dependent on deterministic properties, the network could be vulnerable to delays and other specific attacks against determinism which could interrupt service.

#### 9.2. Private Blockchain Today

Today private blockchain runs in L2 or L3 VPN, in general without guaranteed determinism. The industry players are starting to realize that improving determinism in their blockchain networks could improve the performance of their service, but as of today these goals are not being met.

### 9.3. Private Blockchain Future

Blockchain system performance can be greatly improved through deterministic networking service primarily because it would accelerate the consensus process. It would be valuable to be able to design a private blockchain network with the following properties:

- o Transport of point to multi-point traffic in a coordinated network architecture rather than at the application layer (which typically uses point-to-point connections)
- o Guaranteed transport latency
- o Reduced packet loss (to the point where packet retransmission-incurred delay would be negligible.)

### 9.4. Private Blockchain Asks

- o Layer 2 and Layer 3 multicast of blockchain traffic
- o Item and block delivery with bounded, low latency and negligible packet loss
- o Coexistence in a single network of blockchain and IT traffic.
- o Ability to scale the network by distributing the centralized control of the network across multiple control entities.

## 10. Network Slicing

### 10.1. Use Case Description

Network slicing divides one physical network infrastructure into multiple logical networks. Each slice, corresponding to a logical network, uses resources and network functions independently from each other. Network slicing provides flexibility of resource allocation and service quality customization.

Future services will demand network performance with a wide variety of characteristics such as high data rate, low latency, low loss rate, security and many other parameters. Ideally every service would have its own physical network satisfying its particular performance requirements, however that would be prohibitively expensive. Network slicing can provide a customized slice for a single service, and multiple slices can share the same physical network. This method can optimize the performance for the service at lower cost, and the flexibility of setting up and release the slices also allows the user to allocate the network resources dynamically.

Unlike other DetNet use cases, Network slicing is not a specific application with specific deterministic requirements; it is proposed as a new requirement for the future network, which is still in discussion, and DetNet is a candidate solution for it.

## 10.2. Network Slicing Use Cases

Network Slicing is a core feature of 5G defined in 3GPP, which is currently under development. A Network Slice in mobile network is a complete logical network including Radio Access Network (RAN) and Core Network (CN). It provides telecommunication services and network capabilities, which may vary (or not) from slice to slice.

A 5G bearer network is a typical use case of network slicing, including 3 service scenarios: enhanced Mobile Broadband (eMBB), Ultra-Reliable and Low Latency Communications (URLLC), and massive Machine Type Communications (mMTC). Each of these are described below.

### 10.2.1. Enhanced Mobile Broadband (eMBB)

eMBB focuses on services characterized by high data rates, such as high definition (HD) videos, virtual reality (VR), augmented reality (AR), and fixed mobile convergence (FMC).

### 10.2.2. Ultra-Reliable and Low Latency Communications (URLLC)

URLLC focuses on latency-sensitive services, such as self-driving vehicles, remote surgery, or drone control.

### 10.2.3. massive Machine Type Communications (mMTC)

mMTC focuses on services that have high requirements for connection density, such as those typical for smart city and smart agriculture use cases.

## 10.3. Using DetNet in Network Slicing

One of the requirements discussed for network slicing is the "hard" separation of various users' deterministic performance. That is, it should be impossible for activity, lack of activity, or changes in activity of one or more users to have any appreciable effect on the deterministic performance parameters of any other users. Typical techniques used today, which share a physical network among users, do not offer this kind of insulation. DetNet can supply point-to-point or point-to-multipoint paths that offer bandwidth and latency guarantees to a user that cannot be affected by other users' data traffic.

Thus DetNet is a powerful tool when latency and reliability are required in Network Slicing. However, DetNet cannot cover every Network Slicing use case, and there are some other problems to be solved. Firstly, DetNet is a point-to-point or point-to-multipoint technology while Network Slicing needs multi-point to multi-point guarantee. Second, the number of flows that can be carried by DetNet is limited by DetNet scalability. Flow aggregation and queuing management modification may help to fix the problem. More work and discussions are needed in these topics.

#### 10.4. Network Slicing Today and Future

Network slicing can satisfy the requirements of a lot of future deployment scenario, but it is still a collection of ideas and analysis, without a specific technical solution. A lot of technologies, such as Flex-E, Segment Routing, and DetNet have potential to be used in Network Slicing. For more details please see IETF99 Network Slicing BOF session agenda and materials.

#### 10.5. Network Slicing Asks

- o Isolation from other flows through Queuing Management
- o Service Quality Customization and Guarantee
- o Security

### 11. Use Case Common Themes

This section summarizes the expected properties of a DetNet network, based on the use cases as described in this draft.

#### 11.1. Unified, standards-based network

##### 11.1.1. Extensions to Ethernet

A DetNet network is not "a new kind of network" - it based on extensions to existing Ethernet standards, including elements of IEEE 802.1 AVB/TSN and related standards. Presumably it will be possible to run DetNet over other underlying transports besides Ethernet, but Ethernet is explicitly supported.

##### 11.1.2. Centrally Administered

In general a DetNet network is not expected to be "plug and play" - it is expected that there is some centralized network configuration and control system. Such a system may be in a single central location, or it maybe distributed across multiple control entities

that function together as a unified control system for the network. However, the ability to "hot swap" components (e.g. due to malfunction) is similar enough to "plug and play" that this kind of behavior may be expected in DetNet networks, depending on the implementation.

#### 11.1.3. Standardized Data Flow Information Models

Data Flow Information Models to be used with DetNet networks are to be specified by DetNet.

#### 11.1.4. L2 and L3 Integration

A DetNet network is intended to integrate between Layer 2 (bridged) network(s) (e.g. AVB/TSN LAN) and Layer 3 (routed) network(s) (e.g. using IP-based protocols). One example of this is "making AVB/TSN-type deterministic performance available from Layer 3 applications, e.g. using RTP". Another example is "connecting two AVB/TSN LANs ("islands") together through a standard router".

#### 11.1.5. Guaranteed End-to-End Delivery

Packets sent over DetNet are guaranteed not to be dropped by the network due to congestion. (Packets may however be dropped for intended reasons, e.g. per security measures).

#### 11.1.6. Replacement for Multiple Proprietary Deterministic Networks

There are many proprietary non-interoperable deterministic Ethernet-based networks currently available; DetNet is intended to provide an open-standards-based alternative to such networks.

#### 11.1.7. Mix of Deterministic and Best-Effort Traffic

DetNet is intended to support coexistence of time-sensitive operational (OT) traffic and information (IT) traffic on the same ("unified") network.

#### 11.1.8. Unused Reserved BW to be Available to Best Effort Traffic

If bandwidth reservations are made for a stream but the associated bandwidth is not used at any point in time, that bandwidth is made available on the network for best-effort traffic. If the owner of the reserved stream then starts transmitting again, the bandwidth is no longer available for best-effort traffic, on a moment-to-moment basis. Note that such "temporarily available" bandwidth is not available for time-sensitive traffic, which must have its own reservation.

#### 11.1.1.9. Lower Cost, Multi-Vendor Solutions

The DetNet network specifications are intended to enable an ecosystem in which multiple vendors can create interoperable products, thus promoting device diversity and potentially higher numbers of each device manufactured, promoting cost reduction and cost competition among vendors. The intent is that DetNet networks should be able to be created at lower cost and with greater diversity of available devices than existing proprietary networks.

#### 11.2. Scalable Size

DetNet networks range in size from very small, e.g. inside a single industrial machine, to very large, for example a Utility Grid network spanning a whole country, and involving many "hops" over various kinds of links for example radio repeaters, microwave links, fiber optic links, etc.. However recall that the scope of DetNet is confined to networks that are centrally administered, and explicitly excludes unbounded decentralized networks such as the Internet.

#### 11.3. Scalable Timing Parameters and Accuracy

##### 11.3.1. Bounded Latency

The DetNet Data Flow Information Model is expected to provide means to configure the network that include parameters for querying network path latency, requesting bounded latency for a given stream, requesting worst case maximum and/or minimum latency for a given path or stream, and so on. It is an expected case that the network may not be able to provide a given requested service level, and if so the network control system should reply that the requested services is not available (as opposed to accepting the parameter but then not delivering the desired behavior).

##### 11.3.2. Low Latency

Applications may require "extremely low latency" however depending on the application these may mean very different latency values; for example "low latency" across a Utility grid network is on a different time scale than "low latency" in a motor control loop in a small machine. The intent is that the mechanisms for specifying desired latency include wide ranges, and that architecturally there is nothing to prevent arbitrarily low latencies from being implemented in a given network.

#### 11.3.3. Symmetrical Path Delays

Some applications would like to specify that the transit delay time values be equal for both the transmit and return paths.

#### 11.4. High Reliability and Availability

Reliability is of critical importance to many DetNet applications, in which consequences of failure can be extraordinarily high in terms of cost and even human life. DetNet based systems are expected to be implemented with essentially arbitrarily high availability (for example 99.9999% up time, or even 12 nines). The intent is that the DetNet designs should not make any assumptions about the level of reliability and availability that may be required of a given system, and should define parameters for communicating these kinds of metrics within the network.

A strategy used by DetNet for providing such extraordinarily high levels of reliability is to provide redundant paths that can be seamlessly switched between, while maintaining the required performance of that system.

#### 11.5. Security

Security is of critical importance to many DetNet applications. A DetNet network must be able to be made secure against devices failures, attackers, misbehaving devices, and so on. In a DetNet network the data traffic is expected to be time-sensitive, thus in addition to arriving with the data content as intended, the data must also arrive at the expected time. This may present "new" security challenges to implementers, and must be addressed accordingly. There are other security implications, including (but not limited to) the change in attack surface presented by packet replication and elimination.

#### 11.6. Deterministic Flows

Reserved bandwidth data flows must be isolated from each other and from best-effort traffic, so that even if the network is saturated with best-effort (and/or reserved bandwidth) traffic, the configured flows are not adversely affected.

### 12. Use Cases Explicitly Out of Scope for DetNet

This section contains use case text that has been determined to be outside of the scope of the present DetNet work.

### 12.1. DetNet Scope Limitations

The scope of DetNet is deliberately limited to specific use cases that are consistent with the WG charter, subject to the interpretation of the WG. At the time the DetNet Use Cases were solicited and provided by the authors the scope of DetNet was not clearly defined, and as that clarity has emerged, certain of the use cases have been determined to be outside the scope of the present DetNet work. Such text has been moved into this section to clarify that these use cases will not be supported by the DetNet work.

The text in this section was moved here based on the following "exclusion" principles. Or, as an alternative to moving all such text to this section, some draft text has been modified in situ to reflect these same principles.

The following principles have been established to clarify the scope of the present DetNet work.

- o The scope of network addressed by DetNet is limited to networks that can be centrally controlled, i.e. an "enterprise" aka "corporate" network. This explicitly excludes "the open Internet".
- o Maintaining synchronized time across a DetNet network is crucial to its operation, however DetNet assumes that time is to be maintained using other means, for example (but not limited to) Precision Time Protocol ([IEEE1588]). A use case may state the accuracy and reliability that it expects from the DetNet network as part of a whole system, however it is understood that such timing properties are not guaranteed by DetNet itself. It is currently an open question as to whether DetNet protocols will include a way for an application to communicate such timing expectations to the network, and if so whether they would be expected to materially affect the performance they would receive from the network as a result.

### 12.2. Internet-based Applications

#### 12.2.1. Use Case Description

There are many applications that communicate across the open Internet that could benefit from guaranteed delivery and bounded latency. The following are some representative examples.



#### 12.2.1.1. Media Content Delivery

Media content delivery continues to be an important use of the Internet, yet users often experience poor quality audio and video due to the delay and jitter inherent in today's Internet.

#### 12.2.1.2. Online Gaming

Online gaming is a significant part of the gaming market, however latency can degrade the end user experience. For example "First Person Shooter" (FPS) games are highly delay-sensitive.

#### 12.2.1.3. Virtual Reality

Virtual reality (VR) has many commercial applications including real estate presentations, remote medical procedures, and so on. Low latency is critical to interacting with the virtual world because perceptual delays can cause motion sickness.

#### 12.2.2. Internet-Based Applications Today

Internet service today is by definition "best effort", with no guarantees on delivery or bandwidth.

#### 12.2.3. Internet-Based Applications Future

We imagine an Internet from which we will be able to play a video without glitches and play games without lag.

For online gaming, the maximum round-trip delay can be 100ms and stricter for FPS gaming which can be 10-50ms. Transport delay is the dominate part with a 5-20ms budget.

For VR, 1-10ms maximum delay is needed and total network budget is 1-5ms if doing remote VR.

Flow identification can be used for gaming and VR, i.e. it can recognize a critical flow and provide appropriate latency bounds.

#### 12.2.4. Internet-Based Applications Asks

- o Unified control and management protocols to handle time-critical data flow
- o Application-aware flow filtering mechanism to recognize the timing critical flow without doing 5-tuple matching

- o Unified control plane to provide low latency service on Layer-3 without changing the data plane
- o OAM system and protocols which can help to provide E2E-delay sensitive service provisioning

### 12.3. Pro Audio and Video - Digital Rights Management (DRM)

This section was moved here because this is considered a Link layer topic, not direct responsibility of DetNet.

Digital Rights Management (DRM) is very important to the audio and video industries. Any time protected content is introduced into a network there are DRM concerns that must be maintained (see [CONTENT\_PROTECTION]). Many aspects of DRM are outside the scope of network technology, however there are cases when a secure link supporting authentication and encryption is required by content owners to carry their audio or video content when it is outside their own secure environment (for example see [DCI]).

As an example, two techniques are Digital Transmission Content Protection (DTCP) and High-Bandwidth Digital Content Protection (HDCP). HDCP content is not approved for retransmission within any other type of DRM, while DTCP may be retransmitted under HDCP. Therefore if the source of a stream is outside of the network and it uses HDCP protection it is only allowed to be placed on the network with that same HDCP protection.

### 12.4. Pro Audio and Video - Link Aggregation

Note: The term "Link Aggregation" is used here as defined by the text in the following paragraph, i.e. not following a more common Network Industry definition. Current WG consensus is that this item won't be directly supported by the DetNet architecture, for example because it implies guarantee of in-order delivery of packets which conflicts with the core goal of achieving the lowest possible latency.

For transmitting streams that require more bandwidth than a single link in the target network can support, link aggregation is a technique for combining (aggregating) the bandwidth available on multiple physical links to create a single logical link of the required bandwidth. However, if aggregation is to be used, the network controller (or equivalent) must be able to determine the maximum latency of any path through the aggregate link.

## 13. Contributors

RFC7322 limits the number of authors listed on the front page of a draft to a maximum of 5, far fewer than the 20 individuals below who made important contributions to this draft. The editor wishes to thank and acknowledge each of the following authors for contributing text to this draft. See also Section 14.

Craig Gunther (Harman International)  
10653 South River Front Parkway, South Jordan, UT 84095  
phone +1 801 568-7675, email [craig.gunther@harman.com](mailto:craig.gunther@harman.com)

Pascal Thubert (Cisco Systems, Inc)  
Building D, 45 Allee des Ormes - BP1200, MOUGINS  
Sophia Antipolis 06254 FRANCE  
phone +33 497 23 26 34, email [pthubert@cisco.com](mailto:pthubert@cisco.com)

Patrick Wetterwald (Cisco Systems)  
45 Allee des Ormes, Mougins, 06250 FRANCE  
phone +33 4 97 23 26 36, email [pwetterw@cisco.com](mailto:pwetterw@cisco.com)

Jean Raymond (Hydro-Quebec)  
1500 University, Montreal, H3A3S7, Canada  
phone +1 514 840 3000, email [raymond.jean@hydro.qc.ca](mailto:raymond.jean@hydro.qc.ca)

Jouni Korhonen (Broadcom Corporation)  
3151 Zanker Road, San Jose, 95134, CA, USA  
email [jouni.nospam@gmail.com](mailto:jouni.nospam@gmail.com)

Yu Kaneko (Toshiba)  
1 Komukai-Toshiba-cho, Saiwai-ku, Kasasaki-shi, Kanagawa, Japan  
email [yul.kaneko@toshiba.co.jp](mailto:yul.kaneko@toshiba.co.jp)

Subir Das (Applied Communication Sciences)  
150 Mount Airy Road, Basking Ridge, New Jersey, 07920, USA  
email [sdas@appcomsci.com](mailto:sdas@appcomsci.com)

Yiyong Zha (Huawei Technologies)  
email

Balazs Varga (Ericsson)  
Konyves Kalman krt. 11/B, Budapest, Hungary, 1097  
email [balazs.a.varga@ericsson.com](mailto:balazs.a.varga@ericsson.com)

Janos Farkas (Ericsson)  
Konyves Kalman krt. 11/B, Budapest, Hungary, 1097  
email [janos.farkas@ericsson.com](mailto:janos.farkas@ericsson.com)

Franz-Josef Goetz (Siemens)  
Gleiwitzerstr. 555, Nurnberg, Germany, 90475  
email franz-josef.goetz@siemens.com

Juergen Schmitt (Siemens)  
Gleiwitzerstr. 555, Nurnberg, Germany, 90475  
email juergen.jues.schmitt@siemens.com

Xavier Vilajosana (Worldsensing)  
483 Arago, Barcelona, Catalonia, 08013, Spain  
email xvilajosana@worldsensing.com

Toktam Mahmoodi (King's College London)  
Strand, London WC2R 2LS, United Kingdom  
email toktam.mahmoodi@kcl.ac.uk

Spiros Spirou (Intracom Telecom)  
19.7 km Markopoulou Ave., Peania, Attiki, 19002, Greece  
email spiros.spirou@gmail.com

Petra Vizarreta (Technical University of Munich)  
Maxvorstadt, ArcisstraBe 21, Munich, 80333, Germany  
email petra.stojsavljevic@tum.de

Daniel Huang (ZTE Corporation, Inc.)  
No. 50 Software Avenue, Nanjing, Jiangsu, 210012, P.R. China  
email huang.guangping@zte.com.cn

Xuesong Geng (Huawei Technologies)  
email gengxuesong@huawei.com

Diego Dujovne (Universidad Diego Portales)  
email diego.dujovne@mail.udp.cl

Maik Seewald (Cisco Systems)  
email maseewal@cisco.com

## 14. Acknowledgments

### 14.1. Pro Audio

This section was derived from draft-gunther-detnet-proaudio-req-01.

The editors would like to acknowledge the help of the following individuals and the companies they represent:

Jeff Koftinoff, Meyer Sound

Jouni Korhonen, Associate Technical Director, Broadcom

Pascal Thubert, CTAO, Cisco

Kieran Tyrrell, Sienda New Media Technologies GmbH

#### 14.2. Utility Telecom

This section was derived from draft-wetterwald-detnet-utilities-reqs-02.

Faramarz Maghsoodlou, Ph. D. IoT Connected Industries and Energy Practice Cisco

Pascal Thubert, CTAO Cisco

#### 14.3. Building Automation Systems

This section was derived from draft-bas-usecase-detnet-00.

#### 14.4. Wireless for Industrial

This section was derived from draft-thubert-6tisch-4detnet-01.

This specification derives from the 6TiSCH architecture, which is the result of multiple interactions, in particular during the 6TiSCH (bi)Weekly Interim call, relayed through the 6TiSCH mailing list at the IETF.

The authors wish to thank: Kris Pister, Thomas Watteyne, Xavier Vilajosana, Qin Wang, Tom Phinney, Robert Assimiti, Michael Richardson, Zhuo Chen, Malisa Vucinic, Alfredo Grieco, Martin Turon, Dominique Barthel, Elvis Vogli, Guillaume Gaillard, Herman Storey, Maria Rita Palattella, Nicola Accettura, Patrick Wetterwald, Pouria Zand, Raghuram Sudhaakar, and Shitanshu Shah for their participation and various contributions.

#### 14.5. Cellular Radio

This section was derived from draft-korhonen-detnet-telreq-00.

#### 14.6. Industrial M2M

The authors would like to thank Feng Chen and Marcel Kiessling for their comments and suggestions.

#### 14.7. Internet Applications and CoMP

This section was derived from draft-zha-detnet-use-case-00.

This document has benefited from reviews, suggestions, comments and proposed text provided by the following members, listed in alphabetical order: Jing Huang, Junru Lin, Lehong Niu and Oilver Huang.

#### 14.8. Electrical Utilities

The wind power generation use case has been extracted from the study of Wind Farms conducted within the 5GPPP Virtuwind Project. The project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 671648 (VirtuWind).

#### 14.9. Network Slicing

This section was written by Xuesong Geng, who would like to acknowledge Norm Finn and Mach Chen for their useful comments.

#### 14.10. Mining

This section was written by Diego Dujovne in conjunction with Xavier Vilasojana.

#### 14.11. Private Blockchain

This section was written by Daniel Huang.

### 15. Informative References

- [ACE] IETF, "Authentication and Authorization for Constrained Environments",  
<<https://datatracker.ietf.org/doc/charter-ietf-ace/>>.
- [Ahm14] Ahmed, M. and R. Kim, "Communication network architectures for smart-wind power farms.", *Energies*, p. 3900-3921. , June 2014.
- [bacnetip] ASHRAE, "Annex J to ANSI/ASHRAE 135-1995 - BACnet/IP", January 1999.
- [CCAMP] IETF, "Common Control and Measurement Plane",  
<<https://datatracker.ietf.org/doc/charter-ietf-ccamp/>>.

- [CoMP] NGMN Alliance, "RAN EVOLUTION PROJECT COMP EVALUATION AND ENHANCEMENT", NGMN Alliance NGMN\_RANEV\_D3\_CoMP\_Evaluation\_and\_Enhancement\_v2.0, March 2015, <[https://www.ngmn.org/uploads/media/NGMN\\_RANEV\\_D3\\_CoMP\\_Evaluation\\_and\\_Enhancement\\_v2.0.pdf](https://www.ngmn.org/uploads/media/NGMN_RANEV_D3_CoMP_Evaluation_and_Enhancement_v2.0.pdf)>.
- [CONTENT\_PROTECTION] Olsen, D., "1722a Content Protection", 2012, <[http://grouper.ieee.org/groups/1722/contributions/2012/avtp\\_dolsen\\_1722a\\_content\\_protection.pdf](http://grouper.ieee.org/groups/1722/contributions/2012/avtp_dolsen_1722a_content_protection.pdf)>.
- [CPRI] CPRI Cooperation, "Common Public Radio Interface (CPRI); Interface Specification", CPRI Specification V6.1, July 2014, <[http://www.cpri.info/downloads/CPRI\\_v\\_6\\_1\\_2014-07-01.pdf](http://www.cpri.info/downloads/CPRI_v_6_1_2014-07-01.pdf)>.
- [CPRI-transp] CPRI TWG, "CPRI requirements for Ethernet Fronthaul", November 2015, <<http://www.ieee802.org/1/files/public/docs2015/cm-CPRI-requirements-1115-v01.pdf>>.
- [DCI] Digital Cinema Initiatives, LLC, "DCI Specification, Version 1.2", 2012, <<http://www.dcinovies.com/>>.
- [DICE] IETF, "DTLS In Constrained Environments", <<https://datatracker.ietf.org/doc/charter-ietf-dice/>>.
- [EA12] Evans, P. and M. Annunziata, "Industrial Internet: Pushing the Boundaries of Minds and Machines", November 2012.
- [ESPN\_DC2] Daley, D., "ESPN's DC2 Scales AVB Large", 2014, <<http://sportsvideo.org/main/blog/2014/06/espns-dc2-scales-avb-large>>.
- [flnet] Japan Electrical Manufacturers Association, "JEMA 1479 - English Edition", September 2012.
- [Fronthaul] Chen, D. and T. Mustala, "Ethernet Fronthaul Considerations", IEEE 1904.3, February 2015, <[http://www.ieee1904.org/3/meeting\\_archive/2015/02/tf3\\_1502\\_chen\\_la.pdf](http://www.ieee1904.org/3/meeting_archive/2015/02/tf3_1502_chen_la.pdf)>.
- [HART] [www.hartcomm.org](http://www.hartcomm.org), "Highway Addressable remote Transducer, a group of specifications for industrial process and control devices administered by the HART Foundation".

- [I-D.finn-detnet-architecture]  
Finn, N. and P. Thubert, "Deterministic Networking Architecture", draft-finn-detnet-architecture-08 (work in progress), August 2016.
- [I-D.finn-detnet-problem-statement]  
Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", draft-finn-detnet-problem-statement-05 (work in progress), March 2016.
- [I-D.ietf-6tisch-6top-interface]  
Wang, Q. and X. Vilajosana, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-04 (work in progress), July 2015.
- [I-D.ietf-6tisch-architecture]  
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-13 (work in progress), November 2017.
- [I-D.ietf-6tisch-coap]  
Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-03 (work in progress), March 2015.
- [I-D.ietf-6tisch-terminology]  
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-09 (work in progress), June 2017.
- [I-D.ietf-ipv6-multilink-subnets]  
Thaler, D. and C. Huitema, "Multi-link Subnet Support in IPv6", draft-ietf-ipv6-multilink-subnets-00 (work in progress), July 2002.
- [I-D.ietf-mpls-residence-time]  
Mirsky, G., Ruffini, S., Gray, E., Drake, J., Bryant, S., and S. Vainshtein, "Residence Time Measurement in MPLS network", draft-ietf-mpls-residence-time-15 (work in progress), March 2017.
- [I-D.ietf-roll-rpl-industrial-applicability]  
Phinney, T., Thubert, P., and R. Assimiti, "RPL applicability in industrial networks", draft-ietf-roll-rpl-industrial-applicability-02 (work in progress), October 2013.



- [I-D.ietf-tictoc-1588overmpls]  
Davari, S., Oren, A., Bhatia, M., Roberts, P., and L. Montini, "Transporting Timing messages over MPLS Networks", draft-ietf-tictoc-1588overmpls-07 (work in progress), October 2015.
- [I-D.kh-spring-ip-ran-use-case]  
Khasnabish, B., hu, f., and L. Contreras, "Segment Routing in IP RAN use case", draft-kh-spring-ip-ran-use-case-02 (work in progress), November 2014.
- [I-D.svshah-tsvwg-deterministic-forwarding]  
Shah, S. and P. Thubert, "Deterministic Forwarding PHB", draft-svshah-tsvwg-deterministic-forwarding-04 (work in progress), August 2015.
- [I-D.thubert-6lowpan-backbone-router]  
Thubert, P., "6LoWPAN Backbone Router", draft-thubert-6lowpan-backbone-router-03 (work in progress), February 2013.
- [I-D.wang-6tisch-6top-sublayer]  
Wang, Q. and X. Vilajosana, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-04 (work in progress), November 2015.
- [IEC-60870-5-104]  
International Electrotechnical Commission, "International Standard IEC 60870-5-104: Network access for IEC 60870-5-101 using standard transport profiles", June 2006.
- [IEC61400]  
"International standard 61400-25: Communications for monitoring and control of wind power plants", June 2013.
- [IEC61850-90-12]  
TC57 WG10, IEC., "IEC 61850-90-12 TR: Communication networks and systems for power utility automation - Part 90-12: Wide area network engineering guidelines", 2015.
- [IEC62439-3:2012]  
TC65, IEC., "IEC 62439-3: Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)", 2012.

- [IEEE1588]  
IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008, 2008, <<http://standards.ieee.org/findstds/standard/1588-2008.html>>.
- [IEEE1646]  
"Communication Delivery Time Performance Requirements for Electric Power Substation Automation", IEEE Standard 1646-2004 , Apr 2004.
- [IEEE1722]  
IEEE, "1722-2011 - IEEE Standard for Layer 2 Transport Protocol for Time Sensitive Applications in a Bridged Local Area Network", IEEE Std 1722-2011, 2011, <<http://standards.ieee.org/findstds/standard/1722-2011.html>>.
- [IEEE19043]  
IEEE Standards Association, "IEEE 1904.3 TF", IEEE 1904.3, 2015, <[http://www.ieee1904.org/3/tf3\\_home.shtml](http://www.ieee1904.org/3/tf3_home.shtml)>.
- [IEEE802.1TSNTG]  
IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networks Task Group", March 2013, <<http://www.ieee802.org/1/pages/avbridges.html>>.
- [IEEE802154]  
IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks".
- [IEEE802154e]  
IEEE standard for Information Technology, "IEEE standard for Information Technology, IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks, June 2011 as amended by IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer", April 2012.

- [IEEE802.1AS]  
IEEE, "Timing and Synchronizations (IEEE 802.1AS-2011)",  
IEEE 802.1AS-2001, 2011,  
<<http://standards.ieee.org/getIEEE802/download/802.1AS-2011.pdf>>.
- [IEEE802.1CM]  
Farkas, J., "Time-Sensitive Networking for Fronthaul",  
Unapproved PAR, PAR for a New IEEE Standard;  
IEEE P802.1CM, April 2015,  
<<http://www.ieee802.org/1/files/public/docs2015/new-P802-1CM-dr-aft-PAR-0515-v02.pdf>>.
- [IEEE802.1TSN]  
IEEE 802.1, "The charter of the TG is to provide the  
specifications that will allow time-synchronized low  
latency streaming services through 802 networks.", 2016,  
<<http://www.ieee802.org/1/pages/tsn.html>>.
- [IETFDetNet]  
IETF, "Charter for IETF DetNet Working Group", 2015,  
<<https://datatracker.ietf.org/wg/detnet/charter/>>.
- [ISA100] ISA/ANSI, "ISA100, Wireless Systems for Automation",  
<<https://www.isa.org/isa100/>>.
- [ISA100.11a]  
ISA/ANSI, "Wireless Systems for Industrial Automation:  
Process Control and Related Applications - ISA100.11a-2011  
- IEC 62734", 2011, <<http://www.isa.org/Community/SP100WirelessSystemsforAutomation>>.
- [ISO7240-16]  
ISO, "ISO 7240-16:2007 Fire detection and alarm systems --  
Part 16: Sound system control and indicating equipment",  
2007, <[http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=42978](http://www.iso.org/iso/catalogue_detail.htm?csnumber=42978)>.
- [knx] KNX Association, "ISO/IEC 14543-3 - KNX", November 2006.
- [lontalk] ECHELON, "LonTalk(R) Protocol Specification Version 3.0",  
1994.
- [LTE-Latency]  
Johnston, S., "LTE Latency: How does it compare to other  
technologies", March 2014,  
<<http://opensignal.com/blog/2014/03/10/lte-latency-how-does-it-compare-to-other-technologies>>.

- [MEF] MEF, "Mobile Backhaul Phase 2 Amendment 1 -- Small Cells", MEF 22.1.1, July 2014, <[http://www.mef.net/Assets/Technical\\_Specifications/PDF/MEF\\_22.1.1.pdf](http://www.mef.net/Assets/Technical_Specifications/PDF/MEF_22.1.1.pdf)>.
- [METIS] METIS, "Scenarios, requirements and KPIs for 5G mobile and wireless system", ICT-317669-METIS/D1.1 ICT-317669-METIS/D1.1, April 2013, <[https://www.metis2020.com/wp-content/uploads/deliverables/METIS\\_D1.1\\_v1.pdf](https://www.metis2020.com/wp-content/uploads/deliverables/METIS_D1.1_v1.pdf)>.
- [modbus] Modbus Organization, "MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b", December 2006.
- [MODBUS] Modbus Organization, Inc., "MODBUS Application Protocol Specification", Apr 2012.
- [net5G] Ericsson, "5G Radio Access, Challenges for 2020 and Beyond", Ericsson white paper wp-5g, June 2013, <<http://www.ericsson.com/res/docs/whitepapers/wp-5g.pdf>>.
- [NGMN] NGMN Alliance, "5G White Paper", NGMN 5G White Paper v1.0, February 2015, <[https://www.ngmn.org/uploads/media/NGMN\\_5G\\_White\\_Paper\\_V1\\_0.pdf](https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf)>.
- [NGMN-fronth] NGMN Alliance, "Fronthaul Requirements for C-RAN", March 2015, <[https://www.ngmn.org/uploads/media/NGMN\\_RANEV\\_D1\\_C-RAN\\_Fronthaul\\_Requirements\\_v1.0.pdf](https://www.ngmn.org/uploads/media/NGMN_RANEV_D1_C-RAN_Fronthaul_Requirements_v1.0.pdf)>.
- [OPCXML] OPC Foundation, "OPC XML-Data Access Specification", Dec 2004.
- [PCE] IETF, "Path Computation Element", <<https://datatracker.ietf.org/doc/charter-ietf-pce/>>.
- [profibus] IEC, "IEC 61158 Type 3 - Profibus DP", January 2001.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.

- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <<https://www.rfc-editor.org/info/rfc3393>>.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, DOI 10.17487/RFC3411, December 2002, <<https://www.rfc-editor.org/info/rfc3411>>.
- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <<https://www.rfc-editor.org/info/rfc3444>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<https://www.rfc-editor.org/info/rfc3972>>.
- [RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <<https://www.rfc-editor.org/info/rfc3985>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.

