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-02

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) 16
5.3.1. Networks with multiple technology segments 16
5.3.2. DN-IWF related considerations 17
6. MPLS-based DetNet data plane solution 18
6.1. DetNet specific packet fields 18
6.2. Data plane encapsulation 18
6.3. DetNet control word 19
6.4. Flow identification 20
6.5. Service layer considerations 20
6.5.1. Edge node processing 21
6.5.2. Relay node processing 22
6.5.3. End system processing 24
6.6. Transport node considerations 24
6.6.1. Congestion protection 24
6.6.2. Explicit routes 24
7. IPv6-based DetNet data plane solution 24
7.1. Data plane encapsulation 24

7.2. DetNet destination option 26
7.3. Flow identification 27
7.4. Service layer considerations 27
7.4.1. Edge node processing 28
7.4.2. Relay node processing 30
7.4.3. End system processing 30
7.5. Transport node processing 30
7.5.1. Congestion protection 30
7.5.2. Explicit routes 31
8. Other DetNet data plane considerations 31
8.1. Class of Service 31
8.2. Quality of Service 31
8.3. Cross-DetNet flow resource aggregation 33
8.4. Bidirectional traffic 34
8.5. Layer 2 addressing and QoS Considerations 34
8.6. Interworking between MPLS- and IPv6-based encapsulations 35
8.7. IPv4 considerations 35
9. Time synchronization 35
10. Management and control considerations 37
10.1. MPLS-based data plane 37
10.1.1. S-Label assignment and distribution 37
10.1.2. Explicit routes 37
10.2. IPv6-based data plane 37
10.2.1. Flow Label assignment and distribution 37
10.2.2. Explicit routes 38
10.3. Packet replication and elimination 38
10.4. Congestion protection and latency control 38
10.5. Flow aggregation control 38
11. Security considerations 38
12. IANA considerations 38
13. Acknowledgements 38
14. References 39
14.1. Normative references 39
14.2. Informative references 41
Appendix A. Example of DetNet data plane operation 42
Appendix B. Example of pinned paths using IPv6 43
Authors' Addresses 43

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 enacapsulation 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 eaxtension 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 implment 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 duplaicate 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 formwarding 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.

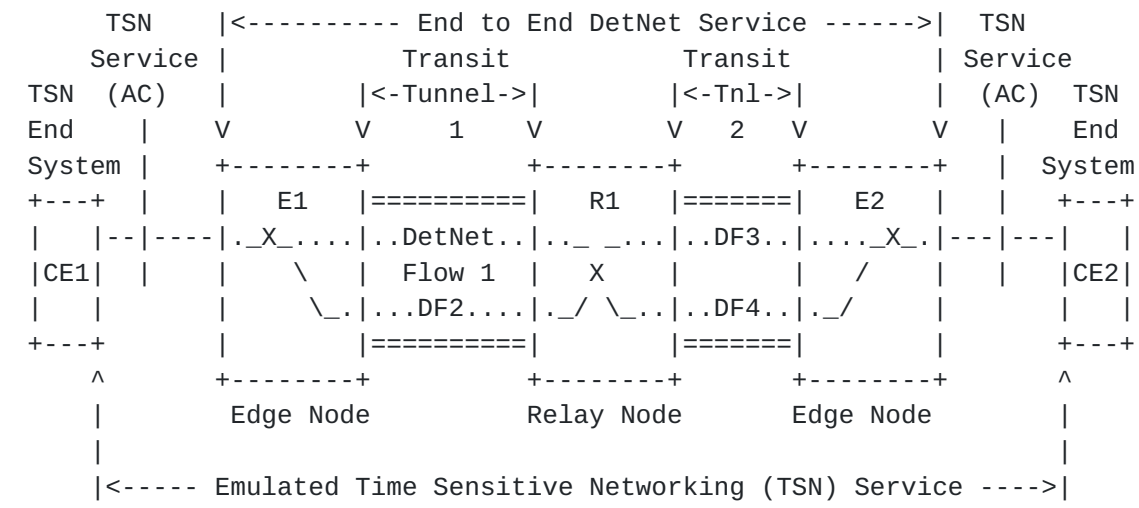


Figure 1: IEEE 802.1TSN over DetNet

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.

