

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

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

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

Abstract

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

Status of This Memo

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

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

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

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

Copyright Notice

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

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

Table of Contents

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

1. Introduction

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

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

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

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

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

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

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

2. Terminology and Definitions

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

3. DetNet bounded latency model

3.1. Flow creation

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

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

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

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

3.1.1. Static flow latency calculation

The static problem:

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

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

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

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

3.1.2. Dynamic flow latency calculation

The dynamic problem:

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

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

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

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

3.2. Relay node model

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

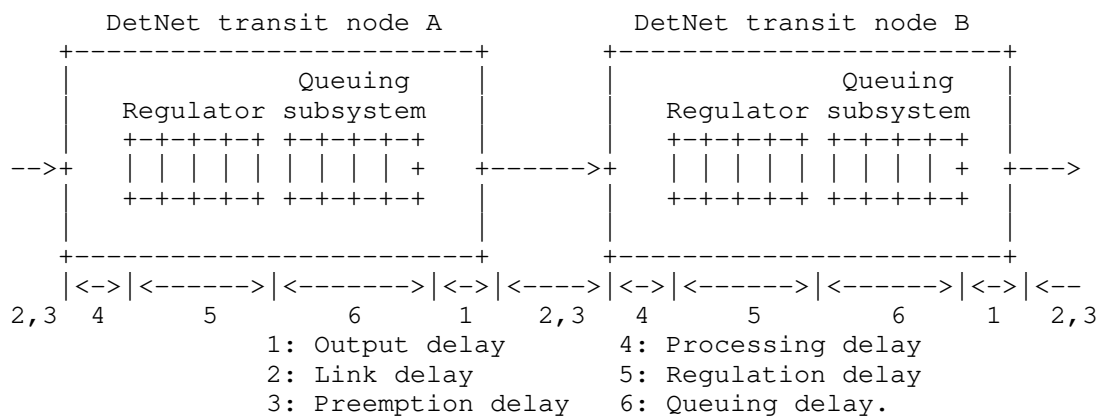


Figure 1: Timing model for DetNet or TSN

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

1. Output delay

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

2. Link delay

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

3. Preemption delay

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

4. Processing delay

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

5. Regulator delay

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

6. Queuing subsystem delay

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

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

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

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

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

4. Computing End-to-end Latency Bounds

4.1. Non-queueing delay bound

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

$$\text{end_to_end_latency_bound} = \text{non_queueing_latency} + \text{queueing_latency}$$

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

4.2. Queueing delay bound

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

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

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

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

4.2.1. Per-flow queuing mechanisms

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

4.2.2. Per-class queuing mechanisms

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

A solution to deal with this issue is to reshape the flows at every hop. This can be done with per-flow regulators (e.g. leaky bucket shapers), but this requires per-flow queuing and defeats the purpose of per-class queuing. An alternative is the interleaved regulator, which reshapes individual flows without per-flow queuing ([Specht2016UBS], [IEEE8021Qcr]). With an interleaved regulator, the packet at the head of the queue is regulated based on its (flow) regulation constraints; it is released at the earliest time at which this is possible without violating the constraint. One key feature of per-flow or interleaved regulator is that, it does not increase worst-case latency bounds [le_boudec_theory_2018]. Specifically, when an interleaved regulator is appended to a FIFO subsystem, it does not increase the worst-case delay of the latter.

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

$$\text{end_to_end_latency_bound_of_flow_f} = C_{12} + C_{23} + C_{34} + S_4$$

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

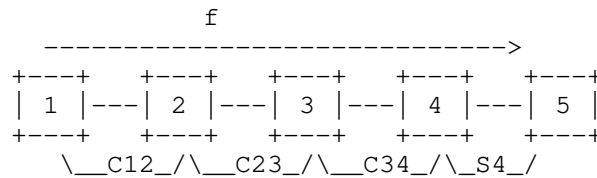


Figure 2: End-to-end latency computation example

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

4.3. Ingress considerations

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

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

4.4. Interspersed non-DetNet transit nodes

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

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

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

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

5. Achieving zero congestion loss

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

5.1. A General Formula

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

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

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

Let

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

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

$$\text{backlog_bound} = (\text{nb_classes} + \text{nb_input_ports}) * \text{max_packet_length} + \text{total_in_rate} * \text{max_delay45}$$