- [RFC4553] Vainshtein, A., Ed. and YJ. Stein, Ed., "Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)", RFC 4553, DOI 10.17487/RFC4553, June 2006, <<https://www.rfc-editor.org/info/rfc4553>>.
- [RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, DOI 10.17487/RFC4903, June 2007, <<https://www.rfc-editor.org/info/rfc4903>>.
- [RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <<https://www.rfc-editor.org/info/rfc4919>>.
- [RFC5086] Vainshtein, A., Ed., Sasson, I., Metz, E., Frost, T., and P. Pate, "Structure-Aware Time Division Multiplexed (TDM) Circuit Emulation Service over Packet Switched Network (CESoPSN)", RFC 5086, DOI 10.17487/RFC5086, December 2007, <<https://www.rfc-editor.org/info/rfc5086>>.
- [RFC5087] Stein, Y(J)., Shashoua, R., Insler, R., and M. Anavi, "Time Division Multiplexing over IP (TDMoIP)", RFC 5087, DOI 10.17487/RFC5087, December 2007, <<https://www.rfc-editor.org/info/rfc5087>>.
- [RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <<https://www.rfc-editor.org/info/rfc6282>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <<https://www.rfc-editor.org/info/rfc6551>>.

- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <<https://www.rfc-editor.org/info/rfc6775>>.
- [RFC7554] Watteyne, T., Ed., Palattella, M., and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement", RFC 7554, DOI 10.17487/RFC7554, May 2015, <<https://www.rfc-editor.org/info/rfc7554>>.
- [Spe09] Sperotto, A., Sadre, R., Vliet, F., and A. Pras, "A First Look into SCADA Network Traffic", IP Operations and Management, p. 518-521. , June 2009.
- [SRP\_LATENCY] Gunther, C., "Specifying SRP Latency", 2014, <<http://www.ieee802.org/1/files/public/docs2014/cc-cgunther-acceptable-latency-0314-v01.pdf>>.
- [STUDIO\_IP] Mace, G., "IP Networked Studio Infrastructure for Synchronized & Real-Time Multimedia Transmissions", 2007, <<http://www.ieee802.org/1/files/public/docs2047/avb-mace-ip-networked-studio-infrastructure-0107.pdf>>.
- [SyncE] ITU-T, "G.8261 : Timing and synchronization aspects in packet networks", Recommendation G.8261, August 2013, <<http://www.itu.int/rec/T-REC-G.8261>>.
- [TEAS] IETF, "Traffic Engineering Architecture and Signaling", <<https://datatracker.ietf.org/doc/charter-ietf-teas/>>.
- [TS23401] 3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", 3GPP TS 23.401 10.10.0, March 2013.
- [TS25104] 3GPP, "Base Station (BS) radio transmission and reception (FDD)", 3GPP TS 25.104 3.14.0, March 2007.
- [TS36104] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception", 3GPP TS 36.104 10.11.0, July 2013.
- [TS36133] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management", 3GPP TS 36.133 12.7.0, April 2015.

- [TS36211] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation", 3GPP TS 36.211 10.7.0, March 2013.
- [TS36300] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2", 3GPP TS 36.300 10.11.0, September 2013.
- [TSNTG] IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networks Task Group", 2013, <<http://www.IEEE802.org/1/pages/avbridges.html>>.
- [UHD-video]
- Holub, P., "Ultra-High Definition Videos and Their Applications over the Network", The 7th International Symposium on VICTORIES Project PetrHolub\_presentation, October 2014, <[http://www.aist-victories.org/jp/7th\\_sympo\\_ws/PetrHolub\\_presentation.pdf](http://www.aist-victories.org/jp/7th_sympo_ws/PetrHolub_presentation.pdf)>.
- [WirelessHART]
- [www.hartcomm.org](http://www.hartcomm.org), "Industrial Communication Networks - Wireless Communication Network and Communication Profiles - WirelessHART - IEC 62591", 2010.

## Author's Address

Ethan Grossman (editor)  
Dolby Laboratories, Inc.  
1275 Market Street  
San Francisco, CA 94103  
USA

Phone: +1 415 645 4726  
Email: [ethan.grossman@dolby.com](mailto:ethan.grossman@dolby.com)  
URI: <http://www.dolby.com>



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: June 22, 2019

E. Grossman, Ed.  
DOLBY  
December 19, 2018

Deterministic Networking Use Cases  
draft-ietf-detnet-use-cases-20

Abstract

This draft presents use cases from diverse industries which have in common a need for "deterministic flows". "Deterministic" in this context means that such flows provide guaranteed bandwidth, bounded latency, and other properties germane to the transport of time-sensitive data. These use cases differ notably in their network topologies and specific desired behavior, providing as a group broad industry context for DetNet. For each use case, this document will identify the use case, identify representative solutions used today, and describe potential improvements that DetNet can enable. The Use Case Common Themes section then extracts and enumerates the set of common properties implied by these use cases.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 22, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	5
2. Pro Audio and Video . . . . .	7
2.1. Use Case Description . . . . .	7
2.1.1. Uninterrupted Stream Playback . . . . .	7
2.1.2. Synchronized Stream Playback . . . . .	8
2.1.3. Sound Reinforcement . . . . .	8
2.1.4. Secure Transmission . . . . .	9
2.1.4.1. Safety . . . . .	9
2.2. Pro Audio Today . . . . .	9
2.3. Pro Audio Future . . . . .	9
2.3.1. Layer 3 Interconnecting Layer 2 Islands . . . . .	9
2.3.2. High Reliability Stream Paths . . . . .	10
2.3.3. Integration of Reserved Streams into IT Networks . . . . .	10
2.3.4. Use of Unused Reservations by Best-Effort Traffic . . . . .	10
2.3.5. Traffic Segregation . . . . .	11
2.3.5.1. Packet Forwarding Rules, VLANs and Subnets . . . . .	11
2.3.5.2. Multicast Addressing (IPv4 and IPv6) . . . . .	11
2.3.6. Latency Optimization by a Central Controller . . . . .	12
2.3.7. Reduced Device Cost Due To Reduced Buffer Memory . . . . .	12
2.4. Pro Audio Asks . . . . .	12
3. Electrical Utilities . . . . .	13
3.1. Use Case Description . . . . .	13
3.1.1. Transmission Use Cases . . . . .	13
3.1.1.1. Protection . . . . .	13
3.1.1.2. Intra-Substation Process Bus Communications . . . . .	18
3.1.1.3. Wide Area Monitoring and Control Systems . . . . .	19
3.1.1.4. IEC 61850 WAN engineering guidelines requirement classification . . . . .	20
3.1.2. Generation Use Case . . . . .	21
3.1.2.1. Control of the Generated Power . . . . .	21
3.1.2.2. Control of the Generation Infrastructure . . . . .	22
3.1.3. Distribution use case . . . . .	27
3.1.3.1. Fault Location Isolation and Service Restoration (FLISR) . . . . .	27
3.2. Electrical Utilities Today . . . . .	28
3.2.1. Security Current Practices and Limitations . . . . .	28
3.3. Electrical Utilities Future . . . . .	30
3.3.1. Migration to Packet-Switched Network . . . . .	31
3.3.2. Telecommunications Trends . . . . .	31

3.3.2.1.	General Telecommunications Requirements . . . . .	31
3.3.2.2.	Specific Network topologies of Smart Grid Applications . . . . .	32
3.3.2.3.	Precision Time Protocol . . . . .	33
3.3.3.	Security Trends in Utility Networks . . . . .	34
3.4.	Electrical Utilities Asks . . . . .	36
4.	Building Automation Systems . . . . .	36
4.1.	Use Case Description . . . . .	36
4.2.	Building Automation Systems Today . . . . .	37
4.2.1.	BAS Architecture . . . . .	37
4.2.2.	BAS Deployment Model . . . . .	38
4.2.3.	Use Cases for Field Networks . . . . .	40
4.2.3.1.	Environmental Monitoring . . . . .	40
4.2.3.2.	Fire Detection . . . . .	40
4.2.3.3.	Feedback Control . . . . .	41
4.2.4.	Security Considerations . . . . .	41
4.3.	BAS Future . . . . .	41
4.4.	BAS Asks . . . . .	42
5.	Wireless for Industrial Applications . . . . .	42
5.1.	Use Case Description . . . . .	42
5.1.1.	Network Convergence using 6TiSCH . . . . .	43
5.1.2.	Common Protocol Development for 6TiSCH . . . . .	43
5.2.	Wireless Industrial Today . . . . .	44
5.3.	Wireless Industrial Future . . . . .	44
5.3.1.	Unified Wireless Network and Management . . . . .	44
5.3.1.1.	PCE and 6TiSCH ARQ Retries . . . . .	46
5.3.2.	Schedule Management by a PCE . . . . .	47
5.3.2.1.	PCE Commands and 6TiSCH CoAP Requests . . . . .	47
5.3.2.2.	6TiSCH IP Interface . . . . .	48
5.3.3.	6TiSCH Security Considerations . . . . .	49
5.4.	Wireless Industrial Asks . . . . .	49
6.	Cellular Radio . . . . .	49
6.1.	Use Case Description . . . . .	49
6.1.1.	Network Architecture . . . . .	49
6.1.2.	Delay Constraints . . . . .	50
6.1.3.	Time Synchronization Constraints . . . . .	52
6.1.4.	Transport Loss Constraints . . . . .	54
6.1.5.	Security Considerations . . . . .	54
6.2.	Cellular Radio Networks Today . . . . .	55
6.2.1.	Fronthaul . . . . .	55
6.2.2.	Midhaul and Backhaul . . . . .	55
6.3.	Cellular Radio Networks Future . . . . .	56
6.4.	Cellular Radio Networks Asks . . . . .	58
7.	Industrial Machine to Machine (M2M) . . . . .	59
7.1.	Use Case Description . . . . .	59
7.2.	Industrial M2M Communication Today . . . . .	60
7.2.1.	Transport Parameters . . . . .	60
7.2.2.	Stream Creation and Destruction . . . . .	61

7.3.	Industrial M2M Future . . . . .	61
7.4.	Industrial M2M Asks . . . . .	62
8.	Mining Industry . . . . .	62
8.1.	Use Case Description . . . . .	62
8.2.	Mining Industry Today . . . . .	63
8.3.	Mining Industry Future . . . . .	63
8.4.	Mining Industry Asks . . . . .	64
9.	Private Blockchain . . . . .	64
9.1.	Use Case Description . . . . .	64
9.1.1.	Blockchain Operation . . . . .	65
9.1.2.	Blockchain Network Architecture . . . . .	65
9.1.3.	Security Considerations . . . . .	66
9.2.	Private Blockchain Today . . . . .	66
9.3.	Private Blockchain Future . . . . .	66
9.4.	Private Blockchain Asks . . . . .	67
10.	Network Slicing . . . . .	67
10.1.	Use Case Description . . . . .	67
10.2.	DetNet Applied to Network Slicing . . . . .	67
10.2.1.	Resource Isolation Across Slices . . . . .	67
10.2.2.	Deterministic Services Within Slices . . . . .	68
10.3.	A Network Slicing Use Case Example - 5G Bearer Network . . . . .	68
10.4.	Non-5G Applications of Network Slicing . . . . .	69
10.5.	Limitations of DetNet in Network Slicing . . . . .	69
10.6.	Network Slicing Today and Future . . . . .	69
10.7.	Network Slicing Asks . . . . .	69
11.	Use Case Common Themes . . . . .	69
11.1.	Unified, standards-based network . . . . .	70
11.1.1.	Extensions to Ethernet . . . . .	70
11.1.2.	Centrally Administered . . . . .	70
11.1.3.	Standardized Data Flow Information Models . . . . .	70
11.1.4.	L2 and L3 Integration . . . . .	70
11.1.5.	Consideration for IPv4 . . . . .	70
11.1.6.	Guaranteed End-to-End Delivery . . . . .	71
11.1.7.	Replacement for Multiple Proprietary Deterministic Networks . . . . .	71
11.1.8.	Mix of Deterministic and Best-Effort Traffic . . . . .	71
11.1.9.	Unused Reserved BW to be Available to Best-Effort Traffic . . . . .	71
11.1.10.	Lower Cost, Multi-Vendor Solutions . . . . .	71
11.2.	Scalable Size . . . . .	71
11.2.1.	Scalable Number of Flows . . . . .	72
11.3.	Scalable Timing Parameters and Accuracy . . . . .	72
11.3.1.	Bounded Latency . . . . .	72
11.3.2.	Low Latency . . . . .	72
11.3.3.	Bounded Jitter (Latency Variation) . . . . .	72
11.3.4.	Symmetrical Path Delays . . . . .	72
11.4.	High Reliability and Availability . . . . .	73
11.5.	Security . . . . .	73

11.6. Deterministic Flows . . . . .	73
12. Security Considerations . . . . .	73
13. Contributors . . . . .	74
14. Acknowledgments . . . . .	75
14.1. Pro Audio . . . . .	75
14.2. Utility Telecom . . . . .	76
14.3. Building Automation Systems . . . . .	76
14.4. Wireless for Industrial Applications . . . . .	76
14.5. Cellular Radio . . . . .	76
14.6. Industrial Machine to Machine (M2M) . . . . .	77
14.7. Internet Applications and CoMP . . . . .	77
14.8. Network Slicing . . . . .	77
14.9. Mining . . . . .	77
14.10. Private Blockchain . . . . .	77
15. IANA Considerations . . . . .	77
16. Informative References . . . . .	77
Appendix A. Use Cases Explicitly Out of Scope for DetNet . . . .	84
A.1. DetNet Scope Limitations . . . . .	85
A.2. Internet-based Applications . . . . .	85
A.2.1. Use Case Description . . . . .	86
A.2.1.1. Media Content Delivery . . . . .	86
A.2.1.2. Online Gaming . . . . .	86
A.2.1.3. Virtual Reality . . . . .	86
A.2.2. Internet-Based Applications Today . . . . .	86
A.2.3. Internet-Based Applications Future . . . . .	86
A.2.4. Internet-Based Applications Asks . . . . .	86
A.3. Pro Audio and Video - Digital Rights Management (DRM) . .	87
A.4. Pro Audio and Video - Link Aggregation . . . . .	87
A.5. Pro Audio and Video - Deterministic Time to Establish Streaming . . . . .	87
Author's Address . . . . .	88

## 1. Introduction

This draft documents use cases in diverse industries which require deterministic flows over multi-hop paths. DetNet flows can be established from either a Layer 2 or Layer 3 (IP) interface, and such flows can co-exist on an IP network with best-effort traffic. DetNet also provides for highly reliable flows through provision for redundant paths.

The DetNet Use Cases explicitly do not suggest any specific design for DetNet architecture or protocols; these are topics of other DetNet drafts.

The DetNet use cases as originally submitted explicitly were not considered by the DetNet Working Group to be concrete requirements; The DetNet Working Group and Design Team considered these use cases,

identifying which elements of them could be feasibly implemented within the charter of DetNet, and as a result certain of the originally submitted use cases (or elements of them) have been moved to the Use Cases Explicitly Out of Scope for DetNet section.

The DetNet Use Cases document provide context regarding DetNet design decisions. It also serves a long-lived purpose of helping those learning (or new to) DetNet to understand the types of applications that can be supported by DetNet. It also allow those WG contributors who are users to ensure that their concerns are addressed by the WG; for them this document both covers their contribution and provides a long term reference to the problems they expect to be served by the technology, both in the short term deliverables and as the technology evolves in the future.

The DetNet Use Cases document has served as a "yardstick" against which proposed DetNet designs can be measured, answering the question "to what extent does a proposed design satisfy these various use cases?"

The Use Case industries covered are professional audio, electrical utilities, building automation systems, wireless for industrial applications, cellular radio, industrial machine-to-machine, mining, private blockchain, and network slicing. For each use case the following questions are answered:

- o What is the use case?
- o How is it addressed today?
- o How should it be addressed in the future?
- o What should the IETF deliver to enable this use case?

The level of detail in each use case is intended to be sufficient to express the relevant elements of the use case, but not greater than that.

DetNet does not directly address clock distribution or time synchronization; these are considered to be part of the overall design and implementation of a time-sensitive network, using existing (or future) time-specific protocols (such as [IEEE8021AS] and/or [RFC5905]).

## 2. Pro Audio and Video

### 2.1. Use Case Description

The professional audio and video industry ("ProAV") includes:

- o Music and film content creation
- o Broadcast
- o Cinema
- o Live sound
- o Public address, media and emergency systems at large venues (airports, stadiums, churches, theme parks).

These industries have already transitioned audio and video signals from analog to digital. However, the digital interconnect systems remain primarily point-to-point with a single (or small number of) signals per link, interconnected with purpose-built hardware.

These industries are now transitioning to packet-based infrastructure to reduce cost, increase routing flexibility, and integrate with existing IT infrastructure.

Today ProAV applications have no way to establish deterministic flows from a standards-based Layer 3 (IP) interface, which is a fundamental limitation to the use cases described here. Today deterministic flows can be created within standards-based layer 2 LANs (e.g. using IEEE 802.1 AVB) however these are not routable via IP and thus are not effective for distribution over wider areas (for example broadcast events that span wide geographical areas).

It would be highly desirable if such flows could be routed over the open Internet, however solutions with more limited scope (e.g. enterprise networks) would still provide a substantial improvement.

The following sections describe specific ProAV use cases.

#### 2.1.1. Uninterrupted Stream Playback

Transmitting audio and video streams for live playback is unlike common file transfer because uninterrupted stream playback in the presence of network errors cannot be achieved by re-trying the transmission; by the time the missing or corrupt packet has been identified it is too late to execute a re-try operation. Buffering can be used to provide enough delay to allow time for one or more

retries, however this is not an effective solution in applications where large delays (latencies) are not acceptable (as discussed below).

Streams with guaranteed bandwidth can eliminate congestion on the network as a cause of transmission errors that would lead to playback interruption. Use of redundant paths can further mitigate transmission errors to provide greater stream reliability.

Additional techniques such as forward error correction can also be used to improve stream reliability.

#### 2.1.2. Synchronized Stream Playback

Latency in this context is the time between when a signal is initially sent over a stream and when it is received. A common example in ProAV is time-synchronizing audio and video when they take separate paths through the playback system. In this case the latency of both the audio and video streams must be bounded and consistent if the sound is to remain matched to the movement in the video. A common tolerance for audio/video sync is one NTSC video frame (about 33ms) and to maintain the audience perception of correct lip sync the latency needs to be consistent within some reasonable tolerance, for example 10%.

A common architecture for synchronizing multiple streams that have different paths through the network (and thus potentially different latencies) is to enable measurement of the latency of each path, and have the data sinks (for example speakers) delay (buffer) all packets on all but the slowest path. Each packet of each stream is assigned a presentation time which is based on the longest required delay. This implies that all sinks must maintain a common time reference of sufficient accuracy, which can be achieved by any of various techniques.

This type of architecture is commonly implemented using a central controller that determines path delays and arbitrates buffering delays.

#### 2.1.3. Sound Reinforcement

Consider the latency (delay) from when a person speaks into a microphone to when their voice emerges from the speaker. If this delay is longer than about 10-15 milliseconds it is noticeable and can make a sound reinforcement system unusable (see slide 6 of [SRP\_LATENCY]). (If you have ever tried to speak in the presence of a delayed echo of your voice you may know this experience).



Note that the 15ms latency bound includes all parts of the signal path, not just the network, so the network latency must be significantly less than 15ms.

In some cases local performers must perform in synchrony with a remote broadcast. In such cases the latencies of the broadcast stream and the local performer must be adjusted to match each other, with a worst case of one video frame (33ms for NTSC video).

In cases where audio phase is a consideration, for example beam-forming using multiple speakers, latency can be in the 10 microsecond range (1 audio sample at 96kHz).

#### 2.1.4. Secure Transmission

##### 2.1.4.1. Safety

Professional audio systems can include amplifiers that are capable of generating hundreds or thousands of watts of audio power which if used incorrectly can cause hearing damage to those in the vicinity. Apart from the usual care required by the systems operators to prevent such incidents, the network traffic that controls these devices must be secured (as with any sensitive application traffic).

#### 2.2. Pro Audio Today

Some proprietary systems have been created which enable deterministic streams at Layer 3 however they are "engineered networks" which require careful configuration to operate, often require that the system be over-provisioned, and it is implied that all devices on the network voluntarily play by the rules of that network. To enable these industries to successfully transition to an interoperable multi-vendor packet-based infrastructure requires effective open standards, and establishing relevant IETF standards is a crucial factor.

#### 2.3. Pro Audio Future

##### 2.3.1. Layer 3 Interconnecting Layer 2 Islands

It would be valuable to enable IP to connect multiple Layer 2 LANs.

As an example, ESPN constructed a state-of-the-art 194,000 sq ft, \$125 million broadcast studio called DC2. The DC2 network is capable of handling 46 Tbps of throughput with 60,000 simultaneous signals. Inside the facility are 1,100 miles of fiber feeding four audio control rooms (see [ESPN\_DC2] ).

In designing DC2 they replaced as much point-to-point technology as they could with packet-based technology. They constructed seven individual studios using layer 2 LANS (using IEEE 802.1 AVB) that were entirely effective at routing audio within the LANs. However to interconnect these layer 2 LAN islands together they ended up using dedicated paths in a custom SDN (Software Defined Networking) router because there is no standards-based routing solution available.

#### 2.3.2. High Reliability Stream Paths

On-air and other live media streams are often backed up with redundant links that seamlessly act to deliver the content when the primary link fails for any reason. In point-to-point systems this is provided by an additional point-to-point link; the analogous requirement in a packet-based system is to provide an alternate path through the network such that no individual link can bring down the system.

#### 2.3.3. Integration of Reserved Streams into IT Networks

A commonly cited goal of moving to a packet based media infrastructure is that costs can be reduced by using off the shelf, commodity network hardware. In addition, economy of scale can be realized by combining media infrastructure with IT infrastructure. In keeping with these goals, stream reservation technology should be compatible with existing protocols, and not compromise use of the network for best-effort (non-time-sensitive) traffic.

#### 2.3.4. Use of Unused Reservations by Best-Effort Traffic

In cases where stream bandwidth is reserved but not currently used (or is under-utilized) that bandwidth must be available to best-effort (i.e. non-time-sensitive) traffic. For example a single stream may be nailed up (reserved) for specific media content that needs to be presented at different times of the day, ensuring timely delivery of that content, yet in between those times the full bandwidth of the network can be utilized for best-effort tasks such as file transfers.

This also addresses a concern of IT network administrators that are considering adding reserved bandwidth traffic to their networks that "users will reserve large quantities of bandwidth and then never un-reserve it even though they are not using it, and soon the network will have no bandwidth left".

### 2.3.5. Traffic Segregation

Sink devices may be low cost devices with limited processing power. In order to not overwhelm the CPUs in these devices it is important to limit the amount of traffic that these devices must process.

As an example, consider the use of individual seat speakers in a cinema. These speakers are typically required to be cost reduced since the quantities in a single theater can reach hundreds of seats. Discovery protocols alone in a one thousand seat theater can generate enough broadcast traffic to overwhelm a low powered CPU. Thus an installation like this will benefit greatly from some type of traffic segregation that can define groups of seats to reduce traffic within each group. All seats in the theater must still be able to communicate with a central controller.

There are many techniques that can be used to support this feature including (but not limited to) the following examples.

#### 2.3.5.1. Packet Forwarding Rules, VLANs and Subnets

Packet forwarding rules can be used to eliminate some extraneous streaming traffic from reaching potentially low powered sink devices, however there may be other types of broadcast traffic that should be eliminated using other means for example VLANs or IP subnets.

#### 2.3.5.2. Multicast Addressing (IPv4 and IPv6)

Multicast addressing is commonly used to keep bandwidth utilization of shared links to a minimum.

Because of the MAC Address forwarding nature of Layer 2 bridges it is important that a multicast MAC address is only associated with one stream. This will prevent reservations from forwarding packets from one stream down a path that has no interested sinks simply because there is another stream on that same path that shares the same multicast MAC address.

Since each multicast MAC Address can represent 32 different IPv4 multicast addresses there must be a process put in place to make sure this does not occur. Requiring use of IPv6 address can achieve this, however due to their continued prevalence, solutions that are effective for IPv4 installations are also desirable.

#### 2.3.6. Latency Optimization by a Central Controller

A central network controller might also perform optimizations based on the individual path delays, for example sinks that are closer to the source can inform the controller that they can accept greater latency since they will be buffering packets to match presentation times of farther away sinks. The controller might then move a stream reservation on a short path to a longer path in order to free up bandwidth for other critical streams on that short path. See slides 3-5 of [SRP\_LATENCY].

Additional optimization can be achieved in cases where sinks have differing latency requirements, for example in a live outdoor concert the speaker sinks have stricter latency requirements than the recording hardware sinks. See slide 7 of [SRP\_LATENCY].

#### 2.3.7. Reduced Device Cost Due To Reduced Buffer Memory

Device cost can be reduced in a system with guaranteed reservations with a small bounded latency due to the reduced requirements for buffering (i.e. memory) on sink devices. For example, a theme park might broadcast a live event across the globe via a layer 3 protocol; in such cases the size of the buffers required is proportional to the latency bounds and jitter caused by delivery, which depends on the worst case segment of the end-to-end network path. For example on today's open internet the latency is typically unacceptable for audio and video streaming without many seconds of buffering. In such scenarios a single gateway device at the local network that receives the feed from the remote site would provide the expensive buffering required to mask the latency and jitter issues associated with long distance delivery. Sink devices in the local location would have no additional buffering requirements, and thus no additional costs, beyond those required for delivery of local content. The sink device would be receiving the identical packets as those sent by the source and would be unaware that there were any latency or jitter issues along the path.

#### 2.4. Pro Audio Asks

- o Layer 3 routing on top of AVB (and/or other high QoS networks)
- o Content delivery with bounded, lowest possible latency
- o IntServ and DiffServ integration with AVB (where practical)
- o Single network for A/V and IT traffic
- o Standards-based, interoperable, multi-vendor

- o IT department friendly
- o Enterprise-wide networks (e.g. size of San Francisco but not the whole Internet (yet...))

### 3. Electrical Utilities

#### 3.1. Use Case Description

Many systems that an electrical utility deploys today rely on high availability and deterministic behavior of the underlying networks. Presented here are use cases in Transmission, Generation and Distribution, including key timing and reliability metrics. In addition, security issues and industry trends which affect the architecture of next generation utility networks are discussed.

##### 3.1.1. Transmission Use Cases

###### 3.1.1.1. Protection

Protection means not only the protection of human operators but also the protection of the electrical equipment and the preservation of the stability and frequency of the grid. If a fault occurs in the transmission or distribution of electricity then severe damage can occur to human operators, electrical equipment and the grid itself, leading to blackouts.

Communication links in conjunction with protection relays are used to selectively isolate faults on high voltage lines, transformers, reactors and other important electrical equipment. The role of the teleprotection system is to selectively disconnect a faulty part by transferring command signals within the shortest possible time.

###### 3.1.1.1.1. Key Criteria

The key criteria for measuring teleprotection performance are command transmission time, dependability and security. These criteria are defined by the IEC standard 60834 as follows:

- o Transmission time (Speed): The time between the moment where state changes at the transmitter input and the moment of the corresponding change at the receiver output, including propagation delay. Overall operating time for a teleprotection system includes the time for initiating the command at the transmitting end, the propagation delay over the network (including equipments) and the selection and decision time at the receiving end, including any additional delay due to a noisy environment.

- o **Dependability:** The ability to issue and receive valid commands in the presence of interference and/or noise, by minimizing the probability of missing command (PMC). Dependability targets are typically set for a specific bit error rate (BER) level.
- o **Security:** The ability to prevent false tripping due to a noisy environment, by minimizing the probability of unwanted commands (PUC). Security targets are also set for a specific bit error rate (BER) level.

Additional elements of the teleprotection system that impact its performance include:

- o Network bandwidth
- o Failure recovery capacity (aka resiliency)

#### 3.1.1.1.2. Fault Detection and Clearance Timing

Most power line equipment can tolerate short circuits or faults for up to approximately five power cycles before sustaining irreversible damage or affecting other segments in the network. This translates to total fault clearance time of 100ms. As a safety precaution, however, actual operation time of protection systems is limited to 70- 80 percent of this period, including fault recognition time, command transmission time and line breaker switching time.

Some system components, such as large electromechanical switches, require particularly long time to operate and take up the majority of the total clearance time, leaving only a 10ms window for the telecommunications part of the protection scheme, independent of the distance to travel. Given the sensitivity of the issue, new networks impose requirements that are even more stringent: IEC standard 61850 limits the transfer time for protection messages to 1/4 - 1/2 cycle or 4 - 8ms (for 60Hz lines) for the most critical messages.

#### 3.1.1.1.3. Symmetric Channel Delay

Teleprotection channels which are differential must be synchronous, which means that any delays on the transmit and receive paths must match each other. Teleprotection systems ideally support zero asymmetric delay; typical legacy relays can tolerate delay discrepancies of up to 750us.

Some tools available for lowering delay variation below this threshold are:

- o For legacy systems using Time Division Multiplexing (TDM), jitter buffers at the multiplexers on each end of the line can be used to offset delay variation by queuing sent and received packets. The length of the queues must balance the need to regulate the rate of transmission with the need to limit overall delay, as larger buffers result in increased latency.
- o For jitter-prone IP packet networks, traffic management tools can ensure that the teleprotection signals receive the highest transmission priority to minimize jitter.
- o Standard packet-based synchronization technologies, such as 1588-2008 Precision Time Protocol (PTP) and Synchronous Ethernet (Sync-E), can help keep networks stable by maintaining a highly accurate clock source on the various network devices.

#### 3.1.1.1.4. Teleprotection Network Requirements (IEC 61850)

The following table captures the main network metrics as based on the IEC 61850 standard.

Teleprotection Requirement	Attribute
One way maximum delay	4-10 ms
Asymmetric delay required	Yes
Maximum jitter	less than 250 us (750 us for legacy IED)
Topology	Point to point, point to Multi-point
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1% to 1%

Table 1: Teleprotection network requirements

#### 3.1.1.1.5. Inter-Trip Protection scheme

"Inter-tripping" is the signal-controlled tripping of a circuit breaker to complete the isolation of a circuit or piece of apparatus in concert with the tripping of other circuit breakers.

Inter-Trip protection Requirement	Attribute
One way maximum delay	5 ms
Asymmetric delay required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 2: Inter-Trip protection network requirements

## 3.1.1.1.6. Current Differential Protection Scheme

Current differential protection is commonly used for line protection, and is typical for protecting parallel circuits. At both end of the lines the current is measured by the differential relays, and both relays will trip the circuit breaker if the current going into the line does not equal the current going out of the line. This type of protection scheme assumes some form of communications being present between the relays at both end of the line, to allow both relays to compare measured current values. Line differential protection schemes assume a very low telecommunications delay between both relays, often as low as 5ms. Moreover, as those systems are often not time-synchronized, they also assume symmetric telecommunications paths with constant delay, which allows comparing current measurement values taken at the exact same time.



Current Differential protection Requirement	Attribute
One way maximum delay	5 ms
Asymmetric delay Required	Yes
Maximum jitter	less than 250 us (750us for legacy IED)
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 3: Current Differential Protection metrics

## 3.1.1.1.7. Distance Protection Scheme

Distance (Impedance Relay) protection scheme is based on voltage and current measurements. The network metrics are similar (but not identical to) Current Differential protection.

Distance protection Requirement	Attribute
One way maximum delay	5 ms
Asymmetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 4: Distance Protection requirements

## 3.1.1.1.8. Inter-Substation Protection Signaling

This use case describes the exchange of Sampled Value and/or GOOSE (Generic Object Oriented Substation Events) message between Intelligent Electronic Devices (IED) in two substations for protection and tripping coordination. The two IEDs are in a master-slave mode.

The Current Transformer or Voltage Transformer (CT/VT) in one substation sends the sampled analog voltage or current value to the Merging Unit (MU) over hard wire. The MU sends the time-synchronized 61850-9-2 sampled values to the slave IED. The slave IED forwards the information to the Master IED in the other substation. The master IED makes the determination (for example based on sampled value differentials) to send a trip command to the originating IED. Once the slave IED/Relay receives the GOOSE trip for breaker tripping, it opens the breaker. It then sends a confirmation message back to the master. All data exchanges between IEDs are either through Sampled Value and/or GOOSE messages.

Inter-Substation protection Requirement	Attribute
One way maximum delay	5 ms
Asymmetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	1%

Table 5: Inter-Substation Protection requirements

## 3.1.1.2. Intra-Substation Process Bus Communications

This use case describes the data flow from the CT/VT to the IEDs in the substation via the MU. The CT/VT in the substation send the analog voltage or current values to the MU over hard wire. The MU converts the analog values into digital format (typically time-synchronized Sampled Values as specified by IEC 61850-9-2) and sends them to the IEDs in the substation. The GPS Master Clock can send

1PPS or IRIG-B format to the MU through a serial port or IEEE 1588 protocol via a network. Process bus communication using 61850 simplifies connectivity within the substation and removes the requirement for multiple serial connections and removes the slow serial bus architectures that are typically used. This also ensures increased flexibility and increased speed with the use of multicast messaging between multiple devices.