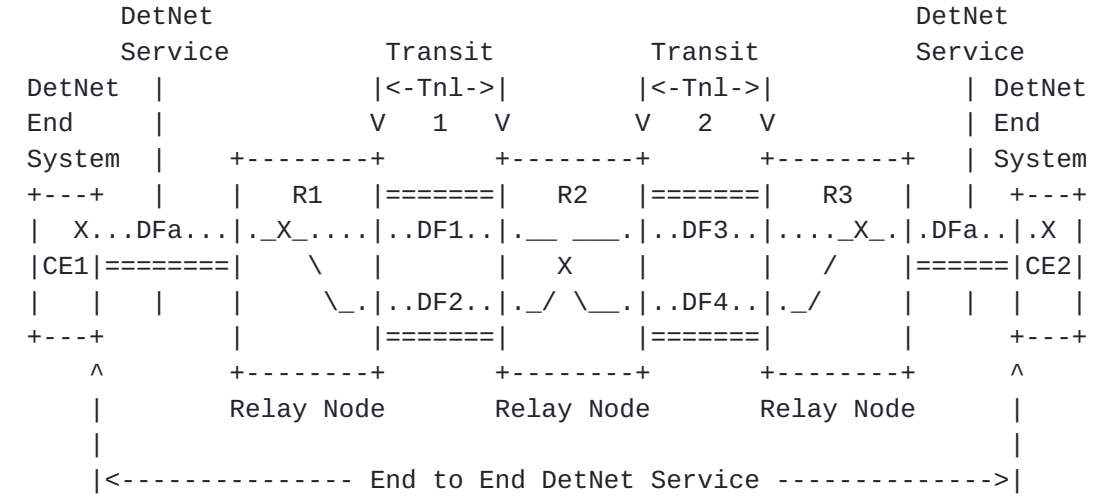


Figure 2: MPLS-Based Native DetNet

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

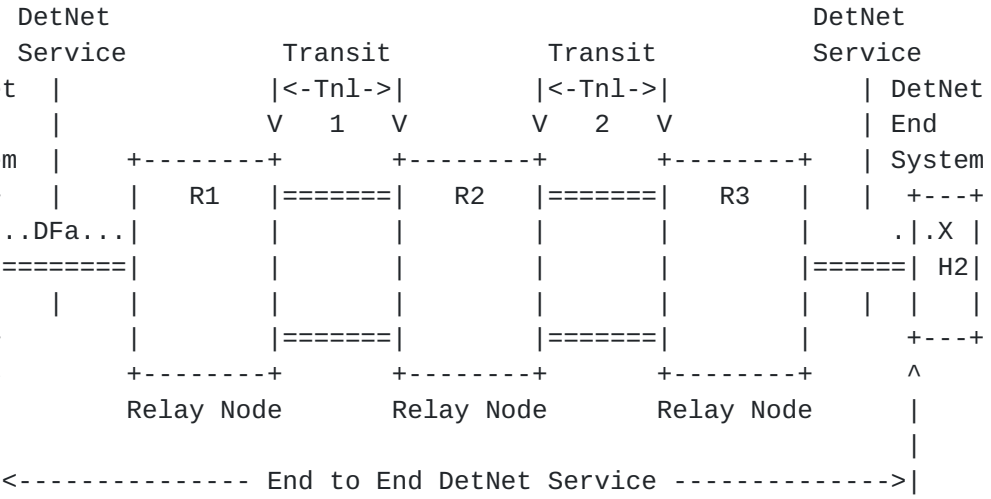


Figure 3: IP-Based Native DetNet

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 an 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 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 reordeing 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:

- L3 (IP) end-system: application over L3

- 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):

