DetNet
Internet-Draft
Intended status: Standards Track
Expires: February 4, 2019

N. Finn
Huawei
P. Thubert
Cisco
B. Varga
J. Farkas
Ericsson

August 3, 2018

Deterministic Networking Architecture
draft-ietf-detnet-architecture-07

Abstract

Deterministic Networking (DetNet) provides a capability to carry specified unicast or multicast data flows for real-time applications with extremely low data loss rates and bounded latency. Techniques used include: 1) reserving data plane resources for individual (or aggregated) DetNet flows in some or all of the intermediate nodes (e.g., bridges or routers) along the path of the flow; 2) providing explicit routes for DetNet flows that do not immediately change with the network topology; and 3) distributing 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.

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 February 4, 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. Terminology 4
2.1. Terms used in this document 4
2.2. IEEE 802.1 TSN to DetNet dictionary 6
3. Providing the DetNet Quality of Service 7
3.1. Primary goals defining the DetNet QoS 7
3.2. Mechanisms to achieve DetNet QoS 9
3.2.1. Congestion protection 9
3.2.1.1. Eliminate congestion loss 9
3.2.1.2. Jitter Reduction 10
3.2.2. Service Protection 11
3.2.2.1. In-Order Delivery 11
3.2.2.2. Packet Replication and Elimination 11
3.2.2.3. Packet encoding for service protection 13
3.2.3. Explicit routes 13
3.3. Secondary goals for DetNet 14
3.3.1. Coexistence with normal traffic 14
3.3.2. Fault Mitigation 15
4. DetNet Architecture 16
4.1. DetNet stack model 16
4.1.1. Representative Protocol Stack Model 16
4.1.2. DetNet Data Plane Overview 18
4.1.3. Network reference model 20
4.2. DetNet systems 21
4.2.1. End system 21
4.2.2. DetNet edge, relay, and transit nodes 22
4.3. DetNet flows 23
4.3.1. DetNet flow types 23
4.3.2. Source transmission behavior 23
4.3.3. Incomplete Networks 25
4.4. Traffic Engineering for DetNet 25
4.4.1. The Application Plane 25
4.4.2. The Controller Plane 26
4.4.3. The Network Plane 26
4.5. Queuing, Shaping, Scheduling, and Preemption 27
4.6. Service instance 28

4.7. Flow identification at technology borders 30
4.7.1. Exporting flow identification 30
4.7.2. Flow attribute mapping between layers 31
4.7.3. Flow-ID mapping examples 32
4.8. Advertising resources, capabilities and adjacencies . . . 34
4.9. Scaling to larger networks 35
4.10. Compatibility with Layer-2 35
5. Security Considerations 35
6. Privacy Considerations 36
7. IANA Considerations 36
8. Acknowledgements 36
9. Informative References 37
Authors' Addresses 40

1. Introduction

Deterministic Networking (DetNet) is a service that can be offered by a network to DetNet flows. DetNet provides these flows with extremely low packet loss rates and assured maximum end-to-end delivery latency. This is accomplished by dedicating network resources such as link bandwidth and buffer space to DetNet flows and/or classes of DetNet flows, and by replicating packets along multiple paths. Unused reserved resources are available to non-DetNet packets.

The Deterministic Networking Problem Statement [I-D.ietf-detnet-problem-statement] introduces Deterministic Networking, and Deterministic Networking Use Cases [I-D.ietf-detnet-use-cases] summarizes the need for it. See [I-D.ietf-detnet-dp-sol-mpls] and [I-D.ietf-detnet-dp-sol-ip] for specific techniques that can be used to identify DetNet flows and assign them to specific paths through a network.

A goal of DetNet is a converged network in all respects. That is, the presence of DetNet flows does not preclude non-DetNet flows, and the benefits offered DetNet flows should not, except in extreme cases, prevent existing QoS mechanisms from operating in a normal fashion, subject to the bandwidth required for the DetNet flows. A single source-destination pair can trade both DetNet and non-DetNet flows. End systems and applications need not instantiate special interfaces for DetNet flows. Networks are not restricted to certain topologies; connectivity is not restricted. Any application that generates a data flow that can be usefully characterized as having a maximum bandwidth should be able to take advantage of DetNet, as long as the necessary resources can be reserved. Reservations can be made by the application itself, via network management, by an application's controller, or by other means, e.g., a dynamic control plane (e.g., [RFC2205]).

Many applications that are intended to be served by Deterministic Networking require the ability to synchronize the clocks in end systems to a sub-microsecond accuracy. Some of the queue control techniques defined in Section 4.5 also require time synchronization among relay and transit nodes. The means used to achieve time synchronization are not addressed in this document. DetNet can accommodate various time synchronization techniques and profiles that are defined elsewhere to address the needs of different market segments.

2. Terminology

2.1. Terms used in this document

The following terms are used in the context of DetNet in this document:

allocation
: Resources are dedicated to support a DetNet flow. Depending on an implementation, the resource may be reused by non-DetNet flows when it is not used by the DetNet flow.

App-flow
: The native format of a DetNet flow.

DetNet destination
: An end system capable of terminating a DetNet flow.

DetNet domain
: The portion of a network that is DetNet aware. It includes end systems and other DetNet nodes.

DetNet flow
: A DetNet flow is a sequence of packets to which the DetNet service is to be provided.

DetNet compound flow and DetNet member flow
: A DetNet compound flow is a DetNet flow that has been separated into multiple duplicate DetNet member flows for service protection at the DetNet service layer. Member flows are merged back into a single DetNet compound flow such that there are no duplicate packets. "Compound" and "member" are strictly relative to each other, not absolutes; a DetNet compound flow comprising multiple DetNet member flows can, in turn, be a member of a higher-order compound.

DetNet intermediate node
: A DetNet relay node or transit node.

DetNet edge node
: An instance of a DetNet relay node that acts as a source and/or destination at the DetNet service layer. For example, it can include 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 terminates congestion protection at the DetNet transport layer, or aggregates DetNet services into new DetNet flows. It is analogous to a Label Edge Router (LER) or a Provider Edge (PE) router.

DetNet-UNI
: User-to-Network Interface with DetNet specific functionalities. It is a packet-based reference point and may provide multiple functions like encapsulation, status, synchronization, etc.

end system
: 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.

link
: A connection between two DetNet nodes. It may be composed of a physical link or a sub-network technology that can provide appropriate traffic delivery for DetNet flows.

DetNet system
: A DetNet aware end system, transit node, or relay node. "DetNet" may be omitted in some text.

DetNet relay node
: 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.

PEF
: 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.

PRF 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.

PREOF Collective name for Packet Replication, Elimination, and Ordering Functions.

POF 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.

reservation
: The set of resources allocated between a source and one or more destinations through transit nodes and subnets associated with a DetNet flow, to provide the provisioned DetNet service.

DetNet service layer
: The layer at which A DetNet service, e.g., service protection is provided.

DetNet service proxy
: Maps between App-flows and DetNet flows.

DetNet source
: An end system capable of originating a DetNet flow.

DetNet transit node
: 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. Typically provides congestion protection over those paths. An MPLS LSR is an example of a DetNet transit node.

DetNet transport layer
: The layer that optionally provides congestion protection for DetNet flows over paths provided by the underlying network.

2.2. IEEE 802.1 TSN to DetNet dictionary

This section also serves as a dictionary for translating from the terms used by the Time-Sensitive Networking (TSN) Task Group [IEEE802.1TSNTG] of the IEEE 802.1 WG to those of the DetNet WG.

Listener
: The IEEE 802.1 term for a destination of a DetNet flow.

relay system
: The IEEE 802.1 term for a DetNet intermediate node.

Stream
: The IEEE 802.1 term for a DetNet flow.

Talker
: The IEEE 802.1 term for the source of a DetNet flow.

3. Providing the DetNet Quality of Service

3.1. Primary goals defining the DetNet QoS

The DetNet Quality of Service can be expressed in terms of:

- Minimum and maximum end-to-end latency from source to destination; timely delivery, and bounded jitter (packet delay variation) derived from these constraints.

- Packet loss ratio, under various assumptions as to the operational states of the nodes and links.

- An upper bound on out-of-order packet delivery. It is worth noting that some DetNet applications are unable to tolerate any out-of-order delivery.

It is a distinction of DetNet that it is concerned solely with worst-case values for the end-to-end latency, jitter, and misordering. Average, mean, or typical values are of little interest, because they do not affect the ability of a real-time system to perform its tasks. In general, a trivial priority-based queuing scheme will give better average latency to a data flow than DetNet, but of course, the worst-case latency can be essentially unbounded.

Three techniques are used by DetNet to provide these qualities of service:

- Congestion protection (Section 3.2.1).

- Service protection (Section 3.2.2).

- Explicit routes (Section 3.2.3).

Congestion protection operates by allocating resources along the path of a DetNet flow, e.g., buffer space or link bandwidth. Congestion

protection greatly reduces, or even eliminates entirely, packet loss due to output packet congestion within the network, but it can only be supplied to a DetNet flow that is limited at the source to a maximum packet size and transmission rate. Note that congestion protection provided via congestion detection and notification is explicitly excluded from consideration in DetNet, as it serves a different set of applications.

Congestion protection addresses two of the DetNet QoS requirements: latency and packet loss. Given that DetNet nodes have a finite amount of buffer space, congestion protection necessarily results in a maximum end-to-end latency. It also addresses the largest contribution to packet loss, which is buffer congestion.