Intra-Substation protection Requirement	Attribute
One way maximum delay	5 ms
Asymmetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on Node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes - No
Packet loss	0.1%

Table 6: Intra-Substation Protection requirements

#### 3.1.1.3. Wide Area Monitoring and Control Systems

The application of synchrophasor measurement data from Phasor Measurement Units (PMU) to Wide Area Monitoring and Control Systems promises to provide important new capabilities for improving system stability. Access to PMU data enables more timely situational awareness over larger portions of the grid than what has been possible historically with normal SCADA (Supervisory Control and Data Acquisition) data. Handling the volume and real-time nature of synchrophasor data presents unique challenges for existing application architectures. Wide Area management System (WAMS) makes it possible for the condition of the bulk power system to be observed and understood in real-time so that protective, preventative, or corrective action can be taken. Because of the very high sampling rate of measurements and the strict requirement for time synchronization of the samples, WAMS has stringent telecommunications requirements in an IP network that are captured in the following table:

WAMS Requirement	Attribute
One way maximum delay	50 ms
Asymmetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point, Multi-point to Multi-point
Bandwidth	100 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on Node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	1%
Consecutive Packet Loss	At least 1 packet per application cycle must be received.

Table 7: WAMS Special Communication Requirements

#### 3.1.1.4. IEC 61850 WAN engineering guidelines requirement classification

The IEC (International Electrotechnical Commission) has published a Technical Report which offers guidelines on how to define and deploy Wide Area Networks for the interconnections of electric substations, generation plants and SCADA operation centers. The IEC 61850-90-12 is providing a classification of WAN communication requirements into 4 classes. Table 8 summarizes these requirements:

WAN Requirement	Class WA	Class WB	Class WC	Class WD
Application field	EHV (Extra High Voltage)	HV (High Voltage)	MV (Medium Voltage)	General purpose
Latency	5 ms	10 ms	100 ms	> 100 ms
Jitter	10 us	100 us	1 ms	10 ms
Latency Asymetry	100 us	1 ms	10 ms	100 ms
Time Accuracy	1 us	10 us	100 us	10 to 100 ms
Bit Error rate	10 <sup>-7</sup> to 10 <sup>-6</sup>	10 <sup>-5</sup> to 10 <sup>-4</sup>	10 <sup>-3</sup>	
Unavailability	10 <sup>-7</sup> to 10 <sup>-6</sup>	10 <sup>-5</sup> to 10 <sup>-4</sup>	10 <sup>-3</sup>	
Recovery delay	Zero	50 ms	5 s	50 s
Cyber security	extremely high	High	Medium	Medium

Table 8: 61850-90-12 Communication Requirements; Courtesy of IEC

### 3.1.2. Generation Use Case

Energy generation systems are complex infrastructures that require control of both the generated power and the generation infrastructure.

#### 3.1.2.1. Control of the Generated Power

The electrical power generation frequency must be maintained within a very narrow band. Deviations from the acceptable frequency range are detected and the required signals are sent to the power plants for frequency regulation.

Automatic Generation Control (AGC) is a system for adjusting the power output of generators at different power plants, in response to changes in the load.

FCAG (Frequency Control Automatic Generation) Requirement	Attribute
One way maximum delay	500 ms
Asymmetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point
Bandwidth	20 Kbps
Availability	99.999
precise timing required	Yes
Recovery time on Node failure	N/A
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	1%

Table 9: FCAG Communication Requirements

#### 3.1.2.2. Control of the Generation Infrastructure

The control of the generation infrastructure combines requirements from industrial automation systems and energy generation systems. This section considers the use case of the control of the generation infrastructure of a wind turbine.

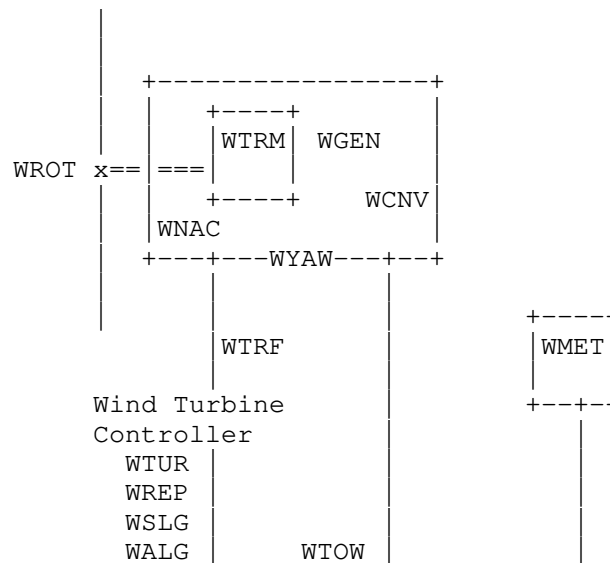


Figure 1: Wind Turbine Control Network

Figure 1 presents the subsystems that operate a wind turbine. These subsystems include

- o WROT (Rotor Control)
- o WNAC (Nacelle Control) (nacelle: housing containing the generator)
- o WTRM (Transmission Control)
- o WGEN (Generator)
- o WYAW (Yaw Controller) (of the tower head)
- o WCNV (In-Turbine Power Converter)
- o WMET (External Meteorological Station providing real time information to the controllers of the tower)

Traffic characteristics relevant for the network planning and dimensioning process in a wind turbine scenario are listed below. The values in this section are based mainly on the relevant references [Ahm14] and [Spe09]. Each logical node (Figure 1) is a part of the metering network and produces analog measurements and status information which must comply with their respective data rate constraints.

Subsystem	Sensor Count	Analog Sample Count	Data Rate (bytes/sec)	Status Sample Count	Data rate (bytes/sec)
WROT	14	9	642	5	10
WTRM	18	10	2828	8	16
WGEN	14	12	73764	2	4
WCNV	14	12	74060	2	4
WTRF	12	5	73740	2	4
WNAC	12	9	112	3	6
WYAW	7	8	220	4	8
WTOW	4	1	8	3	6
WMET	7	7	228	–	–

Table 10: Wind Turbine Data Rate Constraints

Quality of Service (QoS) constraints for different services are presented in Table 11. These constraints are defined by IEEE 1646 standard [IEEE1646] and IEC 61400 standard [IEC61400].

Service	Latency	Reliability	Packet Loss Rate
Analogue measure	16 ms	99.99%	< 10 <sup>-6</sup>
Status information	16 ms	99.99%	< 10 <sup>-6</sup>
Protection traffic	4 ms	100.00%	< 10 <sup>-9</sup>
Reporting and logging	1 s	99.99%	< 10 <sup>-6</sup>
Video surveillance	1 s	99.00%	No specific requirement
Internet connection	60 min	99.00%	No specific requirement
Control traffic	16 ms	100.00%	< 10 <sup>-9</sup>
Data polling	16 ms	99.99%	< 10 <sup>-6</sup>

Table 11: Wind Turbine Reliability and Latency Constraints

#### 3.1.2.2.1. Intra-Domain Network Considerations

A wind turbine is composed of a large set of subsystems including sensors and actuators which require time-critical operation. The reliability and latency constraints of these different subsystems is shown in Table 11. These subsystems are connected to an intra-domain network which is used to monitor and control the operation of the turbine and connect it to the SCADA subsystems. The different



components are interconnected using fiber optics, industrial buses, industrial Ethernet, EtherCat, or a combination of them. Industrial signaling and control protocols such as Modbus, Profibus, Profinet and EtherCat are used directly on top of the Layer 2 transport or encapsulated over TCP/IP.

The Data collected from the sensors and condition monitoring systems is multiplexed onto fiber cables for transmission to the base of the tower, and to remote control centers. The turbine controller continuously monitors the condition of the wind turbine and collects statistics on its operation. This controller also manages a large number of switches, hydraulic pumps, valves, and motors within the wind turbine.

There is usually a controller both at the bottom of the tower and in the nacelle. The communication between these two controllers usually takes place using fiber optics instead of copper links. Sometimes, a third controller is installed in the hub of the rotor and manages the pitch of the blades. That unit usually communicates with the nacelle unit using serial communications.

#### 3.1.2.2.2. Inter-Domain network considerations

A remote control center belonging to a grid operator regulates the power output, enables remote actuation, and monitors the health of one or more wind parks in tandem. It connects to the local control center in a wind park over the Internet (Figure 2) via firewalls at both ends. The AS path between the local control center and the Wind Park typically involves several ISPs at different tiers. For example, a remote control center in Denmark can regulate a wind park in Greece over the normal public AS path between the two locations.

The remote control center is part of the SCADA system, setting the desired power output to the wind park and reading back the result once the new power output level has been set. Traffic between the remote control center and the wind park typically consists of protocols like IEC 60870-5-104 [IEC-60870-5-104], OPC XML-DA [OPCXML], Modbus [MODBUS], and SNMP [RFC3411]. At the time of this writing, traffic flows between the wind farm and the remote control center are best effort. QoS requirements are not strict, so no SLAs or service provisioning mechanisms (e.g., VPN) are employed. In case of events like equipment failure, tolerance for alarm delay is on the order of minutes, due to redundant systems already in place.

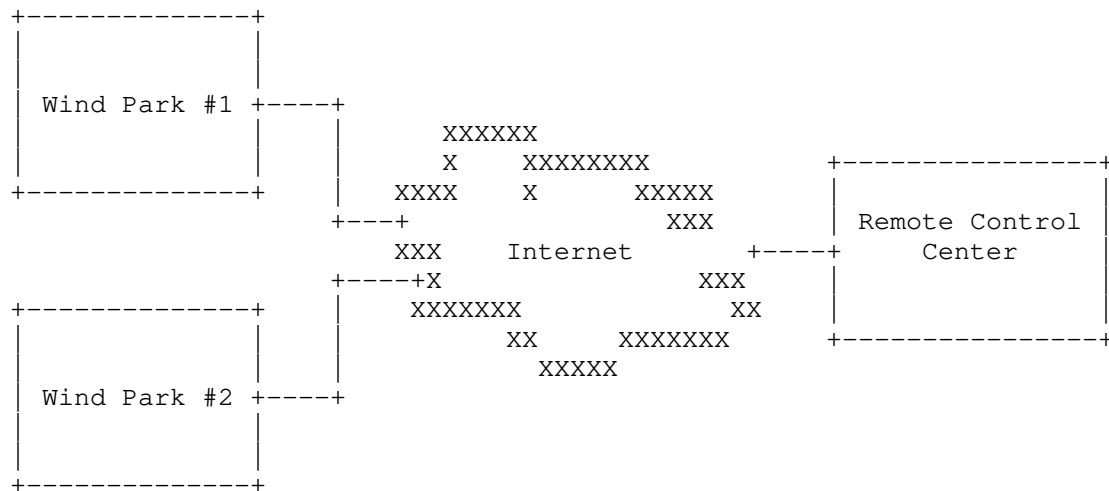


Figure 2: Wind Turbine Control via Internet

Future use cases will require bounded latency, bounded jitter and extraordinary low packet loss for inter-domain traffic flows due to the softwarization and virtualization of core wind farm equipment (e.g. switches, firewalls and SCADA server components). These factors will create opportunities for service providers to install new services and dynamically manage them from remote locations. For example, to enable fail-over of a local SCADA server, a SCADA server in another wind farm site (under the administrative control of the same operator) could be utilized temporarily (Figure 3). In that case local traffic would be forwarded to the remote SCADA server and existing intra-domain QoS and timing parameters would have to be met for inter-domain traffic flows.

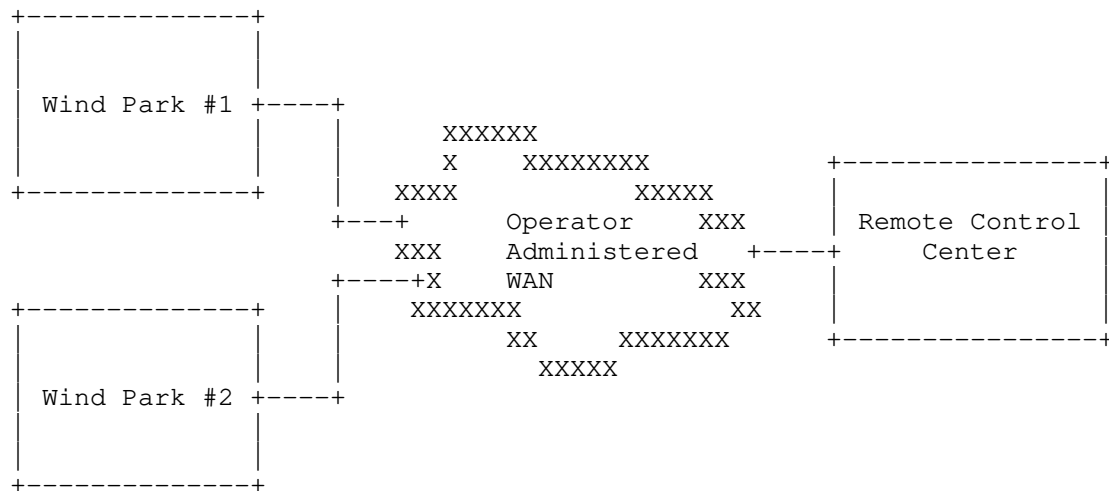


Figure 3: Wind Turbine Control via Operator Administered WAN

### 3.1.3. Distribution use case

#### 3.1.3.1. Fault Location Isolation and Service Restoration (FLISR)

Fault Location, Isolation, and Service Restoration (FLISR) refers to the ability to automatically locate the fault, isolate the fault, and restore service in the distribution network. This will likely be the first widespread application of distributed intelligence in the grid.

Static power switch status (open/closed) in the network dictates the power flow to secondary substations. Reconfiguring the network in the event of a fault is typically done manually on site to energize/de-energize alternate paths. Automating the operation of substation switchgear allows the flow of power to be altered automatically under fault conditions.

FLISR can be managed centrally from a Distribution Management System (DMS) or executed locally through distributed control via intelligent switches and fault sensors.

FLISR Requirement	Attribute
One way maximum delay	80 ms
Asymmetric delay Required	No
Maximum jitter	40 ms
Topology	Point to point, point to Multi-point, Multi-point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on Node failure	Depends on customer impact
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 12: FLISR Communication Requirements

### 3.2. Electrical Utilities Today

Many utilities still rely on complex environments formed of multiple application-specific proprietary networks, including TDM networks.

In this kind of environment there is no mixing of OT and IT applications on the same network, and information is siloed between operational areas.

Specific calibration of the full chain is required, which is costly.

This kind of environment prevents utility operations from realizing the operational efficiency benefits, visibility, and functional integration of operational information across grid applications and data networks.

In addition, there are many security-related issues as discussed in the following section.

#### 3.2.1. Security Current Practices and Limitations

Grid monitoring and control devices are already targets for cyber attacks, and legacy telecommunications protocols have many intrinsic network-related vulnerabilities. For example, DNP3, Modbus,

PROFIBUS/PROFINET, and other protocols are designed around a common paradigm of request and respond. Each protocol is designed for a master device such as an HMI (Human Machine Interface) system to send commands to subordinate slave devices to retrieve data (reading inputs) or control (writing to outputs). Because many of these protocols lack authentication, encryption, or other basic security measures, they are prone to network-based attacks, allowing a malicious actor or attacker to utilize the request-and-respond system as a mechanism for command-and-control like functionality. Specific security concerns common to most industrial control, including utility telecommunication protocols include the following:

- o Network or transport errors (e.g. malformed packets or excessive latency) can cause protocol failure.
- o Protocol commands may be available that are capable of forcing slave devices into inoperable states, including powering-off devices, forcing them into a listen-only state, disabling alarming.
- o Protocol commands may be available that are capable of restarting communications and otherwise interrupting processes.
- o Protocol commands may be available that are capable of clearing, erasing, or resetting diagnostic information such as counters and diagnostic registers.
- o Protocol commands may be available that are capable of requesting sensitive information about the controllers, their configurations, or other need-to-know information.
- o Most protocols are application layer protocols transported over TCP; therefore it is easy to transport commands over non-standard ports or inject commands into authorized traffic flows.
- o Protocol commands may be available that are capable of broadcasting messages to many devices at once (i.e. a potential DoS).
- o Protocol commands may be available to query the device network to obtain defined points and their values (i.e. a configuration scan).
- o Protocol commands may be available that will list all available function codes (i.e. a function scan).

These inherent vulnerabilities, along with increasing connectivity between IT and OT networks, make network-based attacks very feasible.

Simple injection of malicious protocol commands provides control over the target process. Altering legitimate protocol traffic can also alter information about a process and disrupt the legitimate controls that are in place over that process. A man-in-the-middle attack could provide both control over a process and misrepresentation of data back to operator consoles.

### 3.3. Electrical Utilities Future

The business and technology trends that are sweeping the utility industry will drastically transform the utility business from the way it has been for many decades. At the core of many of these changes is a drive to modernize the electrical grid with an integrated telecommunications infrastructure. However, interoperability concerns, legacy networks, disparate tools, and stringent security requirements all add complexity to the grid transformation. Given the range and diversity of the requirements that should be addressed by the next generation telecommunications infrastructure, utilities need to adopt a holistic architectural approach to integrate the electrical grid with digital telecommunications across the entire power delivery chain.

The key to modernizing grid telecommunications is to provide a common, adaptable, multi-service network infrastructure for the entire utility organization. Such a network serves as the platform for current capabilities while enabling future expansion of the network to accommodate new applications and services.

To meet this diverse set of requirements, both today and in the future, the next generation utility telecommunications network will be based on open-standards-based IP architecture. An end-to-end IP architecture takes advantage of nearly three decades of IP technology development, facilitating interoperability and device management across disparate networks and devices, as it has been already demonstrated in many mission-critical and highly secure networks.

IPv6 is seen as a future telecommunications technology for the Smart Grid; the IEC (International Electrotechnical Commission) and different National Committees have mandated a specific adhoc group (AHG8) to define the migration strategy to IPv6 for all the IEC TC57 power automation standards. The AHG8 has finalised the work on the migration strategy and the following Technical Report has been issued: IEC TR 62357-200:2015: Guidelines for migration from Internet Protocol version 4 (IPv4) to Internet Protocol version 6 (IPv6).

Cloud-based SCADA systems will control and monitor the critical and non-critical subsystems of generation systems, for example wind farms.

### 3.3.1. Migration to Packet-Switched Network

Throughout the world, utilities are increasingly planning for a future based on smart grid applications requiring advanced telecommunications systems. Many of these applications utilize packet connectivity for communicating information and control signals across the utility's Wide Area Network (WAN), made possible by technologies such as multiprotocol label switching (MPLS). The data that traverses the utility WAN includes:

- o Grid monitoring, control, and protection data
- o Non-control grid data (e.g. asset data for condition-based monitoring)
- o Physical safety and security data (e.g. voice and video)
- o Remote worker access to corporate applications (voice, maps, schematics, etc.)
- o Field area network backhaul for smart metering, and distribution grid management
- o Enterprise traffic (email, collaboration tools, business applications)

WANs support this wide variety of traffic to and from substations, the transmission and distribution grid, generation sites, between control centers, and between work locations and data centers. To maintain this rapidly expanding set of applications, many utilities are taking steps to evolve present time-division multiplexing (TDM) based and frame relay infrastructures to packet systems. Packet-based networks are designed to provide greater functionalities and higher levels of service for applications, while continuing to deliver reliability and deterministic (real-time) traffic support.

### 3.3.2. Telecommunications Trends

These general telecommunications topics are in addition to the use cases that have been addressed so far. These include both current and future telecommunications related topics that should be factored into the network architecture and design.

#### 3.3.2.1. General Telecommunications Requirements

- o IP Connectivity everywhere
- o Monitoring services everywhere and from different remote centers

- o Move services to a virtual data center
- o Unify access to applications / information from the corporate network
- o Unify services
- o Unified Communications Solutions
- o Mix of fiber and microwave technologies - obsolescence of SONET/SDH or TDM
- o Standardize grid telecommunications protocol to opened standard to ensure interoperability
- o Reliable Telecommunications for Transmission and Distribution Substations
- o IEEE 1588 time synchronization Client / Server Capabilities
- o Integration of Multicast Design
- o QoS Requirements Mapping
- o Enable Future Network Expansion
- o Substation Network Resilience
- o Fast Convergence Design
- o Scalable Headend Design
- o Define Service Level Agreements (SLA) and Enable SLA Monitoring
- o Integration of 3G/4G Technologies and future technologies
- o Ethernet Connectivity for Station Bus Architecture
- o Ethernet Connectivity for Process Bus Architecture
- o Protection, teleprotection and PMU (Phaser Measurement Unit) on IP

#### 3.3.2.2. Specific Network topologies of Smart Grid Applications

Utilities often have very large private telecommunications networks. It covers an entire territory / country. The main purpose of the network, until now, has been to support transmission network monitoring, control, and automation, remote control of generation



sites, and providing FCAPS (Fault, Configuration, Accounting, Performance, Security) services from centralized network operation centers.

Going forward, one network will support operation and maintenance of electrical networks (generation, transmission, and distribution), voice and data services for ten of thousands of employees and for exchange with neighboring interconnections, and administrative services. To meet those requirements, utility may deploy several physical networks leveraging different technologies across the country: an optical network and a microwave network for instance. Each protection and automatism system between two points has two telecommunications circuits, one on each network. Path diversity between two substations is key. Regardless of the event type (hurricane, ice storm, etc.), one path needs to stay available so the system can still operate.

In the optical network, signals are transmitted over more than tens of thousands of circuits using fiber optic links, microwave and telephone cables. This network is the nervous system of the utility's power transmission operations. The optical network represents ten of thousands of km of cable deployed along the power lines, with individual runs as long as 280 km.

#### 3.3.2.3. Precision Time Protocol

Some utilities do not use GPS clocks in generation substations. One of the main reasons is that some of the generation plants are 30 to 50 meters deep under ground and the GPS signal can be weak and unreliable. Instead, atomic clocks are used. Clocks are synchronized amongst each other. Rubidium clocks provide clock and 1ms timestamps for IRIG-B.

Some companies plan to transition to the Precision Time Protocol (PTP, [IEEE1588]), distributing the synchronization signal over the IP/MPLS network. PTP provides a mechanism for synchronizing the clocks of participating nodes to a high degree of accuracy and precision.

PTP operates based on the following assumptions:

It is assumed that the network eliminates cyclic forwarding of PTP messages within each communication path (e.g. by using a spanning tree protocol).

PTP is tolerant of an occasional missed message, duplicated message, or message that arrived out of order. However, PTP assumes that such impairments are relatively rare.

PTP was designed assuming a multicast communication model, however PTP also supports a unicast communication model as long as the behavior of the protocol is preserved.

Like all message-based time transfer protocols, PTP time accuracy is degraded by delay asymmetry in the paths taken by event messages. Asymmetry is not detectable by PTP, however, if such delays are known a priori, PTP can correct for asymmetry.

IEC 61850 defines the use of IEC/IEEE 61850-9-3:2016. The title is: Precision time protocol profile for power utility automation. It is based on Annex B/IEC 62439 which offers the support of redundant attachment of clocks to Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR) networks.

### 3.3.3. Security Trends in Utility Networks

Although advanced telecommunications networks can assist in transforming the energy industry by playing a critical role in maintaining high levels of reliability, performance, and manageability, they also introduce the need for an integrated security infrastructure. Many of the technologies being deployed to support smart grid projects such as smart meters and sensors can increase the vulnerability of the grid to attack. Top security concerns for utilities migrating to an intelligent smart grid telecommunications platform center on the following trends:

- o Integration of distributed energy resources
- o Proliferation of digital devices to enable management, automation, protection, and control
- o Regulatory mandates to comply with standards for critical infrastructure protection
- o Migration to new systems for outage management, distribution automation, condition-based maintenance, load forecasting, and smart metering
- o Demand for new levels of customer service and energy management

This development of a diverse set of networks to support the integration of microgrids, open-access energy competition, and the use of network-controlled devices is driving the need for a converged security infrastructure for all participants in the smart grid, including utilities, energy service providers, large commercial and industrial, as well as residential customers. Securing the assets of electric power delivery systems (from the control center to the

substation, to the feeders and down to customer meters) requires an end-to-end security infrastructure that protects the myriad of telecommunications assets used to operate, monitor, and control power flow and measurement.

"Cyber security" refers to all the security issues in automation and telecommunications that affect any functions related to the operation of the electric power systems. Specifically, it involves the concepts of:

- o Integrity : data cannot be altered undetectably
- o Authenticity (data origin authentication): the telecommunications parties involved must be validated as genuine
- o Authorization : only requests and commands from the authorized users can be accepted by the system
- o Confidentiality : data must not be accessible to any unauthenticated users

When designing and deploying new smart grid devices and telecommunications systems, it is imperative to understand the various impacts of these new components under a variety of attack situations on the power grid. Consequences of a cyber attack on the grid telecommunications network can be catastrophic. This is why security for smart grid is not just an ad hoc feature or product, it's a complete framework integrating both physical and Cyber security requirements and covering the entire smart grid networks from generation to distribution. Security has therefore become one of the main foundations of the utility telecom network architecture and must be considered at every layer with a defense-in-depth approach. Migrating to IP based protocols is key to address these challenges for two reasons:

- o IP enables a rich set of features and capabilities to enhance the security posture
- o IP is based on open standards, which allows interoperability between different vendors and products, driving down the costs associated with implementing security solutions in OT networks.

Securing OT (Operation technology) telecommunications over packet-switched IP networks follow the same principles that are foundational for securing the IT infrastructure, i.e., consideration must be given to enforcing electronic access control for both person-to-machine and machine-to-machine communications, and providing the appropriate

levels of data privacy, device and platform integrity, and threat detection and mitigation.

### 3.4. Electrical Utilities Asks

- o Mixed L2 and L3 topologies
- o Deterministic behavior
- o Bounded latency and jitter
- o Tight feedback intervals
- o High availability, low recovery time
- o Redundancy, low packet loss
- o Precise timing
- o Centralized computing of deterministic paths
- o Distributed configuration may also be useful

## 4. Building Automation Systems

### 4.1. Use Case Description

A Building Automation System (BAS) manages equipment and sensors in a building for improving residents' comfort, reducing energy consumption, and responding to failures and emergencies. For example, the BAS measures the temperature of a room using sensors and then controls the HVAC (heating, ventilating, and air conditioning) to maintain a set temperature and minimize energy consumption.

A BAS primarily performs the following functions:

- o Periodically measures states of devices, for example humidity and illuminance of rooms, open/close state of doors, FAN speed, etc.
- o Stores the measured data.
- o Provides the measured data to BAS systems and operators.
- o Generates alarms for abnormal state of devices.
- o Controls devices (e.g. turn off room lights at 10:00 PM).

## 4.2. Building Automation Systems Today

### 4.2.1. BAS Architecture

A typical BAS architecture of today is shown in Figure 4.

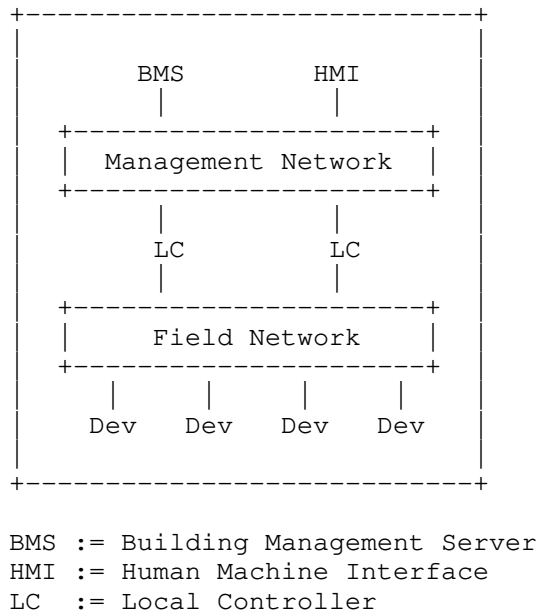


Figure 4: BAS architecture

There are typically two layers of network in a BAS. The upper one is called the Management Network and the lower one is called the Field Network. In management networks an IP-based communication protocol is used, while in field networks non-IP based communication protocols ("field protocols") are mainly used. Field networks have specific timing requirements, whereas management networks can be best-effort.

A Human Machine Interface (HMI) is typically a desktop PC used by operators to monitor and display device states, send device control commands to Local Controllers (LCs), and configure building schedules (for example "turn off all room lights in the building at 10:00 PM").

A Building Management Server (BMS) performs the following operations.

- o Collect and store device states from LCs at regular intervals.
- o Send control values to LCs according to a building schedule.

- o Send an alarm signal to operators if it detects abnormal devices states.

The BMS and HMI communicate with LCs via IP-based "management protocols" (see standards [bacnetip], [knx]).

A LC is typically a Programmable Logic Controller (PLC) which is connected to several tens or hundreds of devices using "field protocols". An LC performs the following kinds of operations:

- o Measure device states and provide the information to BMS or HMI.
- o Send control values to devices, unilaterally or as part of a feedback control loop.

There are many field protocols used at the time of this writing; some are standards-based and others are proprietary (see standards [lontalk], [modbus], [profibus] and [flnet]). The result is that BASs have multiple MAC/PHY modules and interfaces. This makes BASs more expensive, slower to develop, and can result in "vendor lock-in" with multiple types of management applications.

#### 4.2.2. BAS Deployment Model

An example BAS for medium or large buildings is shown in Figure 5. The physical layout spans multiple floors, and there is a monitoring room where the BAS management entities are located. Each floor will have one or more LCs depending upon the number of devices connected to the field network.

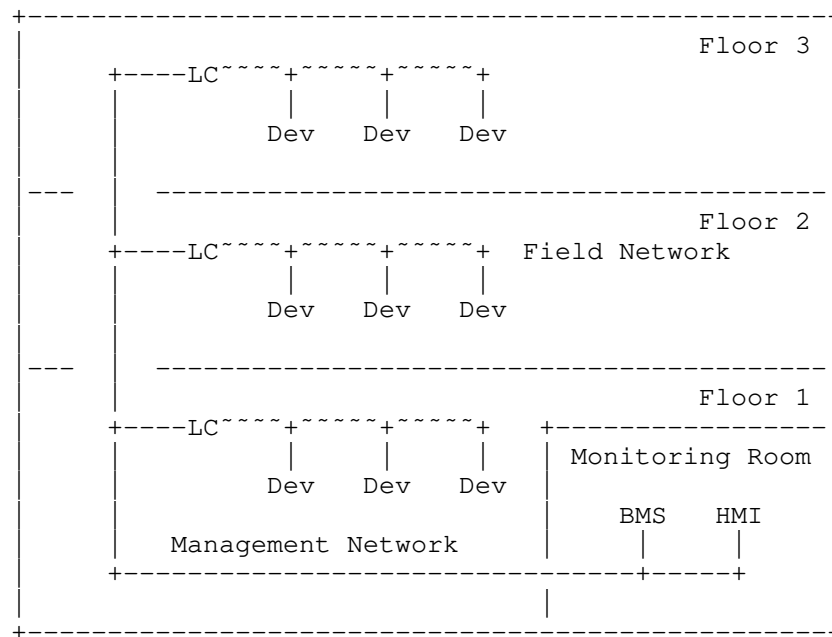


Figure 5: BAS Deployment model for Medium/Large Buildings

Each LC is connected to the monitoring room via the Management network, and the management functions are performed within the building. In most cases, fast Ethernet (e.g. 100BASE-T) is used for the management network. Since the management network is non-realtime, use of Ethernet without quality of service is sufficient for today's deployment.

In the field network a variety of physical interfaces such as RS232C and RS485 are used, which have specific timing requirements. Thus if a field network is to be replaced with an Ethernet or wireless network, such networks must support time-critical deterministic flows.

In Figure 6, another deployment model is presented in which the management system is hosted remotely. This is becoming popular for small office and residential buildings in which a standalone monitoring system is not cost-effective.

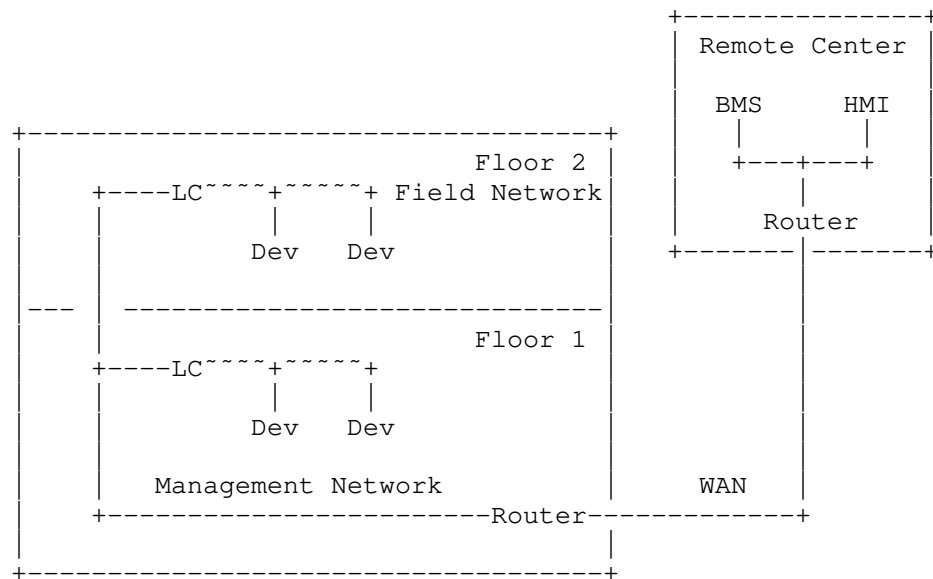


Figure 6: Deployment model for Small Buildings

Some interoperability is possible today in the Management Network, but not in today's field networks due to their non-IP-based design.