- L3 end-systems peer with L3 end-systems and

- L2 end-systems peer with L2 end-systems

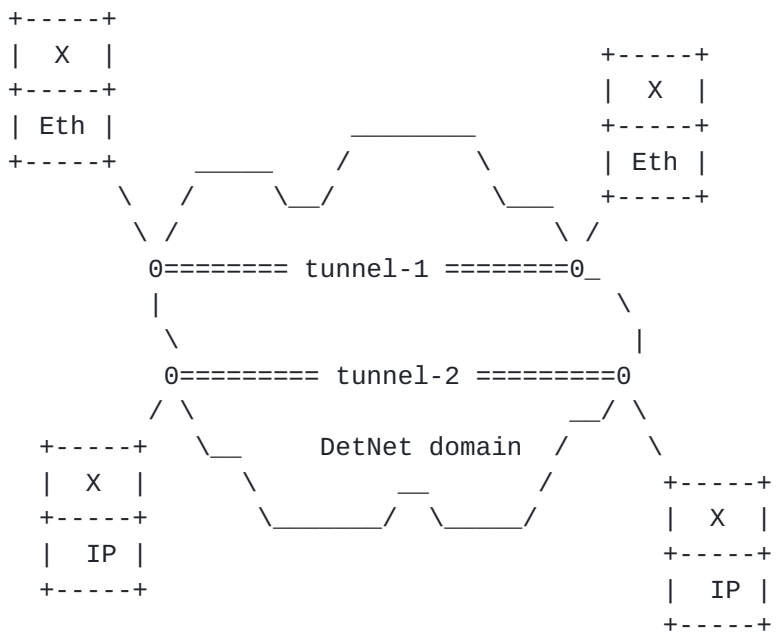


Figure 4: End-systems and the DetNet domain

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.

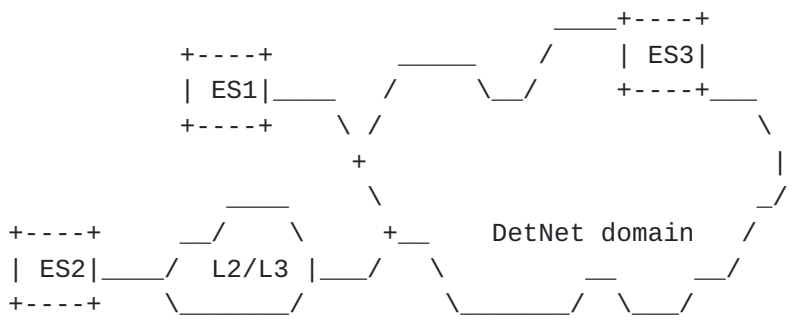


Figure 5: Connection types of L3 end-systems

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.