After congestion, the most important contributions to packet loss are typically from random media errors and equipment failures. Service protection is the name for the mechanisms used by DetNet to address these losses. The mechanisms employed are constrained by the requirement to meet the users' latency requirements. Packet replication and elimination (Section 3.2.2) and packet encoding (Section 3.2.2.3) are described in this document to provide service protection; others may be found. For instance, packet encoding can be used to provide service protection against random media errors, packet replication and elimination can be used to provide service protection against equipment failures. This mechanism distributes the contents of DetNet flows over multiple paths in time and/or space, so that the loss of some of the paths does need not cause the loss of any packets.

The paths are typically (but not necessarily) explicit routes, so that they do not normally suffer temporary interruptions caused by the convergence of routing or bridging protocols.

These three techniques can be applied independently, giving eight possible combinations, including none (no DetNet), although some combinations are of wider utility than others. This separation keeps the protocol stack coherent and maximizes interoperability with existing and developing standards in this (IETF) and other Standards Development Organizations. Some examples of typical expected combinations:

- Explicit routes plus service protection are exactly the techniques employed by seamless redundancy mechanisms applied on a ring topology as described, e.g., in [IEC62439-3-2016]. In this example, explicit routes are achieved by limiting the physical topology of the network to a ring. Sequentialization, replication, and duplicate elimination are facilitated by packet

tags added at the front or the end of Ethernet frames. [RFC8227] provides another example in the context of MPLS.

- Congestion protection alone is offered by IEEE 802.1 Audio Video bridging [IEEE802.1BA]. As long as the network suffers no failures, zero congestion loss can be achieved through the use of a reservation protocol (MSRP [IEEE802.1Q-2018]), shapers in every bridge, and proper dimensioning.

- Using all three together gives maximum protection.

There are, of course, simpler methods available (and employed, today) to achieve levels of latency and packet loss that are satisfactory for many applications. Prioritization and over-provisioning is one such technique. However, these methods generally work best in the absence of any significant amount of non-critical traffic in the network (if, indeed, such traffic is supported at all), or work only if the critical traffic constitutes only a small portion of the network's theoretical capacity, or work only if all systems are functioning properly, or in the absence of actions by end systems that disrupt the network's operations.

There are any number of methods in use, defined, or in progress for accomplishing each of the above techniques. It is expected that this DetNet Architecture will assist various vendors, users, and/or "vertical" Standards Development Organizations (dedicated to a single industry) to make selections among the available means of implementing DetNet networks.

3.2. Mechanisms to achieve DetNet QoS

3.2.1. Congestion protection

3.2.1.1. Eliminate congestion loss

The primary means by which DetNet achieves its QoS assurances is to reduce, or even completely eliminate, congestion within a node as a cause of packet loss. Given that a DetNet flow cannot be throttled, this can be achieved only by the provision of sufficient buffer storage at each hop through the network to ensure that no packets are dropped due to a lack of buffer storage.

Ensuring adequate buffering requires, in turn, that the source, and every intermediate node along the path to the destination (or nearly every node, see Section 4.3.3) be careful to regulate its output to not exceed the data rate for any DetNet flow, except for brief periods when making up for interfering traffic. Any packet sent ahead of its time potentially adds to the number of buffers required

by the next hop and may thus exceed the resources allocated for a particular DetNet flow.

The low-level mechanisms described in Section 4.5 provide the necessary regulation of transmissions by an end system or intermediate node to provide congestion protection. The allocation of the bandwidth and buffers for a DetNet flow requires provisioning A DetNet node may have other resources requiring allocation and/or scheduling, that might otherwise be over-subscribed and trigger the rejection of a reservation.

3.2.1.2. Jitter Reduction

A core objective of DetNet is to enable the convergence of sensitive non-IP networks onto a common network infrastructure. This requires the accurate emulation of currently deployed mission-specific networks, which for example rely on point-to-point analog (e.g., 4-20mA modulation) and serial-digital cables (or buses) for highly reliable, synchronized and jitter-free communications. While the latency of analog transmissions is basically the speed of light, legacy serial links are usually slow (in the order of Kbps) compared to, say, GigE, and some latency is usually acceptable. What is not acceptable is the introduction of excessive jitter, which may, for instance, affect the stability of control systems.

Applications that are designed to operate on serial links usually do not provide services to recover the jitter, because jitter simply does not exist there. DetNet flows are generally expected to be delivered in-order and the precise time of reception influences the processes. In order to converge such existing applications, there is a desire to emulate all properties of the serial cable, such as clock transportation, perfect flow isolation and fixed latency. While minimal jitter (in the form of specifying minimum, as well as maximum, end-to-end latency) is supported by DetNet, there are practical limitations on packet-based networks in this regard. In general, users are encouraged to use, instead of, "do this when you get the packet," a combination of:

- Sub-microsecond time synchronization among all source and destination end systems, and

- Time-of-execution fields in the application packets.

Jitter reduction is provided by the mechanisms described in Section 4.5 that also provide congestion protection.

3.2.2. Service Protection

Service protection aims to mitigate or eliminate packet loss due to equipment failures, random media and/or memory faults. These types of packet loss can be greatly reduced by spreading the data over multiple disjoint forwarding paths. Various service protection methods are described in [RFC6372], e.g., 1+1 linear protection. This section describes the functional details of an additional method in Section 3.2.2.2, which can be implemented as described in Section 3.2.2.3 or as specified in [I-D.ietf-detnet-dp-sol-mpls] in order to provide 1+n hitless protection. The appropriate service protection mechanism depends on the scenario and the requirements.

3.2.2.1. In-Order Delivery

Out-of-order packet delivery can be a side effect of service protection. Packets delivered out-of-order impact the amount of buffering needed at the destination to properly process the received data. Such packets also influence the jitter of a flow. The DetNet service includes maximum allowed misordering as a constraint. Zero misordering would be a valid service constraint to reflect that the end system(s) of the flow cannot tolerate any out-of-order delivery. DetNet Packet Ordering Functionality (POF) (Section 3.2.2.2) can be used to provide in-order delivery.

3.2.2.2. Packet Replication and Elimination

This section describes a service protection method that sends copies of the same packets over multiple paths.

The DetNet service layer includes the packet replication (PRF), the packet elimination (PEF), and the packet ordering functionality (POF) for use in DetNet edge, relay node, and end system packet processing. Either of these functions can be enabled in a DetNet edge node, relay node or end system. The collective name for all three functions is PREOF. The packet replication and elimination service protection method altogether involves four capabilities:

- Providing sequencing information to the packets of a DetNet compound flow. This may be done by adding a sequence number or time stamp as part of DetNet, or may be inherent in the packet, e.g., in a transport protocol, or associated to other physical properties such as the precise time (and radio channel) of reception of the packet. This is typically done once, at or near the source.

- The Packet Replication Function (PRF) replicates these packets into multiple DetNet member flows and typically sends them along

multiple different paths to the destination(s), e.g., over the explicit routes of Section 3.2.3. The location within a node, and the mechanism used for the PRF is implementation specific.

- The Packet Elimination Function (PEF) eliminates duplicate packets of a DetNet flow based on the sequencing information and a history of received packets. The output of the PEF is always a single packet. This may be done at any node along the path to save network resources further downstream, in particular if multiple Replication points exist. But the most common case is to perform this operation at the very edge of the DetNet network, preferably in or near the receiver. The location within a node, and mechanism used for the PEF is implementation specific.

- The Packet Ordering Function (POF) uses the sequencing information to re-order a DetNet flow's packets that are received out of order.

The order in which a node applies PEF, POF, and PRF to a DetNet flow is implementation specific.

Some service protection mechanisms rely on switching from one flow to another when a failure of a flow is detected. Contrarily, packet replication and elimination combines the DetNet member flows sent along multiple different paths, and performs a packet-by-packet selection of which to discard, e.g., based on sequencing information.

In the simplest case, this amounts to replicating each packet in a source that has two interfaces, and conveying them through the network, along separate (disjoint non-SRLG) paths, to the similarly dual-homed destinations, that discard the extras. This ensures that one path (with zero congestion loss) remains, even if some intermediate node fails. The sequencing information can also be used for loss detection and for re-ordering.

DetNet relay nodes in the network can provide replication and elimination facilities at various points in the network, so that multiple failures can be accommodated.

This is shown in Figure 1, where the two relay nodes each replicate (R) the DetNet flow on input, sending the DetNet member flows to both the other relay node and to the end system, and eliminate duplicates (E) on the output interface to the right-hand end system. Any one link in the network can fail, and the DetNet compound flow can still get through. Furthermore, two links can fail, as long as they are in different segments of the network.

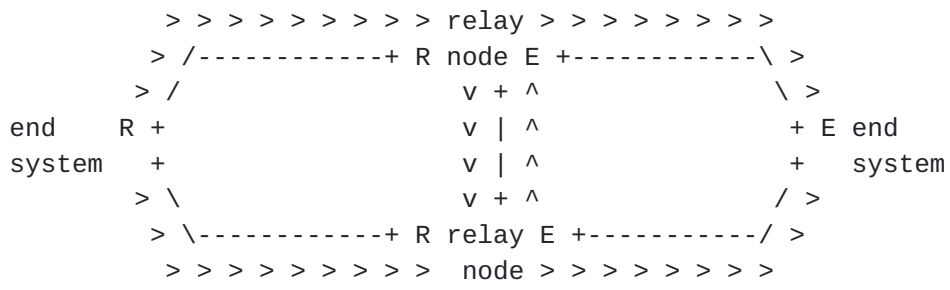


Figure 1: Packet replication and elimination

Packet replication and elimination does not react to and correct failures; it is entirely passive. Thus, intermittent failures, mistakenly created packet filters, or misrouted data is handled just the same as the equipment failures that are handled by typical routing and bridging protocols.