#### 4.2.3. Use Cases for Field Networks

Below are use cases for Environmental Monitoring, Fire Detection, and Feedback Control, and their implications for field network performance.

##### 4.2.3.1. Environmental Monitoring

The BMS polls each LC at a maximum measurement interval of 100ms (for example to draw a historical chart of 1 second granularity with a 10x sampling interval) and then performs the operations as specified by the operator. Each LC needs to measure each of its several hundred sensors once per measurement interval. Latency is not critical in this scenario as long as all sensor values are completed in the measurement interval. Availability is expected to be 99.999 %.

##### 4.2.3.2. Fire Detection

On detection of a fire, the BMS must stop the HVAC, close the fire shutters, turn on the fire sprinklers, send an alarm, etc. There are typically ~10s of sensors per LC that BMS needs to manage. In this



scenario the measurement interval is 10-50ms, the communication delay is 10ms, and the availability must be 99.9999 %.

#### 4.2.3.3. Feedback Control

BAS systems utilize feedback control in various ways; the most time-critical is control of DC motors, which require a short feedback interval (1-5ms) with low communication delay (10ms) and jitter (1ms). The feedback interval depends on the characteristics of the device and a target quality of control value. There are typically ~10s of such devices per LC.

Communication delay is expected to be less than 10ms, jitter less than 1ms while the availability must be 99.9999% .

#### 4.2.4. Security Considerations

When BAS field networks were developed it was assumed that the field networks would always be physically isolated from external networks and therefore security was not a concern. In today's world many BASs are managed remotely and are thus connected to shared IP networks and so security is definitely a concern, yet security features are not available in the majority of BAS field network deployments .

The management network, being an IP-based network, has the protocols available to enable network security, but in practice many BAS systems do not implement even the available security features such as device authentication or encryption for data in transit.

#### 4.3. BAS Future

In the future more fine-grained environmental monitoring and lower energy consumption will emerge which will require more sensors and devices, thus requiring larger and more complex building networks.

Building networks will be connected to or converged with other networks (Enterprise network, Home network, and Internet).

Therefore better facilities for network management, control, reliability and security are critical in order to improve resident and operator convenience and comfort. For example the ability to monitor and control building devices via the internet would enable (for example) control of room lights or HVAC from a resident's desktop PC or phone application.

#### 4.4. BAS Asks

The community would like to see an interoperable protocol specification that can satisfy the timing, security, availability and QoS constraints described above, such that the resulting converged network can replace the disparate field networks. Ideally this connectivity could extend to the open Internet.

This would imply an architecture that can guarantee

- o Low communication delays (from <10ms to 100ms in a network of several hundred devices)
- o Low jitter (< 1 ms)
- o Tight feedback intervals (1ms - 10ms)
- o High network availability (up to 99.9999% )
- o Availability of network data in disaster scenario
- o Authentication between management and field devices (both local and remote)
- o Integrity and data origin authentication of communication data between field and management devices
- o Confidentiality of data when communicated to a remote device

### 5. Wireless for Industrial Applications

#### 5.1. Use Case Description

Wireless networks are useful for industrial applications, for example when portable, fast-moving or rotating objects are involved, and for the resource-constrained devices found in the Internet of Things (IoT).

Such network-connected sensors, actuators, control loops (etc.) typically require that the underlying network support real-time quality of service (QoS), as well as specific classes of other network properties such as reliability, redundancy, and security.

These networks may also contain very large numbers of devices, for example for factories, "big data" acquisition, and the IoT. Given the large numbers of devices installed, and the potential pervasiveness of the IoT, this is a huge and very cost-sensitive

market such that small cost reductions can save large amounts of money.

#### 5.1.1. Network Convergence using 6TiSCH

Some wireless network technologies support real-time QoS, and are thus useful for these kinds of networks, but others do not.

This use case focuses on one specific wireless network technology which provides the required deterministic QoS, which is "IPv6 over the TSCH mode of IEEE 802.15.4e" (6TiSCH, where TSCH stands for "Time-Slotted Channel Hopping", see [I-D.ietf-6tisch-architecture], [IEEE802154], [IEEE802154e], and [RFC7554]).

There are other deterministic wireless busses and networks available today, however they are incompatible with each other, and incompatible with IP traffic (for example [ISA100], [WirelessHART]).

Thus the primary goal of this use case is to apply 6TiSCH as a converged IP- and standards-based wireless network for industrial applications, i.e. to replace multiple proprietary and/or incompatible wireless networking and wireless network management standards.

#### 5.1.2. Common Protocol Development for 6TiSCH

Today there are a number of protocols required by 6TiSCH which are still in development, and a second intent of this use case is to highlight the ways in which these "missing" protocols share goals in common with DetNet. Thus it is possible that some of the protocol technology developed for DetNet will also be applicable to 6TiSCH.

These protocol goals are identified here, along with their relationship to DetNet. It is likely that ultimately the resulting protocols will not be identical, but will share design principles which contribute to the efficiency of enabling both DetNet and 6TiSCH.

One such commonality is that although at a different time scale, in both TSN [IEEE802.1TSNTG] and TSCH a packet crosses the network from node to node follows a precise schedule, as a train that leaves intermediate stations at precise times along its path. This kind of operation reduces collisions, saves energy, and enables engineering the network for deterministic properties.

Another commonality is remote monitoring and scheduling management of a TSCH network by a Path Computation Element (PCE) and Network Management Entity (NME). The PCE/NME manage timeslots and device resources in a manner that minimizes the interaction with and the

load placed on resource-constrained devices. For example, a tiny IoT device may have just enough buffers to store one or a few IPv6 packets, and will have limited bandwidth between peers such that it can maintain only a small amount of peer information, and will not be able to store many packets waiting to be forwarded. It is advantageous then for it to only be required to carry out the specific behavior assigned to it by the PCE/NME (as opposed to maintaining its own IP stack, for example).

It is possible that there will be some peer-to-peer communication, for example the PCE may communicate only indirectly with some devices in order to enable hierarchical configuration of the system.

6TiSCH depends on [PCE] and [I-D.ietf-detnet-architecture].

6TiSCH also depends on the fact that DetNet will maintain consistency with [IEEE802.1TSNTG].

## 5.2. Wireless Industrial Today

Today industrial wireless is accomplished using multiple deterministic wireless networks which are incompatible with each other and with IP traffic.

6TiSCH is not yet fully specified, so it cannot be used in today's applications.

## 5.3. Wireless Industrial Future

### 5.3.1. Unified Wireless Network and Management

DetNet and 6TiSCH together can enable converged transport of deterministic and best-effort traffic flows between real-time industrial devices and wide area networks via IP routing. A high level view of a basic such network is shown in Figure 7.

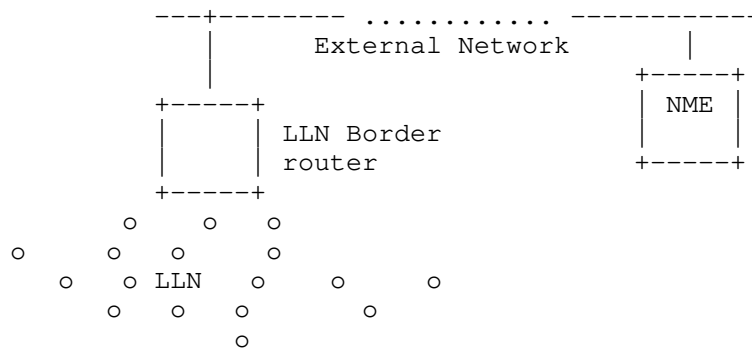


Figure 7: Basic 6TiSCH Network

Figure 8 shows a backbone router federating multiple synchronized 6TiSCH subnets into a single subnet connected to the external network.

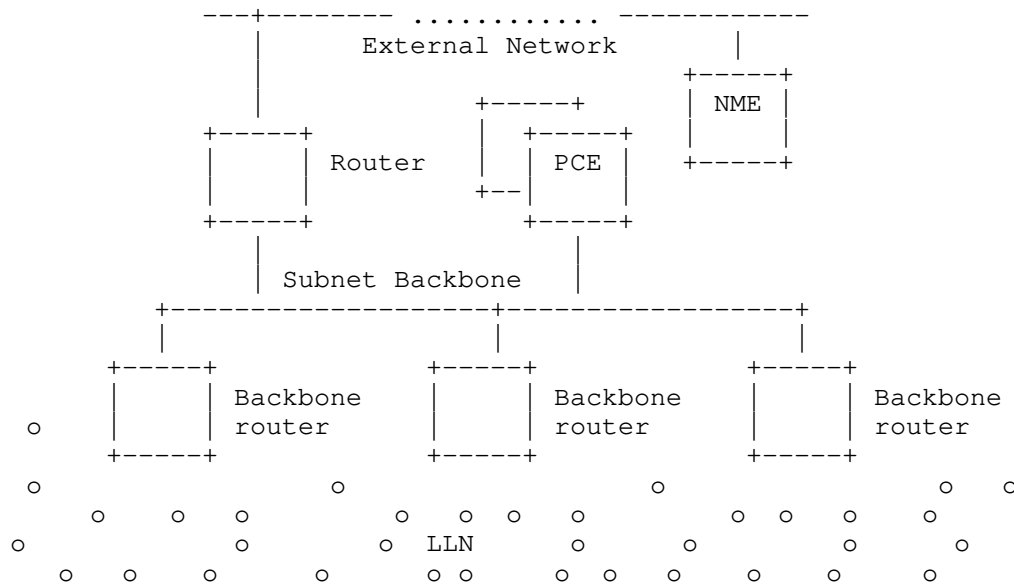


Figure 8: Extended 6TiSCH Network

The backbone router must ensure end-to-end deterministic behavior between the LLN and the backbone. This should be accomplished in conformance with the work done in [I-D.ietf-detnet-architecture] with respect to Layer-3 aspects of deterministic networks that span multiple Layer-2 domains.

The PCE must compute a deterministic path end-to-end across the TSCH network and IEEE802.1 TSN Ethernet backbone, and DetNet protocols are expected to enable end-to-end deterministic forwarding.

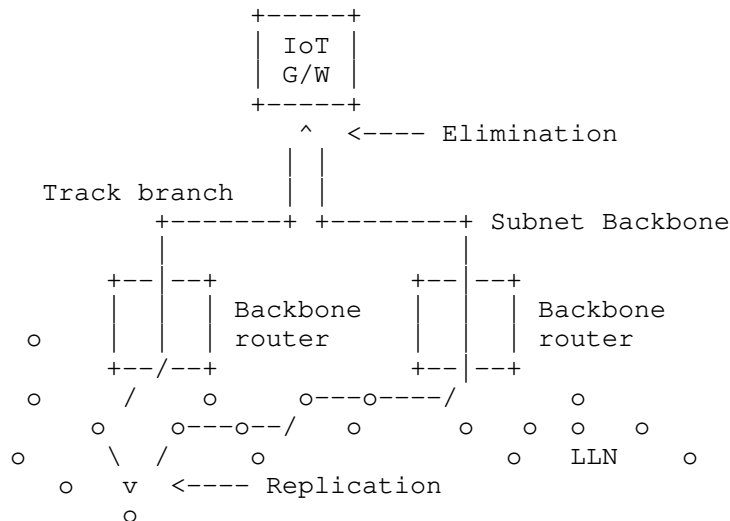


Figure 9: 6TiSCH Network with PRE

#### 5.3.1.1. PCE and 6TiSCH ARQ Retries

6TiSCH uses the IEEE802.15.4 Automatic Repeat-reQuest (ARQ) mechanism to provide higher reliability of packet delivery. ARQ is related to packet replication and elimination because there are two independent paths for packets to arrive at the destination, and if an expected packet does not arrive on one path then it checks for the packet on the second path.

Although to date this mechanism is only used by wireless networks, this may be a technique that would be appropriate for DetNet and so aspects of the enabling protocol could be co-developed.

For example, in Figure 9, a Track is laid out from a field device in a 6TiSCH network to an IoT gateway that is located on a IEEE802.1 TSN backbone.

In ARQ the Replication function in the field device sends a copy of each packet over two different branches, and the PCE schedules each hop of both branches so that the two copies arrive in due time at the gateway. In case of a loss on one branch, hopefully the other copy

of the packet still arrives within the allocated time. If two copies make it to the IoT gateway, the Elimination function in the gateway ignores the extra packet and presents only one copy to upper layers.

At each 6TiSCH hop along the Track, the PCE may schedule more than one timeSlot for a packet, so as to support Layer-2 retries (ARQ).

In deployments at the time of this writing, a TSCH Track does not necessarily support PRE but is systematically multi-path. This means that a Track is scheduled so as to ensure that each hop has at least two forwarding solutions, and the forwarding decision is to try the preferred one and use the other in case of Layer-2 transmission failure as detected by ARQ.

#### 5.3.2. Schedule Management by a PCE

A common feature of 6TiSCH and DetNet is the action of a PCE to configure paths through the network. Specifically, what is needed is a protocol and data model that the PCE will use to get/set the relevant configuration from/to the devices, as well as perform operations on the devices. This protocol should be developed by DetNet with consideration for its reuse by 6TiSCH. The remainder of this section provides a bit more context from the 6TiSCH side.

##### 5.3.2.1. PCE Commands and 6TiSCH CoAP Requests

The 6TiSCH device does not expect to place the request for bandwidth between itself and another device in the network. Rather, an operation control system invoked through a human interface specifies the required traffic specification and the end nodes (in terms of latency and reliability). Based on this information, the PCE must compute a path between the end nodes and provision the network with per-flow state that describes the per-hop operation for a given packet, the corresponding timeslots, and the flow identification that enables recognizing that a certain packet belongs to a certain path, etc.

For a static configuration that serves a certain purpose for a long period of time, it is expected that a node will be provisioned in one shot with a full schedule, which incorporates the aggregation of its behavior for multiple paths. 6TiSCH expects that the programming of the schedule will be done over COAP as discussed in [I-D.ietf-6tisch-coap].

6TiSCH expects that the PCE commands will be mapped back and forth into CoAP by a gateway function at the edge of the 6TiSCH network. For instance, it is possible that a mapping entity on the backbone transforms a non-CoAP protocol such as PCEP into the RESTful

interfaces that the 6TiSCH devices support. This architecture will be refined to comply with DetNet [I-D.ietf-detnet-architecture] when the work is formalized. Related information about 6TiSCH can be found at [I-D.ietf-6tisch-6top-interface] and RPL [RFC6550].

A protocol may be used to update the state in the devices during runtime, for example if it appears that a path through the network has ceased to perform as expected, but in 6TiSCH that flow was not designed and no protocol was selected. DetNet should define the appropriate end-to-end protocols to be used in that case. The implication is that these state updates take place once the system is configured and running, i.e. they are not limited to the initial communication of the configuration of the system.

A "slotFrame" is the base object that a PCE would manipulate to program a schedule into an LLN node ([I-D.ietf-6tisch-architecture]).

The PCE should read energy data from devices and compute paths that will implement policies on how energy in devices is consumed, for instance to ensure that the spent energy does not exceed the available energy over a period of time. Note: this statement implies that an extensible protocol for communicating device info to the PCE and enabling the PCE to act on it will be part of the DetNet architecture, however for subnets with specific protocols (e.g. CoAP) a gateway may be required.

6TiSCH devices can discover their neighbors over the radio using a mechanism such as beacons, but even though the neighbor information is available in the 6TiSCH interface data model, 6TiSCH does not describe a protocol to proactively push the neighborhood information to a PCE. DetNet should define such a protocol; one possible design alternative is that it could operate over CoAP, alternatively it could be converted to/from CoAP by a gateway. Such a protocol could carry multiple metrics, for example similar to those used for RPL operations [RFC6551]

#### 5.3.2.2. 6TiSCH IP Interface

"6top" ([I-D.wang-6tisch-6top-sublayer]) is a logical link control sitting between the IP layer and the TSCH MAC layer which provides the link abstraction that is required for IP operations. The 6top data model and management interfaces are further discussed in [I-D.ietf-6tisch-6top-interface] and [I-D.ietf-6tisch-coap].

An IP packet that is sent along a 6TiSCH path uses the Differentiated Services Per-Hop-Behavior Group called Deterministic Forwarding, as described in [I-D.svshah-tsvwg-deterministic-forwarding].



### 5.3.3. 6TiSCH Security Considerations

On top of the classical requirements for protection of control signaling, it must be noted that 6TiSCH networks operate on limited resources that can be depleted rapidly in a DoS attack on the system, for instance by placing a rogue device in the network, or by obtaining management control and setting up unexpected additional paths.

### 5.4. Wireless Industrial Asks

6TiSCH depends on DetNet to define:

- o Configuration (state) and operations for deterministic paths
- o End-to-end protocols for deterministic forwarding (tagging, IP)
- o Protocol for packet replication and elimination

## 6. Cellular Radio

### 6.1. Use Case Description

This use case describes the application of deterministic networking in the context of cellular telecom transport networks. Important elements include time synchronization, clock distribution, and ways of establishing time-sensitive streams for both Layer-2 and Layer-3 user plane traffic.

#### 6.1.1. Network Architecture

Figure 10 illustrates a 3GPP-defined cellular network architecture typical at the time of this writing, which includes "Fronthaul", "Midhaul" and "Backhaul" network segments. The "Fronthaul" is the network connecting base stations (baseband processing units) to the remote radio heads (antennas). The "Midhaul" is the network inter-connecting base stations (or small cell sites). The "Backhaul" is the network or links connecting the radio base station sites to the network controller/gateway sites (i.e. the core of the 3GPP cellular network).

In Figure 10 "eNB" ("E-UTRAN Node B") is the hardware that is connected to the mobile phone network which communicates directly with mobile handsets ([TS36300]).

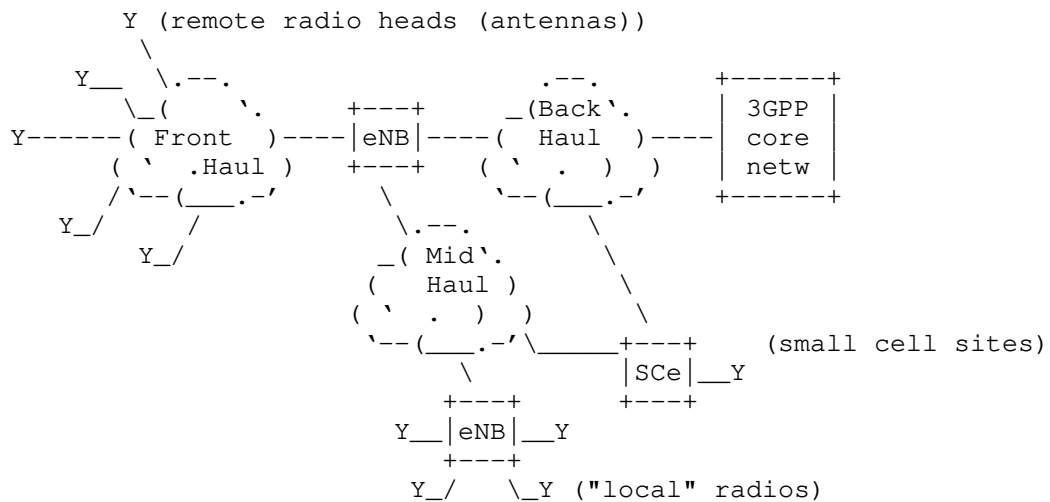


Figure 10: Generic 3GPP-based Cellular Network Architecture

#### 6.1.2. Delay Constraints

The available processing time for Fronthaul networking overhead is limited to the available time after the baseband processing of the radio frame has completed. For example in Long Term Evolution (LTE) radio, processing of a radio frame is allocated 3ms but typically the processing uses most of it, allowing only a small fraction to be used by the Fronthaul network (e.g. up to 250us one-way delay, though the existing spec ([NGMN-fronth]) supports delay only up to 100us). This ultimately determines the distance the remote radio heads can be located from the base stations (e.g., 100us equals roughly 20 km of optical fiber-based transport). Allocation options of the available time budget between processing and transport are under heavy discussions in the mobile industry.

For packet-based transport the allocated transport time (e.g. CPRI would allow for 100us delay [CPRI]) is consumed by all nodes and buffering between the remote radio head and the baseband processing unit, plus the distance-incurred delay.

The baseband processing time and the available "delay budget" for the fronthaul is likely to change in the forthcoming "5G" due to reduced radio round trip times and other architectural and service requirements [NGMN].

The transport time budget, as noted above, places limitations on the distance that remote radio heads can be located from base stations (i.e. the link length). In the above analysis, the entire transport

time budget is assumed to be available for link propagation delay. However the transport time budget can be broken down into three components: scheduling /queueing delay, transmission delay, and link propagation delay. Using today's Fronthaul networking technology, the queuing, scheduling and transmission components might become the dominant factors in the total transport time rather than the link propagation delay. This is especially true in cases where the Fronthaul link is relatively short and it is shared among multiple Fronthaul flows, for example in indoor and small cell networks, massive MIMO antenna networks, and split Fronthaul architectures.

DetNet technology can improve this application by controlling and reducing the time required for the queuing, scheduling and transmission operations by properly assigning the network resources, thus leaving more of the transport time budget available for link propagation, and thus enabling longer link lengths. However, link length is usually a given parameter and is not a controllable network parameter, since RRH and BBU sites are usually located in predetermined locations. However, the number of antennas in an RRH site might increase for example by adding more antennas, increasing the MIMO capability of the network or support of massive MIMO. This means increasing the number of the fronthaul flows sharing the same fronthaul link. DetNet can now control the bandwidth assignment of the fronthaul link and the scheduling of fronthaul packets over this link and provide adequate buffer provisioning for each flow to reduce the packet loss rate.

Another way in which DetNet technology can aid Fronthaul networks is by providing effective isolation from best-effort (and other classes of) traffic, which can arise as a result of network slicing in 5G networks where Fronthaul traffic generated in different network slices might have differing performance requirements. DetNet technology can also dynamically control the bandwidth assignment, scheduling and packet forwarding decisions and the buffer provisioning of the Fronthaul flows to guarantee the end-to-end delay of the Fronthaul packets and minimize the packet loss rate.

[METIS] documents the fundamental challenges as well as overall technical goals of the future 5G mobile and wireless system as the starting point. These future systems should support much higher data volumes and rates and significantly lower end-to-end latency for 100x more connected devices (at similar cost and energy consumption levels as today's system).

For Midhaul connections, delay constraints are driven by Inter-Site radio functions like Coordinated Multipoint Processing (CoMP, see [CoMP]). CoMP reception and transmission is a framework in which multiple geographically distributed antenna nodes cooperate to

improve the performance of the users served in the common cooperation area. The design principal of CoMP is to extend single-cell to multi-UE (User Equipment) transmission to a multi-cell-to-multi-UEs transmission by base station cooperation.

CoMP has delay-sensitive performance parameters, which are "midhaul latency" and "CSI (Channel State Information) reporting and accuracy". The essential feature of CoMP is signaling between eNBs, so Midhaul latency is the dominating limitation of CoMP performance. Generally, CoMP can benefit from coordinated scheduling (either distributed or centralized) of different cells if the signaling delay between eNBs is within 1-10ms. This delay requirement is both rigid and absolute because any uncertainty in delay will degrade the performance significantly.

Inter-site CoMP is one of the key requirements for 5G and is also a goal for 4.5G network architecture.

### 6.1.3. Time Synchronization Constraints

Fronthaul time synchronization requirements are given by [TS25104], [TS36104], [TS36211], and [TS36133]. These can be summarized for the 3GPP LTE-based networks as:

#### Delay Accuracy:

$\pm 8\text{ns}$  (i.e.  $\pm 1/32 T_c$ , where  $T_c$  is the UMTS Chip time of  $1/3.84\text{MHz}$ ) resulting in a round trip accuracy of  $\pm 16\text{ns}$ . The value is this low to meet the 3GPP Timing Alignment Error (TAE) measurement requirements. Note: performance guarantees of low nanosecond values such as these are considered to be below the DetNet layer - it is assumed that the underlying implementation, e.g. the hardware, will provide sufficient support (e.g. buffering) to enable this level of accuracy. These values are maintained in the use case to give an indication of the overall application.

#### Timing Alignment Error:

Timing Alignment Error (TAE) is problematic to Fronthaul networks and must be minimized. If the transport network cannot guarantee low enough TAE then additional buffering has to be introduced at the edges of the network to buffer out the jitter. Buffering is not desirable as it reduces the total available delay budget. Packet Delay Variation (PDV) requirements can be derived from TAE for packet based Fronthaul networks.

- \* For multiple input multiple output (MIMO) or TX diversity transmissions, at each carrier frequency, TAE shall not exceed 65 ns (i.e.  $1/4 T_c$ ).
- \* For intra-band contiguous carrier aggregation, with or without MIMO or TX diversity, TAE shall not exceed 130 ns (i.e.  $1/2 T_c$ ).
- \* For intra-band non-contiguous carrier aggregation, with or without MIMO or TX diversity, TAE shall not exceed 260 ns (i.e. one  $T_c$ ).
- \* For inter-band carrier aggregation, with or without MIMO or TX diversity, TAE shall not exceed 260 ns.

Transport link contribution to radio frequency error:

+/-2 PPB. This value is considered to be "available" for the Fronthaul link out of the total 50 PPB budget reserved for the radio interface. Note: the reason that the transport link contributes to radio frequency error is as follows. At the time of this writing, Fronthaul communication is from the radio unit to remote radio head directly. The remote radio head is essentially a passive device (without buffering etc.) The transport drives the antenna directly by feeding it with samples and everything the transport adds will be introduced to radio as-is. So if the transport causes additional frequency error that shows immediately on the radio as well. Note: performance guarantees of low nanosecond values such as these are considered to be below the DetNet layer - it is assumed that the underlying implementation, e.g. the hardware, will provide sufficient support to enable this level of performance. These values are maintained in the use case to give an indication of the overall application.

The above listed time synchronization requirements are difficult to meet with point-to-point connected networks, and more difficult when the network includes multiple hops. It is expected that networks must include buffering at the ends of the connections as imposed by the jitter requirements, since trying to meet the jitter requirements in every intermediate node is likely to be too costly. However, every measure to reduce jitter and delay on the path makes it easier to meet the end-to-end requirements.

In order to meet the timing requirements both senders and receivers must remain time synchronized, demanding very accurate clock distribution, for example support for IEEE 1588 transparent clocks or boundary clocks in every intermediate node.

In cellular networks from the LTE radio era onward, phase synchronization is needed in addition to frequency synchronization ([TS36300], [TS23401]). Time constraints are also important due to their impact on packet loss. If a packet is delivered too late, then the packet may be dropped by the host.

#### 6.1.4. Transport Loss Constraints

Fronthaul and Midhaul networks assume almost error-free transport. Errors can result in a reset of the radio interfaces, which can cause reduced throughput or broken radio connectivity for mobile customers.

For packetized Fronthaul and Midhaul connections packet loss may be caused by BER, congestion, or network failure scenarios. Different fronthaul functional splits are being considered by 3GPP, requiring strict frame loss ratio (FLR) guarantees. As one example (referring to the legacy CPRI split which is option 8 in 3GPP) lower layers splits may imply an FLR of less than  $10E-7$  for data traffic and less than  $10E-6$  for control and management traffic.

Many of the tools available for eliminating packet loss for Fronthaul and Midhaul networks have serious challenges, for example retransmitting lost packets and/or using forward error correction (FEC) to circumvent bit errors is practically impossible due to the additional delay incurred. Using redundant streams for better guarantees for delivery is also practically impossible in many cases due to high bandwidth requirements of Fronthaul and Midhaul networks. Protection switching is also a candidate but at the time of this writing, available technologies for the path switch are too slow to avoid reset of mobile interfaces.

Fronthaul links are assumed to be symmetric, and all Fronthaul streams (i.e. those carrying radio data) have equal priority and cannot delay or pre-empt each other. This implies that the network must guarantee that each time-sensitive flow meets their schedule.

#### 6.1.5. Security Considerations

Establishing time-sensitive streams in the network entails reserving networking resources for long periods of time. It is important that these reservation requests be authenticated to prevent malicious reservation attempts from hostile nodes (or accidental misconfiguration). This is particularly important in the case where the reservation requests span administrative domains. Furthermore, the reservation information itself should be digitally signed to reduce the risk of a legitimate node pushing a stale or hostile configuration into another networking node.

Note: This is considered important for the security policy of the network, but does not affect the core DetNet architecture and design.

## 6.2. Cellular Radio Networks Today

### 6.2.1. Fronthaul

Today's Fronthaul networks typically consist of:

- o Dedicated point-to-point fiber connection is common
- o Proprietary protocols and framings
- o Custom equipment and no real networking

At the time of this writing, solutions for Fronthaul are direct optical cables or Wavelength-Division Multiplexing (WDM) connections.

### 6.2.2. Midhaul and Backhaul

Today's Midhaul and Backhaul networks typically consist of:

- o Mostly normal IP networks, MPLS-TP, etc.
- o Clock distribution and sync using 1588 and SyncE

Telecommunication networks in the Mid- and Backhaul are already heading towards transport networks where precise time synchronization support is one of the basic building blocks. While the transport networks themselves have practically transitioned to all-IP packet-based networks to meet the bandwidth and cost requirements, highly accurate clock distribution has become a challenge.

In the past, Mid- and Backhaul connections were typically based on Time Division Multiplexing (TDM-based) and provided frequency synchronization capabilities as a part of the transport media. Alternatively other technologies such as Global Positioning System (GPS) or Synchronous Ethernet (SyncE) are used [SyncE].

Both Ethernet and IP/MPLS [RFC3031] (and PseudoWires (PWE) [RFC3985] for legacy transport support) have become popular tools to build and manage new all-IP Radio Access Networks (RANs) [I-D.kh-spring-ip-ran-use-case]. Although various timing and synchronization optimizations have already been proposed and implemented including 1588 PTP enhancements [I-D.ietf-tictoc-1588overmpls] and [RFC8169], these solution are not necessarily sufficient for the forthcoming RAN architectures nor do

they guarantee the more stringent time-synchronization requirements such as [CPRI].

There are also existing solutions for TDM over IP such as [RFC4553], [RFC5086], and [RFC5087], as well as TDM over Ethernet transports such as [MEF8].

### 6.3. Cellular Radio Networks Future

Future Cellular Radio Networks will be based on a mix of different xHaul networks (xHaul = front-, mid- and backhaul), and future transport networks should be able to support all of them simultaneously. It is already envisioned today that:

- o Not all "cellular radio network" traffic will be IP, for example some will remain at Layer 2 (e.g. Ethernet based). DetNet solutions must address all traffic types (Layer 2, Layer 3) with the same tools and allow their transport simultaneously.
- o All forms of xHaul networks will need some form of DetNet solutions. For example with the advent of 5G some Backhaul traffic will also have DetNet requirements, for example traffic belonging to time-critical 5G applications.
- o Different splits of the functionality run on the base stations and the on-site units could co-exist on the same Fronthaul and Backhaul network.

Future Cellular Radio networks should contain the following:

- o Unified standards-based transport protocols and standard networking equipment that can make use of underlying deterministic link-layer services
- o Unified and standards-based network management systems and protocols in all parts of the network (including Fronthaul)

New radio access network deployment models and architectures may require time- sensitive networking services with strict requirements on other parts of the network that previously were not considered to be packetized at all. Time and synchronization support are already topical for Backhaul and Midhaul packet networks [MEF22.1.1] and are becoming a real issue for Fronthaul networks also. Specifically in Fronthaul networks the timing and synchronization requirements can be extreme for packet based technologies, for example, on the order of sub +-20 ns packet delay variation (PDV) and frequency accuracy of +0.002 PPM [Fronthaul].



The actual transport protocols and/or solutions to establish required transport "circuits" (pinned-down paths) for Fronthaul traffic are still undefined. Those are likely to include (but are not limited to) solutions directly over Ethernet, over IP, and using MPLS/PseudoWire transport.

Interesting and important work for time-sensitive networking has been done for Ethernet [TSNTG], which specifies the use of IEEE 1588 time precision protocol (PTP) [IEEE1588] in the context of IEEE 802.1D and IEEE 802.1Q. [IEEE8021AS] specifies a Layer 2 time synchronizing service, and other specifications such as IEEE 1722 [IEEE1722] specify Ethernet-based Layer-2 transport for time-sensitive streams.

However even these Ethernet TSN features may not be sufficient for Fronthaul traffic. Therefore, having specific profiles that take the requirements of Fronthaul into account is desirable [IEEE8021CM].

New promising work seeks to enable the transport of time-sensitive fronthaul streams in Ethernet bridged networks [IEEE8021CM]. Analogous to IEEE 1722 there is an ongoing standardization effort to define the Layer-2 transport encapsulation format for transporting radio over Ethernet (RoE) in the IEEE 1904.3 Task Force [IEEE19143].