6. Queuing techniques

6.1. Queuing data model

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

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

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

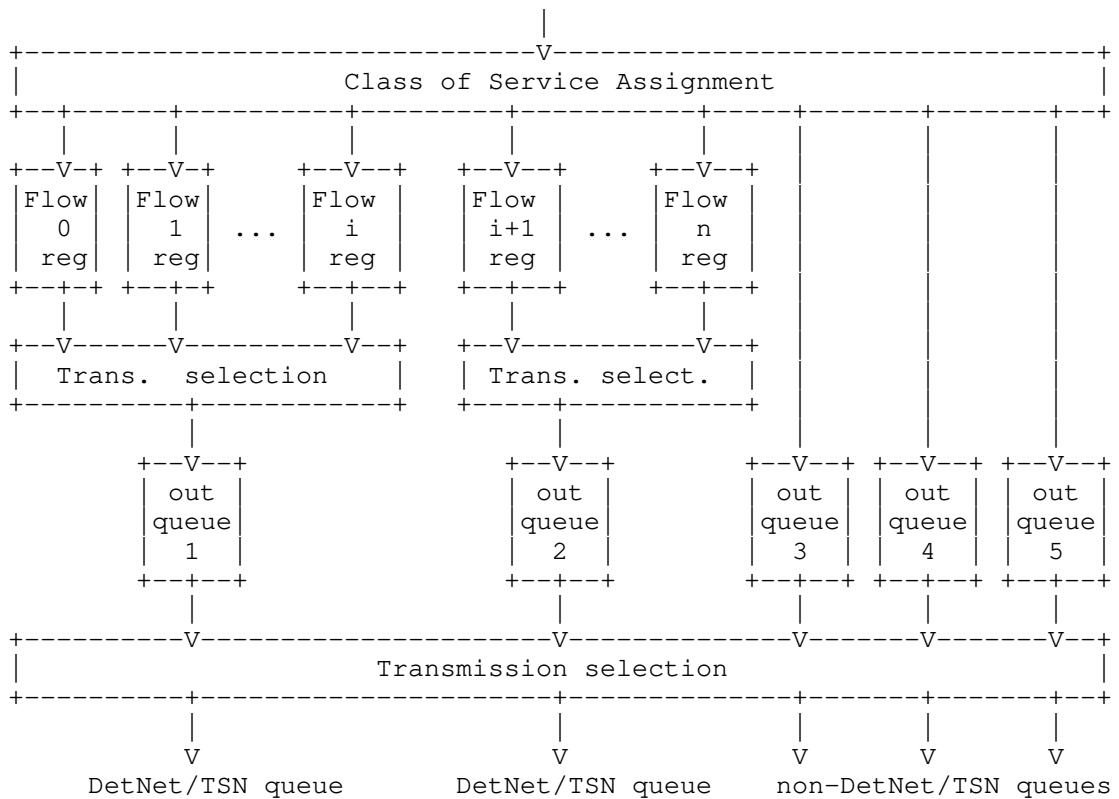


Figure 3: IEEE 802.1Q Queuing Model: Data flow

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

- o Discarding packets because a queue is full.

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

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

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

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

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

6.2. Preemption

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

6.3. Time-scheduled queuing

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

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

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

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

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

6.4. Credit-Based Shaper with Asynchronous Traffic Shaping

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

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

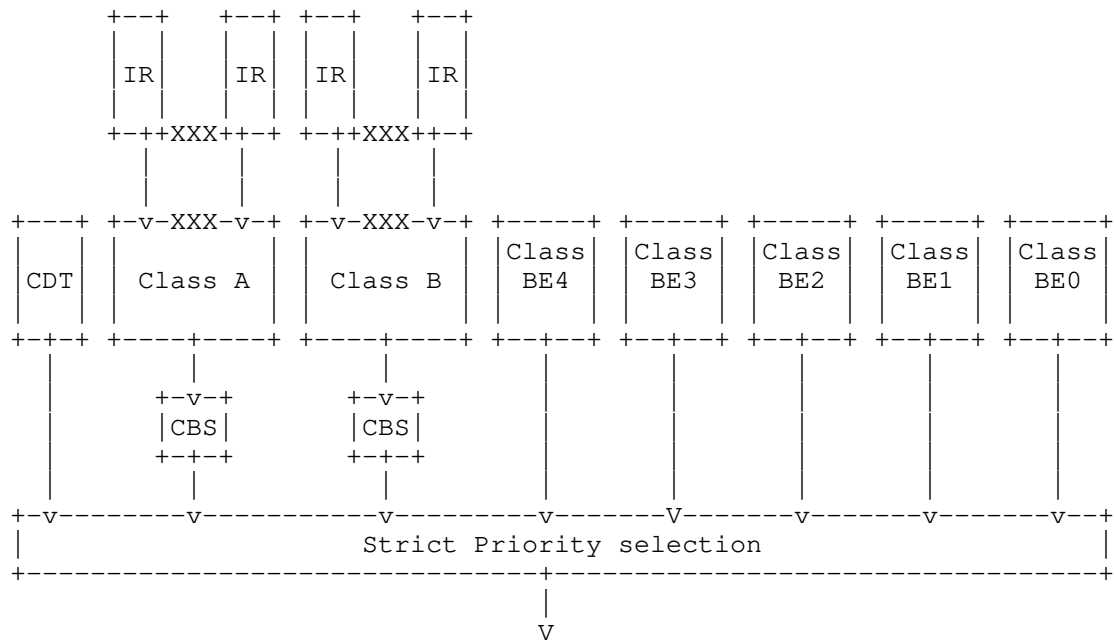


Figure 4: Architecture of a TSN node output port with interleaved regulators (IRs)

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

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

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

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

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

If the condition holds, the delay bound is:

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

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

If f is of class A:

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

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

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

If f is of class B:

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

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

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

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

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

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

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

6.4.1. Flow Admission

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

The static admission control:

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

The dynamic admission control:

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

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

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

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

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

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

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

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

6.5. IntServ

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

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

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

The traffic specification can be described as an arrival curve:

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

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

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

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

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

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

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

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

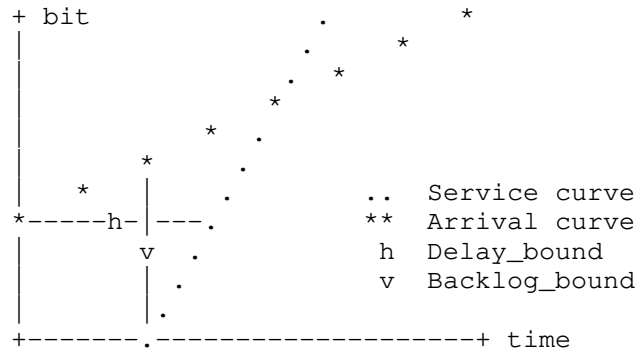


Figure 5: Computation of backlog bound and delay bound. Note that arrival and service curves are not necessary to be linear.

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

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

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

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

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

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

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

6.6.1. CQF timing sequence

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

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

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

6.6.2. CQF latency calculation

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

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

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

7. References

7.1. Normative References

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

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

7.2. Informative References

- [bennett2002delay]
J.C.R. Bennett, K. Benson, A. Charny, W.F. Courtney, and J.-Y. Le Boudec, "Delay Jitter Bounds and Packet Scale Rate Guarantee for Expedited Forwarding", <<https://dl.acm.org/citation.cfm?id=581870>>.
- [charny2000delay]
A. Charny and J.-Y. Le Boudec, "Delay Bounds in a Network with Aggregate Scheduling", <https://link.springer.com/chapter/10.1007/3-540-39939-9_1>.
- [IEEE8021Q]
IEEE 802.1, "IEEE Std 802.1Q-2018: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks", 2018, <<http://ieeexplore.ieee.org/document/8403927>>.
- [IEEE8021Qcr]
IEEE 802.1, "IEEE P802.1Qcr: IEEE Draft Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment: Asynchronous Traffic Shaping", 2017, <<http://www.ieee802.org/1/files/private/cr-drafts/>>.
- [IEEE8021TSN]
IEEE 802.1, "IEEE 802.1 Time-Sensitive Networking (TSN) Task Group", <<http://www.ieee802.org/1/>>.
- [IEEE8023]
IEEE 802.3, "IEEE Std 802.3-2018: IEEE Standard for Ethernet", 2018, <<http://ieeexplore.ieee.org/document/8457469>>.
- [le_boudec_theory_2018]
J.-Y. Le Boudec, "A Theory of Traffic Regulators for Deterministic Networks with Application to Interleaved Regulators", <<http://arxiv.org/abs/1801.08477>>.