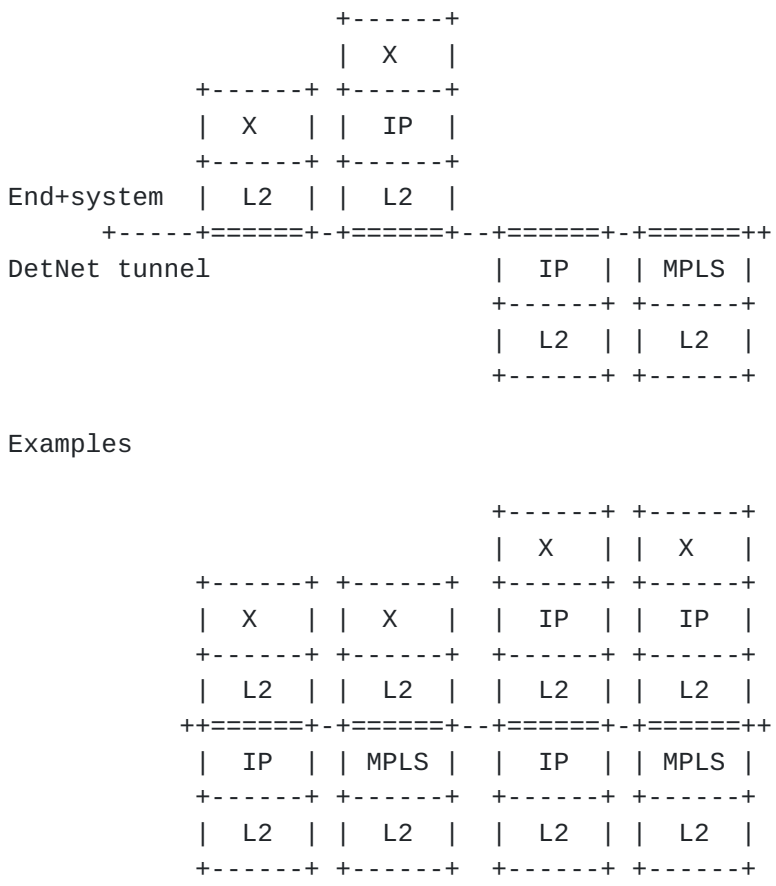


Figure 6: Encapsulation format for DetNet Bridging

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.

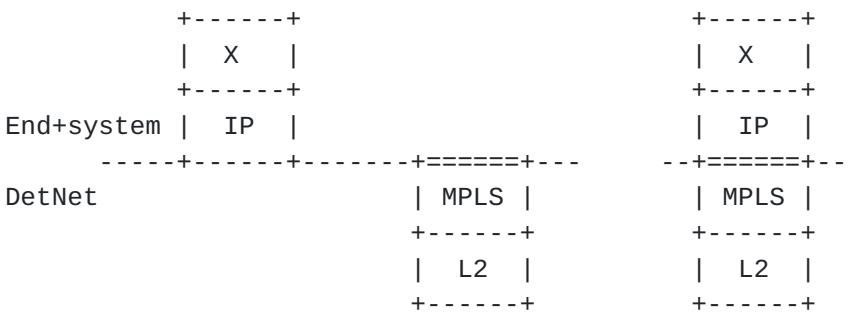


Figure 7: Encapsulation format for DetNet Routing in MPLS PSN for L3 end-systems

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.

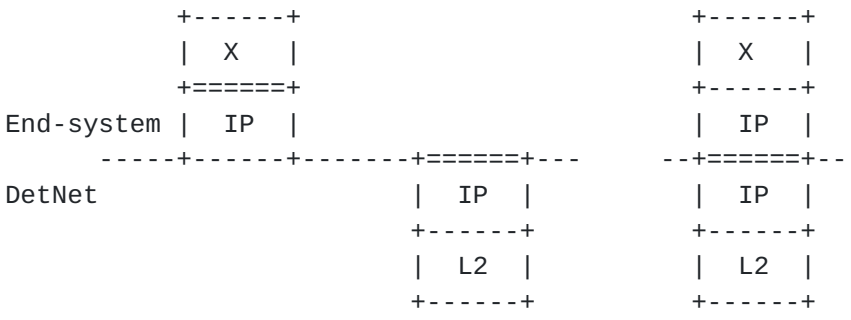


Figure 8: Encapsulation format for DetNet Routing in IP PSN for L3 end-systems

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.).

5.3. 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 stich 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.