As mentioned in Section 6.1.2, 5G communications will provide one of the most challenging cases for delay sensitive networking. In order to meet the challenges of ultra-low latency and ultra-high throughput, 3GPP has studied various "functional splits" for 5G, i.e., physical decomposition of the gNodeB base station and deployment of its functional blocks in different locations [TR38801].

These splits are numbered from split option 1 (Dual Connectivity, a split in which the radio resource control is centralized and other radio stack layers are in distributed units) to split option 8 (a PHY-RF split in which RF functionality is in a distributed unit and the rest of the radio stack is in the centralized unit), with each intermediate split having its own data rate and delay requirements. Packetized versions of different splits have been proposed including eCPRI [eCPRI] and RoE (as previously noted). Both provide Ethernet encapsulations, and eCPRI is also capable of IP encapsulation.

All-IP RANs and xHaul networks would benefit from time synchronization and time-sensitive transport services. Although Ethernet appears to be the unifying technology for the transport, there is still a disconnect providing Layer 3 services. The protocol stack typically has a number of layers below the Ethernet Layer 2 that shows up to the Layer 3 IP transport. It is not uncommon that on top of the lowest layer (optical) transport there is the first layer of Ethernet followed one or more layers of MPLS, PseudoWires

and/or other tunneling protocols finally carrying the Ethernet layer visible to the user plane IP traffic.

While there are existing technologies to establish circuits through the routed and switched networks (especially in MPLS/PWE space), there is still no way to signal the time synchronization and time-sensitive stream requirements/reservations for Layer-3 flows in a way that addresses the entire transport stack, including the Ethernet layers that need to be configured.

Furthermore, not all "user plane" traffic will be IP. Therefore, the same solution also must address the use cases where the user plane traffic is a different layer, for example Ethernet frames.

There is existing work describing the problem statement [I-D.ietf-detnet-problem-statement] and the architecture [I-D.ietf-detnet-architecture] for deterministic networking (DetNet) that targets solutions for time-sensitive (IP/transport) streams with deterministic properties over Ethernet-based switched networks.

#### 6.4. Cellular Radio Networks Asks

A standard for data plane transport specification which is:

- o Unified among all xHauls (meaning that different flows with diverse DetNet requirements can coexist in the same network and traverse the same nodes without interfering with each other)
- o Deployed in a highly deterministic network environment
- o Capable of supporting multiple functional splits simultaneously, including existing Backhaul and CPRI Fronthaul and potentially new modes as defined for example in 3GPP; these goals can be supported by the existing DetNet Use Case Common Themes, notably "Mix of Deterministic and Best-Effort Traffic", "Bounded Latency", "Low Latency", "Symmetrical Path Delays", and "Deterministic Flows".
- o Capable of supporting Network Slicing and Multi-tenancy; these goals can be supported by the same DetNet themes noted above.
- o Capable of transporting both in-band and out-band control traffic (OAM info, ...).
- o Deployable over multiple data link technologies (e.g., IEEE 802.3, mmWave, etc.).

A standard for data flow information models that are:

- o Aware of the time sensitivity and constraints of the target networking environment
- o Aware of underlying deterministic networking services (e.g., on the Ethernet layer)

## 7. Industrial Machine to Machine (M2M)

### 7.1. Use Case Description

Industrial Automation in general refers to automation of manufacturing, quality control and material processing. This "machine to machine" (M2M) use case considers machine units in a plant floor which periodically exchange data with upstream or downstream machine modules and/or a supervisory controller within a local area network.

The actors of M2M communication are Programmable Logic Controllers (PLCs). Communication between PLCs and between PLCs and the supervisory PLC (S-PLC) is achieved via critical control/data streams Figure 11.

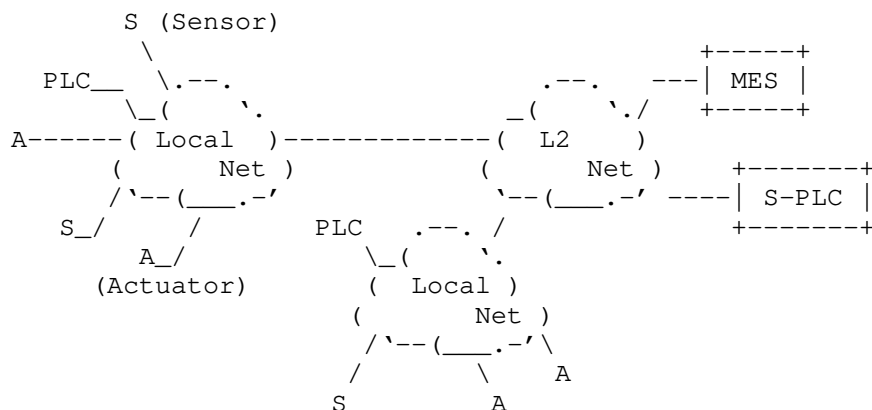


Figure 11: Current Generic Industrial M2M Network Architecture

This use case focuses on PLC-related communications; communication to Manufacturing-Execution-Systems (MESs) are not addressed.

This use case covers only critical control/data streams; non-critical traffic between industrial automation applications (such as communication of state, configuration, set-up, and database communication) are adequately served by prioritizing techniques available at the time of this writing. Such traffic can use up to

80% of the total bandwidth required. There is also a subset of non-time-critical traffic that must be reliable even though it is not time-sensitive.

In this use case the primary need for deterministic networking is to provide end-to-end delivery of M2M messages within specific timing constraints, for example in closed loop automation control. Today this level of determinism is provided by proprietary networking technologies. In addition, standard networking technologies are used to connect the local network to remote industrial automation sites, e.g. over an enterprise or metro network which also carries other types of traffic. Therefore, flows that should be forwarded with deterministic guarantees need to be sustained regardless of the amount of other flows in those networks.

## 7.2. Industrial M2M Communication Today

Today, proprietary networks fulfill the needed timing and availability for M2M networks.

The network topologies used today by industrial automation are similar to those used by telecom networks: Daisy Chain, Ring, Hub and Spoke, and Comb (a subset of Daisy Chain).

PLC-related control/data streams are transmitted periodically and carry either a pre-configured payload or a payload configured during runtime.

Some industrial applications require time synchronization at the end nodes. For such time-coordinated PLCs, accuracy of 1 microsecond is required. Even in the case of "non-time-coordinated" PLCs time sync may be needed e.g. for timestamping of sensor data.

Industrial network scenarios require advanced security solutions. At the time of this writing, many industrial production networks are physically separated. Preventing critical flows from being leaked outside a domain is handled by filtering policies that are typically enforced in firewalls.

### 7.2.1. Transport Parameters

The Cycle Time defines the frequency of message(s) between industrial actors. The Cycle Time is application dependent, in the range of 1ms - 100ms for critical control/data streams.

Because industrial applications assume deterministic transport for critical Control-Data-Stream parameters (instead of defining latency and delay variation parameters) it is sufficient to fulfill the upper

bound of latency (maximum latency). The underlying networking infrastructure must ensure a maximum end-to-end delivery time of messages in the range of 100 microseconds to 50 milliseconds depending on the control loop application.

The bandwidth requirements of control/data streams are usually calculated directly from the bytes-per-cycle parameter of the control loop. For PLC-to-PLC communication one can expect 2 - 32 streams with packet size in the range of 100 - 700 bytes. For S-PLC to PLCs the number of streams is higher - up to 256 streams. Usually no more than 20% of available bandwidth is used for critical control/data streams. In today's networks 1Gbps links are commonly used.

Most PLC control loops are rather tolerant of packet loss, however critical control/data streams accept no more than 1 packet loss per consecutive communication cycle (i.e. if a packet gets lost in cycle "n", then the next cycle ("n+1") must be lossless). After two or more consecutive packet losses the network may be considered to be "down" by the Application.

As network downtime may impact the whole production system the required network availability is rather high (99.999%).

Based on the above parameters some form of redundancy will be required for M2M communications, however any individual solution depends on several parameters including cycle time, delivery time, etc.

#### 7.2.2. Stream Creation and Destruction

In an industrial environment, critical control/data streams are created rather infrequently, on the order of ~10 times per day / week / month. Most of these critical control/data streams get created at machine startup, however flexibility is also needed during runtime, for example when adding or removing a machine. Going forward as production systems become more flexible, there will be a significant increase in the rate at which streams are created, changed and destroyed.

#### 7.3. Industrial M2M Future

We foresee a converged IP-standards-based network with deterministic properties that can satisfy the timing, security and reliability constraints described above. Today's proprietary networks could then be interfaced to such a network via gateways or, in the case of new installations, devices could be connected directly to the converged network.

For this use case time synchronization accuracy on the order of 1us is expected.

#### 7.4. Industrial M2M Asks

- o Converged IP-based network
- o Deterministic behavior (bounded latency and jitter )
- o High availability (presumably through redundancy) (99.999 %)
- o Low message delivery time (100us - 50ms)
- o Low packet loss (with bounded number of consecutive lost packets)
- o Security (e.g. prevent critical flows from being leaked between physically separated networks)

### 8. Mining Industry

#### 8.1. Use Case Description

The mining industry is highly dependent on networks to monitor and control their systems both in open-pit and underground extraction, transport and refining processes. In order to reduce risks and increase operational efficiency in mining operations, a number of processes have migrated the operators from the extraction site to remote control and monitoring.

In the case of open pit mining, autonomous trucks are used to transport the raw materials from the open pit to the refining factory where the final product (e.g. Copper) is obtained. Although the operation is autonomous, the trucks are remotely monitored from a central facility.

In pit mines, the monitoring of the tailings or mine dumps is critical in order to minimize environmental pollution. In the past, monitoring has been conducted through manual inspection of pre-installed dataloggers. Cabling is not usually exploited in such scenarios due to the cost and complex deployment requirements. At the time of this writing, wireless technologies are being employed to monitor these cases permanently. Slopes are also monitored in order to anticipate possible mine collapse. Due to the unstable terrain, cable maintenance is costly and complex and hence wireless technologies are employed.

In the underground monitoring case, autonomous vehicles with extraction tools travel autonomously through the tunnels, but their

operational tasks (such as excavation, stone breaking and transport) are controlled remotely from a central facility. This generates video and feedback upstream traffic plus downstream actuator control traffic.

## 8.2. Mining Industry Today

At the time of this writing, the mining industry uses a packet switched architecture supported by high speed ethernet. However in order to achieve the delay and packet loss requirements the network bandwidth is overestimated, thus providing very low efficiency in terms of resource usage.

QoS is implemented at the Routers to separate video, management, monitoring and process control traffic for each stream.

Since mobility is involved in this process, the connection between the backbone and the mobile devices (e.g. trucks, trains and excavators) is solved using a wireless link. These links are based on 802.11 for open-pit mining and "leaky feeder" communications for underground mining. (A "leaky feeder" communication system consists of a coaxial cable run along tunnels which emits and receives radio waves, functioning as an extended antenna. The cable is "leaky" in that it has gaps or slots in its outer conductor to allow the radio signal to leak into or out of the cable along its entire length.)

Lately in pit mines the use of LPWAN technologies has been extended: Tailings, slopes and mine dumps are monitored by battery-powered dataloggers that make use of robust long range radio technologies. Reliability is usually ensured through retransmissions at L2. Gateways or concentrators act as bridges forwarding the data to the backbone ethernet network. Deterministic requirements are biased towards reliability rather than latency as events are slowly triggered or can be anticipated in advance.

At the mineral processing stage, conveyor belts and refining processes are controlled by a SCADA system, which provides the in-factory delay-constrained networking requirements.

At the time of this writing, voice communications are served by a redundant trunking infrastructure, independent from data networks.

## 8.3. Mining Industry Future

Mining operations and management are converging towards a combination of autonomous operation and teleoperation of transport and extraction machines. This means that video, audio, monitoring and process

control traffic will increase dramatically. Ideally, all activities on the mine will rely on network infrastructure.

Wireless for open-pit mining is already a reality with LPWAN technologies and it is expected to evolve to more advanced LPWAN technologies such as those based on LTE to increase last hop reliability or novel LPWAN flavours with deterministic access.

One area in which DetNet can improve this use case is in the wired networks that make up the "backbone network" of the system, which connect together many wireless access points (APs). The mobile machines (which are connected to the network via wireless) transition from one AP to the next as they move about. A deterministic, reliable, low latency backbone can enable these transitions to be more reliable.

Connections which extend all the way from the base stations to the machinery via a mix of wired and wireless hops would also be beneficial, for example to improve remote control responsiveness of digging machines. However to guarantee deterministic performance of a DetNet, the end-to-end underlying network must be deterministic. Thus for this use case if a deterministic wireless transport is integrated with a wire-based DetNet network, it could create the desired wired plus wireless end-to-end deterministic network.

#### 8.4. Mining Industry Asks

- o Improved bandwidth efficiency
- o Very low delay to enable machine teleoperation
- o Dedicated bandwidth usage for high resolution video streams
- o Predictable delay to enable realtime monitoring
- o Potential to construct a unified DetNet network over a combination of wired and deterministic wireless links

### 9. Private Blockchain

#### 9.1. Use Case Description

Blockchain was created with bitcoin as a 'public' blockchain on the open Internet, however blockchain has also spread far beyond its original host into various industries such as smart manufacturing, logistics, security, legal rights and others. In these industries blockchain runs in designated and carefully managed networks in which



deterministic networking requirements could be addressed by DetNet. Such implementations are referred to as 'private' blockchain.

The sole distinction between public and private blockchain is defined by who is allowed to participate in the network, execute the consensus protocol, and maintain the shared ledger.

Today's networks treat the traffic from blockchain on a best-effort basis, but blockchain operation could be made much more efficient if deterministic networking services were available to minimize latency and packet loss in the network.

#### 9.1.1. Blockchain Operation

A 'block' runs as a container of a batch of primary items such as transactions, property records etc. The blocks are chained in such a way that the hash of the previous block works as the pointer to the header of the new block. Confirmation of each block requires a consensus mechanism. When an item arrives at a blockchain node, the latter broadcasts this item to the rest of the nodes which receive and verify it and put it in the ongoing block. The block confirmation process begins as the number of items reaches the predefined block capacity, at which time the node broadcasts its proved block to the rest of the nodes, to be verified and chained. The result is that block N+1 of each chain transitively vouches for blocks N and before of that chain.

#### 9.1.2. Blockchain Network Architecture

Blockchain node communication and coordination is achieved mainly through frequent point-to-multi-point communication, however persistent point-to-point connections are used to transport both the items and the blocks to the other nodes. For example, consider the following implementation.

When a node is initiated, it first requests the other nodes' address from a specific entity such as DNS, then it creates persistent connections each of with other nodes. If a node confirms an item, it sends the item to the other nodes via these persistent connections.

As a new block in a node is completed and is proven by the surrounding nodes, it propagates towards its neighbor nodes. When node A receives a block, it verifies it, then sends an invite message to its neighbor B. Neighbor B checks to see if the designated block is available, and responds to A if it is unavailable, then A sends the complete block to B. B repeats the process (as done by A above) to start the next round of block propagation.

The challenge of blockchain network operation is not overall data rates, since the volume from both block and item stays between hundreds of bytes to a couple of megabytes per second, but is in transporting the blocks with minimum latency to maximize efficiency of the blockchain consensus process. The efficiency of differing implementations of the consensus process may be affected to a differing degree by the latency (and variation of latency) of the network.

#### 9.1.3. Security Considerations

Security is crucial to blockchain applications, and at the time of this writing, blockchain systems address security issues mainly at the application level, where cryptography as well as hash-based consensus play a leading role in preventing both double-spending and malicious service attacks. However, there is concern that in the proposed use case of a private blockchain network which is dependent on deterministic properties, the network could be vulnerable to delays and other specific attacks against determinism which could interrupt service.

#### 9.2. Private Blockchain Today

Today private blockchain runs in L2 or L3 VPN, in general without guaranteed determinism. The industry players are starting to realize that improving determinism in their blockchain networks could improve the performance of their service, but as of today these goals are not being met.

#### 9.3. Private Blockchain Future

Blockchain system performance can be greatly improved through deterministic networking service primarily because it would accelerate the consensus process. It would be valuable to be able to design a private blockchain network with the following properties:

- o Transport of point-to-multi-point traffic in a coordinated network architecture rather than at the application layer (which typically uses point-to-point connections)
- o Guaranteed transport latency
- o Reduced packet loss (to the point where packet retransmission-incurred delay would be negligible.)

#### 9.4. Private Blockchain Asks

- o Layer 2 and Layer 3 multicast of blockchain traffic
- o Item and block delivery with bounded, low latency and negligible packet loss
- o Coexistence in a single network of blockchain and IT traffic.
- o Ability to scale the network by distributing the centralized control of the network across multiple control entities.

### 10. Network Slicing

#### 10.1. Use Case Description

Network Slicing divides one physical network infrastructure into multiple logical networks. Each slice, corresponding to a logical network, uses resources and network functions independently from each other. Network Slicing provides flexibility of resource allocation and service quality customization.

Future services will demand network performance with a wide variety of characteristics such as high data rate, low latency, low loss rate, security and many other parameters. Ideally every service would have its own physical network satisfying its particular performance requirements, however that would be prohibitively expensive. Network Slicing can provide a customized slice for a single service, and multiple slices can share the same physical network. This method can optimize the performance for the service at lower cost, and the flexibility of setting up and release the slices also allows the user to allocate the network resources dynamically.

Unlike the other use cases presented here, Network Slicing is not a specific application that depends on specific deterministic properties; rather it is introduced as an area of networking to which DetNet might be applicable.

#### 10.2. DetNet Applied to Network Slicing

##### 10.2.1. Resource Isolation Across Slices

One of the requirements discussed for Network Slicing is the "hard" separation of various users' deterministic performance. That is, it should be impossible for activity, lack of activity, or changes in activity of one or more users to have any appreciable effect on the deterministic performance parameters of any other slices. Typical techniques used today, which share a physical network among users, do

not offer this level of isolation. DetNet can supply point-to-point or point-to-multipoint paths that offer bandwidth and latency guarantees to a user that cannot be affected by other users' data traffic. Thus DetNet is a powerful tool when latency and reliability are required in Network Slicing.

#### 10.2.2. Deterministic Services Within Slices

Slices may need to provide services with DetNet-type performance guarantees, however note that a system can be implemented to provide such services in more than one way. For example the slice itself might be implemented using DetNet, and thus the slice can provide service guarantees and isolation to its users without any particular DetNet awareness on the part of the users' applications. Alternatively, a "non-DetNet-aware" slice may host an application that itself implements DetNet services and thus can enjoy similar service guarantees.

#### 10.3. A Network Slicing Use Case Example - 5G Bearer Network

Network Slicing is a core feature of 5G defined in 3GPP, which is under development at the time of this writing [TR38501]. A network slice in a mobile network is a complete logical network including Radio Access Network (RAN) and Core Network (CN). It provides telecommunication services and network capabilities, which may vary from slice to slice. A 5G bearer network is a typical use case of Network Slicing; for example consider three 5G service scenarios: eMMB, URLLC, and mMTC.

- o eMBB (Enhanced Mobile Broadband) focuses on services characterized by high data rates, such as high definition videos, virtual reality, augmented reality, and fixed mobile convergence.
- o URLLC (Ultra-Reliable and Low Latency Communications) focuses on latency-sensitive services, such as self-driving vehicles, remote surgery, or drone control.
- o mMTC (massive Machine Type Communications) focuses on services that have high requirements for connection density, such as those typical for smart city and smart agriculture use cases.

A 5G bearer network could use DetNet to provide hard resource isolation across slices and within the slice. For example consider Slice-A and Slice-B, with DetNet used to transit services URLLC-A and URLLC-B over them. Without DetNet, URLLC-A and URLLC-B would compete for bandwidth resource, and latency and reliability would not be guaranteed. With DetNet, URLLC-A and URLLC-B have separate bandwidth

reservation and there is no resource conflict between them, as though they were in different logical networks.

#### 10.4. Non-5G Applications of Network Slicing

Although operation of services not related to 5G is not part of the 5G Network Slicing definition and scope, Network Slicing is likely to become a preferred approach to providing various services across a shared physical infrastructure. Examples include providing electrical utilities services and pro audio services via slices. Use cases like these could become more common once the work for the 5G core network evolves to include wired as well as wireless access.

#### 10.5. Limitations of DetNet in Network Slicing

DetNet cannot cover every Network Slicing use case. One issue is that DetNet is a point-to-point or point-to-multipoint technology, however Network Slicing ultimately needs multi-point to multi-point guarantees. Another issue is that the number of flows that can be carried by DetNet is limited by DetNet scalability; flow aggregation and queuing management modification may help address this. Additional work and discussion are needed to address these topics.

#### 10.6. Network Slicing Today and Future

Network Slicing has the promise to satisfy many requirements of future network deployment scenarios, but it is still a collection of ideas and analysis, without a specific technical solution. DetNet is one of various technologies that have potential to be used in Network Slicing, along with for example Flex-E and Segment Routing. For more information please see the IETF99 Network Slicing BOF session agenda and materials.

#### 10.7. Network Slicing Asks

- o Isolation from other flows through Queuing Management
- o Service Quality Customization and Guarantee
- o Security

### 11. Use Case Common Themes

This section summarizes the expected properties of a DetNet network, based on the use cases as described in this draft.

### 11.1. Unified, standards-based network

#### 11.1.1. Extensions to Ethernet

A DetNet network is not "a new kind of network" - it based on extensions to existing Ethernet standards, including elements of IEEE 802.1 AVB/TSN and related standards. Presumably it will be possible to run DetNet over other underlying transports besides Ethernet, but Ethernet is explicitly supported.

#### 11.1.2. Centrally Administered

In general a DetNet network is not expected to be "plug and play" - it is expected that there is some centralized network configuration and control system. Such a system may be in a single central location, or it maybe distributed across multiple control entities that function together as a unified control system for the network. However, the ability to "hot swap" components (e.g. due to malfunction) is similar enough to "plug and play" that this kind of behavior may be expected in DetNet networks, depending on the implementation.

#### 11.1.3. Standardized Data Flow Information Models

Data Flow Information Models to be used with DetNet networks are to be specified by DetNet.

#### 11.1.4. L2 and L3 Integration

A DetNet network is intended to integrate between Layer 2 (bridged) network(s) (e.g. AVB/TSN LAN) and Layer 3 (routed) network(s) (e.g. using IP-based protocols). One example of this is "making AVB/TSN-type deterministic performance available from Layer 3 applications, e.g. using RTP". Another example is "connecting two AVB/TSN LANs ("islands") together through a standard router".

#### 11.1.5. Consideration for IPv4

This Use Cases draft explicitly does not specify any particular implementation or protocol, however it has been observed that various of the use cases described (and their associated industries) are explicitly based on IPv4 (as opposed to IPv6) and it is not considered practical to expect them to migrate to IPv6 in order to use DetNet. Thus the expectation is that even if not every feature of DetNet is available in an IPv4 context, at least some of the significant benefits (such as guaranteed end-to-end delivery and low latency) are expected to be available.

#### 11.1.6. Guaranteed End-to-End Delivery

Packets in a DetNet flow are guaranteed not to be dropped by the network due to congestion. However, the network may drop packets for intended reasons, e.g. per security measures. Similarly best-effort traffic on a DetNet is subject to being dropped (as on a non-DetNet IP network). Also note that this guarantee applies to the actions of DetNet protocol software, and does not provide any guarantee against lower level errors such as media errors or checksum errors.

#### 11.1.7. Replacement for Multiple Proprietary Deterministic Networks

There are many proprietary non-interoperable deterministic Ethernet-based networks available; DetNet is intended to provide an open-standards-based alternative to such networks.

#### 11.1.8. Mix of Deterministic and Best-Effort Traffic

DetNet is intended to support coexistence of time-sensitive operational (OT) traffic and information (IT) traffic on the same ("unified") network.

#### 11.1.9. Unused Reserved BW to be Available to Best-Effort Traffic

If bandwidth reservations are made for a stream but the associated bandwidth is not used at any point in time, that bandwidth is made available on the network for best-effort traffic. If the owner of the reserved stream then starts transmitting again, the bandwidth is no longer available for best-effort traffic, on a moment-to-moment basis. Note that such "temporarily available" bandwidth is not available for time-sensitive traffic, which must have its own reservation.

#### 11.1.10. Lower Cost, Multi-Vendor Solutions

The DetNet network specifications are intended to enable an ecosystem in which multiple vendors can create interoperable products, thus promoting device diversity and potentially higher numbers of each device manufactured, promoting cost reduction and cost competition among vendors. The intent is that DetNet networks should be able to be created at lower cost and with greater diversity of available devices than existing proprietary networks.

### 11.2. Scalable Size

DetNet networks range in size from very small, e.g. inside a single industrial machine, to very large, for example a Utility Grid network spanning a whole country, and involving many "hops" over various

kinds of links for example radio repeaters, microwave links, fiber optic links, etc.. However recall that the scope of DetNet is confined to networks that are centrally administered, and explicitly excludes unbounded decentralized networks such as the Internet.

#### 11.2.1. Scalable Number of Flows

The number of flows in a given network application can potentially be large, and can potentially grow faster than the number of nodes and hops. So the network should provide a sufficient (perhaps configurable) maximum number of flows for any given application.

### 11.3. Scalable Timing Parameters and Accuracy

#### 11.3.1. Bounded Latency

The DetNet Data Flow Information Model is expected to provide means to configure the network that include parameters for querying network path latency, requesting bounded latency for a given stream, requesting worst case maximum and/or minimum latency for a given path or stream, and so on. It is an expected case that the network may not be able to provide a given requested service level, and if so the network control system should reply that the requested services is not available (as opposed to accepting the parameter but then not delivering the desired behavior).

#### 11.3.2. Low Latency

Applications may require "extremely low latency" however depending on the application these may mean very different latency values; for example "low latency" across a Utility grid network is on a different time scale than "low latency" in a motor control loop in a small machine. The intent is that the mechanisms for specifying desired latency include wide ranges, and that architecturally there is nothing to prevent arbitrarily low latencies from being implemented in a given network.

#### 11.3.3. Bounded Jitter (Latency Variation)

As with the other Latency-related elements noted above, parameters should be available to determine or request the allowed variation in latency.

#### 11.3.4. Symmetrical Path Delays

Some applications would like to specify that the transit delay time values be equal for both the transmit and return paths.



#### 11.4. High Reliability and Availability

Reliability is of critical importance to many DetNet applications, in which consequences of failure can be extraordinarily high in terms of cost and even human life. DetNet based systems are expected to be implemented with essentially arbitrarily high availability (for example 99.9999% up time, or even 12 nines). The intent is that the DetNet designs should not make any assumptions about the level of reliability and availability that may be required of a given system, and should define parameters for communicating these kinds of metrics within the network.

A strategy used by DetNet for providing such extraordinarily high levels of reliability is to provide redundant paths that can be seamlessly switched between, while maintaining the required performance of that system.

#### 11.5. Security

Security is of critical importance to many DetNet applications. A DetNet network must be able to be made secure against devices failures, attackers, misbehaving devices, and so on. In a DetNet network the data traffic is expected to be time-sensitive, thus in addition to arriving with the data content as intended, the data must also arrive at the expected time. This may present "new" security challenges to implementers, and must be addressed accordingly. There are other security implications, including (but not limited to) the change in attack surface presented by packet replication and elimination.

#### 11.6. Deterministic Flows

Reserved bandwidth data flows must be isolated from each other and from best-effort traffic, so that even if the network is saturated with best-effort (and/or reserved bandwidth) traffic, the configured flows are not adversely affected.

### 12. Security Considerations

This document covers a number of representative applications and network scenarios that are expected to make use of DetNet technologies. Each of the potential DetNet uses cases will have security considerations from both the use-specific and DetNet technology perspectives. While some use-specific security considerations are discussed above, a more comprehensive discussion of such considerations is captured in DetNet Security Considerations [I-D.ietf-detnet-security]. Readers are encouraged to review this

document to gain a more complete understanding of DetNet related security considerations.

### 13. Contributors

RFC7322 limits the number of authors listed on the front page of a draft to a maximum of 5, far fewer than the 20 individuals below who made important contributions to this draft. The editor wishes to thank and acknowledge each of the following authors for contributing text to this draft. See also Section 14.

Craig Gunther (Harman International)  
10653 South River Front Parkway, South Jordan, UT 84095  
phone +1 801 568-7675, email [craig.gunther@harman.com](mailto:craig.gunther@harman.com)

Pascal Thubert (Cisco Systems, Inc)  
Building D, 45 Allee des Ormes - BP1200, MOUGINS  
Sophia Antipolis 06254 FRANCE  
phone +33 497 23 26 34, email [pthubert@cisco.com](mailto:pthubert@cisco.com)

Patrick Wetterwald (Cisco Systems)  
45 Allee des Ormes, Mougins, 06250 FRANCE  
phone +33 4 97 23 26 36, email [pwetterw@cisco.com](mailto:pwetterw@cisco.com)

Jean Raymond (Hydro-Quebec)  
1500 University, Montreal, H3A3S7, Canada  
phone +1 514 840 3000, email [raymond.jean@hydro.qc.ca](mailto:raymond.jean@hydro.qc.ca)

Jouni Korhonen (Broadcom Corporation)  
3151 Zanker Road, San Jose, 95134, CA, USA  
email [jouni.nospam@gmail.com](mailto:jouni.nospam@gmail.com)

Yu Kaneko (Toshiba)  
1 Komukai-Toshiba-cho, Saiwai-ku, Kasasaki-shi, Kanagawa, Japan  
email [yul.kaneko@toshiba.co.jp](mailto:yul.kaneko@toshiba.co.jp)

Subir Das (Vencore Labs)  
150 Mount Airy Road, Basking Ridge, New Jersey, 07920, USA  
email [sdas@appcomsci.com](mailto:sdas@appcomsci.com)

Balazs Varga (Ericsson)  
Konyves Kalman krt. 11/B, Budapest, Hungary, 1097  
email [balazs.a.varga@ericsson.com](mailto:balazs.a.varga@ericsson.com)

Janos Farkas (Ericsson)  
Konyves Kalman krt. 11/B, Budapest, Hungary, 1097  
email [janos.farkas@ericsson.com](mailto:janos.farkas@ericsson.com)

Franz-Josef Goetz (Siemens)  
Gleiwitzerstr. 555, Nurnberg, Germany, 90475  
email franz-josef.goetz@siemens.com

Juergen Schmitt (Siemens)  
Gleiwitzerstr. 555, Nurnberg, Germany, 90475  
email juergen.jues.schmitt@siemens.com

Xavier Vilajosana (Worldsensing)  
483 Arago, Barcelona, Catalonia, 08013, Spain  
email xvilajosana@worldsensing.com

Toktam Mahmoodi (King's College London)  
Strand, London WC2R 2LS, United Kingdom  
email toktam.mahmoodi@kcl.ac.uk

Spiros Spirou (Intracom Telecom)  
19.7 km Markopoulou Ave., Peania, Attiki, 19002, Greece  
email spiros.spirou@gmail.com

Petra Vizarreta (Technical University of Munich)  
Maxvorstadt, ArcisstraBe 21, Munich, 80333, Germany  
email petra.stojasavljevic@tum.de

Daniel Huang (ZTE Corporation, Inc.)  
No. 50 Software Avenue, Nanjing, Jiangsu, 210012, P.R. China  
email huang.guangping@zte.com.cn

Xuesong Geng (Huawei Technologies)  
email gengxuesong@huawei.com

Diego Dujovne (Universidad Diego Portales)  
email diego.dujovne@mail.udp.cl

Maik Seewald (Cisco Systems)  
email maseewal@cisco.com

## 14. Acknowledgments

### 14.1. Pro Audio

This section was derived from draft-gunther-detnet-proaudio-req-01.

The editors would like to acknowledge the help of the following individuals and the companies they represent:

Jeff Koftinoff, Meyer Sound

Jouni Korhonen, Associate Technical Director, Broadcom

Pascal Thubert, CTAO, Cisco

Kieran Tyrrell, Sienda New Media Technologies GmbH

#### 14.2. Utility Telecom

This section was derived from draft-wetterwald-detnet-utilities-reqs-02.

Faramarz Maghsoodlou, Ph. D. IoT Connected Industries and Energy Practice Cisco

Pascal Thubert, CTAO Cisco

The wind power generation use case has been extracted from the study of Wind Farms conducted within the 5GPPP Virtuwind Project. The project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 671648 (VirtuWind).

#### 14.3. Building Automation Systems

This section was derived from draft-bas-usecase-detnet-00.

#### 14.4. Wireless for Industrial Applications

This section was derived from draft-thubert-6tisch-4detnet-01.

This specification derives from the 6TiSCH architecture, which is the result of multiple interactions, in particular during the 6TiSCH (bi)Weekly Interim call, relayed through the 6TiSCH mailing list at the IETF.

The authors wish to thank: Kris Pister, Thomas Watteyne, Xavier Vilajosana, Qin Wang, Tom Phinney, Robert Assimiti, Michael Richardson, Zhuo Chen, Malisa Vucinic, Alfredo Grieco, Martin Turon, Dominique Barthel, Elvis Vogli, Guillaume Gaillard, Herman Storey, Maria Rita Palattella, Nicola Accettura, Patrick Wetterwald, Pouria Zand, Raghuram Sudhaakar, and Shitanshu Shah for their participation and various contributions.