If packet replication and elimination is used over paths providing congestion protection (Section 3.2.1), and member flows that take different-length paths through the network are combined, a merge point may require extra buffering to equalize the delays over the different paths. This equalization ensures that the resultant compound flow will not exceed its contracted bandwidth even after one or the other of the paths is restored after a failure. The extra buffering can be also used to provide in-order delivery.

3.2.2.3. Packet encoding for service protection

There are methods for using multiple paths to provide service protection that involve encoding the information in a packet belonging to a DetNet flow into multiple transmission units, combining information from multiple packets into any given transmission unit. Such techniques, also known as "network coding", can be used as a DetNet service protection technique.

3.2.3. Explicit routes

In networks controlled by typical dynamic control protocols such as IS-IS or OSPF, a network topology event in one part of the network can impact, at least briefly, the delivery of data in parts of the network remote from the failure or recovery event. Even the use of redundant paths through a network defined, e.g., as defined by [RFC6372] do not eliminate the chances of packet loss. Furthermore, out-of-order packet delivery can be a side effect of route changes.

Many real-time networks rely on physical rings of two-port devices, with a relatively simple ring control protocol. This supports redundant paths for service protection with a minimum of wiring. As

an additional benefit, ring topologies can often utilize different topology management protocols than those used for a mesh network, with a consequent reduction in the response time to topology changes. Of course, this comes at some cost in terms of increased hop count, and thus latency, for the typical path.

In order to get the advantages of low hop count and still ensure against even very brief losses of connectivity, DetNet employs explicit routes, where the path taken by a given DetNet flow does not change, at least immediately, and likely not at all, in response to network topology events. Service protection (Section 3.2.2 or Section 3.2.2.3) over explicit routes provides a high likelihood of continuous connectivity. Explicit routes can be established in various ways, e.g., with RSVP-TE [RFC3209], with Segment Routing (SR) [RFC8402], via a Software Defined Networking approach [RFC7426], with IS-IS [RFC7813], etc. Explicit routes are typically used in MPLS TE LSPs.

Out-of-order packet delivery can be a side effect of distributing a single flow over multiple paths especially when there is a change from one path to another when combining the flow. This is irrespective of the distribution method used, and also applies to service protection over explicit routes. As described in Section 3.2.2.1, out-of-order packets influence the jitter of a flow and impact the amount of buffering needed to process the data; therefore, DetNet service includes maximum allowed misordering as a constraint. The use of explicit routes helps to provide in-order delivery because there is no immediate route change with the network topology, but the changes are plannable as they are between the different explicit routes.

3.3. Secondary goals for DetNet

Many applications require DetNet to provide additional services, including coexistence with other QoS mechanisms Section 3.3.1 and protection against misbehaving transmitters Section 3.3.2.

3.3.1. Coexistence with normal traffic

A DetNet network supports the dedication of a high proportion (e.g. 75%) of the network bandwidth to DetNet flows. But, no matter how much is dedicated for DetNet flows, it is a goal of DetNet to coexist with existing Class of Service schemes (e.g., DiffServ). It is also important that non-DetNet traffic not disrupt the DetNet flow, of course (see Section 3.3.2 and Section 5). For these reasons:

- Bandwidth (transmission opportunities) not utilized by a DetNet flow is available to non-DetNet packets (though not to other DetNet flows).

- DetNet flows can be shaped or scheduled, in order to ensure that the highest-priority non-DetNet packet is also ensured a worst-case latency (at any given hop).

- When transmission opportunities for DetNet flows are scheduled in detail, then the algorithm constructing the schedule should leave sufficient opportunities for non-DetNet packets to satisfy the needs of the users of the network. Detailed scheduling can also permit the time-shared use of buffer resources by different DetNet flows.

Ideally, the net effect of the presence of DetNet flows in a network on the non-DetNet packets is primarily a reduction in the available bandwidth.

3.3.2. Fault Mitigation

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 allowing for 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.

4. DetNet Architecture

4.1. DetNet stack model

DetNet functionality (Section 3) is implemented in two adjacent layers in the protocol stack: the DetNet service layer and the DetNet transport layer. The DetNet service layer provides DetNet service, e.g., service protection, to higher layers in the protocol stack and applications. The DetNet transport layer supports DetNet service in the underlying network, e.g., by providing explicit routes and congestion protection to DetNet flows.

4.1.1. Representative Protocol Stack Model

Figure 2 illustrates a conceptual DetNet data plane layering model. One may compare it to that in [IEEE802.1CB], Annex C.

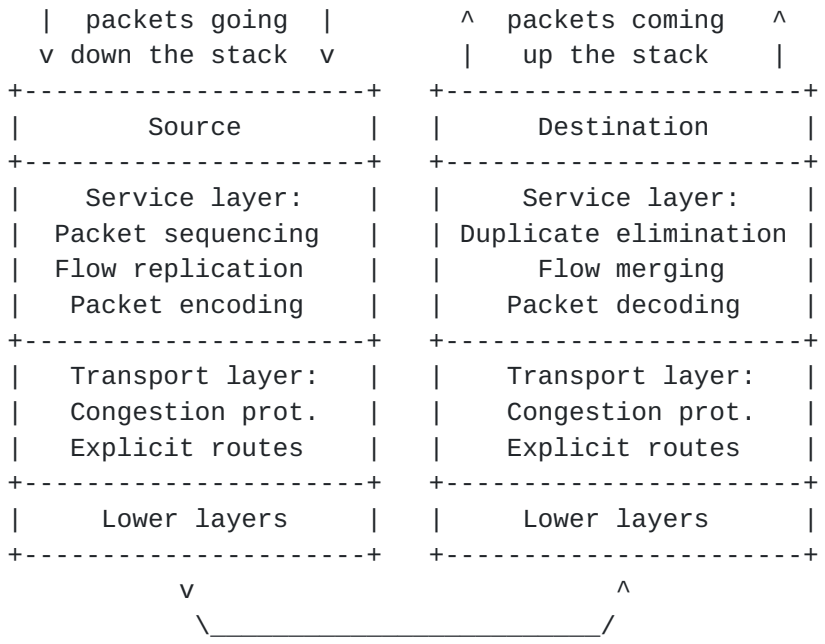


Figure 2: DetNet data plane protocol stack

Not all layers are required for any given application, or even for any given network. The functionality shown in Figure 2 is:

Application
: Shown as "source" and "destination" in the diagram.

Packet sequencing
: As part of DetNet service protection, supplies the sequence number for packet replication and elimination (Section 3.2.2). Peers with Duplicate elimination. This

layer is not needed if a higher-layer transport protocol is expected to perform any packet sequencing and duplicate elimination required by the DetNet flow replication.

Duplicate elimination
: As part of the DetNet service layer, based on the sequenced number supplied by its peer, packet sequencing, Duplicate elimination discards any duplicate packets generated by DetNet flow replication. It can operate on member flows, compound flows, or both. The replication may also be inferred from other information such as the precise time of reception in a scheduled network. The duplicate elimination layer may also perform resequencing of packets to restore packet order in a flow that was disrupted by the loss of packets on one or another of the multiple paths taken.

Flow replication
: As part of DetNet service protection, packets that belong to a DetNet compound flow are replicated into two or more DetNet member flows. This function is separate from packet sequencing. Flow replication can be an explicit replication and remarking of packets, or can be performed by, for example, techniques similar to ordinary multicast replication, albeit with resource allocation implications. Peers with DetNet flow merging.

Flow merging
: As part of DetNet service protection, merges DetNet member flows together for packets coming up the stack belonging to a specific DetNet compound flow. Peers with DetNet flow replication. DetNet flow merging, together with packet sequencing, duplicate elimination, and DetNet flow replication perform packet replication and elimination (Section 3.2.2).

Packet encoding
: As part of DetNet service protection, as an alternative to packet sequencing and flow replication, packet encoding combines the information in multiple DetNet packets, perhaps from different DetNet compound flows, and transmits that information in packets on different DetNet member Flows. Peers with Packet decoding.

Packet decoding
: As part of DetNet service protection, as an alternative to flow merging and duplicate elimination, packet decoding takes packets from different DetNet member flows, and computes from

those packets the original DetNet packets from the compound flows input to packet encoding. Peers with Packet encoding.

Congestion protection
: The DetNet transport layer provides congestion protection. See Section 4.5. The actual queuing and shaping mechanisms are typically provided by underlying subnet layers, these can be closely associated with the means of providing paths for DetNet flows, the path and the congestion protection are conflated in this figure.

Explicit routes
: The DetNet transport layer provides mechanisms to ensure that fixed paths are provided for DetNet flows. These explicit paths avoid the impact of network convergence.

Operations, Administration, and Maintenance (OAM) leverages in-band and out-of-band signaling that validates whether the service is effectively obtained within QoS constraints. OAM is not shown in Figure 2; it may reside in any number of the layers. OAM can involve specific tagging added in the packets for tracing implementation or network configuration errors; traceability enables to find whether a packet is a replica, which relay node performed the replication, and which segment was intended for the replica. Active and hybrid OAM methods require additional bandwidth to perform fault management and performance monitoring of the DetNet domain. OAM may, for instance, generate special test probes or add OAM information into the data packet.