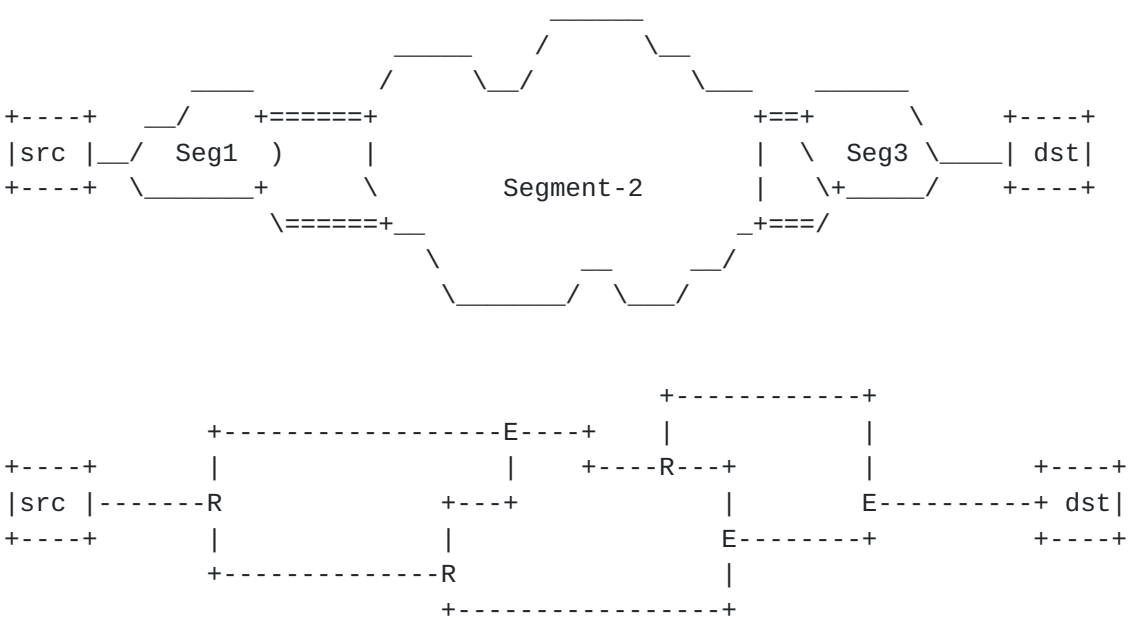


Figure 9: Optimal replication and elimination placement across technology segments example

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:

- Flow-ID: it is same in all packets of a flow

- 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 later case the DN-IWF can simple 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 10: 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 11 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:

- 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.

- 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 lable (S-Label) that represents DetNet Service LSP used between DetNet Egde 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 the actual DetNet data plane encapsulation.

DetNet MPLS-based encapsulation

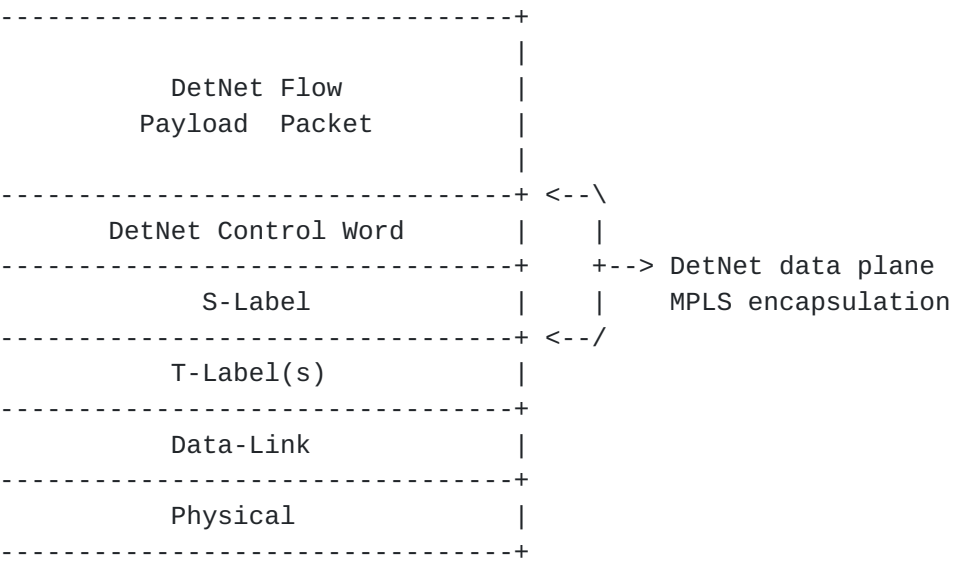


Figure 11: Encapsulation of a DetNet flow in an MPLS(-TP) PSN

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 12. 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 seqeunce number and the value of the sequence number MUST be set to zero (0).

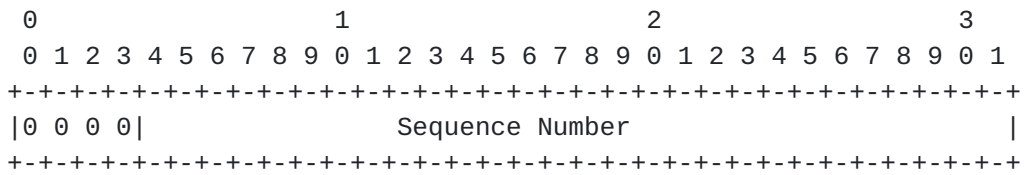


Figure 12: DetNet Control Word

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 errorneously 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. Edge node processing

TBD.