#### 14.5. Cellular Radio

This section was derived from draft-korhonen-detnet-telreq-00.

#### 14.6. Industrial Machine to Machine (M2M)

The authors would like to thank Feng Chen and Marcel Kiessling for their comments and suggestions.

#### 14.7. Internet Applications and CoMP

This section was derived from draft-zha-detnet-use-case-00 by Yiyong Zha.

This document has benefited from reviews, suggestions, comments and proposed text provided by the following members, listed in alphabetical order: Jing Huang, Junru Lin, Lehong Niu and Oilver Huang.

#### 14.8. Network Slicing

This section was written by Xuesong Geng, who would like to acknowledge Norm Finn and Mach Chen for their useful comments.

#### 14.9. Mining

This section was written by Diego Dujovne in conjunction with Xavier Vilasojana.

#### 14.10. Private Blockchain

This section was written by Daniel Huang.

### 15. IANA Considerations

This memo includes no requests from IANA.

### 16. Informative References

- [Ahm14] Ahmed, M. and R. Kim, "Communication network architectures for smart-wind power farms.", *Energies*, p. 3900-3921. , June 2014.
- [bacnetip] ASHRAE, "Annex J to ANSI/ASHRAE 135-1995 - BACnet/IP", January 1999.
- [CoMP] NGMN Alliance, "RAN EVOLUTION PROJECT COMP EVALUATION AND ENHANCEMENT", NGMN Alliance NGMN\_RANEV\_D3\_CoMP\_Evaluation\_and\_Enhancement\_v2.0, March 2015, <[https://www.ngmn.org/uploads/media/NGMN\\_RANEV\\_D3\\_CoMP\\_Evaluation\\_and\\_Enhancement\\_v2.0.pdf](https://www.ngmn.org/uploads/media/NGMN_RANEV_D3_CoMP_Evaluation_and_Enhancement_v2.0.pdf)>.

- [CONTENT\_PROTECTION] Olsen, D., "1722a Content Protection", 2012, <[http://grouper.ieee.org/groups/1722/contributions/2012/avtp\\_dolsen\\_1722a\\_content\\_protection.pdf](http://grouper.ieee.org/groups/1722/contributions/2012/avtp_dolsen_1722a_content_protection.pdf)>.
- [CPRI] CPRI Cooperation, "Common Public Radio Interface (CPRI); Interface Specification", CPRI Specification V6.1, July 2014, <[http://www.cpri.info/downloads/CPRI\\_v\\_6\\_1\\_2014-07-01.pdf](http://www.cpri.info/downloads/CPRI_v_6_1_2014-07-01.pdf)>.
- [DCI] Digital Cinema Initiatives, LLC, "DCI Specification, Version 1.2", 2012, <<http://www.dcinovies.com/>>.
- [eCPRI] IEEE Standards Association, "Common Public Radio Interface, "Common Public Radio Interface: eCPRI Interface Specification V1.0", 2017, <<http://www.cpri.info/>>.
- [ESPN\_DC2] Daley, D., "ESPN's DC2 Scales AVB Large", 2014, <<http://sportsvideo.org/main/blog/2014/06/espns-dc2-scales-avb-large>>.
- [flnet] Japan Electrical Manufacturers Association, "JEMA 1479 - English Edition", September 2012.
- [Fronthaul] Chen, D. and T. Mustala, "Ethernet Fronthaul Considerations", IEEE 1904.3, February 2015, <[http://www.ieee1904.org/3/meeting\\_archive/2015/02/tf3\\_1502\\_chen\\_la.pdf](http://www.ieee1904.org/3/meeting_archive/2015/02/tf3_1502_chen_la.pdf)>.
- [I-D.ietf-6tisch-6top-interface] Wang, Q. and X. Vilajosana, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-04 (work in progress), July 2015.
- [I-D.ietf-6tisch-architecture] Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-19 (work in progress), December 2018.
- [I-D.ietf-6tisch-coap] Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-03 (work in progress), March 2015.

- [I-D.ietf-detnet-architecture]  
Finn, N., Thubert, P., Varga, B., and J. Farkas,  
"Deterministic Networking Architecture", draft-ietf-  
detnet-architecture-09 (work in progress), October 2018.
- [I-D.ietf-detnet-problem-statement]  
Finn, N. and P. Thubert, "Deterministic Networking Problem  
Statement", draft-ietf-detnet-problem-statement-08 (work  
in progress), December 2018.
- [I-D.ietf-detnet-security]  
Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell,  
J., Austad, H., Stanton, K., and N. Finn, "Deterministic  
Networking (DetNet) Security Considerations", draft-ietf-  
detnet-security-03 (work in progress), October 2018.
- [I-D.ietf-tictoc-1588overmpls]  
Davari, S., Oren, A., Bhatia, M., Roberts, P., and L.  
Montini, "Transporting Timing messages over MPLS  
Networks", draft-ietf-tictoc-1588overmpls-07 (work in  
progress), October 2015.
- [I-D.kh-spring-ip-ran-use-case]  
Khasnabish, B., hu, f., and L. Contreras, "Segment Routing  
in IP RAN use case", draft-kh-spring-ip-ran-use-case-02  
(work in progress), November 2014.
- [I-D.svshah-tsvwg-deterministic-forwarding]  
Shah, S. and P. Thubert, "Deterministic Forwarding PHB",  
draft-svshah-tsvwg-deterministic-forwarding-04 (work in  
progress), August 2015.
- [I-D.wang-6tisch-6top-sublayer]  
Wang, Q. and X. Vilajosana, "6TiSCH Operation Sublayer  
(6top)", draft-wang-6tisch-6top-sublayer-04 (work in  
progress), November 2015.
- [IEC-60870-5-104]  
International Electrotechnical Commission, "International  
Standard IEC 60870-5-104: Network access for IEC  
60870-5-101 using standard transport profiles", June 2006.
- [IEC61400]  
"International standard 61400-25: Communications for  
monitoring and control of wind power plants", June 2013.

- [IEEE1588]  
IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008, 2008,  
<<http://standards.ieee.org/findstds/standard/1588-2008.html>>.
- [IEEE1646]  
"Communication Delivery Time Performance Requirements for Electric Power Substation Automation", IEEE Standard 1646-2004 , Apr 2004.
- [IEEE1722]  
IEEE, "1722-2011 - IEEE Standard for Layer 2 Transport Protocol for Time Sensitive Applications in a Bridged Local Area Network", IEEE Std 1722-2011, 2011,  
<<http://standards.ieee.org/findstds/standard/1722-2011.html>>.
- [IEEE19143]  
IEEE Standards Association, "P1914.3/D3.1 Draft Standard for Radio over Ethernet Encapsulations and Mappings", IEEE 1914.3, 2018,  
<<https://standards.ieee.org/develop/project/1914.3.html>>.
- [IEEE802.1TSNTG]  
IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networks Task Group", March 2013,  
<<http://www.ieee802.org/1/pages/avbridges.html>>.
- [IEEE802154]  
IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks".
- [IEEE802154e]  
IEEE standard for Information Technology, "IEEE standard for Information Technology, IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks, June 2011 as amended by IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer", April 2012.



- [IEEE8021AS] IEEE, "Timing and Synchronizations (IEEE 802.1AS-2011)", IEEE 802.1AS-2001, 2011, <<http://standards.ieee.org/getIEEE802/download/802.1AS-2011.pdf>>.
- [IEEE8021CM] Farkas, J., "Time-Sensitive Networking for Fronthaul", Unapproved PAR, PAR for a New IEEE Standard; IEEE P802.1CM, April 2015, <<http://www.ieee802.org/1/files/public/docs2015/new-P802-1CM-dr-aft-PAR-0515-v02.pdf>>.
- [ISA100] ISA/ANSI, "ISA100, Wireless Systems for Automation", <<https://www.isa.org/isa100/>>.
- [knx] KNX Association, "ISO/IEC 14543-3 - KNX", November 2006.
- [lontalk] ECHELON, "LonTalk(R) Protocol Specification Version 3.0", 1994.
- [MEF22.1.1] MEF, "Mobile Backhaul Phase 2 Amendment 1 -- Small Cells", MEF 22.1.1, July 2014, <[http://www.mef.net/Assets/Technical\\_Specifications/PDF/MEF\\_22.1.1.pdf](http://www.mef.net/Assets/Technical_Specifications/PDF/MEF_22.1.1.pdf)>.
- [MEF8] MEF, "Implementation Agreement for the Emulation of PDH Circuits over Metro Ethernet Networks", MEF 8, October 2004, <[https://www.mef.net/Assets/Technical\\_Specifications/PDF/MEF\\_8.pdf](https://www.mef.net/Assets/Technical_Specifications/PDF/MEF_8.pdf)>.
- [METIS] METIS, "Scenarios, requirements and KPIs for 5G mobile and wireless system", ICT-317669-METIS/D1.1 ICT-317669-METIS/D1.1, April 2013, <[https://www.metis2020.com/wp-content/uploads/deliverables/METIS\\_D1.1\\_v1.pdf](https://www.metis2020.com/wp-content/uploads/deliverables/METIS_D1.1_v1.pdf)>.
- [modbus] Modbus Organization, "MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b", December 2006.
- [MODBUS] Modbus Organization, Inc., "MODBUS Application Protocol Specification", Apr 2012.
- [NGMN] NGMN Alliance, "5G White Paper", NGMN 5G White Paper v1.0, February 2015, <[https://www.ngmn.org/uploads/media/NGMN\\_5G\\_White\\_Paper\\_V1\\_0.pdf](https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf)>.

- [NGMN-fronth] NGMN Alliance, "Fronthaul Requirements for C-RAN", March 2015, <[https://www.ngmn.org/uploads/media/NGMN\\_RAN\\_EV\\_D1\\_C-RAN\\_Fronthaul\\_Requirements\\_v1.0.pdf](https://www.ngmn.org/uploads/media/NGMN_RAN_EV_D1_C-RAN_Fronthaul_Requirements_v1.0.pdf)>.
- [OPCXML] OPC Foundation, "OPC XML-Data Access Specification", Dec 2004.
- [PCE] IETF, "Path Computation Element", <<https://datatracker.ietf.org/doc/charter-ietf-pce/>>.
- [profibus] IEC, "IEC 61158 Type 3 - Profibus DP", January 2001.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, DOI 10.17487/RFC3411, December 2002, <<https://www.rfc-editor.org/info/rfc3411>>.
- [RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <<https://www.rfc-editor.org/info/rfc3985>>.
- [RFC4553] Vainshtein, A., Ed. and YJ. Stein, Ed., "Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)", RFC 4553, DOI 10.17487/RFC4553, June 2006, <<https://www.rfc-editor.org/info/rfc4553>>.
- [RFC5086] Vainshtein, A., Ed., Sasson, I., Metz, E., Frost, T., and P. Pate, "Structure-Aware Time Division Multiplexed (TDM) Circuit Emulation Service over Packet Switched Network (CESoPSN)", RFC 5086, DOI 10.17487/RFC5086, December 2007, <<https://www.rfc-editor.org/info/rfc5086>>.
- [RFC5087] Stein, Y(J)., Shashoua, R., Insler, R., and M. Anavi, "Time Division Multiplexing over IP (TDMoIP)", RFC 5087, DOI 10.17487/RFC5087, December 2007, <<https://www.rfc-editor.org/info/rfc5087>>.

- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <<https://www.rfc-editor.org/info/rfc6551>>.
- [RFC7554] Watteyne, T., Ed., Palattella, M., and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement", RFC 7554, DOI 10.17487/RFC7554, May 2015, <<https://www.rfc-editor.org/info/rfc7554>>.
- [RFC8169] Mirsky, G., Ruffini, S., Gray, E., Drake, J., Bryant, S., and A. Vainshtein, "Residence Time Measurement in MPLS Networks", RFC 8169, DOI 10.17487/RFC8169, May 2017, <<https://www.rfc-editor.org/info/rfc8169>>.
- [Spe09] Sperotto, A., Sadre, R., Vliet, F., and A. Pras, "A First Look into SCADA Network Traffic", IP Operations and Management, p. 518-521. , June 2009.
- [SRP\_LATENCY] Gunther, C., "Specifying SRP Latency", 2014, <<http://www.ieee802.org/1/files/public/docs2014/cc-cgunther-acceptable-latency-0314-v01.pdf>>.
- [SyncE] ITU-T, "G.8261 : Timing and synchronization aspects in packet networks", Recommendation G.8261, August 2013, <<http://www.itu.int/rec/T-REC-G.8261>>.
- [TR38501] 3GPP, "3GPP TS 38.501, Technical Specification System Architecture for the 5G System (Release 15)", 2017, <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>>.

- [TR38801] 3GPP, "3GPP TR 38.801, Technical Specification Group Radio Access Network; Study on new radio access technology: Radio access architecture and interfaces (Release 14)", 2017,  
<<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3056>>.
- [TS23401] 3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", 3GPP TS 23.401 10.10.0, March 2013.
- [TS25104] 3GPP, "Base Station (BS) radio transmission and reception (FDD)", 3GPP TS 25.104 3.14.0, March 2007.
- [TS36104] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception", 3GPP TS 36.104 10.11.0, July 2013.
- [TS36133] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management", 3GPP TS 36.133 12.7.0, April 2015.
- [TS36211] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation", 3GPP TS 36.211 10.7.0, March 2013.
- [TS36300] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2", 3GPP TS 36.300 10.11.0, September 2013.
- [TSNTG] IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networks Task Group", 2013,  
<<http://www.IEEE802.org/1/pages/avbridges.html>>.
- [WirelessHART]  
www.hartcomm.org, "Industrial Communication Networks - Wireless Communication Network and Communication Profiles - WirelessHART - IEC 62591", 2010.

#### Appendix A. Use Cases Explicitly Out of Scope for DetNet

This section contains use case text that has been determined to be outside of the scope of the present DetNet work.

### A.1. DetNet Scope Limitations

The scope of DetNet is deliberately limited to specific use cases that are consistent with the WG charter, subject to the interpretation of the WG. At the time the DetNet Use Cases were solicited and provided by the authors the scope of DetNet was not clearly defined, and as that clarity has emerged, certain of the use cases have been determined to be outside the scope of the present DetNet work. Such text has been moved into this section to clarify that these use cases will not be supported by the DetNet work.

The text in this section was moved here based on the following "exclusion" principles. Or, as an alternative to moving all such text to this section, some draft text has been modified in situ to reflect these same principles.

The following principles have been established to clarify the scope of the present DetNet work.

- o The scope of network addressed by DetNet is limited to networks that can be centrally controlled, i.e. an "enterprise" aka "corporate" network. This explicitly excludes "the open Internet".
- o Maintaining synchronized time across a DetNet network is crucial to its operation, however DetNet assumes that time is to be maintained using other means, for example (but not limited to) Precision Time Protocol ([IEEE1588]). A use case may state the accuracy and reliability that it expects from the DetNet network as part of a whole system, however it is understood that such timing properties are not guaranteed by DetNet itself. At the time of this writing it is an open question as to whether DetNet protocols will include a way for an application to communicate such timing expectations to the network, and if so whether they would be expected to materially affect the performance they would receive from the network as a result.

### A.2. Internet-based Applications

There are many applications that communicate over the open Internet that could benefit from guaranteed delivery and bounded latency. However as noted above, all such applications when run over the open Internet are out of scope for DetNet. These same applications may be in-scope when run in constrained environments, i.e. within a centrally controlled DetNet network. The following are some examples of such applications.

#### A.2.1. Use Case Description

##### A.2.1.1. Media Content Delivery

Media content delivery continues to be an important use of the Internet, yet users often experience poor quality audio and video due to the delay and jitter inherent in today's Internet.

##### A.2.1.2. Online Gaming

Online gaming is a significant part of the gaming market, however latency can degrade the end user experience. For example "First Person Shooter" games are highly delay-sensitive.

##### A.2.1.3. Virtual Reality

Virtual reality has many commercial applications including real estate presentations, remote medical procedures, and so on. Low latency is critical to interacting with the virtual world because perceptual delays can cause motion sickness.

#### A.2.2. Internet-Based Applications Today

Internet service today is by definition "best-effort", with no guarantees on delivery or bandwidth.

#### A.2.3. Internet-Based Applications Future

An Internet from which one can play a video without glitches and play games without lag.

For online gaming, the maximum round-trip delay can be 100ms and stricter for FPS gaming which can be 10-50ms. Transport delay is the dominate part with a 5-20ms budget.

For VR, 1-10ms maximum delay is needed and total network budget is 1-5ms if doing remote VR.

Flow identification can be used for gaming and VR, i.e. it can recognize a critical flow and provide appropriate latency bounds.

#### A.2.4. Internet-Based Applications Asks

- o Unified control and management protocols to handle time-critical data flow
- o Application-aware flow filtering mechanism to recognize the timing critical flow without doing 5-tuple matching

- o Unified control plane to provide low latency service on Layer-3 without changing the data plane
- o OAM system and protocols which can help to provide E2E-delay sensitive service provisioning

#### A.3. Pro Audio and Video - Digital Rights Management (DRM)

This section was moved here because this is considered a Link layer topic, not direct responsibility of DetNet.

Digital Rights Management (DRM) is very important to the audio and video industries. Any time protected content is introduced into a network there are DRM concerns that must be maintained (see [CONTENT\_PROTECTION]). Many aspects of DRM are outside the scope of network technology, however there are cases when a secure link supporting authentication and encryption is required by content owners to carry their audio or video content when it is outside their own secure environment (for example see [DCI]).

As an example, two techniques are Digital Transmission Content Protection (DTCP) and High-Bandwidth Digital Content Protection (HDCP). HDCP content is not approved for retransmission within any other type of DRM, while DTCP may be retransmitted under HDCP. Therefore if the source of a stream is outside of the network and it uses HDCP protection it is only allowed to be placed on the network with that same HDCP protection.

#### A.4. Pro Audio and Video - Link Aggregation

Note: The term "Link Aggregation" is used here as defined by the text in the following paragraph, i.e. not following a more common Network Industry definition.

For transmitting streams that require more bandwidth than a single link in the target network can support, link aggregation is a technique for combining (aggregating) the bandwidth available on multiple physical links to create a single logical link of the required bandwidth. However, if aggregation is to be used, the network controller (or equivalent) must be able to determine the maximum latency of any path through the aggregate link.

#### A.5. Pro Audio and Video - Deterministic Time to Establish Streaming

The DetNet Working Group has decided that guidelines for establishing a deterministic time to establish stream startup are not within scope of DetNet. If bounded timing of establishing or re-establish streams

is required in a given use case, it is up to the application/system to achieve this.

Author's Address

Ethan Grossman (editor)  
Dolby Laboratories, Inc.  
1275 Market Street  
San Francisco, CA 94103  
USA

Phone: +1 415 645 4726  
Email: [ethan.grossman@dolby.com](mailto:ethan.grossman@dolby.com)  
URI: <http://www.dolby.com>



Network Working Group  
Internet-Draft  
Intended status: Informational

Y. Jiang  
N. Finn  
Huawei  
J. Ryoo  
ETRI  
B. Varga  
Ericsson  
L. Geng  
China Mobile  
January 24, 2018

Expires: July 2018

Deterministic Networking Application in Ring Topologies  
draft-jiang-detnet-ring-00

Abstract

Deterministic Networking (DetNet) provides a capability to carry data flows for real-time applications with extremely low data loss rates and bounded latency. This document describes how DetNet can be used in ring topologies to support Point-to-Point (P2P) and Point-to-Multipoint (P2MP) real-time services.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on July 24, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1.	Introduction .....	3
1.1.	Conventions used in this document .....	4
1.2.	Terminology .....	4
2.	P2P DetNet Ring .....	4
2.1.	DetNet applications on a single ring for P2P traffic ....	4
2.2.	Implementation implications of a DetNet ring for P2P traffic .....	5
3.	P2MP DetNet Ring .....	5
3.1.	DetNet applications on a single ring for P2MP traffic ...	5
3.2.	Section LSPs as underlay (Service layer replication) ....	6
3.3.	P2MP LSP tunnels as underlay (LSP layer replication) ....	7
4.	DetNet Ring Interconnections .....	8
4.1.	Single node interconnection .....	8
4.1.1.	DetNet relay node as interconnection node .....	9
4.1.2.	Elimination first approach .....	9
4.2.	Dual node interconnection .....	10
4.2.1.	Dual node interconnection for P2P traffic .....	10
4.2.2.	Elimination first approach in dual node interconnection for P2P traffic .....	11
4.2.3.	Dual node interconnection for P2MP traffic using section LSP .....	11
4.2.4.	Elimination first approach in dual node interconnection for P2MP traffic using section LSP .....	12
4.2.5.	Dual node interconnection for P2MP traffic using P2MP LSP	13
5.	Resource reservation .....	13
6.	Security Considerations .....	13
7.	IANA Considerations .....	13

8.	References .....	13
1.1.	Informative References .....	13
9.	Acknowledgments .....	15

## 1. Introduction

An overview of Deterministic Networking (DetNet) architecture is given in [I-D.ietf-detnet-architecture], and DetNet data plane encapsulations are specified in [I-D.ietf-detnet-dp-sol]. But there is not any discussion on a ring topology in [I-D.ietf-detnet-architecture] yet. Furthermore, [I-D.ietf-detnet-use-cases] outlines several Detnet use cases where multicast capability is needed. If a multicast service replicates all of its packets from the source (as a traditional Virtual Private LAN Service (VPLS) does), the requirements of deterministic delay and high availability for all these replicated packets will pose a great challenge to the Detnet network.

In fact, ring topologies have been very popular and widely deployed in network arrangements for various transport networks, such as Synchronous Digital Hierarchy, Synchronous Optical Network, Optical Transport Network, and Ethernet. The IETF has done some work on ring protection in Multi-Protocol Label Switching - Transport Profile (MPLS-TP), such as [RFC6974] and [RFC8227]. All these works, except Ethernet ring protection, typically use swapping or steering as the protection mechanism. As ring topologies are widely deployed for transport networks, it is also necessary for DetNet to support ring topologies (currently, there is not any discussion on a ring topology in [I-D.ietf-detnet-architecture] yet).

This draft demonstrates how DetNet can be used in a ring topology. Specifically, DetNet ring supports for Point-to-Point (P2P) and Point-to-Multipoint (P2MP, for multicast services) are discussed in details. This document assumes that MPLS encapsulation for DetNet is supported as specified in [I-D.ietf-detnet-dp-sol] and all nodes in a ring network can support the Multi-Protocol Label Switching (MPLS) functionalities. It should be noted that it is more convenient for DetNet to support a ring topology with the intrinsic duplication and elimination mechanism, as there is no need of swapping or steering operations (consequently, Operations, Administration and Maintenance is not needed either for its working) for any service protection.

### 1.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

### 1.2. Terminology

DetNet    Deterministic Networking

LSP      Label Switched Path

MPLS     Multi-Protocol Label Switching

MPLS-TP Multi-Protocol Label Switching - Transport Profile

P2MP     Point-to-Point

P2P      Point-to-Multipoint

PW       Pseudowire

## 2. P2P DetNet Ring

### 2.1. DetNet applications on a single ring for P2P traffic

Figure 1 depicts an example of the DetNet ring for P2P real time traffic. Nodes A and C are DetNet aware devices, and P2P DetNet traffic is transported from node A to node C.

A clockwise and a counter clockwise Pseudowire (PW) and Label Switched Path (LSP) tunnel are configured from node A to node C respectively. The DetNet traffic is replicated on node A, encapsulated with the specific PW and LSP labels, and transported on both LSP paths towards node C. Upon reception of the traffic, node C terminates the LSP and is aware of the DetNet traffic by inspection of the PW label carried in each packet. An elimination function in node C guarantees that only one copy of the DetNet service exits on egress with the help of the DetNet sequence number.

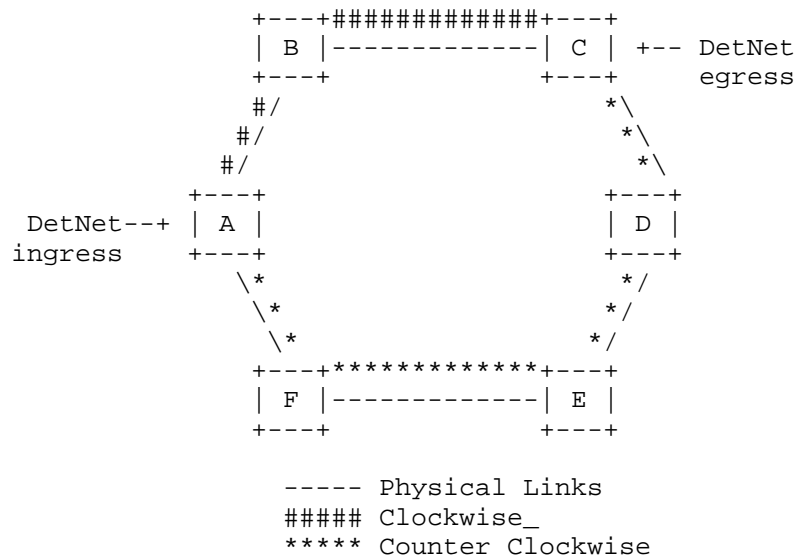


Figure 1: DetNet Ring for P2P traffic

## 2.2. Implementation implications of a DetNet ring for P2P traffic

In a DetNet ring for P2P traffic, one path may be far longer than the other path for the DetNet (this is a DetNet issue more general than a ring).

The buffer need to be large enough to accommodate for the sequence number difference between these two paths. Otherwise, some packets may get lost when a link fault causes traffic switching from a path to another path.

## 3. P2MP DetNet Ring

### 3.1. DetNet applications on a single ring for P2MP traffic

Figure 2 further depicts an example of the DetNet ring for P2MP real time traffic. Nodes A, B, C, E and F are DetNet aware devices, and P2MP DetNet traffic is transported from head-end node A to multiple tail-end nodes C, E and F.

Two approaches are described in Section 3.2 and 3.3 for P2MP traffic.

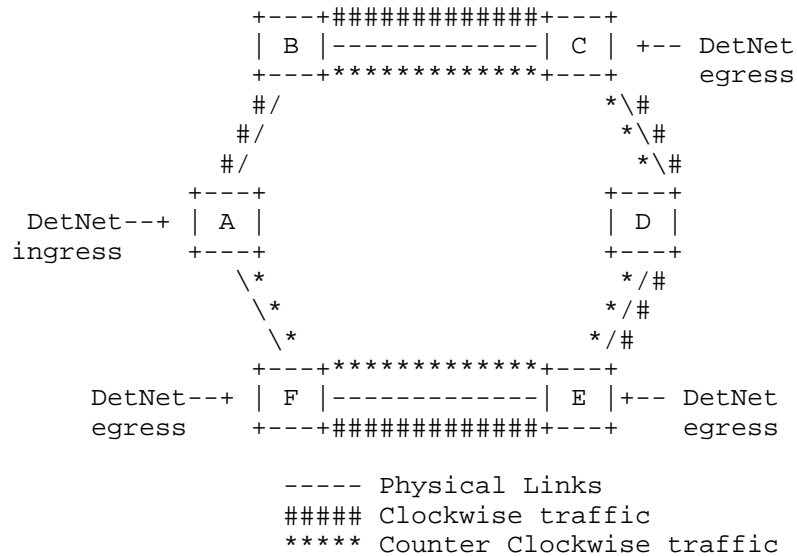


Figure 2: DetNet Ring for P2MP traffic

### 3.2. Section LSPs as underlay (Service layer replication)

If section LSPs are used as an underlay for DetNet services, a bidirectional section LSP tunnel is set up between each pair of neighboring nodes in the ring (e.g., node A and node B, ..., node F and node A). In this case, DetNet PW layer replicates the DetNet packets from one tail-end to another neighboring tail-end.

The DetNet head-end (i.e., node A) in the ring needs to support DetNet replication function. Upon reception on node A, the DetNet traffic is replicated in node A, encapsulated with the specific PW and section LSP labels, and then transported on both section LSPs (i.e., A-B and A-F) originated from the head-end.

All intermediate nodes (non tail-ends) on the ring SHOULD transparently forward the DetNet traffic with a specific PW to the next hop on the ring in the same direction.

All DetNet tail-ends except the penultimate node (egress nodes such as nodes C and E in the clockwise, and node F, E and C in the counter clockwise) on the ring MUST support both DetNet

replication and elimination functions. For example, upon reception of the clockwise traffic, node C terminates the section LSP and is aware of the DetNet traffic by inspection of the PW label in the packet. Firstly, node C needs to transparently forward the DetNet traffic with a specific PW to the next hop on the ring in the same direction. Secondly, DetNet traffic is directed to a DetNet elimination function associated with a specific PW, only one copy of the DetNet service exits on egress by inspection of the DetNet sequence number.

If multiple endpoints are attached to a tail-end node, a multicast module can be used to forward the filtered DetNet traffic to all these endpoints.

To avoid a loop of DetNet service, the penultimate node in the ring (such as node B on the counter clock-wise LSP) needs to terminate the DetNet flow. For example, upon reception of the clockwise DetNet traffic, node F terminates the DetNet traffic by inspection of the PW label in the packet. As an alternative, the last DetNet tail-end (such as node C on the counter clock-wise LSP) may terminate the DetNet flow, so that the bandwidth from this node to the penultimate node can be saved.

### 3.3. P2MP LSP tunnels as underlay (LSP layer replication)

If P2MP LSPs are used as an underlay for the DetNet service, a P2MP unidirectional LSP tunnel in clockwise is set up from head-end (ingress node A) to all the tail-ends (egress nodes C, E and F) for the ring, and another P2MP unidirectional LSP tunnel in counter clockwise is set up from head-end (ingress node A) to all the tail-ends (egress nodes F, E and C) for the ring. Thus, LSP layer replicates the DetNet packets from one tail-end to another neighboring tail-end.

The DetNet head-end (i.e., node A) in the ring needs to support DetNet replication function. Upon reception on node A, the DetNet traffic is replicated, encapsulated with the specific PW and P2MP LSP labels, and transported on both P2MP LSP tunnels in the ring.

All DetNet tail-ends (egress nodes such as node C, E and F in Figure 2) on the ring need to support the DetNet elimination function. For example, upon reception of the traffic, node C pops the P2MP LSP label and is aware of the DetNet traffic by inspection of the PW label in the label stack. Traffic from both directions with the same PW is directed to the same DetNet elimination

function so that only one copy of the DetNet service exits on egress by inspection of the DetNet sequence number.

If multiple endpoints are attached to a tail-end node, a multicast module can be used to forward the filtered DetNet traffic to all these endpoints.

#### 4. DetNet Ring Interconnections

Two DetNet rings can be connected via one or more interconnection nodes. Figures 3a and 3b show ring interconnection scenarios with a single node and dual nodes, respectively. In the interconnected rings, each ring operates in the same way as described in Sections 2 and 3 except the nodes that are used to interconnect two rings.

In this section, we describe the behavior of interconnection nodes with the traffic going from Ring L to Ring R. Symmetrical description is assumed for the traffic in the other direction.

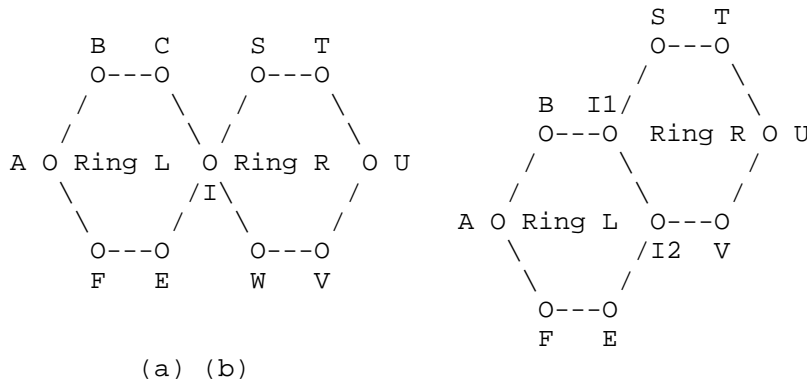


Figure 3: DetNet ring interconnection with: a) single node (node I), and b) dual nodes (nodes I1 and I2).

##### 4.1. Single node interconnection

In the case of the single node interconnection, as shown in Figure 3(a), both P2P and P2MP DetNet traffic that needs to be transported between Ring L and Ring R uses the single interconnection node between two rings. Two approaches are described in the following subsections.



#### 4.1.1.1. DetNet relay node as interconnection node

In this approach, the interconnection node acts as a DetNet relay node, which provides packet replication and elimination.

For P2P DetNet traffic going from Ring L to Ring R, interconnection node I performs packet replication on input and sends the packet to the outputs connected to the links on Ring R clockwise and counter-clockwise. Then, after each output of interconnection node I eliminates any duplicates, the packet is transported over Ring R. In Figure 3(a), when interconnection node I receives traffic on input from node C, node I replicates the traffic and send it to both outputs to nodes S and W. For the traffic from input from node E, node I also replicates the traffic and send it to both outputs to nodes S and W. Then, the output to node S eliminates any duplicates, and sends only one copy to node S. Similarly, the output to node W eliminates any duplicates, and sends only one copy to node W.