The packet sequencing and replication elimination functions at the source and destination ends of a DetNet compound flow may be performed either in the end system or in a DetNet relay node.

4.1.2. DetNet Data Plane Overview

A "Deterministic Network" will be composed of DetNet enabled end systems and nodes, i.e., edge nodes, relay nodes and collectively deliver DetNet services. DetNet enabled nodes are interconnected via transit nodes (e.g., LSRs) which support DetNet, but are not DetNet service aware. All DetNet enabled nodes are connected to sub-networks, where a point-to-point link is also considered as a simple sub-network. These sub-networks will provide DetNet compatible service for support of DetNet traffic. Examples of sub-networks include MPLS TE, IEEE 802.1 TSN and OTN. Of course, multi-layer DetNet systems may also be possible, where one DetNet appears as a sub-network, and provides service to, a higher layer DetNet system. A simple DetNet concept network is shown in Figure 3.

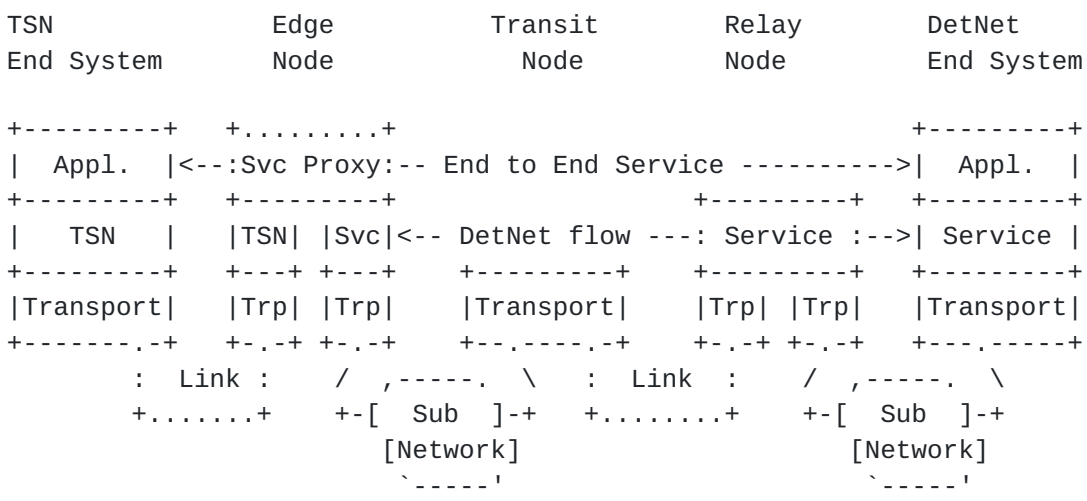


Figure 3: A Simple DetNet Enabled Network

Distinguishing the function of two DetNet data plane layers, the DetNet service layer and the DetNet transport layer, helps to explore and evaluate various combinations of the data plane solutions available, some are illustrated in Figure 4. This separation of DetNet layers, while helpful, should not be considered as formal requirement. For example, some technologies may violate these strict layers and still be able to deliver a DetNet service.

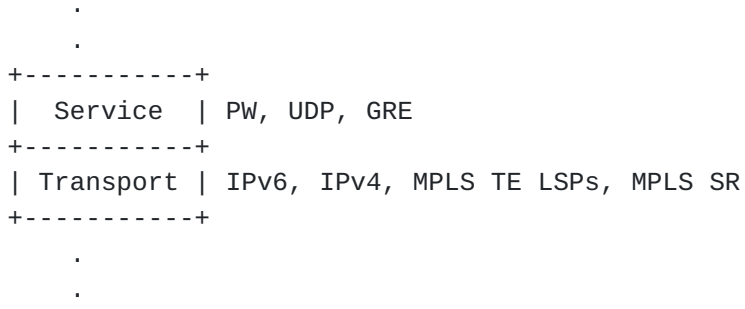


Figure 4: DetNet adaptation to data plane

In some networking scenarios, the end system initially provides a DetNet flow encapsulation, which contains all information needed by DetNet nodes (e.g., Real-time Transport Protocol (RTP) [RFC3550] based DetNet flow transported over a native UDP/IP network or PseudoWire). In other scenarios, the encapsulation formats might differ significantly.

There are many valid options to create a data plane solution for DetNet traffic by selecting a technology approach for the DetNet service layer and also selecting a technology approach for the DetNet transport layer. There are a high number of valid combinations.

One of the most fundamental differences between different potential data plane options is the basic headers used by DetNet nodes. For example, the basic service can be delivered based on an MPLS label or an IP header. This decision impacts the basic forwarding logic for the DetNet service layer. Note that in both cases, IP addresses are used to address DetNet nodes. The selected DetNet transport layer technology also needs to be mapped to the sub-net technology used to interconnect DetNet nodes. For example, DetNet flows will need to be mapped to TSN Streams.

4.1.3. Network reference model

Figure 5 shows another view of the DetNet service related reference points and main components.

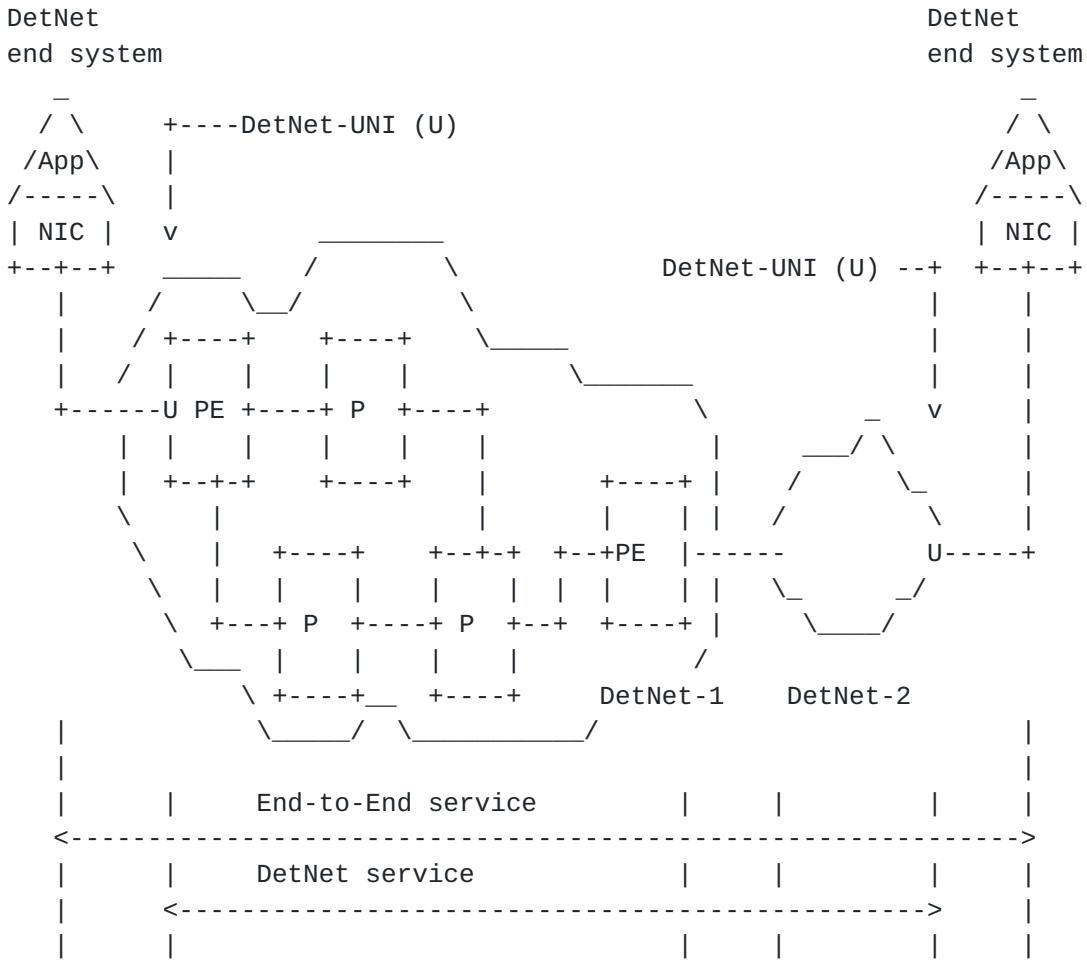


Figure 5: DetNet Service Reference Model (multi-domain)

DetNet-UNIs ("U" in Figure 5) are assumed in this document to be packet-based reference points and provide connectivity over the packet network. A DetNet-UNI may provide multiple functions, e.g.,

it may add networking technology specific encapsulation to the DetNet flows if necessary; it may provide status of the availability of the resources associated with a reservation; it may provide a synchronization service for the end system; it may carry enough signaling to place the reservation in a network without a controller, or if the controller only deals with the network but not the end systems. Internal reference points of end systems (between the application and the NIC) are more challenging from control perspective and they may have extra requirements (e.g., in-order delivery is expected in end system internal reference points, whereas it is considered optional over the DetNet-UNI).