[Editor's note: Since we are not defining the inner workings and implementation of the DetNet Egde 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.]

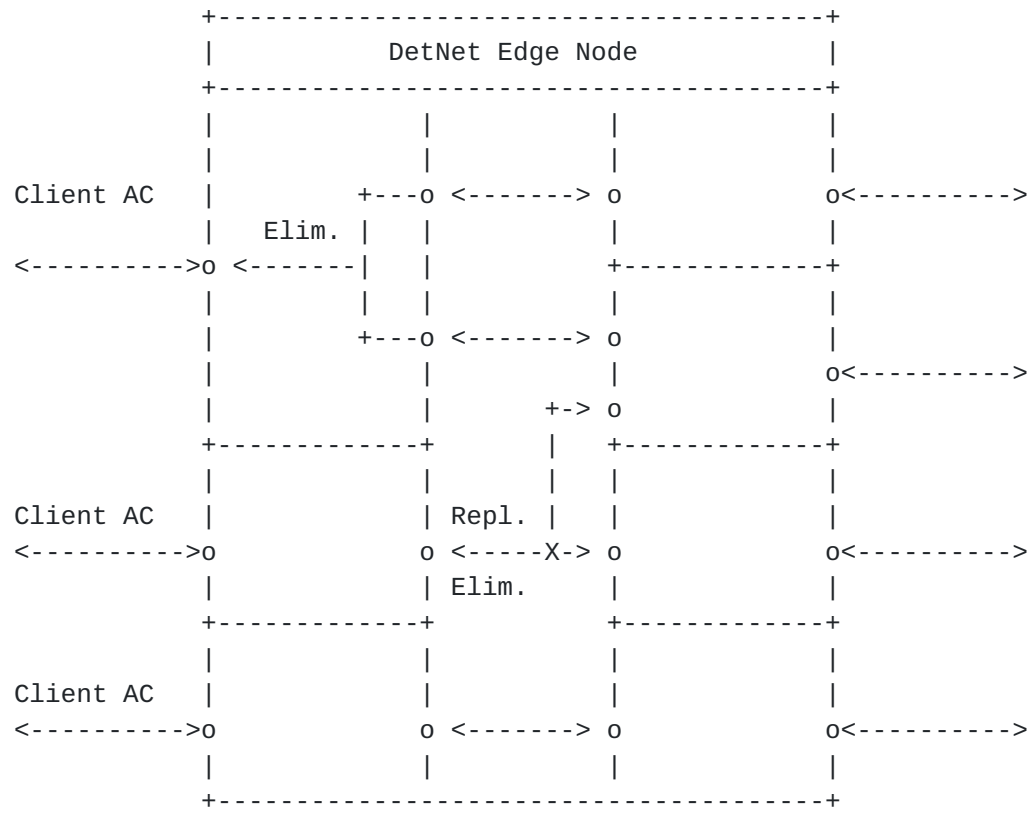


Figure 13: DetNet Edge Node processing

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 termonilogy 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.