[NetCalBook]

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

[Specht2016UBS]

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

[TSNwithATS]

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

Authors' Addresses

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

Phone: +1 925 980 6430
Email: nfinn@nfinnconsulting.com

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

Email: jean-yves.leboudec@epfl.ch

Ehsan Mohammadpour
EPFL
IC Station 14
Lausanne EPFL 1015
Switzerland

Email: ehsan.mohammadpour@epfl.ch

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

Email: zhangjiayi11@huawei.com

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

Email: balazs.a.varga@ericsson.com

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

Email: janos.farkas@ericsson.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: January 5, 2020

X. Geng
M. Chen
Huawei
Y. Zhu
China Telecom
July 04, 2019

DetNet SRv6 Data Plane Encapsulation
draft-geng-detnet-dp-sol-srv6-01

Abstract

This document specifies Deterministic Networking data plane operation for SRv6 encapsulated user data.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 5, 2020.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology and Conventions	3
2.1. Terminology	3
2.2. Conventions	4
3. SRv6 DetNet Data Plane Overview	4
3.1. SRv6 DetNet Data Plane Layers	5
3.2. SRv6 DetNet Data Plane Scenarios	5
4. SRv6 DetNet Data Plane Solution Considerations	7
5. SRv6 DetNet Data Plane Solution for Service Sub-layer	8
5.1. TLV Based SRv6 Data Plane Solution	8
5.1.1. Encapsulation	8
5.1.2. SRv6 Network Programming new Functions	10
5.2. SID Based SRv6 Data Plane Solution	11
5.2.1. Encapsulation	11
5.2.2. Functions	12
5.3. DetNet SID Based SRv6 Data Plane Solution	13
5.3.1. Encapsulation	13
5.3.2. Functions	14
6. SRv6 DetNet Data Plane Solution for Transport Sub-layer	14
7. IANA Considerations	14
8. Security Considerations	14
9. Acknowledgements	14
10. Normative References	14
Authors' Addresses	15

1. Introduction

Deterministic Networking (DetNet), as described in [I-D.ietf-detnet-architecture] provides a capability to carry specified data flows with extremely low data loss rates and bounded latency within a network domain. DetNet is enabled by a group of technologies, such as resource allocation, service protection and explicit routes.

Segment Routing(SR) leverages the source routing paradigm. An ingress node steers a packet through an ordered list of instructions, called "segments". SR can be applied over IPv6 data plane using the Segment Routing Extension Header (SRH, [I-D.ietf-6man-segment-routing-header]). A segment in segment routing terminology is not limited to a routing/forwarding function.

A segment can be associated to an arbitrary processing of the packet in the node identified by the segment. In other words, an SRv6 Segment can indicate functions that are executed locally in the node where they are defined. SRv6 network Programming [I-D.filsfils-spring-srv6-network-programming] describe the different segments and functions associated to them.

This document describes how to implement DetNet in an SRv6 enabled domain, including :

- o Source routing, which steers the DetNet flows through the network according to an explicit path with allocated resources;
- o Network programming, which applies instructions (functions) to packets in some special nodes (or even all the nodes) along the path in order to guarantee, e.g., service protection and congestion protection.

DetNet SRv6 encapsulation and new SRv6 functions ([I-D.filsfils-spring-srv6-network-programming]) for DetNet are defined in this document. Control plane and OAM are not in the scope of this document.

Control plane and OAM are not in the scope of this document.

2. Terminology and Conventions

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].

2.1. Terminology

Terminologies for DetNet go along with the definition in [I-D.ietf-detnet-architecture] and [RFC8402]. Other terminologies are defined as follows:

- o NH: The IPv6 next-header field.
- o SID: A Segment Identifier ([RFC8402]).
- o SRH: The Segment Routing Header ([I-D.ietf-6man-segment-routing-header]).

2.2. Conventions

Conventions in the document are defined as follows:

- o NH=SRH means that NH is 43 with routing type 4 which is (as defined in [I-D.ietf-6man-segment-routing-header], the values representing the SRH.
- o A SID list is represented as <S1, S2, S3> where S1 is the first SID to visit, S2 is the second SID to visit and S3 is the last SID to visit along the SR path.
- o SRH[SL] represents the SID pointed by the SL field in the first SRH. In our example, SRH[2] represents S1, SRH[1] represents S2 and SRH[0] represents S3. It has to be noted that [I-D.ietf-6man-segment-routing-header] defines the segment list encoding in the reverse order of the path. A path represented by <S1,S2,S3>, will be encoded in the SRH as follows:

SegmentList[0]=S3

SegmentList[1]=S2

SegmentList[2]=S1

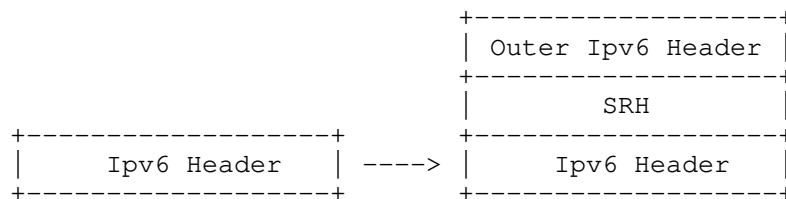
The reverse encoding has been defined in order to optimise the processing time of the segment list. See [draft-ietf-6man-segment-routing-header] for more details.