4.2. DetNet systems

4.2.1. End system

The native data flow between the source/destination end systems is referred to as application-flow (App-flow). The traffic characteristics of an App-flow can be CBR (constant bit rate) or VBR (variable bit rate) and can have L1 or L2 or L3 encapsulation (e.g., TDM (time-division multiplexing), Ethernet, IP). These characteristics are considered as input for resource reservation and might be simplified to ensure determinism during transport (e.g., making reservations for the peak rate of VBR traffic, etc.).

An end system may or may not be DetNet transport layer aware or DetNet service layer aware. That is, an end system may or may not contain DetNet specific functionality. End systems with DetNet functionalities may have the same or different transport layer as the connected DetNet domain. Categorization of end systems are shown in Figure 6.

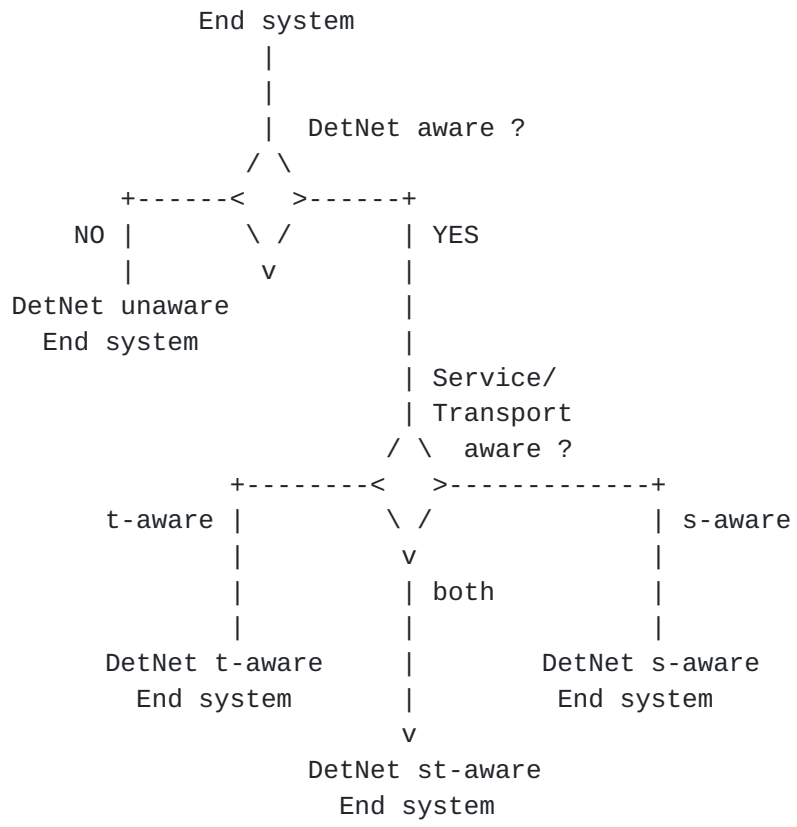


Figure 6: Categorization of end systems

Note some known use case examples for end systems:

- DetNet unaware: The classic case requiring service proxies.

- DetNet t-aware: A DetNet transport-aware system. It knows about some TSN functions (e.g., reservation), but not about service protection.

- DetNet s-aware: A DetNet service-aware system. It supplies sequence numbers, but doesn't know about zero congestion loss.

- DetNet st-aware: A full functioning DetNet end system, it has DetNet functionalities and usually the same forwarding paradigm as the connected DetNet domain. It can be treated as an integral part of the DetNet domain.

4.2.2. DetNet edge, relay, and transit nodes

As shown in Figure 3, DetNet edge nodes providing proxy service and DetNet relay nodes providing the DetNet service layer are DetNet-aware, and DetNet transit nodes need only be aware of the DetNet transport layer.

In general, if a DetNet flow passes through one or more DetNet-unaware network nodes between two DetNet nodes providing the DetNet transport layer for that flow, there is a potential for disruption or failure of the DetNet QoS. A network administrator needs to ensure that the DetNet-unaware network nodes are configured to minimize the chances of packet loss and delay, and provision enough extra buffer space in the DetNet transit node following the DetNet-unaware network nodes to absorb the induced latency variations.

4.3. DetNet flows

4.3.1. DetNet flow types

A DetNet flow can have different formats while it is transported between the peer end systems. Therefore, the following possible types / formats of a DetNet flow are distinguished in this document:

- App-flow: native format of the data carried over a DetNet flow. It does not contain any DetNet related attributes.

- DetNet-t-flow: specific format of a DetNet flow. Only requires the congestion / latency features provided by the DetNet transport layer.

- DetNet-s-flow: specific format of a DetNet flow. Only requires the service protection feature ensured by the DetNet service layer.

- DetNet-st-flow: specific format of a DetNet flow. It requires both DetNet service layer and DetNet transport layer functions during forwarding.

4.3.2. Source transmission behavior

For the purposes of congestion protection, DetNet flows can be synchronous or asynchronous. In synchronous DetNet flows, at least the intermediate nodes (and possibly the end systems) are closely time synchronized, typically to better than 1 microsecond. By transmitting packets from different DetNet flows or classes of DetNet flows at different times, using repeating schedules synchronized among the intermediate nodes, resources such as buffers and link bandwidth can be shared over the time domain among different DetNet flows. There is a tradeoff among techniques for synchronous DetNet flows between the burden of fine-grained scheduling and the benefit of reducing the required resources, especially buffer space.

In contrast, asynchronous DetNet flows are not coordinated with a fine-grained schedule, so relay and end systems must assume worst-

case interference among DetNet flows contending for buffer resources. Asynchronous DetNet flows are characterized by:

- A maximum packet size;
- An observation interval; and
- A maximum number of transmissions during that observation interval.

These parameters, together with knowledge of the protocol stack used (and thus the size of the various headers added to a packet), limit the number of bit times per observation interval that the DetNet flow can occupy the physical medium.

The source is required not to exceed these limits in order to obtain DetNet service. If the source transmits less data than this limit allows, the unused resource such as link bandwidth can be made available by the system to non-DetNet packets. However, making those resources available to DetNet packets in other DetNet flows would serve no purpose. Those other DetNet flows have their own dedicated resources, on the assumption that all DetNet flows can use all of their resources over a long period of time.

There is no provision in DetNet for throttling DetNet flows (reducing end-to-end transmission rate via any explicit congestion notification); the assumption is that a DetNet flow, to be useful, must be delivered in its entirety. That is, while any useful application is written to expect a certain number of lost packets, the real-time applications of interest to DetNet demand that the loss of data due to the network is an extraordinarily event.

Although DetNet strives to minimize the changes required of an application to allow it to shift from a special-purpose digital network to an Internet Protocol network, one fundamental shift in the behavior of network applications is impossible to avoid: the reservation of resources before the application starts. In the first place, a network cannot deliver finite latency and practically zero packet loss to an arbitrarily high offered load. Secondly, achieving practically zero packet loss for unthrottled (though bandwidth limited) DetNet flows means that bridges and routers have to dedicate buffer resources to specific DetNet flows or to classes of DetNet flows. The requirements of each reservation have to be translated into the parameters that control each system's queuing, shaping, and scheduling functions and delivered to the hosts, bridges, and routers.

4.3.3. Incomplete Networks

The presence in the network of transit nodes or subnets that are not fully capable of offering DetNet services complicates the ability of the intermediate nodes and/or controller to allocate resources, as extra buffering must be allocated at points downstream from the non-DetNet intermediate node for a DetNet flow. This extra buffering may increase latency and/or jitter.

4.4. Traffic Engineering for DetNet

Traffic Engineering Architecture and Signaling (TEAS) [TEAS] defines traffic-engineering architectures for generic applicability across packet and non-packet networks. From a TEAS perspective, Traffic Engineering (TE) refers to techniques that enable operators to control how specific traffic flows are treated within their networks.

Because if its very nature of establishing explicit optimized paths, Deterministic Networking can be seen as a new, specialized branch of Traffic Engineering, and inherits its architecture with a separation into planes.

The Deterministic Networking architecture is thus composed of three planes, a (User) Application Plane, a Controller Plane, and a Network Plane, which echoes that of Figure 1 of Software-Defined Networking (SDN): Layers and Architecture Terminology [RFC7426].:

4.4.1. The Application Plane

Per [RFC7426], the Application Plane includes both applications and services. In particular, the Application Plane incorporates the User Agent, a specialized application that interacts with the end user / operator and performs requests for Deterministic Networking services via an abstract Flow Management Entity, (FME) which may or may not be collocated with (one of) the end systems.

At the Application Plane, a management interface enables the negotiation of flows between end systems. An abstraction of the flow called a Traffic Specification (TSpec) provides the representation. This abstraction is used to place a reservation over the (Northbound) Service Interface and within the Application plane. It is associated with an abstraction of location, such as IP addresses and DNS names, to identify the end systems and eventually specify intermediate nodes.

4.4.2. The Controller Plane

The Controller Plane corresponds to the aggregation of the Control and Management Planes in [RFC7426], though Common Control and Measurement Plane (CCAMP) [CCAMP] makes an additional distinction between management and measurement. When the logical separation of the Control, Measurement and other Management entities is not relevant, the term Controller Plane is used for simplicity to represent them all, and the term Controller Plane Function (CPF) refers to any device operating in that plane, whether is it a Path Computation Element (PCE) [RFC4655], or a Network Management entity (NME), or a distributed control plane. The CPF is a core element of a controller, in charge of computing Deterministic paths to be applied in the Network Plane.