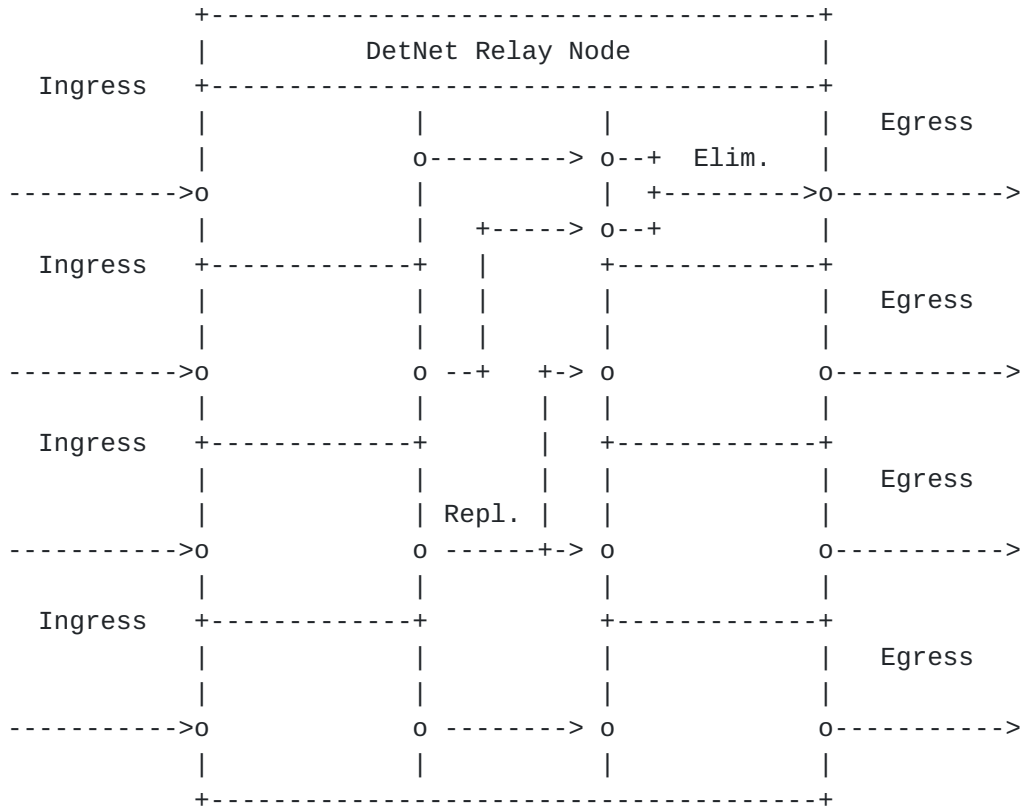


Figure 14: DetNet Relay Node processing

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 kinf 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 15 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:

- IPv6 header as the transport protocol.
- 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.

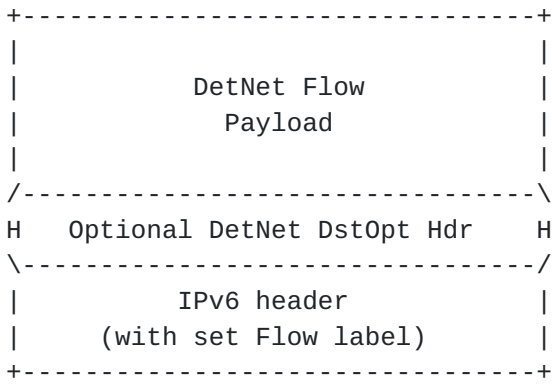


Figure 15: Encapsulation of a native IPv6 DetNet-s- or DetNet-st-flow without a routing header

Figure 16 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).

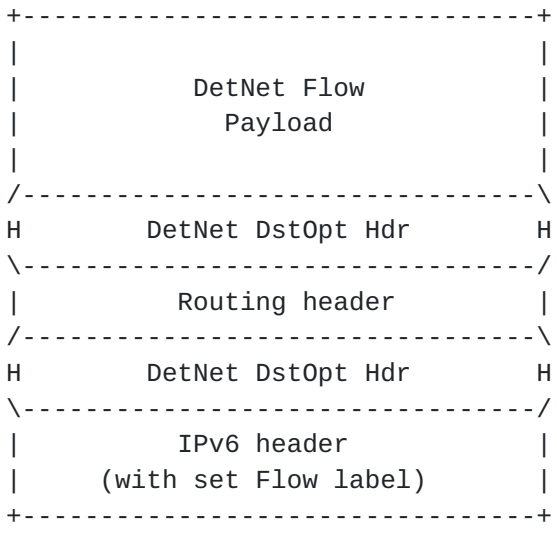


Figure 16: Encapsulation of a native IPv6 DetNet-s- or DetNet-st-flows with routing header

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 17.