- o (SA,DA) (S3, S2, S1; SL) represents an IPv6 packet with:
 - IPv6 header with source and destination addresses SA and DA respectively, and next-header set to SRH (i.e.: 43 with type 4), with a list of segments(SIDs) <S1, S2, S3> with SegmentsLeft = SL
 - The payload of the packet is not represented
 - (S3, S2, S1; SL) represents the same SID list as <S1, S2, S3>, but encoded in the SRH format where the rightmost SID in the SRH is the first SID and the leftmost SID in the SRH is the last SID

3. SRv6 DetNet Data Plane Overview

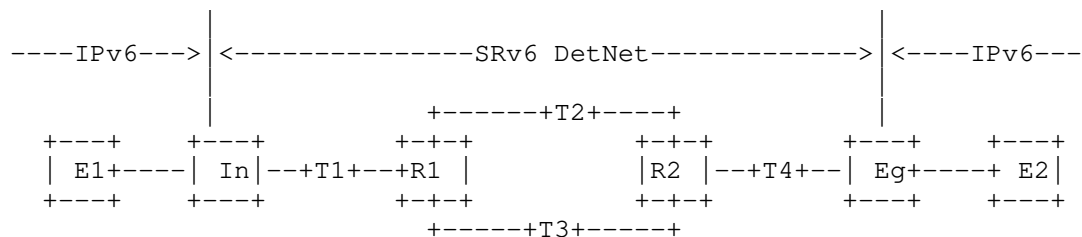
3.1. SRv6 DetNet Data Plane Layers

[I-D.ietf-detnet-architecture]decomposes the DetNet data plane into two sub-layers: service sub-layer and transport sub-layer. Different from DetNet MPLS data plane solution, which uses DetNet Control Word(d-CW) and S-Label to support service sub-layer and uses T-Label to support transport sub-layer, no explicit sub-layer division exists in SRv6 data plane. A classical SRv6 DetNet data plane solution is showed in the picture below:



The outer IPv6 Header with the SRH is used for carrying DetNet flows. Traffic Engineering is instantiated in the segment list of SRH, and other functions and arguments for service protection (packet replication, elimination and ordering) and congestion control (packet queuing and forwarding) are also defined in the SRH.

3.2. SRv6 DetNet Data Plane Scenarios



The figure above shows that an IPv6 flow is sent out from the end station E1. The packet of the flow is encapsulated in an outer IPv6+SRH header as a DetNet SRv6 packet in the Ingress(In) and transported through an SRv6 DetNet domain. In the Egress(Eg), the outer IPv6 header+SRH of the packet is popped, and the packet is sent to the destination E2.

The figure above shows that an IPv6 flow is sent our from the end station: E1. The packet of the flow is encapsulated as a DetNet SRv6 packet in the Ingress(In) and transported through an SRv6 DetNet domain. In the Egress(Eg), the upper IPv6 header with SRH of the packet is popped, and the packet is transmitted to the destination(E2).

The DetNet packet processing is as follows:

Ingress:

Inserts the SRv6 Policy that will steer the packet from Ingress to the destination

The methods and mechanisms used for defining, instantiating and applying the policy are outside of this document. An example of policies are described in [I-D.ietf-spring-segment-routing-policy]

Flow Identification and Sequence Number are carried in the SRH.

Relay Node 1 (Replication Node):

Replicates the payload and IPv6 Header with the SRH. This is a new function in the context of SRv6 Network Programming which will associate a given SID to a replication instruction in the node originating and advertising the SID. The replication instruction includes:

- * The removal of the existing IPv6+SRH header
- * The encapsulation into a new outer IPv6+SRH header. Each packet (the original and the duplicated) are encapsulated into respectively new outer IPv6+SRH headers.

Binding two different SRv6 Policies respectively to the original packet and the replicated packet, which can steer the packets from Relay Node 1 to Relay Node 2 through two tunnels.

Relay Node 2 (Elimination Node):

Eliminates the redundant packets.

Binds a new SRv6 Policy to the survival packet, which steers the packet from Relay Node 2 to Egress.

Egress:

Decapsulates the outer Ipv6 header.

Sends the inter packet to the End Station 2.

The DetNet packet encapsulation is illustrated here below. It has to be noted that, in the example below, the R2 address is a SRH SID

associated to a TBD function related to the packet replication the node R1 has to perform. The same (or reverse) apply to node R2 which is in charge of the discard of the duplicated packet. Here also a new function will have a new SID allocated to it and representing the delete of the duplication in R2.

End Station1 output packet: (E1,E2)

Ingress output packet: (In, T1) (R1,T1, SL=2) (E1,E2)

Transit Node1 output packet: (In, R1) (R1,T1,SL=1) (E1,E2)

Relay Node1 output packets : (R1,T2) (R2,T2,SL=2) (E1,E2),
(R1,T3) (R2,T3,SL=2) (E1,E2)

Transit Node2 output packet: (R1, R2) (R2,T2,SL=1) (E1,E2)

Transit Node3 output packet: (R1, R2) (R2,T3,SL=1) (E1,E2)

Relay Node2 output packet: (R2, T4) (Eg,T4,SL=2) (E1,E2)

Transit Node4 output packet: (R2, Eg) (Eg,T4,SL=1) (E1,E2)

Egress out : (E1,E2)

4. SRv6 DetNet Data Plane Solution Considerations

To carry DetNet over SRv6, the following elements are required:

1. A method of identifying the SRv6 payload type;
2. A suitable explicit path to deliver the DetNet flow ;
3. A method of indicating packet processing, such as PREOF (Packet Replication, Elimination and Ordering as defined in [I-D.ietf-detnet-architecture]);
4. A method of identifying the DetNet flow;
5. A method of carrying DetNet sequence number;
6. A method of carrying queuing and forwarding indication to do congestion protection;

In this design, DetNet flows are encapsulated in an outer IPv6+SRH header at the Ingress Node. The SR policy identified in the SRH steers the DetNet flow along a selected path. The explicit path followed by a DetNet flow, which protect it from temporary

interruptions caused by the convergence of routing, is encoded within the SID list of the SR policy. The network device inside the DetNet domain forwards the packet according to IPv6 Destination Address (DA), and the IPv6 DA is updated with the SID List according to SRv6 forwarding procedures defined in [I-D.ietf-6man-segment-routing-header] and [I-D.filsfils-spring-srv6-network-programming]

With SRv6 network programming, the SID list can also give instruments representing a function to be called at the node in the DetNet domain. Therefore DetNet specific functions defined in [I-D.ietf-detnet-architecture], corresponding to local packet processing in the network, can also be implemented by SRv6. New functions associated with SIDs for DetNet are defined in this document.

This document describes how DetNet flows are encapsulated/identified, and how functions of Packet Replication/Elimination/Ordering are implemented in an SRv6 domain. Congestion protection is also in the scope of this document.

Editor: This version only covers the functions of service protection and the congestion protection considerations will be added in the following versions.

5. SRv6 DetNet Data Plane Solution for Service Sub-layer

This section defines options of SRv6 data plane solution to support DetNet Service Sub-layer.