A (Northbound) Service Interface enables applications in the Application Plane to communicate with the entities in the Controller Plane as illustrated in Figure 7.

One or more CPF(s) collaborate to implement the requests from the FME as Per-Flow Per-Hop Behaviors installed in the intermediate nodes for each individual flow. The CPFs place each flow along a deterministic sequence of intermediate nodes so as to respect per-flow constraints such as security and latency, and optimize the overall result for metrics such as an abstract aggregated cost. The deterministic sequence can typically be more complex than a direct sequence and include redundancy path, with one or more packet replication and elimination points.

4.4.3. The Network Plane

The Network Plane represents the network devices and protocols as a whole, regardless of the Layer at which the network devices operate. It includes Forwarding Plane (data plane), Application, and Operational Plane (e.g., OAM) aspects.

The network Plane comprises the Network Interface Cards (NIC) in the end systems, which are typically IP hosts, and intermediate nodes, which are typically IP routers and switches. Network-to-Network Interfaces such as used for Traffic Engineering path reservation in [RFC5921], as well as User-to-Network Interfaces (UNI) such as provided by the Local Management Interface (LMI) between network and end systems, are both part of the Network Plane, both in the control plane and the data plane.

A Southbound (Network) Interface enables the entities in the Controller Plane to communicate with devices in the Network Plane as

illustrated in Figure 7. This interface leverages and extends TEAS to describe the physical topology and resources in the Network Plane.

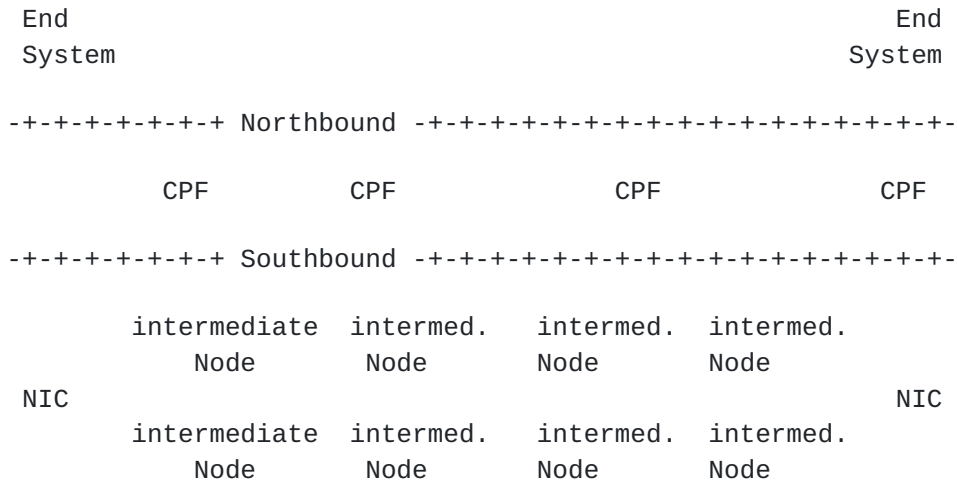


Figure 7: Northbound and Southbound interfaces

The intermediate nodes (and eventually the end systems NIC) expose their capabilities and physical resources to the controller (the CPF), and update the CPFs with their dynamic perception of the topology, across the Southbound Interface. In return, the CPFs set the per-flow paths up, providing a Flow Characterization that is more tightly coupled to the intermediate node Operation than a TSpec.

At the Network plane, intermediate nodes may exchange information regarding the state of the paths, between adjacent systems and eventually with the end systems, and forward packets within constraints associated to each flow, or, when unable to do so, perform a last resort operation such as drop or declassify.

This document focuses on the Southbound interface and the operation of the Network Plane.

4.5. Queuing, Shaping, Scheduling, and Preemption

DetNet achieves congestion protection and bounded delivery latency by reserving bandwidth and buffer resources at every hop along the path of the DetNet flow. The reservation itself is not sufficient, however. Implementors and users of a number of proprietary and standard real-time networks have found that standards for specific data plane techniques are required to enable these assurances to be made in a multi-vendor network. The fundamental reason is that latency variation in one system results in the need for extra buffer space in the next-hop system(s), which in turn, increases the worst-case per-hop latency.

Standard queuing and transmission selection algorithms allow a central controller to compute the latency contribution of each transit node to the end-to-end latency, to compute the amount of buffer space required in each transit node for each incremental DetNet flow, and most importantly, to translate from a flow specification to a set of values for the managed objects that control each relay or end system. For example, the IEEE 802.1 WG has specified (and is specifying) a set of queuing, shaping, and scheduling algorithms that enable each transit node (bridge or router), and/or a central controller, to compute these values. These algorithms include:

- A credit-based shaper [IEEE802.1Qav] (superseded by [IEEE802.1Q-2018]).

- Time-gated queues governed by a rotating time schedule, synchronized among all transit nodes [IEEE802.1Qbv] (superseded by [IEEE802.1Q-2018]).

- Synchronized double (or triple) buffers driven by synchronized time ticks. [IEEE802.1Qch] (superseded by [IEEE802.1Q-2018]).

- Pre-emption of an Ethernet packet in transmission by a packet with a more stringent latency requirement, followed by the resumption of the preempted packet [IEEE802.1Qbu] (superseded by [IEEE802.1Q-2018]), [IEEE802.3br] (superseded by [IEEE802.3-2018]).

While these techniques are currently embedded in Ethernet [IEEE802.3-2018] and bridging standards, we can note that they are all, except perhaps for packet preemption, equally applicable to other media than Ethernet, and to routers as well as bridges. Other media may have its own methods, see, e.g., [I-D.ietf-6tisch-architecture], [RFC7554]. DetNet may include such definitions in the future, or may define how these techniques can be used by DetNet nodes.

4.6. Service instance

A Service instance represents all the functions required on a node to allow the end-to-end service between the UNIs.

The DetNet network general reference model is shown in Figure 8 for a DetNet service scenario (i.e., between two DetNet-UNIs). In this figure, end systems ("A" and "B") are connected directly to the edge nodes of an IP/MPLS network ("PE1" and "PE2"). End systems participating in DetNet communication may require connectivity before setting up an App-flow that requires the DetNet service. Such a

connectivity related service instance and the one dedicated for DetNet service share the same access. Packets belonging to a DetNet flow are selected by a filter configured on the access ("F1" and "F2"). As a result, data flow specific access ("access-A + F1" and "access-B + F2") are terminated in the flow specific service instance ("SI-1" and "SI-2"). A tunnel is used to provide connectivity between the service instances.

The tunnel is used to transport exclusively the packets of the DetNet flow between "SI-1" and "SI-2". The service instances are configured to implement DetNet functions and a flow specific DetNet transport. The service instance and the tunnel may or may not be shared by multiple DetNet flows. Sharing the service instance by multiple DetNet flows requires properly populated forwarding tables of the service instance.

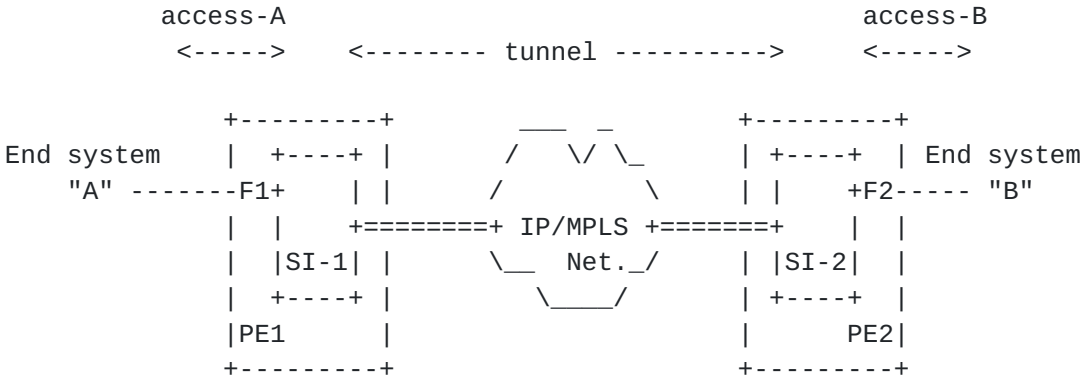


Figure 8: DetNet network general reference model

The tunnel between the service instances may have some special characteristics. For example, in case of a DetNet L3 service, there are differences in the usage of the PW for DetNet traffic compared to the network model described in [RFC6658]. In the DetNet scenario, the PW is likely to be used exclusively by the DetNet flow, whereas [RFC6658] states: "The packet PW appears as a single point-to-point link to the client layer. Network-layer adjacency formation and maintenance between the client equipment will follow the normal practice needed to support the required relationship in the client layer ... This packet PseudoWire is used to transport all of the required Layer-2 and Layer-3 protocols between LSR1 and LSR2". Further details are network technology specific and can be found in [I-D.ietf-detnet-dp-sol-mpls] and [I-D.ietf-detnet-dp-sol-ip].

4.7. Flow identification at technology borders

4.7.1. Exporting flow identification

A DetNet node may need to map specific flows to lower layer flows (or Streams) in order to provide specific queuing and shaping services for specific flows. For example:

- A non-IP, strictly L2 source end system X may be sending multiple flows to the same L2 destination end system Y. Those flows may include DetNet flows with different QoS requirements, and may include non-DetNet flows.

- A router may be sending any number of flows to another router. Again, those flows may include DetNet flows with different QoS requirements, and may include non-DetNet flows.