For P2MP DetNet traffic going from Ring L to Ring R, the input of interconnection node I performs the same packet replication as described for P2P DetNet traffic going from Ring L to Ring R. In addition, the third copy is sent to the other ring port on Ring L, in order to deliver the P2MP DetNet traffic to the remaining tail-end nodes that reside in the other side of Ring L over the interconnected node. The outputs to nodes S and W perform the same duplicate elimination as described for P2P DetNet traffic going from Ring L to Ring R.

#### 4.1.1.2. Elimination first approach

This approach uses two "logical" DetNet relay nodes (or, DA-\*-PE as described in [I-D.ietf-detnet-dp-sol]) coupled back-to-back, such that interconnection node I performs the duplicate elimination function first.

For the Detnet traffic arrived from both node C and node E, the interconnection node I performs duplicate elimination first, and then replicates the traffic in both clockwise and counter-clockwise directions of Ring R, i.e., one copy to node S and the other copy to node W. Therefore, this approach reduces the bandwidth used inside the interconnection node when there is a central unit that eliminates any duplicate among the packets arrived from two ring ports before replication.

#### 4.2. Dual node interconnection

In order to prevent a single point of failure, two interconnection nodes can be used as shown in Figure 3(b). To provide high availability for DetNet services, dual node interconnection is recommended. Two interconnection nodes act as DetNet relay nodes, which provide packet replication and elimination.

##### 4.2.1. Dual node interconnection for P2P traffic

For the P2P DetNet traffic that flows from Ring L to Ring R, the operation of interconnection nodes I1 and I2 follows the description on relay nodes shown in Figure 1 of Section 3.2.4 in [I-D.ietf-detnet-architecture]. In the following, the operation is explained with Figure 3(a).

When interconnection node I1 receives clockwise traffic from node B, it replicates the traffic and sends one copy to interconnection node I2 and the other copy to output towards node S.

When interconnection node I1 receives counter-clockwise traffic from interconnection node I2, it forwards the traffic to the output that is connected to node S.

At the output of interconnection node I1 facing to node S, duplicate elimination is performed for the clockwise traffic from node B and the counter-clockwise traffic from interconnection node I2, and only one copy is sent to the clockwise direction of Ring R (i.e., sent towards node S).

When interconnection node I2 receives counter-clockwise traffic from node E, it replicates the traffic and sends one copy to interconnection node I1 and the other copy to the output that is connected to node V.

When interconnection node I2 receives clockwise traffic from interconnection node I1, it forwards the traffic to the output that is connected to node V.

At the output of interconnection node I2 facing to node V, duplicate elimination is performed for the counter-clockwise traffic from node E and the clockwise traffic from interconnection node I1, and only one copy is sent to the counter-clockwise direction of Ring R (i.e., sent towards node V).

#### 4.2.2. Elimination first approach in dual node interconnection for P2P traffic

The elimination first approach described in Section 4.1.2 can also be used for dual node interconnection, so that each interconnection node performs the duplicate elimination function first.

For the traffic arrived from both node B and interconnection node I2, the interconnection node I1 performs duplicate elimination first, and replicates the traffic in both clockwise and counter-clockwise directions of Ring R, i.e., one copy to node S and the other copy to interconnection node I2.

For the traffic arrived from both node E and interconnection node I1, the interconnection node I2 performs duplicate elimination first, and replicates the traffic in both clockwise and counter-clockwise directions of Ring R, i.e., one copy to interconnection node I1 and the other copy to node V.

#### 4.2.3. Dual node interconnection for P2MP traffic using section LSP

For the P2MP traffic that flows from Ring L to Ring R, each ring is configured and operated as described in Section 3.2 except the interconnection nodes, whose operations are described below.

When interconnection node I1 receives clockwise traffic from node B, it replicates the traffic and sends one copy to interconnection node I2 and the other copy to the output that is connected to node S.

When interconnection node I1 receives the counter-clockwise traffic from interconnection node I2, it replicates the traffic and sends one copy to node B and the other copy to the output that is connected to node S unless interconnection node I1 is the penultimate node for the counter-clockwise traffic on Ring L. In the case that interconnection node I1 is the penultimate node for the counter-clockwise traffic on Ring L, the counter-clockwise traffic from interconnection node I2 is forwarded to the output that is connected to node S.

At the output interface of I1 facing to node S, duplicate elimination is performed for the clockwise traffic from node B and the counter-clockwise traffic from interconnection node I2, and only one copy is sent to the clockwise direction of Ring R (i.e., sent towards node S).

When interconnection node I2 receives the counter-clockwise traffic from node E, it replicates the traffic and sends one copy to interconnection node I1 and the other copy to the output that is connected to node V.

When interconnection node I2 receives the clockwise traffic from interconnection node I1, it replicates the traffic and sends one copy to node E and the other copy to the output that is connected to node V unless interconnection node I2 is the penultimate node for the clockwise traffic in Ring L. In the case that interconnection node I2 is the penultimate node for the clockwise traffic in Ring L, the clockwise traffic from interconnection node I1 is forwarded to the output that is connected to node V.

At the output interface of I2 facing to node V, duplicate elimination is performed for the counter-clockwise traffic from node E and the clockwise traffic from interconnection node I1, and only one copy is sent to the counter-clockwise direction of Ring R (i.e., sent towards node V).

#### 4.2.4. Elimination first approach in dual node interconnection for P2MP traffic using section LSP

The elimination first approach described in Section 4.2.2 is applied without modification for dual node interconnection for P2MP traffic using section LSP only if interconnection nodes I1 and I2 are the penultimate nodes for the counter-clockwise traffic and the clockwise traffic on Ring L, respectively.

When an interconnection node is not the penultimate node for either clockwise or counter-clockwise traffic, the interconnection node replicates the traffic in three ways; one for the remaining tail-ends on Ring L and two for the tail-ends in both clockwise and counter-clockwise directions on Ring R.

For example, assume that interconnection node I2 is not the penultimate node for the clock-wise traffic on Ring L. For the traffic arrived from both node E and interconnection node I1, interconnection node I2 performs duplicate elimination first, and replicates the traffic for the following three outputs; one copy to the output towards node E, another copy to the output towards interconnection node I1, and the other copy to the output towards node V.

#### 4.2.5. Dual node interconnection for P2MP traffic using P2MP LSP

If P2MP LSPs are used in the interconnected rings, two P2MP unidirectional LSP tunnels are used on each ring for the clockwise and counter-clockwise directions.

When the P2MP traffic is forwarded from one ring to another ring, for example from Ring L to Ring R in Figure 3(b), each P2MP LSP in Ring L MUST include interconnection nodes I1 and I2 as tail-ends. For Ring R, one P2MP LSP is set up from interconnection node I1 to all the tail-ends in the clockwise direction on Ring R, and the other P2MP LSP is set up from interconnection node I2 to all the tail-ends in the counter-clockwise direction on Ring R. Therefore, an interconnection node acts as a tail-end for one ring and a head-end for another ring in one direction, and performs the same operation of tail-end and head-end as specified in Section 3.3.

#### 5. Resource reservation

In order to guarantee that DetNet flows don't suffer from network congestion, resource reservation considerations as outlined in Section 4.3.2 of [I-D.ietf-detnet-architecture] apply here.

#### 6. Security Considerations

This document describes the application of DetNet on general ring topologies. Thus the security considerations as described in [I-D.ietf-detnet-dp-sol] also apply to this document.

#### 7. IANA Considerations

There are no IANA actions required by this document.

#### 8. References

##### 1.1. Informative References

[I-D.ietf-detnet-architecture] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture (work in progress), October 2017

- [I-D.ietf-detnet-dp-sol] Korhonen, J., Andersson, L., Jiang, Y., and etc., "DetNet Data Plane Encapsulation", draft-ietf-detnet-dp-sol (work in progress), October, 2017
- [I-D.ietf-detnet-use-cases] Grossman, E., and etc., "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases (work in progress), October, 2017
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997
- [RFC6974] Weingarten, Y., Bryant, S., and etc., "Applicability of MPLS Transport Profile for Ring Topologies", RFC 6974, July 2013
- [RFC8227] Cheng, W., Wang, L., and etc., "MPLS-TP Shared-Ring Protection (MSRP) Mechanism for Ring Topology", RFC 8227, August 2017

## 9. Acknowledgments

TBD

## Authors' Addresses

Yuanlong Jiang  
Huawei Technologies Co., Ltd.  
Bantian, Longgang district  
Shenzhen 518129, China  
Phone: +86-18926415311  
Email: jiangyuanlong@huawei.com

Norman Finn  
Huawei Technologies Co. Ltd  
3755 Avocado Blvd,  
California 91941, USA  
Phone: +1 925 980 6430  
Email: norman.finn@mail01.huawei.com

Jeong-dong Ryoo  
ETRI  
218 Gajeongno  
Yuseong-gu, Daejeon 305-700, South Korea  
Phone: +82-42-860-5384  
Email: ryoo@etri.re.kr

Balazs Varga  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary  
Email: balazs.a.varga@ericsson.com

Liang Geng  
China Mobile  
Beijing, China  
Email: gengliang@chinamobile.com





DetNet Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 14, 2021

Y. Jiang  
N. Finn  
Huawei Technologies  
J. Ryoo  
ETRI  
B. Varga  
Ericsson  
L. Geng  
China Mobile  
July 13, 2020

Deterministic Networking Application in Ring Topologies  
draft-jiang-detnet-ring-06

Abstract

Deterministic Networking (DetNet) provides a capability to carry data flows for real-time applications with extremely low data loss rates and bounded latency. This document describes how DetNet can be used in ring topologies to support Point-to-Point (P2P) and Point-to-Multipoint (P2MP) real-time services.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 14, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions Used in This Document . . . . .	3
3. Abbreviations . . . . .	3
4. P2P DetNet Ring . . . . .	4
4.1. DetNet applications on a single ring for P2P traffic . .	4
4.2. Implementation implications of a DetNet ring for P2P traffic . . . . .	5
5. P2MP DetNet Ring . . . . .	5
5.1. DetNet applications on a single ring for P2MP traffic . .	5
5.2. Section LSPs as underlay (service sub-layer replication)	6
5.3. P2MP LSP tunnels as underlay (forwarding sub-layer replication) . . . . .	7
6. DetNet Ring Interconnections . . . . .	8
6.1. Single node interconnection . . . . .	8
6.2. Dual node interconnection . . . . .	9
6.2.1. Dual node interconnection for P2P traffic . . . . .	9
6.2.2. Dual node interconnection for P2MP traffic using section LSP . . . . .	10
6.2.3. Dual node interconnection for P2MP traffic using P2MP LSP . . . . .	11
7. Resource Reservation . . . . .	11
8. IANA Considerations . . . . .	11
9. Security Considerations . . . . .	11
10. Editor's Note . . . . .	12
11. References . . . . .	12
11.1. Normative References . . . . .	12
11.2. Informative References . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

The overall architecture for Deterministic Networking (DetNet), which provides a capability to carry specified unicast or multicast data flows for real-time applications with extremely low data loss rates and bounded latency, is specified in [RFC8655], and the generic data plane framework, which is common to any DetNet data plane implementations, is provided at [I-D.ietf-detnet-data-plane-framework]. In addition to the DetNet architecture documents, RFC 8578 [RFC8578] outlines several DetNet

use cases where multicast capability is needed. If a multicast service replicates all of its packets from the source (as a traditional Virtual Private LAN Service (VPLS) does), the requirements of deterministic delay and high availability for all these replicated packets will pose a great challenge to the DetNet network.

Ring topologies have been very popular and widely deployed in network arrangements for various transport networks, such as Synchronous Digital Hierarchy, Synchronous Optical Network, Optical Transport Network, and Ethernet. For Multi-Protocol Label Switching - Transport Profile (MPLS-TP), the applicability of the MPLS-TP linear protection [RFC6378][RFC7271] for ring topologies and the ring-specific protection mechanism are specified in RFC 6974 [RFC6974] and RFC 8227 [RFC8227], respectively. All these works, except Ethernet ring protection, typically use swapping or steering as the protection mechanism. As ring topologies are widely deployed for transport networks, it is also necessary for the DetNet to support ring topologies.

This document demonstrates how the DetNet can be used in a ring topology. Specifically, DetNet ring supports for Point-to-Point (P2P) and Point-to-Multipoint (P2MP, for multicast services) are discussed in details. This document assumes that the Multi-Protocol Label Switching (MPLS) encapsulation for DetNet is supported as specified in [I-D.ietf-detnet-mpls] and all nodes in a ring network can support the MPLS functionalities. It should be noted that it is more convenient for the DetNet to support a ring topology with the intrinsic duplication and elimination mechanism, as there is no need of swapping or steering operations (consequently, its Operations, Administration and Maintenance (OAM) can also be simplified) for service protection.

## 2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Abbreviations

This document uses the following abbreviations:

- DetNet
- Deterministic Networking
- LSP
- Label Switched Path
- MPLS
- Multi-Protocol Label Switching
- MPLS-TP
- Multi-Protocol Label Switching - Transport Profile
- P2MP
- Point-to-Multipoint
- P2P
- Point-to-Point
- PEF
- Packet Elimination Function
- POF
- Packet Ordering Function
- PRF
- Packet Replication Function
- PW
- Pseudowire

4. P2P DetNet Ring

This section describes how the DetNet can deliver P2P traffic over a single ring.

4.1. DetNet applications on a single ring for P2P traffic

Figure 1 shows an example of the DetNet ring for P2P real time traffic. Nodes A and C are DetNet aware devices, and P2P DetNet traffic is transported from node A to node C.

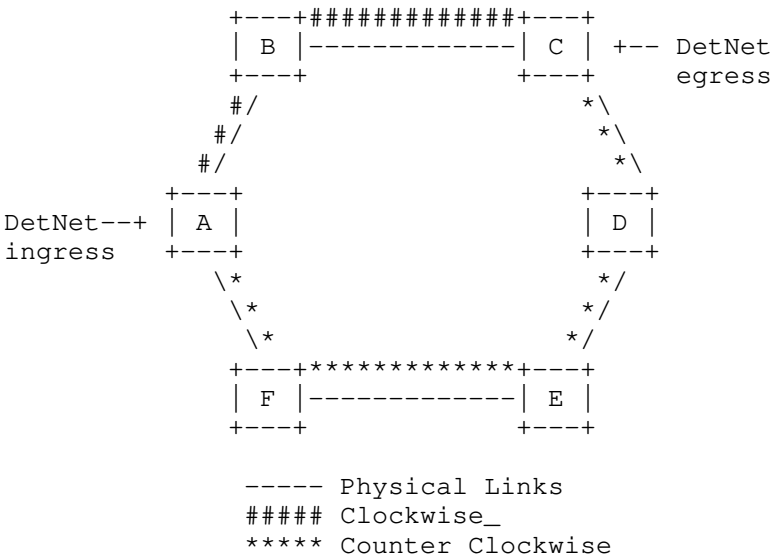


Figure 1: DetNet Ring for P2P traffic

A clockwise and a counter clockwise Label Switched Paths (LSPs) are configured from node A to node C using the DetNet forwarding labels

(F-Labels) are configured from node A to node C. The DetNet service sub-layer functions are provided at nodes A and C utilizing the DetNet service label(s) (S-Label) and DetNet control word (d-CW) as described in [I-D.ietf-detnet-mpls]. The P2P traffic is replicated by a Packet Replication Function (PRF) in node A, encapsulated with the d-CW and specific S-Label and F-Label(s), and transported on both LSP paths towards node C. Upon reception of the traffic, node C terminates the LSP and is aware of the DetNet traffic by inspection of the S-Label carried in each packet. A Packet Elimination Function (PEF) in node C guarantees that only one copy of the DetNet service exits on egress with the help of the DetNet sequence number. A Packet Ordering Function (POF) can further reorder packets in node C before transport of these packets to the destination.

#### 4.2. Implementation implications of a DetNet ring for P2P traffic

In a DetNet ring for P2P traffic, one path may be far longer than the other path. The buffer for reordering at the egress needs to be large enough to accommodate for the sequence number difference between these two paths.

### 5. P2MP DetNet Ring

#### 5.1. DetNet applications on a single ring for P2MP traffic

Figure 2 shows an example of the DetNet ring for P2MP real time traffic. Nodes A, B, C, E and F are DetNet aware devices, and P2MP DetNet traffic is transported from head-end node A to multiple tail-end nodes C, E and F.

Two approaches are described in Section 5.2 and Section 5.3 for P2MP traffic.

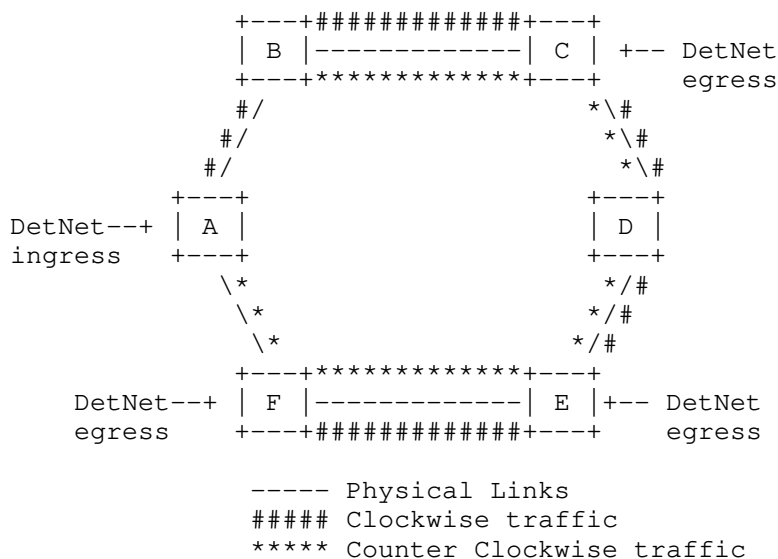


Figure 2: DetNet Ring for P2MP traffic

## 5.2. Section LSPs as underlay (service sub-layer replication)

If section LSPs are used as an underlay for DetNet services, a bidirectional section LSP tunnel is set up between each pair of neighboring nodes in the ring (e.g., node A and node B, ..., node F and node A). In this case, the DetNet sub-layer replicates the DetNet packets from one tail-end to another neighboring tail-end.

The DetNet head-end (i.e., node A) in the ring needs to support DetNet replication function. Upon reception on node A, the DetNet traffic is replicated with a d-CW, encapsulated with a S-Label and a section LSP label per DetNet member flow, and transported on both section LSPs (i.e., A-B and A-F).

All intermediate nodes (non tail-ends) on the ring MUST transparently forward the DetNet packet, which contains a d-CW and S-Label, to the next hop on the ring.

All DetNet tail-ends except the penultimate node (egress nodes such as nodes C and E in the clockwise, and nodes F, E and C in the counter clockwise) on the ring MUST support both DetNet PRF and PEF functions, and MAY further support a DetNet POF function. For the example of Figure 2, upon reception of the clockwise traffic, node C terminates the section LSP and recognizes the DetNet flow by inspection of the S-label in the packet. Firstly, node C needs to forward the DetNet packet to the next hop on the ring in the

clockwise direction. Secondly, the DetNet packet is also directed to a DetNet PEF associated with the DetNet flow, only one copy is egressed from the ring by inspection of the sequence number in the d-CW. Furthermore, if the DetNet POF function is enabled, the packets in the DetNet flow are reordered before exit to DetNet egress.

If multiple endpoints are attached to a tail-end node, a multicast module can be used to forward the traffic to all these endpoints.

To avoid a loop of DetNet service, the penultimate node in the ring (such as node B on the counter clock-wise LSP) MUST terminate the DetNet flow. For example, upon reception of the clockwise DetNet traffic, node F terminates the DetNet traffic by inspection of the S-Label in the packet. As an alternative, the last DetNet tail-end (such as node C on the counter clock-wise LSP) MAY terminate the DetNet flow, so that the bandwidth from this node to the penultimate node can be saved.

### 5.3. P2MP LSP tunnels as underlay (forwarding sub-layer replication)

If P2MP LSPs are used as an underlay for the DetNet service, a P2MP unidirectional LSP tunnel in clockwise is set up from head-end (ingress node A) to all the tail-ends (egress nodes C, E and F) for the ring, and another P2MP unidirectional LSP tunnel in counter clockwise is set up from head-end (ingress node A) to all the tail-ends (egress nodes F, E and C) for the ring. Thus, a PRF in LSP layer replicates the DetNet packets from one tail-end to another neighboring tail-end.

The DetNet head-end (i.e., node A) in the ring needs to support the DetNet PRF function. Upon reception on node A, the DetNet traffic is replicated with a d-CW, encapsulated with a S-Label per DetNet member flow, and transported on both P2MP LSP tunnels in the ring.

All DetNet tail-ends (egress nodes such as nodes C, E and F in Figure 2) on the ring need to support the DetNet PEF function. For example, upon reception of the traffic, node C pops the P2MP LSP label and is aware of the DetNet traffic by inspection of the S-Label label in the label stack. Two DetNet member flows are identified with their S-Labels and directed to the same PEF so that only one copy of the DetNet service is selected by inspection of the DetNet sequence number in the d-CW. Furthermore, if DetNet POF function is enabled, the packets in the DetNet flow are reordered before exit to DetNet egress.

If multiple endpoints are attached to a tail-end node, a multicast module can be used to forward the filtered DetNet traffic to all these endpoints

## 6. DetNet Ring Interconnections

Two DetNet rings can be connected via one or more interconnection nodes. Figure 3 shows the ring interconnection scenarios with a single node and dual nodes. In the interconnected rings, each ring operates in the same way as described in Section 4 and Section 5 except the node or nodes that are used to interconnect two rings.

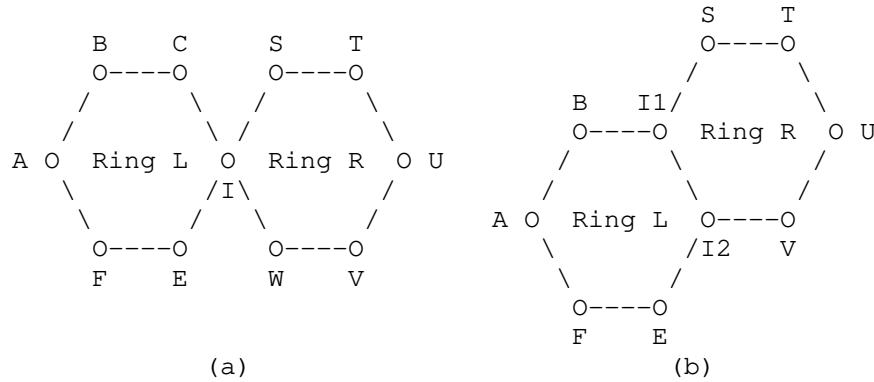


Figure 3: DetNet ring interconnection with: (a) single node (node I), and (b) dual nodes (nodes I1 and I2)

In this section, we describe the behavior of interconnection nodes with the traffic going from Ring L to Ring R. Symmetrical description is assumed for the traffic in the other direction (i.e., from Ring R to Ring L).

### 6.1. Single node interconnection

In the case of the single node interconnection, as shown in Figure 3(a), both P2P and P2MP DetNet traffic that needs to be transported between Ring L and Ring R use a single interconnection node between two rings. Thus, the interconnection node acts as a DetNet relay node, which provides both PRF and PEF functions.

For P2P DetNet traffic going from Ring L to Ring R, interconnection node I receives the same DetNet flow traffic from both node C and node E (i.e., clockwise and counter-clockwise), a PEF in node I performs packet elimination, and a PRF in node I replicates the packet, node I then sends one copy to node S and another copy to node W.



For P2MP DetNet traffic going from Ring L to Ring R, interconnection node I performs the same packet elimination and replication functions as described above. In addition, node I further transparently forwards the P2MP DetNet traffic on Ring L in the same direction if it is not the last tail-end node.

## 6.2. Dual node interconnection

In order to prevent a single point of failure, two interconnection nodes can be used as shown in Figure 3(b). To provide high availability for DetNet services, dual node interconnection is recommended. Two interconnection nodes act as DetNet relay nodes, each provides both packet replication and elimination functions.

### 6.2.1. Dual node interconnection for P2P traffic

For the P2P DetNet traffic that flows from Ring L to Ring R in Figure 3(b), the operations of interconnection nodes I1 and I2 are described below.

When interconnection node I1 receives clockwise traffic from node B, it replicates the traffic and sends one copy to interconnection node I2 and the other copy to a PEF in interconnection node I1.

When interconnection node I1 receives counter-clockwise traffic from interconnection node I2, it forwards the traffic to the PEF of interconnection node I1.

At the PEF of interconnection node I1, duplicate elimination is performed for the clockwise traffic from node B and the counter-clockwise traffic from interconnection node I2, and only one copy is sent to the clockwise direction of Ring R (i.e., sent towards node S). Furthermore, if DetNet POF function is enabled on interconnection node I1, the packets in the DetNet flow are reordered before being forwarded to Ring R.

When interconnection node I2 receives counter-clockwise traffic from node E, it replicates the traffic and sends one copy to interconnection node I1 and the other copy to a PEF in interconnection node I2.

When interconnection node I2 receives clockwise traffic from interconnection node I1, it forwards the traffic to the PEF of interconnection node I2.

At the PEF of interconnection node I2, duplicate elimination is performed for the counter-clockwise traffic from node E and the clockwise traffic from interconnection node I1, and only one copy is

sent to the counter-clockwise direction of Ring R (i.e., sent towards node V). Furthermore, if DetNet POF function is enabled on interconnection node I2, the packets in the DetNet flow are reordered before being forwarded to Ring R.

#### 6.2.2. Dual node interconnection for P2MP traffic using section LSP

For the P2MP traffic that flows from Ring L to Ring R in Figure 3(b), each ring is configured and operated as described in Section 5.2 except the interconnection nodes, whose operations are described below.

When interconnection node I1 receives clockwise traffic from node B, its PRF replicates the traffic and sends one copy to interconnection node I2 and the other copy to interconnection node I1's PEF.

When interconnection node I1 receives the counter-clockwise traffic from interconnection node I2, its PRF replicates the traffic and sends one copy to node B and the other copy to interconnection node I1's PEF unless interconnection node I1 is the penultimate node for the counter-clockwise traffic on Ring L. In the case that interconnection node I1 is the penultimate node for the counter-clockwise traffic on Ring L, the counter-clockwise traffic from interconnection node I2 is only forwarded to interconnection node I1's PEF.

At interconnection node I1's PEF, duplicate elimination is performed for the clockwise traffic from node B and the counter-clockwise traffic from interconnection node I2, and only one copy is sent to the clockwise direction of Ring R (i.e., sent towards node S). Furthermore, if DetNet POF function is enabled on node I1, the packets in the DetNet flow are reordered before being forwarded to Ring R.

When interconnection node I2 receives the counter-clockwise traffic from node E, its PRF replicates the traffic and sends one copy to interconnection node I1 and the other copy to node I2's PEF.

When interconnection node I2 receives the clockwise traffic from interconnection node I1, its PRF replicates the traffic and sends one copy to node E and the other copy to interconnection node I2's PEF unless interconnection node I2 is the penultimate node for the clockwise traffic on Ring L. In the case that interconnection node I2 is the penultimate node for the clockwise traffic on Ring L, the clockwise traffic from interconnection node I1 is only forwarded to node I2's PEF.

At node I2's PEF, duplicate elimination is performed for the counter-clockwise traffic from node E and the clockwise traffic from interconnection node I1, and only one copy is sent to the counter-clockwise direction of Ring R (i.e., sent towards node V). Furthermore, if DetNet POF function is enabled on interconnection node I2, the packets in the DetNet flow are reordered before being forwarded to Ring R.

#### 6.2.3. Dual node interconnection for P2MP traffic using P2MP LSP

If P2MP LSPs are used in the interconnected rings, two P2MP unidirectional LSP tunnels are used on each ring for the clockwise and counter-clockwise directions.

When the P2MP traffic is forwarded from one ring to another ring, for example from Ring L to Ring R in Figure 3(b), each P2MP LSP in Ring L MUST include interconnection nodes I1 and I2 as its tail-ends. For Ring R, one P2MP LSP is set up from interconnection node I1 to all the tail-ends in the clockwise direction on Ring R, and the other P2MP LSP is set up from interconnection node I2 to all the tail-ends in the counter-clockwise direction on Ring R. Therefore, an interconnection node acts as a tail-end for one ring and a head-end for another ring in one direction, and performs the same operation of tail-end and head-end as specified in Section 5.3.

### 7. Resource Reservation

In order to guarantee that DetNet flows do not suffer from network congestion, the DetNet data plane considerations on resource reservation and allocation as described in [I-D.ietf-detnet-data-plane-framework] apply here.

### 8. IANA Considerations

There are no IANA actions required by this document

### 9. Security Considerations

This document describes the application of DetNet MPLS on ring topologies. Thus, the security considerations described in [I-D.ietf-detnet-mpls] are also applied to this document. If any new security considerations specific to ring topologies are identified, they will be added in a future version of this draft.

## 10. Editor's Note

This section lists current issues raised by experts in DetNet and other ring protection technologies. This section will be removed once the issues are addressed.

- o See if Resilient MPLS Ring (RMR) can be used for automatic configuration of a DetNet ring topology network.
- o Consideration of coexistence with existing ring protection solutions in the DetNet forwarding sublayer.
- o Consideration on scalability
- o Explain why this document is needed when the DetNet architecture and data plane documents exist.

## 11. References

### 11.1. Normative References

- [I-D.ietf-detnet-data-plane-framework]  
Varga, B., Farkas, J., Berger, L., Malis, A., and S. Bryant, "DetNet Data Plane Framework", draft-ietf-detnet-data-plane-framework-06 (work in progress), May 2020.
- [I-D.ietf-detnet-mpls]  
Varga, B., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS", draft-ietf-detnet-mpls-09 (work in progress), July 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.

## 11.2. Informative References

- [RFC6378] Weingarten, Y., Ed., Bryant, S., Osborne, E., Sprecher, N., and A. Fulignoli, Ed., "MPLS Transport Profile (MPLS-TP) Linear Protection", RFC 6378, DOI 10.17487/RFC6378, October 2011, <<https://www.rfc-editor.org/info/rfc6378>>.
- [RFC6974] Weingarten, Y., Bryant, S., Ceccarelli, D., Caviglia, D., Fondelli, F., Corsi, M., Wu, B., and X. Dai, "Applicability of MPLS Transport Profile for Ring Topologies", RFC 6974, DOI 10.17487/RFC6974, July 2013, <<https://www.rfc-editor.org/info/rfc6974>>.
- [RFC7271] Ryoo, J., Ed., Gray, E., Ed., van Helvoort, H., D'Alessandro, A., Cheung, T., and E. Osborne, "MPLS Transport Profile (MPLS-TP) Linear Protection to Match the Operational Expectations of Synchronous Digital Hierarchy, Optical Transport Network, and Ethernet Transport Network Operators", RFC 7271, DOI 10.17487/RFC7271, June 2014, <<https://www.rfc-editor.org/info/rfc7271>>.
- [RFC8227] Cheng, W., Wang, L., Li, H., van Helvoort, H., and J. Dong, "MPLS-TP Shared-Ring Protection (MSRP) Mechanism for Ring Topology", RFC 8227, DOI 10.17487/RFC8227, August 2017, <<https://www.rfc-editor.org/info/rfc8227>>.
- [RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <<https://www.rfc-editor.org/info/rfc8578>>.

## Authors' Addresses

Yuanlong Jiang  
Huawei Technologies  
Bantian, Longgang district  
Shenzhen 518129  
China

Phone: +86-18926415311  
Email: [jiangyuanlong@huawei.com](mailto:jiangyuanlong@huawei.com)

Norman Finn  
Huawei Technologies  
3755 Avocado Blvd  
California 91941  
USA

Phone: +1 925 980 6430  
Email: norman.finn@mail01.huawei.com

Jeong-dong Ryoo  
ETRI  
218 Gajeongno  
Yuseong-gu, Daejeon 34129  
South Korea

Phone: +82-42-860-5384  
Email: ryoo@etri.re.kr

Balazs Varga  
Ericsson  
Konyves Kalman krt. 11/B  
Budapest 1097  
Hungary  
  
Email: balazs.a.varga@ericsson.com

Liang Geng  
China Mobile  
Beijing  
China  
  
Email: gengliang@chinamobile.com

DetNet Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 6, 2018

A. Malis  
S. Bryant  
M. Chen  
Huawei Technologies  
B. Varga  
Ericsson  
March 05, 2018

DetNet IP Encapsulation  
draft-malis-detnet-ip-dp-00

Abstract

This document specifies Deterministic Networking data plane operation over an IP network. It is primarily aimed at IPv4, but would work with IPv6 as well if a unified solution is desired.

This document is a derivative work from draft-ietf-detnet-dp-sol-01, to augment or replace the text currently contained in section 5.2.2.

Whether this is published as a stand-alone text, or serves as a focal point to refine the IP design and subsequently remerged with draft-ietf-detnet-dp-sol-01 is a matter for the DETNET WG.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
2.1. Terms used in this document . . . . .	3
2.2. Abbreviations . . . . .	3
3. Requirements language . . . . .	3
4. DetNet Over an IP Network . . . . .	3
5. DetNet over IP Encapsulation . . . . .	5
6. Security considerations . . . . .	7
7. IANA considerations . . . . .	7
8. Acknowledgements . . . . .	7
9. References . . . . .	7
9.1. Normative References . . . . .	7
9.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

This document is a derivative work from [I-D.ietf-detnet-dp-sol].