5.1. TLV Based SRv6 Data Plane Solution

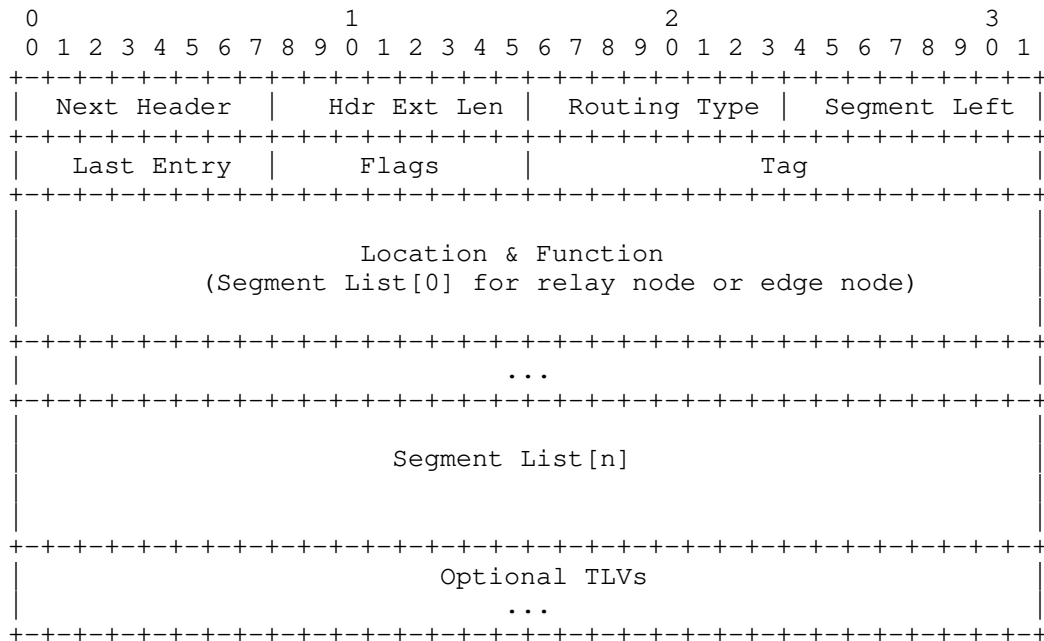
5.1.1. Encapsulation

An SRv6 Segment is a 128-bit value. SID is used as a shorter reference for "SRv6 Segment Identifier" or "SRv6 Segment". SRv6 SID can also be represented as LOC:FUNCT, where:

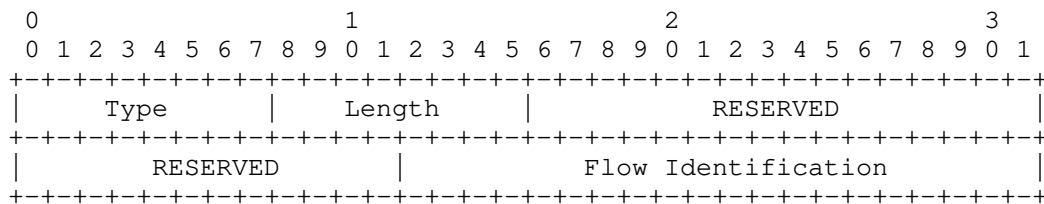
LOC, means "LOCATION" and defines the node associated with the SID (i.e.: represented by the SID).

FUNCT, means "FUNCTION", and identifies the processing that the node specified in LOC applies to the packet. See [I-D.filsfils-spring-srv6-network-programming] for details on SRv6 Network Programming.

The SRH for DetNet in the outer IPv6 header is showed as follows, according to [I-D.ietf-6man-segment-routing-header] and [I-D.filsfils-spring-srv6-network-programming]:



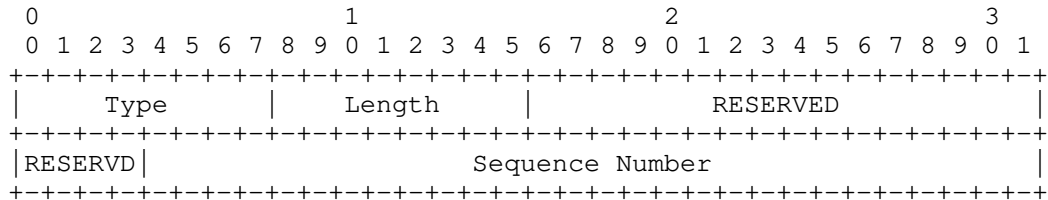
The SRH specification allows the use of optional TLVs. Two new TLVs are defined to support DetNet service protection. DetNet Flow Identification TLV is used to uniquely identify a DetNet flow in an SRv6 DetNet node. DetNet sequence number is used to discriminate packets in the same DetNet flow. They are defined as follows:



where:

- o Type: 8bits, to be assigned by IANA.
- o Length: 8 octets.

- o RESERVED: 28 bits, MUST be 0 on transmission and ignored on receipt.
- o Flow Identification: 20 bits, which is used for identifying DetNet flow.



where:

- o Type: 8 bits, to be assigned by IANA.
- o Length: 8.
- o RESERVED: 20 bits. MUST be 0 on transmission and ignored on receipt.
- o Sequence Number: 28 bits, which is used for indicating sequence number of a DetNet flow.

5.1.2. SRv6 Network Programming new Functions

New SRv6 Network Programming functions are defined as follows:

5.1.2.1. End. B.Replicationreserve the value of argument field(Inherited argument)of segment[0] of SRH n: Packet Replication Function

1. IF NH=SRH & SL>0 THEN
2. extract the DetNet TLV values from the SRH
3. create two new outer IPv6+SRH headers: IPv6-SRH-1 and IPv6-SRH-2
Insert the policy-instructed segment lists in each newly created SRH (SRH-1 and SRH-2). Also, add the extracted DetNet TLVs into SRH-1 and SRH-2.
4. remove the incoming outer IPv6+SRH header.
5. create a duplication of the incoming packet.
6. encapsulate the original packet into the first outer IPv6+SRH header: (IPv6-SRH-1) (original packet)

7. encapsulate the duplicate packet into the second outer IPv6+SRH header: (IPv6-SRH-2) (duplicate packet)
8. set the IPv6 SA as the local address of this node.
9. set the IPv6 DA of IPv6-SRH-1 to the first segment of the SRv6 Policy in of SRH-1 segment list.
10. set the IPv6 DA of IPv6-SRH-2 to the first segment of the SRv6 Policy in of SRH-2 segment list.
11. ELSE
12. drop the packet