- Two routers may be separated by bridges. For these bridges to perform any required per-flow queuing and shaping, they must be able to identify the individual flows.

- A Label Edge Router (LER) may have a Label Switched Path (LSP) set up for handling traffic destined for a particular IP address carrying only non-DetNet flows. If a DetNet flow to that same address is requested, a separate LSP may be needed, in order that all of the Label Switch Routers (LSRs) along the path to the destination give that flow special queuing and shaping.

The need for a lower-layer node to be aware of individual higher-layer flows is not unique to DetNet. But, given the endless complexity of layering and relayering over tunnels that is available to network designers, DetNet needs to provide a model for flow identification that is better than packet inspection. That is not to say that packet inspection to layer 4 or 5 addresses will not be used, or the capability standardized; but, there are alternatives.

A DetNet relay node can connect DetNet flows on different paths using different flow identification methods. For example:

- A single unicast DetNet flow passing from router A through a bridged network to router B may be assigned a TSN Stream identifier that is unique within that bridged network. The bridges can then identify the flow without accessing higher-layer headers. Of course, the receiving router must recognize and accept that TSN Stream.

- A DetNet flow passing from LSR A to LSR B may be assigned a different label than that used for other flows to the same IP destination.

In any of the above cases, it is possible that an existing DetNet flow can be an aggregate carrying multiple other DetNet flows. (Not to be confused with DetNet compound vs. member flows.) Of course, this requires that the aggregate DetNet flow be provisioned properly to carry the aggregated flows.

Thus, rather than packet inspection, there is the option to export higher-layer information to the lower layer. The requirement to support one or the other method for flow identification (or both) is a complexity that is part of DetNet control models.

4.7.2. Flow attribute mapping between layers

Transport of DetNet flows over multiple technology domains may require that lower layers are aware of specific flows of higher layers. Such an "exporting of flow identification" is needed each time when the forwarding paradigm is changed on the transport path (e.g., two LSRs are interconnected by a L2 bridged domain, etc.). The three representative forwarding methods considered for deterministic networking are:

- IP routing
- MPLS label switching
- Ethernet bridging

A packet with corresponding Flow-IDs is illustrated in Figure 9, which also indicates where each Flow-ID can be added or removed.

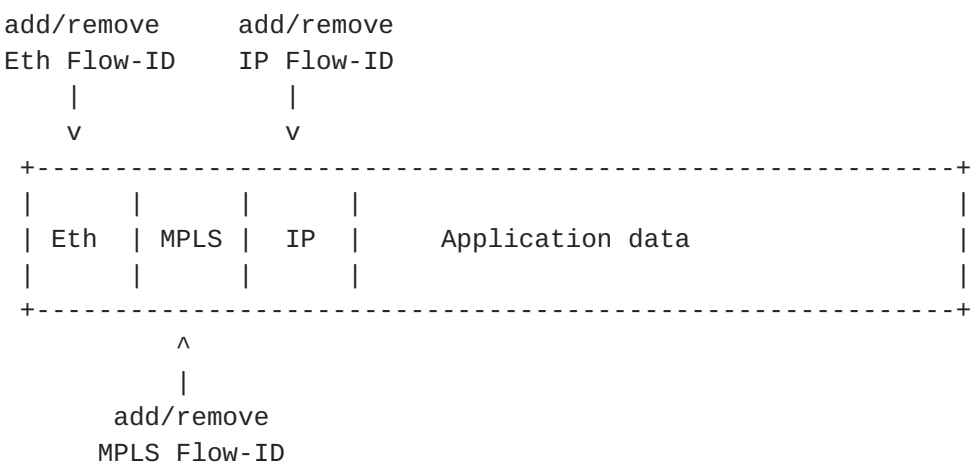


Figure 9: Packet with multiple Flow-IDs

The additional (domain specific) Flow-ID can be

- created by a domain specific function or
- derived from the Flow-ID added to the App-flow.

The Flow-ID must be unique inside a given domain. Note that the Flow-ID added to the App-flow is still present in the packet, but transport nodes may lack the function to recognize it; that's why the additional Flow-ID is added.

4.7.3. Flow-ID mapping examples

IP nodes and MPLS nodes are assumed to be configured to push such an additional (domain specific) Flow-ID when sending traffic to an Ethernet switch (as shown in the examples below).

Figure 10 shows a scenario where an IP end system ("IP-A") is connected via two Ethernet switches ("ETH-n") to an IP router ("IP-1").

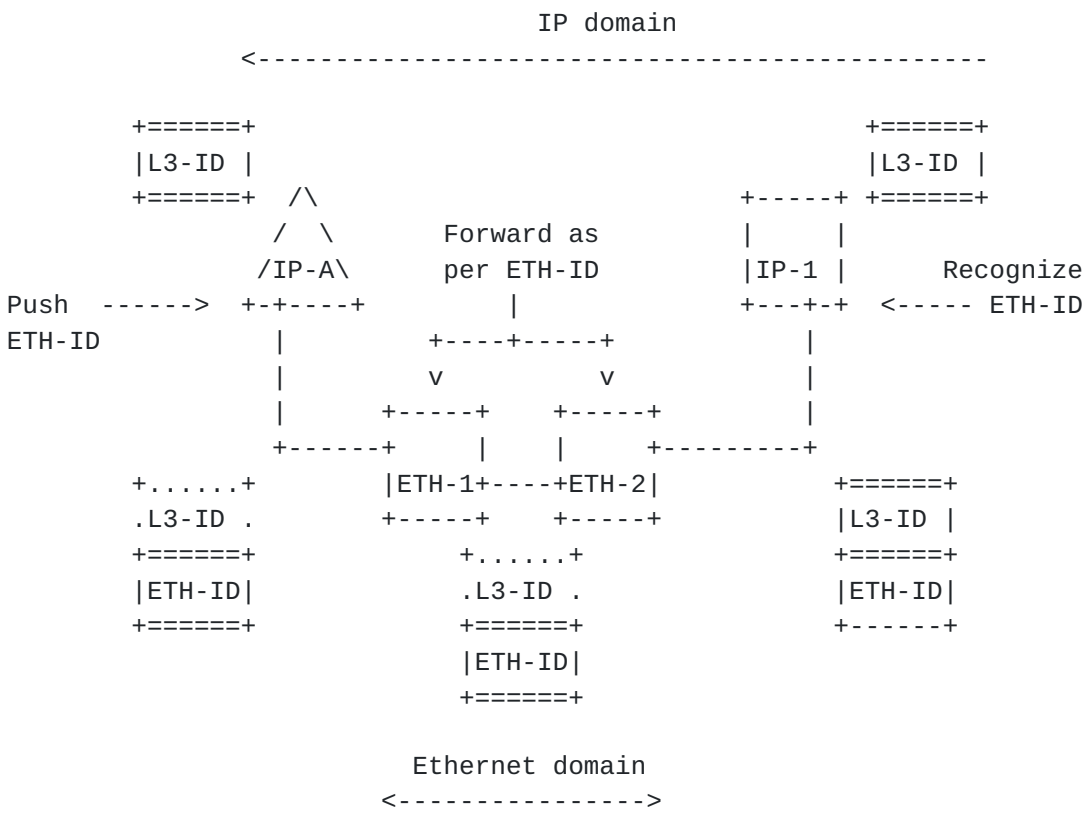


Figure 10: IP nodes interconnected by an Ethernet domain

End system "IP-A" uses the original App-flow specific ID ("L3-ID"), but as it is connected to an Ethernet domain it has to push an Ethernet-domain specific flow-ID ("VID + multicast MAC address", referred as "ETH-ID") before sending the packet to "ETH-1" node. Ethernet switch "ETH-1" can recognize the data flow based on the "ETH-ID" and it does forwarding toward "ETH-2". "ETH-2" switches the packet toward the IP router. "IP-1" must be configured to receive the Ethernet Flow-ID specific multicast flow, but (as it is an L3 node) it decodes the data flow ID based on the "L3-ID" fields of the received packet.

Figure 11 shows a scenario where MPLS domain nodes ("PE-n" and "P-m") are connected via two Ethernet switches ("ETH-n").