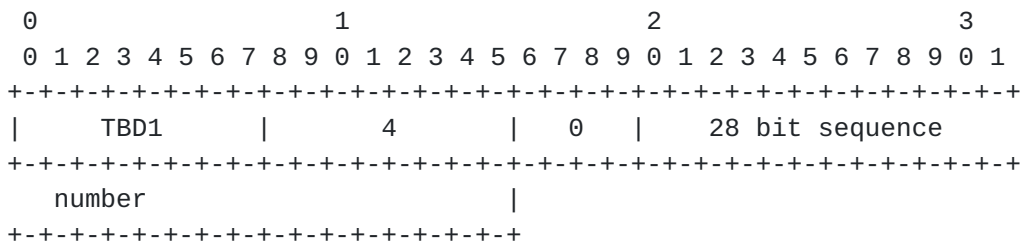


Figure 17: DetNet Destination Option

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 uniquelly 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.]

- PREF - requires both flow identification and sequence numbering.

- 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 18. Essentially every time an igress 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.

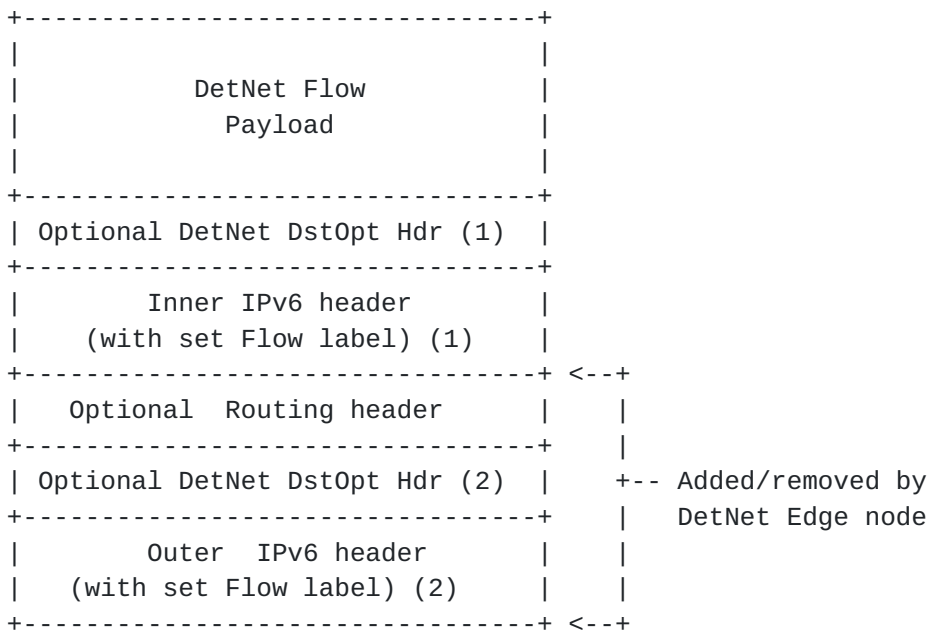


Figure 18: Encapsulation of a DetNet-flow IPv6 packet at the DetNet Edge node

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:

- 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.

- The flow label of the outer IPv6 header SHOULD be set to a value maintained by the edge node.

- 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 reordeing 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:

- 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.

- The flow label of the outer IPv6 header SHOULD be set to a value maintained by the edge node.

- The DetNet Destination Option with the edge node managed per DetNet flow sequence number value MAY be inserted into the outer IPv6 header.

- 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 sequece 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. Engress 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:

- Defend the DetNet QoS by discarding or remarking (to a non-DetNet CoS) packets received that are not the subject of a completed reservation.

- 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 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 applicaton(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:

- 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.

- 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.

- 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.

- 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 sould 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

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
Leganes, Madrid 28911
Spain

Phone: +34 91624 6236
Email: cjbc@it.uc3m.es
URI: http://www.it.uc3m.es/cjbc/

Tal Mizrahi
Marvell
6 Hamada st.
Yokneam
Israel

Email: talmi@marvell.com

Lou Berger
LabN Consulting, L.L.C.

Email: lberger@labn.net