5.1.2.2. End. B. Elimination: Packet Elimination Function

1. IF NH=SRH & SL>0 & "the packet is not a redundant packet" THEN
2. do not decrement SL nor update the IPv6 DA with SRH[SL]
3. extract the value of DetNet TLVs from the SRH
4. create a new outer IPv6+SRH header
5. insert the policy-instructed segment lists in the newly created SRH and add the retrieved DetNet TLVs in the newly created SRH
6. remove the incoming outer IPv6+SRH header.
7. set the IPv6 DA to the first segment of the SRv6 Policy in the newly created SRH
8. ELSE
9. drop the packet

5.2. SID Based SRv6 Data Plane Solution

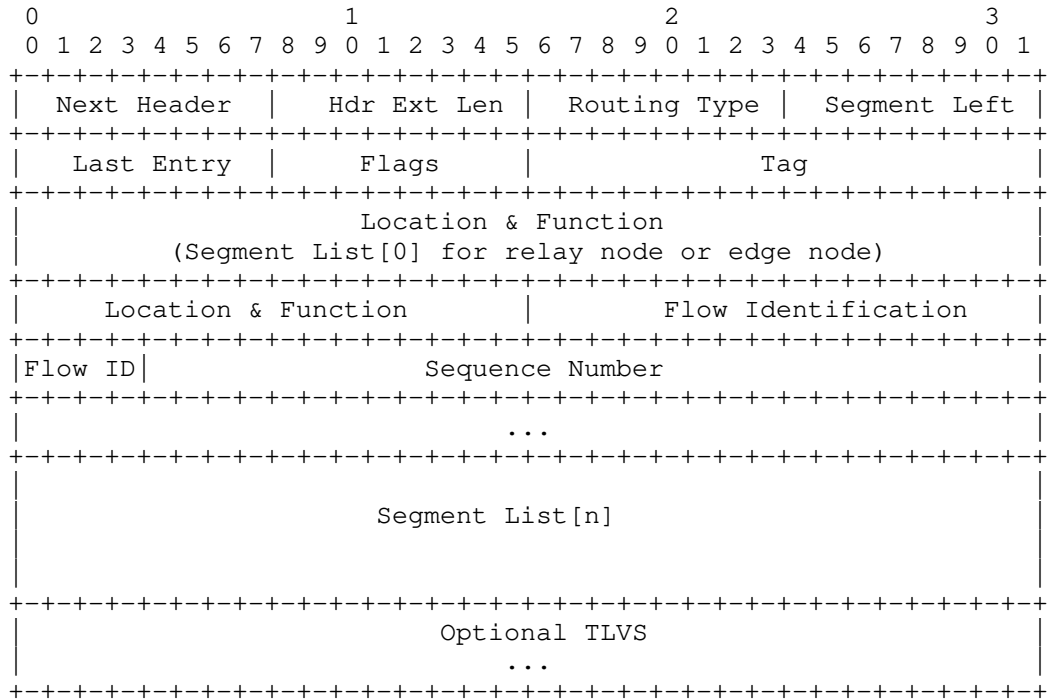
5.2.1. Encapsulation

SRv6 SID can be represented as LOC:FUNCT:ARG::, where:

LOC, means "LOCATION" and defines the node associated with the SID (i.e.: represented by the SID).

FUNCT, means "FUNCTION", and identifies the processing that the node specified in LOC applies to the packet.

ARG, means "ARGUMENTS" and provides the additional arguments for the function. New SID functions for DetNet is defined in section 5.2.2. See [I-D.filsfils-spring-srv6-network-programming] for details on SRV6 Network Programming. The SRH for DetNet in the outer IPv6 header is illustrated as follows



where:

- o LOCATION&FUNCTION: the 80 most significant bits that are used for routing the packet towards the LOCATION (as defined in [I-D.filsfils-spring-srv6-network-programming]);
- o FLOW IDENTIFICATION: 20 bits, in the DetNet TLVs in the SRH, used for DetNet flow identification in the DetNet relay node;
- o SEQUENCE NUMBER : 28 bits, in the DetNet TLVs, used for dis crime packets in the same DetNet flow;

5.2.2. Functions

New SID functions are defined as follows:

5.2.2.1. End. B.Replication: Packet Replication Function

The function is similar as that has been defined in section 5.1.2.1. The only difference is that instead of retrieving the TLV values, this function retrieves the argument.

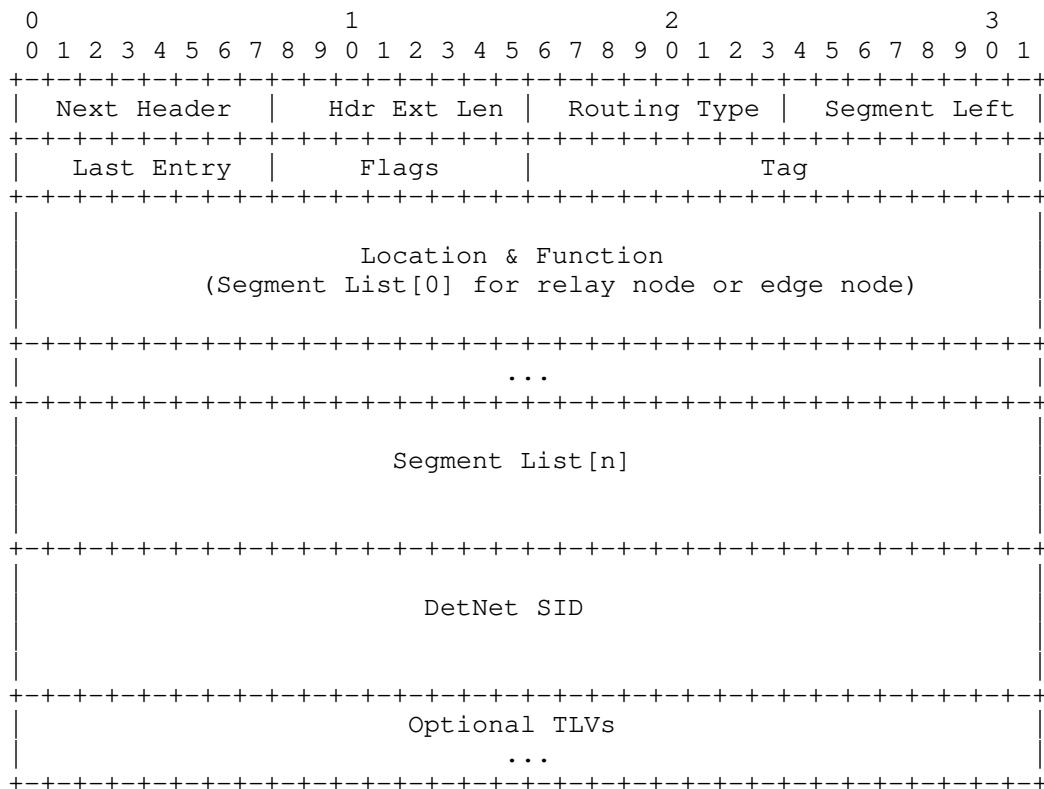
5.2.2.2. End. B. Elimination: Packet Elimination Function

The function is similar as that has been defined in section 5.1.2.2. The only difference is that instead of retrieving the TLV values, this function retrieves the argument.