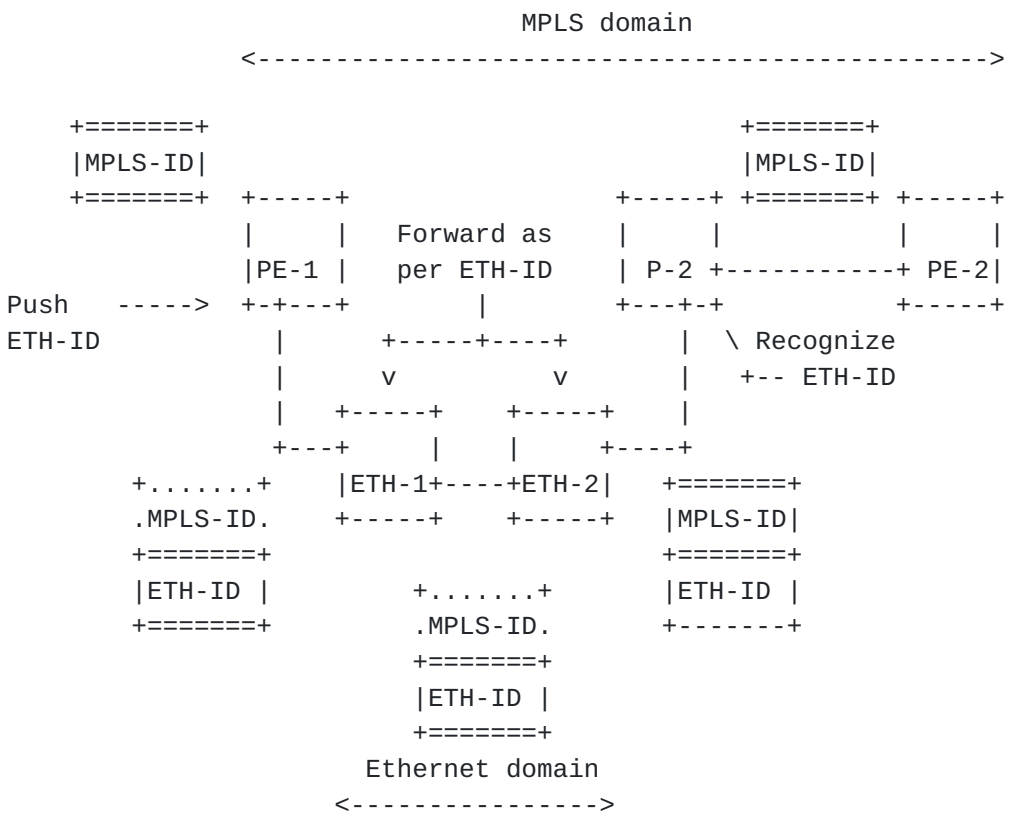


Figure 11: MPLS nodes interconnected by an Ethernet domain

"PE-1" uses the MPLS specific ID ("MPLS-ID"), but as it is connected to an Ethernet domain it has to push an Ethernet-domain specific flow-ID ("VID + multicast MAC address", referred as "ETH-ID") before sending the packet to "ETH-1". Ethernet switch "ETH-1" can recognize the data flow based on the "ETH-ID" and it does forwarding toward "ETH-2". "ETH-2" switches the packet toward the MPLS node ("P-2"). "P-2" must be configured to receive the Ethernet Flow-ID specific multicast flow, but (as it is an MPLS node) it decodes the data flow ID based on the "MPLS-ID" fields of the received packet.

One can appreciate from the above example that, when the means used for DetNet flow identification is altered or exported, the means for encoding the sequence number information must similarly be altered or exported.

4.8. Advertising resources, capabilities and adjacencies

There are three classes of information that a central controller or distributed control plane needs to know that can only be obtained from the end systems and/or nodes in the network. When using a peer-to-peer control plane, some of this information may be required by a system's neighbors in the network.

- Details of the system's capabilities that are required in order to accurately allocate that system's resources, as well as other systems' resources. This includes, for example, which specific queuing and shaping algorithms are implemented (Section 4.5), the number of buffers dedicated for DetNet allocation, and the worst-case forwarding delay and misordering.

- The dynamic state of a node's DetNet resources.

- The identity of the system's neighbors, and the characteristics of the link(s) between the systems, including the length (in nanoseconds) of the link(s).

4.9. Scaling to larger networks

Reservations for individual DetNet flows require considerable state information in each transit node, especially when adequate fault mitigation (Section 3.3.2) is required. The DetNet data plane, in order to support larger numbers of DetNet flows, must support the aggregation of DetNet flows. Such aggregated flows can be viewed by the transit nodes' data plane largely as individual DetNet flows. Without such aggregation, the per-relay system may limit the scale of DetNet networks. Example techniques that may be used include MPLS hierarchy and IP DiffServ Code Points (DSCPs).

4.10. Compatibility with Layer-2

Standards providing similar capabilities for bridged networks (only) have been and are being generated in the IEEE 802 LAN/MAN Standards Committee. The present architecture describes an abstract model that can be applicable both at Layer-2 and Layer-3, and over links not defined by IEEE 802.

DetNet enabled end systems and intermediate nodes can be interconnected by sub-networks, i.e., Layer-2 technologies. These sub-networks will provide DetNet compatible service for support of DetNet traffic. Examples of sub-networks include MPLS TE, 802.1 TSN, and a point-to-point OTN link. Of course, multi-layer DetNet systems may be possible too, where one DetNet appears as a sub-network, and provides service to, a higher layer DetNet system.

5. Security Considerations

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 complex 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. See also Section 3.3.2.

Security must cover:

- the protection of the signaling protocol
- the authentication and authorization of the controlling systems
- the identification and shaping of the DetNet flows

6. Privacy Considerations

DetNet is provides a Quality of Service (QoS), and as such, does not directly raise any new privacy considerations.

However, the requirement for every (or almost every) node along the path of a DetNet flow to identify DetNet flows may present an additional attack surface for privacy, should the DetNet paradigm be found useful in broader environments.

7. IANA Considerations

This document does not require an action from IANA.

8. Acknowledgements

The authors wish to thank Lou Berger, David Black, Stewart Bryant, Rodney Cummings, Ethan Grossman, Craig Gunther, Marcel Kiessling, Rudy Klecka, Jouni Korhonen, Erik Nordmark, Shitanshu Shah, Wilfried Steiner, George Swallow, Michael Johas Teener, Pat Thaler, Thomas Watteyne, Patrick Wetterwald, Karl Weber, Anca Zamfir, for their various contribution with this work.

9. Informative References

[CCAMP] IETF, "Common Control and Measurement Plane Working Group", <https://datatracker.ietf.org/doc/charter-ietf-ccamp/>.

[I-D.ietf-6tisch-architecture] Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-14 (work in progress), April 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-problem-statement] Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", draft-ietf-detnet-problem-statement-06 (work in progress), July 2018.

[I-D.ietf-detnet-use-cases] Grossman, E., "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-17 (work in progress), June 2018.

[IEC62439-3-2016] International Electrotechnical Commission (IEC) TC 65/SC 65C - Industrial networks, "IEC 62439-3:2016 Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)", 2016, <https://webstore.iec.ch/publication/24447>.

[IEEE802.1BA] IEEE Standards Association, "IEEE Std 802.1BA-2011 Audio Video Bridging (AVB) Systems", 2011, <https://ieeexplore.ieee.org/document/6032690/>.

[IEEE802.1CB] IEEE Standards Association, "IEEE Std 802.1CB-2017 Frame Replication and Elimination for Reliability", 2017, <https://ieeexplore.ieee.org/document/8091139/>.

[IEEE802.1Q-2018]
IEEE Standards Association, "IEEE Std 802.1Q-2018 Bridges and Bridged Networks", 2018, <https://standards.ieee.org/findstds/standard/802.1Q-2018.html>.

[IEEE802.1Qav]
IEEE Standards Association, "IEEE Std 802.1Qav-2009 Bridges and Bridged Networks - Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams", 2009, <https://ieeexplore.ieee.org/document/5375704/>.

[IEEE802.1Qbu]
IEEE Standards Association, "IEEE Std 802.1Qbu-2016 Bridges and Bridged Networks - Amendment 26: Frame Preemption", 2016, <https://ieeexplore.ieee.org/document/7553415/>.

[IEEE802.1Qbv]
IEEE Standards Association, "IEEE Std 802.1Qbv-2015 Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", 2015, <https://ieeexplore.ieee.org/document/7572858/>.

[IEEE802.1Qch]
IEEE Standards Association, "IEEE Std 802.1Qch-2017 Bridges and Bridged Networks - Amendment 29: Cyclic Queuing and Forwarding", 2017, <https://ieeexplore.ieee.org/document/7961303/>.

[IEEE802.1TSNTG]
IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networking Task Group", 2013, <http://www.ieee802.org/1/tsn>.

[IEEE802.3-2018]
IEEE Standards Association, "IEEE Std 802.3-2018 Standard for Ethernet", 2018, <http://standards.ieee.org/findstds/standard/802.3-2018.html>.

[IEEE802.3br]
IEEE Standards Association, "IEEE Std 802.3br-2016 Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic", 2016, <http://ieeexplore.ieee.org/document/7900321/>.

[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>.

[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>.

[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>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC4655] Farrel, A., Vasseur, J., 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>.

[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>.

[RFC6372] Sprecher, N., Ed. and A. Farrel, Ed., "MPLS Transport Profile (MPLS-TP) Survivability Framework", RFC 6372, DOI 10.17487/RFC6372, September 2011, <https://www.rfc-editor.org/info/rfc6372>.

[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>.

[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>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <https://www.rfc-editor.org/info/rfc7426>.

[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>.

[RFC7813] Farkas, J., Ed., Bragg, N., Unbehagen, P., Parsons, G., Ashwood-Smith, P., and C. Bowers, "IS-IS Path Control and Reservation", RFC 7813, DOI 10.17487/RFC7813, June 2016, <https://www.rfc-editor.org/info/rfc7813>.

[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>.

[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>.

[TEAS] IETF, "Traffic Engineering Architecture and Signaling Working Group", <https://datatracker.ietf.org/doc/charter-ietf-teas/>.

Authors' Addresses

Norman Finn
Huawei
3755 Avocado Blvd.
PMB 436
La Mesa, California 91941
US

Phone: +1 925 980 6430
Email: norman.finn@mail01.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

Balazs Varga
Ericsson
Magyar tudosok korutja 11
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

Janos Farkas
Ericsson
Magyar tudosok korutja 11
Budapest 1117
Hungary

Email: janos.farkas@ericsson.com