Whether this is published as a stand-alone text, or serves as a focal point to refine the IP design and subsequently remerged with draft-ietf-detnet-dp-sol-01 is a matter for the DetNet WG.

Deterministic Networking (DetNet) is a service that can be offered by a network to DetNet flows. DetNet provides these flows extremely low packet loss rates and assured maximum end-to-end delivery latency. General background and concepts of DetNet can be found in [I-D.ietf-detnet-architecture].

This document specifies the encapsulation and operation of deterministic networking over an IP data-plane. The approach is modeled on the operation of PseudoWires (PW) over an IP Packet Switched Network (PSN) [RFC3985][RFC4385][RFC7510].

It is based on the "simplified model" discussed during the DetNet Interim Meeting held on 14 February 2018



[<http://etherpad.tools.ietf.org:9000/p/notes-ietf-interim-2018-detnet-03>].

It is also based on the MPLS encapsulation described in draft-bryant-detnet-mpls-dp (this reference to be updated once draft is available).

The DetNet transport layer functionality that provides congestion protection for DetNet flows is assumed to be in place in a DetNet node.

This document does not currently define the associated control plane functions, or Operations, Administration, and Maintenance (OAM). It also does not currently specify traffic handling capabilities required to deliver congestion protection and latency control for DetNet flows at the DetNet transport layer. These aspects may be included in future revisions of this draft, or in other DetNet documents.

## 2. Terminology

### 2.1. Terms used in this document

This document uses the same terminology as [I-D.ietf-detnet-dp-sol]. Please see that document for the definitions.

### 2.2. Abbreviations

This document uses the same abbreviations as [I-D.ietf-detnet-dp-sol]. Please see that document for the list of abbreviations.

## 3. Requirements language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 4. DetNet Over an IP Network

The "simplified model" of DetNet, as discussed during the interim meeting, is carried over an IP network is shown in Figure 1:

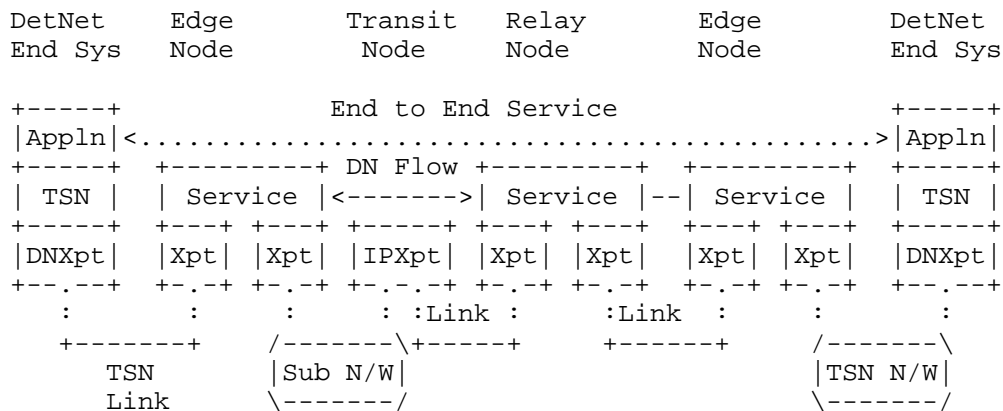


Figure 1: Simplified Model of a DetNet Enabled Network

In this figure, "DNXpt" and "Xpt" are abbreviations for "DetNet Transport" and "IPXpt" is an abbreviation for "IP Transport".

DetNet End Systems sent and receive packets over the DetNet. The supported packet types are IP and Ethernet.

Note that in the Simplified Model, while the DetNet service is end-to-end, the End Systems are not DetNet-aware and do not include DetNet information to their packet headers. Rather, the packets between the End Systems and the Edge Nodes may typically consist of application information, L4 Transport (such as TCP or UDP), IP, and Ethernet headers, transmitted over a TSN link or (sub)-Network between the End System and the Edge Node.

Alternatively, the packets could contain Ethernet-native applications, with the application information directly encapsulated within Ethernet without L4 Transport or IP headers.

Because the End Systems are not DetNet-aware, Edge Nodes are responsible for the imposition and disposition of the required DetNet encapsulation. This functionality is similar to that in pseudowire (PW) Provider Edge (PE) routers.

Relay Nodes are also strategically placed and used enhance the reliability of delivery by enabling the DetNet-layer replication of packets so that multiple copies, possibly over multiple paths. They also reduce the impact of replication by the eliminating surplus copies of DetNet packets. These functions may not be performed in End Systems, as they are not DetNet-aware.

Relay Nodes are aware of the needs of particular DetNet flows and take care to process them in accordance with the required performance needs.

Transit nodes are normal IP routers. They are unaware of DetNet flows per se, although they may be configured to provide congestion protection and delay control in order to meet the required DetNet service level agreement (SLA) via non-DetNet-specific means (IP traffic engineering, queuing mechanisms based on DiffServ markings, etc.).

## 5. DetNet over IP Encapsulation

To carry DetNet over IP the following is required:

1. A method of identifying the DetNet flow to the processing element.
2. A method of carrying the DetNet sequence number.

These latter two pieces of information are already carried in the DetNet over MPLS Encapsulation, as shown in Figure 1, where the Control Word contains a 28-bit sequence number, and the S-Label is used to identify the particular flow.

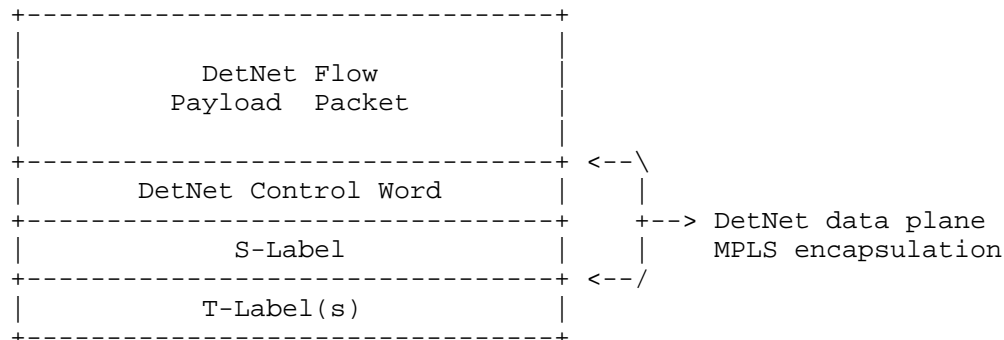


Figure 2: MPLS Encapsulation of DetNet

To simplify operations and implementations, rather than inventing a brand new encapsulation, the IP encapsulation can take advantage of the MPLS encapsulation. By using the specification of MPLS over UDP and IP in [RFC7510], the T-Label(s) in Figure 2 can be replaced by UDP and IP, resulting in the following encapsulation:

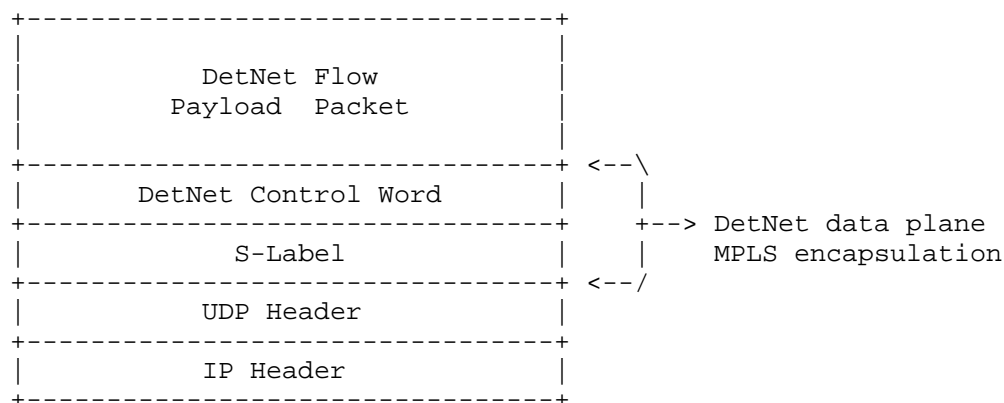


Figure 3: IP Encapsulation of DetNet

Where the UDP header is used as defined in Section 3 of [RFC7510].

In ingress Edge Nodes, the encapsulation in Figure 3 will be imposed on Detnet Flow Payload Packets as received from DetNet End Systems, and the encapsulation will be removed in egress Edge Nodes as they transmit the Payload Packets to the End Systems.

Note that this encapsulation works equally well with IPv4 and IPv6.

This encapsulation can also be used in conjunction with segment routing as specified in [I-D.ietf-spring-segment-routing-mpls]. In this case, the T-Label(s) in Figure 2 should be retained, and at each hop, the top T-label is popped and mapped to a corresponding UDP/IP tunnel, resulting in the following encapsulation:

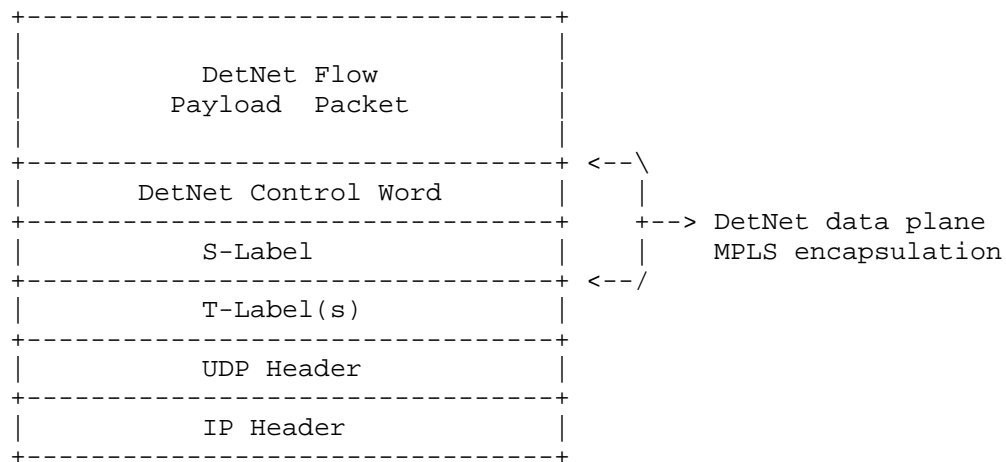


Figure 4: IP Encapsulation of DetNet with MPLS-SR

Again, the UDP header is used as defined in Section 3 of [RFC7510].

## 6. Security considerations

The security considerations of DetNet in general are discussed in [I-D.ietf-detnet-security]. Other security considerations will be added in a future version of this draft.

## 7. IANA considerations

This document makes no IANA requests.

## 8. Acknowledgements

## 9. References

### 9.1. Normative References

[I-D.ietf-detnet-dp-sol]

Korhonen, J., Andersson, L., Jiang, Y., Finn, N., Varga, B., Farkas, J., Bernardos, C., Mizrahi, T., and L. Berger, "DetNet Data Plane Encapsulation", draft-ietf-detnet-dp-sol-01 (work in progress), January 2018.

[I-D.ietf-spring-segment-routing-mpls]

Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-12 (work in progress), February 2018.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <<https://www.rfc-editor.org/info/rfc7510>>.

## 9.2. Informative References

- [I-D.ietf-detnet-architecture] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-04 (work in progress), October 2017.
- [I-D.ietf-detnet-security] Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell, J., Austad, H., Stanton, K., and N. Finn, "Deterministic Networking (DetNet) Security Considerations", draft-ietf-detnet-security-01 (work in progress), October 2017.
- [RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <<https://www.rfc-editor.org/info/rfc3985>>.
- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<https://www.rfc-editor.org/info/rfc4385>>.

## Authors' Addresses

Andrew G. Malis  
Huawei Technologies

Email: [agmalis@gmail.com](mailto:agmalis@gmail.com)

Stewart Bryant  
Huawei Technologies

Email: [stewart.bryant@gmail.com](mailto:stewart.bryant@gmail.com)

Mach Chen  
Huawei Technologies

Email: mach.chen@huawei.com

Balazs Varga  
Ericsson

Email: balazs.a.varga@ericsson.com

BIER  
Internet-Draft  
Intended status: Standards Track  
Expires: September 4, 2018

P. Thubert, Ed.  
Cisco  
T. Eckert  
Huawei  
Z. Brodard  
Ecole Polytechnique  
H. Jiang  
Telecom Bretagne  
March 3, 2018

BIER-TE extensions for Packet Replication and Elimination Function  
(PREF) and OAM  
draft-thubert-bier-replication-elimination-03

Abstract

This specification extends Bit Index Explicit Replication - Traffic Engineering (BIER-TE) forwarding to support in the data plane the DetNet Packet Replication and Elimination Functions (PREF). It also provides traceability of links/adjacencies where replication and loss happen, in a manner that is agnostic to the forwarding information (OAM).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 4, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents



(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. On BIER - Traffic Engineering . . . . .	3
4. BIER-TE-based Replication and Elimination Control . . . . .	4
5. Elimination Function (Normative) . . . . .	9
6. Summary . . . . .	11
7. Implementation Status . . . . .	12
8. Security considerations . . . . .	12
9. IANA Considerations . . . . .	12
10. Acknowledgements . . . . .	12
11. References . . . . .	13
11.1. Normative References . . . . .	13
11.2. Informative References . . . . .	13
Authors' Addresses . . . . .	14

## 1. Introduction

Deterministic Networking (DetNet) [I-D.ietf-detnet-problem-statement] provides a capability to carry unicast or multicast data flows for real-time applications with extremely low data loss rates and known upper bound maximum latency [I-D.ietf-detnet-architecture].

DetNet applies to multiple environments where there is a desire to replace a point to point serial cable or a multidrop bus by a switched or routed infrastructure, in order to scale, lower costs, and simplify management. One classical use case is found in particular in the context of the convergence of IT with Operational Technology (OT), also referred to as the Industrial Internet. But there are many others use cases [I-D.ietf-detnet-use-cases], for instance in professional audio and video, automotive, radio fronthauls, etc..

The DetNet data plane alternatives [I-D.dt-detnet-dp-alt] studies the applicability of existing and emerging dataplane techniques that can be leveraged to enable DetNet properties in IP networks. One critical feature in the dataplane is traceability, the capability to control the activity of intermediate nodes on a packet. For instance, if Replication and Elimination is applied to a packet, then it is desirable to determine which node performed a certain copy of

that packet that is circulating in the network. Likewise, engineered paths are required to support redundant transmission across disjoint paths in support of DetNets PREF functions.

Traceability belongs to Operations, Administration, and Maintenance (OAM) which is the toolset for fault detection and isolation, and for performance measurement. More can be found on OAM Tools in "An Overview of Operations, Administration and Maintenance (OAM) Tools" [I-D.ietf-opsawg-oam-overview].

This document proposes a new set to mechanisms based on [RFC8279] (BIER) and more specifically BIER Traffic Engineering [I-D.ietf-bier-te-arch] (BIER-TE) to control the process or Packet Replication and Elimination Functions (PREF), and provide traceability of these operations, in the DetNet dataplane. An adjacency, which is represented by a bit in the BIER header, can correspond in the dataplane to an Ethernet hop, a Label Switched Path, or it can correspond to an IPv6 loose or strict source routed path.

BIER-TE was primarily designed to carry multicast traffic, but there is nothing prohibiting for it to be used with unicast traffic, and the authors of this document think that for networks whose size requirement match the supportable bitstring length (BSL) in BIER, it can be a good choice as the forwarding plane specifically for DetNet type traffic for both multicast and unicast traffic because it would be a common solution for unicast and multicast (limiting the number of different technologies a DetNet solution requires) and likely provides the most flexible support for path engineering, replication and elimination (PREF) and the novel OAM method described in this document.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. On BIER - Traffic Engineering

[RFC8279] (BIER) is a network plane replication technique that was initially intended as a new method for multicast distribution. In a nutshell, a BIER header includes a bitmap that explicitly signals the listeners that are intended for a particular packet, which means that 1) the sender is aware of the individual listeners and 2) the BIER control plane is a simple extension of the unicast routing as opposed to a dedicated multicast data plane, which represents a considerable

reduction in OPEX. For this reason, the technology faces a lot of traction from Service Providers.

The simplicity of the BIER technology makes it very versatile as a network plane signaling protocol. Already, a new Traffic Engineering variation is emerging that uses bits to signal segments along a TE path.

While BIER-TE was like BIER primarily developed for multicast traffic, the authors think that it can equally be attractive for unicast traffic requiring the DetNet resilience of multiple transitions. If the topology of the network can well be represented by standard BIER-TE bitstring sizes of e.g.: up to 256 bits, then this would allow for a single technology for both unicast and multicast.

BIER-TE supports a Traffic Engineered forwarding plane by explicit hop-by-hop forwarding and loose hop forwarding of packets.

From the BIER-TE architecture, the key differences over BIER are:

- o BIER-TE replaces in-network autonomous path calculation by explicit paths calculated off path for example by a BIER-TE controller host.
- o In BIER-TE every BitPosition of the BitString of a BIER-TE packet indicates one or more adjacencies - instead of a BFER as in BIER. processing packets as a destination (BFER) is one of the possible adjacency types.
- o BIER-TE in each BFR has no routing table but only a BIER-TE Forwarding Table (BIFT) indexed by SI:BitPosition and populated with only those adjacencies to which the BFR should replicate packets to.

The generic view of an adjacency can be over a link, a tunnel or along a path segment.

#### 4. BIER-TE-based Replication and Elimination Control

This document only needs to introduce new functionality to support the Elimination Function and OAM. Creation of appropriate BIER-TE packets is subject to to existing work.

In the solution described below, the encapsulation/insertion of flow-identification and sequence number into packets is performed by a function on the BFIR outside the scope of this document. A companion document draft-huang-bier-te-encapsulation defines an encapsulation for BIER-TE and BIER that can support flow-id and sequence-number ID. Other encapsulations can be used as well, as long as they provide

these signaling elements and are supported by the Elimination Function described in this document (e.g.: that the EF can read these fields and therefore remove duplicates). In the remainder of this document we will call this the extended BIER encapsulation and assume that it is used when describing examples. Unless otherwise noted, we assume that the BFIR performs encapsulation of some data flow packets with an extended BIER header, indicates BIER-TE forwarding in it and fills in flow-id and sequence number. It then fills in the bitstring with two (or more) alternative paths/DAGs and sends off the packets into the BIER-TE domain, replicating it itself if so indicated by the bitstring.

In a nutshell, BIER-TE is used as follows:

- o A controller computes a complex path, sometimes called a track, which takes the general form of a ladder. The steps and the side rails between them are the adjacencies that can be activated on demand on a per-packet basis using bits in the BIER header.

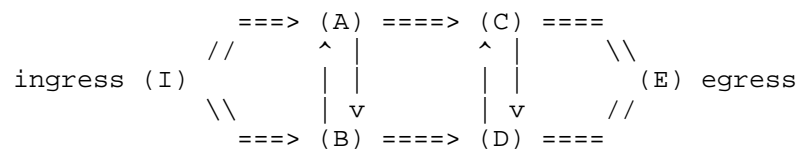


Figure 1: Ladder Shape with Replication and Elimination Points

- o The controller assigns a BIER domain, and inside that domain, assigns bits to the adjacencies. The controller assigns each bit to a replication node that sends towards the adjacency, for instance the ingress router into a segment that will insert a routing header in the packet. A single bit may be used for a step in the ladder, indicating the other end of the step in both directions.

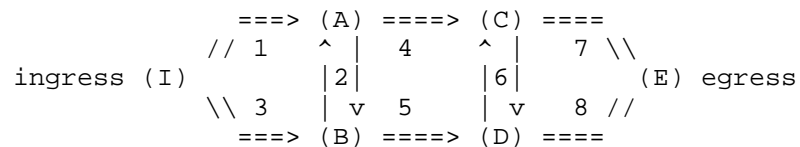


Figure 2: Assigning Bits

- o The controller activates the replication by deciding the setting of the bits associated with the adjacencies. This decision can be modified at any time, but takes the latency of a controller round trip to effectively take place. Below is an example that uses Replication and Elimination to protect the A->C adjacency. The "(EF)" in the following pictures Owner column indicates the fact that that BFR will perform the "Elimination Function" for received BIER-TE packets before further processing/copying them. In this example, only C performs EF. A (1) in the Example Bitstring indicates that the bit is set, but that the actual adjacency is not used by packets because this bit is shared with another adjacency and the overall bitstring will make the packet only use that other adjacency. This applies to bits 2 and 6.

Bit #	Adjacency	Owner	Example Bitstring
1	I->A	I	1
2	A->B	A	1
	B->A	B	(1)
3	I->B	I	0
4	A->C	A	1
5	B->D	B	1
6	C->D	C (EF)	(1)
	D->C	D	
7	C->E	C (EF)	1
8	D->E	D	0

Replication and Elimination Protecting A-&gt;C

Table 1: Controlling Replication

- o The BIER header with the controlling BitString , flow-id and sequence number is injected in the packet by the ingress node I (BFIR). That node may act as a replication point, in which case it may issue multiple copies of the packet, but for the purpose of this example it will not do it, so that the two paths used in this example only go from A to C, and therefore require explicit path engineering. For example, bandwidth I-A and I-B may be more limited and those paths being non long-haul may not warrant the dual transmission.

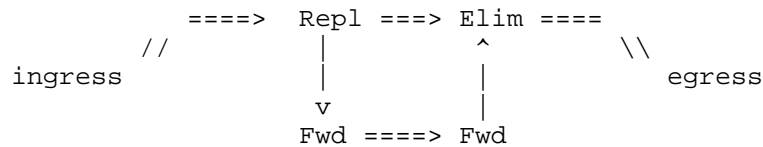


Figure 3: Enabled Adjacencies

- o For each of its bits that is set in the BIER header, the owner replication point resets the bit used for a copy and transmits towards the associated adjacency; to achieve this, the replication point copies the packet and inserts the relevant data plane information, such as next-hop label, MAC-address or source route header (for a BIER-TE routed adjacency), towards the adjacency that corresponds to the bit

Adjacency	BIER BitString
I->A	01011110
A->B	00011110
B->D	00010110
D->C	00010010
A->C	01001110

BitString in BIER Header as Packet Progresses

Table 2: BIER-TE in Action

- o Adversely, an elimination node on the path performs the Elimination Function which will remove duplicate packets (same flow-id, same sequence number) and performs a bitwise AND on the BitStrings from the various copies of the packet that it has received, before it forwards the packet with the resulting BitString. Details of the Elimination Function are described below.

Operation	BIER BitString
D->C	00010010
A->C	01001110
AND in C	00000010
C->E	00000000

BitString Processing at Elimination Point C

Table 3: BIER-TE in Action (cont.)

- o In this example, all the transmissions succeeded and the BitString at arrival has all the bits reset - note that the egress may be an Elimination Point in which case this is evaluated after this node has performed its AND operation on the received BitStrings).

Failing Adjacency	Egress BIER BitString
I->A	Frame Lost
I->B	Not Tried
A->C	00010000
A->B	01001100
B->D	01001100
D->C	01001100
C->E	Frame Lost
D->E	Not Tried

BitString indicating failures

Table 4: BIER-TE in Action (cont.)

- o But if a transmission failed along the way, one (or more) bit is never cleared. Table 4 provides the possible outcomes of a transmission. If the frame is lost, then it is probably due to a failure in either I->A or C->E, and the controller should enable I->B and D->E to find out. A BitString of 00010000 indicates unequivocally a transmission error on the A->C adjacency, and a BitString of 01001100 indicates a loss in either A->B, B->D or D->C; enabling D->E on the next packets may provide more information to sort things out.

In more details:

The BIER header is of variable size, and a DetNet network of a limited size can use a model with 64 bits if 64 adjacencies are enough, whereas a larger deployment may be able to signal up to 256 adjacencies for use in very complex paths. The format of this header is common to BIER and BIER-TE.

For the DetNet data plane, a replication point is an ingress point for more than one adjacency, and an elimination point is an egress point for more than one adjacency.

A pre-populated state in a replication node indicates which bits are served by this node and to which adjacency each of these bits corresponds. With DetNet, the state is typically installed by a controller entity such as a PCE. The way the adjacency is signaled in the packet is fully abstracted in the bit representation and must be provisioned to the replication nodes and maintained as a local state, together with the timing or shaping information for the associated flow.

The DetNet data plane uses BIER-TE to control which adjacencies are used for a given packet. This is signaled from the path ingress, which sets the appropriate bits in the BIER BitString to indicate which replication must happen.

The replication point clears the bit associated to the adjacency where the replica is placed, and the elimination points perform a logical AND of the BitStrings of the copies that it gets before forwarding.

As is apparent in the examples above, clearing the bits enables to trace a packet to the replication points that made any particular copy. BIER-TE also enables to detect the failing adjacencies or sequences of adjacencies along a path and to activate additional replications to counter balance the failures.

Finally, using the same BIER-TE bit for both directions of the steps of the ladder enables to avoid replication in both directions along the crossing adjacencies. At the time of sending along the step of the ladder, the bit may have been already reset by performing the AND operation with the copy from the other side, in which case the transmission is not needed and does not occur (since the control bit is now off).

## 5. Elimination Function (Normative)

This section defines the normative behavior of the Elimination Function with optional OAM sub-function.



The Elimination Function is performed logically on reception of BIER-TE packets. It is therefore not part of the adjacencies or otherwise assigned to a specific bit. "Logically" means that this specification does not constrain implementations, especially on multi-linecard/multi-chassis systems to perform EF on a physical egress module. It just implies that it has to happen before replication to the bits in the bitstring.

TBD: In addition to being an ingress, EF could as well be modelled as a new adjacency assigned to bits. The full adjacency of a bit could then be a sequence of EF followed by one (or more) of existing adjacencies. This is currently not considered by this document due to the lack of identified need to support this option - e.g.: problems that can not be equally/better be solved with EF logically on ingress.

The Elimination Function is more formally written as EF(OAM, BIFT, {flows}/\*), and is configured like BIFTs from the BIER-TE controller host and/or other future mechanisms.

OAM is boolean and indicates whether OAM function of bitwise AND of received packet copies is performed. This OAM function requires additional memory/processing over EF without OAM. Note that the OAM function does not change the effect of the Elimination Function for BFR/receivers - they will continue to just receive the first copy of a packet. Instead, it will continue to track further copies solely for the purpose of providing OAM information. This also requires some timeout or sequence number advancement to decide when to terminate waiting for further copies of packets before considering the OAM analysis of this packet to be complete. BFR supporting this document SHOULD support the OAM sub-function.

BIFT indicates the <SD,SI,BSL> for which to perform EF. Devices SHOULD support enabling per EF. {flows}/\* indicates the set of flows for which EF operates (using the specified BIFT). Duplicate elimination has to create per-flow state to remember which sequence number packets for this flow were already received. In the case of OAM also what bits were set in that received prior copy of the packet.

When a device supports "\*", then it will automatically allocate such a flow-state for every new recognized flow and expire such flow state after an operator determined timeout of activity - for example with a default of 10 seconds. Dynamic allocation of flow-state may cause some initial duplicates before this state is working and it makes the BFR more vulnerable to state DOS attacks, but it will allow BIER applications to send flows with the benefit of EF without the help of the controller having to know and program every flow.

In the {flows} option, control procedures (e.g.: BIER-TE controller host) indicate to the BFR explicitly the set of flows for which it should install/operate the EF function. Note that the flow-id in the extended BIER encapsulation is the combination of BFIR-ID and entropy field of the BIER header.

BFR supporting this document MUST support the {flows} option and MAY support the "\*" option.

The following picture explains the results of EF being performed on ingres in a typical example:

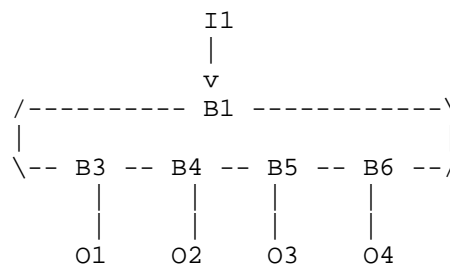


Figure 4: EF with Rings

Consider a simple ring where BFIR I1 generates BIER-TE packets. The bitstring indicates that the packet is sent hop-by-hop counterclockwise B1->B3->B4->B6 and counterclockwise B1->B6->B5->B4->B3. Bits for BFER O1, O2, O3 and O4 are also set. B3,B4,B5,B6,B7 perform EF. The result of this setup is that B2 creates two copies of the packets received from I1, one going to B6, the other to B3. Assume B4 first received the counter-clockwise copy from B3 and B5 the clockwise copy from B6. They will both forward these packets to each other because those were the first copies they saw, but they would block these second copies. Therefore only the link B4->B5 will have carried the packet copy twice (once in each direction). All the other ring links will only carry one copy of the packet.

This is notably different from schemes where EF is not performed before replication, but afterwards. In those schemes, both copies of the packets would flow counterclockwise around (most of) the ring, occupying more bandwidth.

## 6. Summary

With the addition of the functions of this document, BIER-TE becomes a potential option for the DetNet dataplane specifically beneficial when PREF (replication and elimination) is required for resilience

(to reduce packet loss). For DetNet multicast but also DetNet unicast. The unique capabilities of this approach are:

- o Explicit per-packet path selection for packet. Multicast and Unicast.
- o Control which replication take place on a per packet basis, so that replication points can be configured but not actually utilized
- o Trace the replication activity and determine which node replicated a particular packet
- o Measure the quality of transmission of the actual data packet along the replication segments and use that in a control loop to adapt the setting of the bits and maintain the reliability.

## 7. Implementation Status

A research-stage implementation of the forwarding plane for a 6TiSCH IOT use case was developed at Cisco's Paris Innovation Lab (PIRL) by Zacharie Brodard. It was implemented on OpenWSN Open-source firmware and tested on the OpenMote-CC2538 hardware. It implements the header types 15,16, 17, 18 and 19 (bit-by-bit encoding without group ID) in order to allow a BIER-TE protocol over IEEE802.15.4e.

This work was complemented with a Controller-based control loop by Hao Jiang. The controller builds the complex paths (called Tracks in 6TiSCH) and decides the setting of the BitStrings in real time in order to optimize the delivery ratio within a minimal energy budget.

Links:

github: <https://github.com/zach-b/openwsn-fw/tree/BIER>  
OpenWSN firmware: <https://openwsn.atlassian.net/wiki/pages/viewpage.action?pageId=688187>  
OpenMote hardware: <http://www.openmote.com/>

## 8. Security considerations

TBD.

## 9. IANA Considerations

This document has no IANA considerations.

## 10. Acknowledgements

The method presented in this document was discussed and worked out together with the DetNet Data Plane Design Team:

Jouni Korhonen  
Janos Farkas  
Norman Finn  
Olivier Marce  
Gregory Mirsky  
Pascal Thubert  
Zhuangyan Zhuang

The authors also like to thank the DetNet chairs Lou Berger and Pat Thaler, as well as Thomas Watteyne, 6TiSCH co-chair, for their contributions and support to this work.

## 11. References

### 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

### 11.2. Informative References

- [I-D.dt-detnet-dp-alt]  
Korhonen, J., Farkas, J., Mirsky, G., Thubert, P., Zhuangyan, Z., and L. Berger, "DetNet Data Plane Protocol and Solution Alternatives", draft-dt-detnet-dp-alt-04 (work in progress), September 2016.
- [I-D.ietf-bier-te-arch]  
Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication (BIER-TE)", draft-ietf-bier-te-arch-00 (work in progress), January 2018.
- [I-D.ietf-detnet-architecture]  
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-04 (work in progress), October 2017.
- [I-D.ietf-detnet-problem-statement]  
Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", draft-ietf-detnet-problem-statement-02 (work in progress), September 2017.

## [I-D.ietf-detnet-use-cases]

Grossman, E., "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-14 (work in progress), February 2018.

## [I-D.ietf-opsawg-oam-overview]

Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", draft-ietf-opsawg-oam-overview-16 (work in progress), March 2014.

## [I-D.ietf-spring-segment-routing]

Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.

## Authors' Addresses

Pascal Thubert (editor)  
Cisco Systems  
Village d'Entreprises Green Side  
400, Avenue de Roumanille  
Batiment T3  
Biot - Sophia Antipolis 06410  
FRANCE

Phone: +33 4 97 23 26 34  
Email: [pthubert@cisco.com](mailto:pthubert@cisco.com)

Toerless Eckert  
Huawei USA - Futurewei Technologies Inc.  
2330 Central Expy  
Santa Clara 95050  
USA

Email: [tte+ietf@cs.fau.de](mailto:tte+ietf@cs.fau.de)

Zacharie Brodard  
Ecole Polytechnique  
Route de Saclay  
Palaiseau 91128  
FRANCE

Phone: +33 6 73 73 35 09  
Email: zacharie.brodard@polytechnique.edu

Hao Jiang  
Telecom Bretagne  
2, rue de la Chataigneraie  
Cesson-Sevigne 35510  
FRANCE

Phone: +33 7 53 70 97 34  
Email: hao.jiang@telecom-bretagne.eu