5.3. DetNet SID Based SRv6 Data Plane Solution

5.3.1. Encapsulation

A non-forwarding DetNet SID is defined to carry Flow Identification and Sequence Number.



5.3.2. Functions

TBD

6. SRv6 DetNet Data Plane Solution for Transport Sub-layer

TBD

7. IANA Considerations

TBD

8. Security Considerations

TBD

9. Acknowledgements

Thank you for valuable comments from James Guichard and Andrew Mails.

10. Normative References

[I-D.filsfils-spring-srv6-network-programming]

Filsfils, C., Camarillo, P., Leddy, J.,
daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6
Network Programming", draft-filsfils-spring-srv6-network-
programming-07 (work in progress), February 2019.

[I-D.ietf-6man-segment-routing-header]

Filsfils, C., Dukes, D., Previdi, S., Leddy, J.,
Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment
Routing Header (SRH)", draft-ietf-6man-segment-routing-
header-21 (work in progress), June 2019.

[I-D.ietf-detnet-architecture]

Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", draft-ietf-
detnet-architecture-13 (work in progress), May 2019.

[I-D.ietf-detnet-dp-sol-mpls]

Korhonen, J. and B. Varga, "DetNet MPLS Data Plane
Encapsulation", draft-ietf-detnet-dp-sol-mpls-02 (work in
progress), March 2019.

- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d.,
bogdanov@google.com, b., and P. Mattes, "Segment Routing
Policy Architecture", draft-ietf-spring-segment-routing-
policy-03 (work in progress), May 2019.
- [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>>.
- [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>>.

Authors' Addresses

Xuesong Geng
Huawei

Email: gengxuesong@huawei.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Yongqing Zhu
China Telecom

Email: zhuyq@gsta.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2020

L. Geng
China Mobile
P. Willis
BT
S. Homma
NTT
L. Qiang
Huawei
July 8, 2019

Requirements of Layer 3 Deterministic Latency Service
draft-geng-detnet-requirements-bounded-latency-03

Abstract

This document specifies some technical, operational and management requirements of deploying deterministic latency service on layer 3 networks from the perspective of the service provider.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

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

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

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
1.2. Terminology & Abbreviations	3
2. Technical Requirements	3
2.1. Requirement 1: Must support the dynamic creation, modification and deletion of deterministic services	3
2.2. Requirement 2: Should tolerate a certain degree of time variance	4
2.2.1. Sub-requirement 2.1: Should support asynchronous clocks across domains	4
2.2.2. Sub-requirement 2.2: Should tolerate clock jitter & wander within a clock synchronous domain	4
2.3. Requirement 3: Must support Inter-Continental propagation delay	5
3. Operational and Management Requirements	6
3.1. Requirement 4: Should have self-monitoring capability . .	6
3.2. Requirement 5: Should be robust against denial of service attacks	6
3.3. Requirement 6: Must tolerate failures of links or nodes and topology changes	7
3.4. Requirement 7: Must be scalable	7
3.4.1. Sub-requirement 7.1: Must be scalable to numerous network devices	7
3.4.2. Sub-requirement 7.2: Must be scalable to massive traffic flows	7
4. IANA Considerations	7
5. Security Considerations	8
6. Acknowledgements	8
7. Normative References	8
Authors' Addresses	9

1. Introduction

DetNet is chartered to provide bounds on latency, jitter (delay variation) and packet loss [draft-ietf-detnet-problem-statement]. Where latency and jitter are two closely related performance characteristics, this document refers to bounded latency and bounded jitter collectively as deterministic latency.

This document does not intend to define any specific mechanism, but specifies some technical, operational and management requirements

that must be considered when deploying deterministic latency service at layer 3.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology & Abbreviations

This document uses the terminology defined in [draft-ietf-detnet-architecture].

TSN: Time Sensitive Network

2. Technical Requirements

2.1. Requirement 1: Must support the dynamic creation, modification and deletion of deterministic services

There are at least two modes to provide deterministic service over an operator's network: 1) deterministic VPN; 2) point-to-point deterministic tunnel. No matter in which mode, the layer 3 deterministic latency mechanisms must be able to support the dynamic creation, modification and deletion of deterministic services without affecting any deterministic services that are already running in the network.

In a local area network such as a factory, the information about when a deterministic service will start, how long the service will last, can be known in advance, or can even be planned. Based on this information, the local area network can adopt a global programming mechanism to calculate the accurate processing behaviors for each device, and achieve a global optimal performance. However, such global programming mechanisms are unsuitable for service providers' networks. Many deterministic applications are expected to running on a service provider's network simultaneously. Different deterministic applications may have different lifecycles and SLA requirements, hence the network state changes dynamically. If a mechanism relies on a stable network state for global computing, any change in network state (e.g., new application starts, or an application finishes, or SLA requirement changes) will lead to re-computing, even worse if all devices need to stop work and install the recomputed results, then this mechanism is hard to be deployed on service provider's network.

2.2. Requirement 2: Should tolerate a certain degree of time variance

2.2.1. Sub-requirement 2.1: Should support asynchronous clocks across domains

One of DetNet's objectives is to stitch TSN islands together. All devices inside a TSN domain are time-synchronized, and most of TSN technologies rely on precise time synchronization. However, different TSN islands may have different clocks which are not synchronized as shown in Figure 1, where the time difference of two TSN domain is D . DetNet needs to connect these two TSN domains together and provide end-to-end deterministic latency service. The mechanism adopted by DetNet should be able to support the interaction across time domains by using a fine controlled buffer (the time difference ' D ' may be us-level) to absorb the time difference, or relying on the mechanism itself to implement cross domain time mapping.

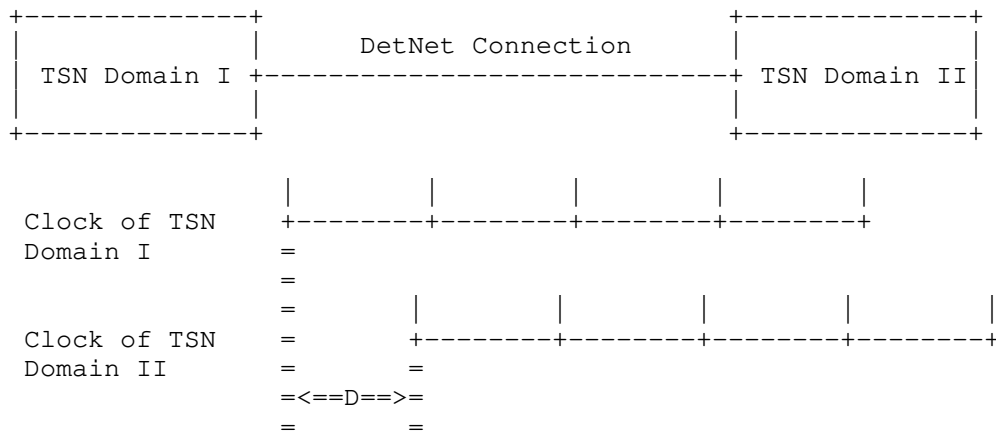


Figure 1: TSN islands interconnecting

2.2.2. Sub-requirement 2.2: Should tolerate clock jitter & wander within a clock synchronous domain

DetNet domain itself can be time synchronous or asynchronous, depending on actual deployment, application, use cases, etc. Even within a time synchronous domain, the synchronized clocks may also experience jitter & wander. Different areas have different clock accuracy, for example the crystal oscillator in Ethernet is specified at 100 ppm [Fast-Ethernet-MII-clock], SyncE can achieve 50 ppb [G.8262], and more precise time synchronization [G.8273] is expected in 5G mobile backhaul. Hence the mechanisms adopted by DetNet should

be able to tolerate a certain degree of clock jitter & wander accordingly.

2.3. Requirement 3: Must support Inter-Continental propagation delay

In contrast to Layer 2 TSN that is deployed on a LAN [IEEE-TSN], DetNet is expected to be deployed in a larger scale carrier networks that have long link propagation delay which means that DetNet must work on network links that have inter-continental propagation delays. Long link propagation delay can cause some troubles to cyclic forwarding type mechanisms. In order to guarantee deterministic latency, cyclic forwarding type mechanisms usually require a packet be sent out (or received) at a particular cycle, rather than be sent out (or received) randomly. There is a mapping between the sending cycle of upstream node and the receiving cycle of downstream node. In a local area network that with short link propagation delay, the cycle mapping relationship could be very simple.

As an example shown in Figure 2, where the length of a cycle is 10 us, and the sending cycle of upstream node X and the receiving cycle of downstream node Y correspond to the same period of time (e.g., 0~10 us). Packets sent from X at sending cycle will arrive at downstream node Y at receiving cycle, and the link propagation delay between X and Y is smaller than the length of a cycle (i.e., 10 us).

Suppose a large scale network wants to keep using this simple cycle mapping relationship, however the link distance between two nodes is longer. Moreover, a downstream node may have many upstream nodes each with different link propagation delays (e.g., 9 us, 10 us, 11 us, 15 us, 20 us). In order to absorb the longest link propagation delay, then the length of cycle must be set to at least 20 us. However since packet's arrival time varies within the receiving cycle, larger cycle length means larger delay variance.

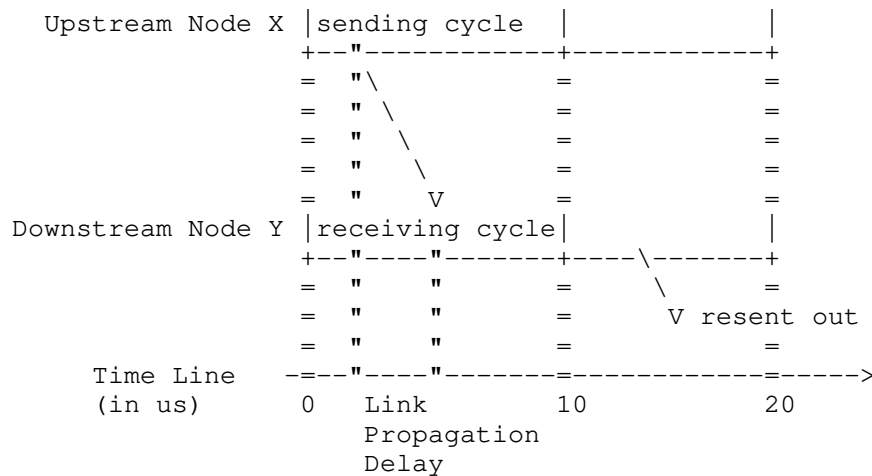


Figure 2: An example of limited link

3. Operational and Management Requirements

[Authors note: more explanations for each requirement need to be added in later versions.]

3.1. Requirement 4: Should have self-monitoring capability

Both network operators and end-users need to be able to measure if the deterministic latency service is working correctly and meeting its SLA. Once the monitored results exceed the expected bounds, network operators and end-users should be notified, and some service protection mechanisms should also be triggered accordingly. In addition, network operators can input the collected results into a reporting system and produce the latency (and jitter) distribution over a period of time, which would be helpful for operators in understanding their networks performance.

However, such fine-granularity monitoring is costly. Hence although the self-monitoring is an important capability to both operators and customers, it is not recommended as a mandatory requirement until the use cases are clear.

3.2. Requirement 5: Should be robust against denial of service attacks

To protect the services requiring deterministic latency, the mechanisms implemented by DetNet should be robust to denial of service attacks. This includes robustness against attacks on the mechanisms to generate clock synchronization.

[draft-ietf-detnet-architecture] has discussed using the traffic

admission control at the ingress or through the fault mitigation methods, to prevent (or mitigate) the excess traffic caused by malicious or malfunction devices. DetNet also considers using authentication and authorization to mitigate man-in-the middle attacks

3.3. Requirement 6: Must tolerate failures of links or nodes and topology changes

A step change in latency due to a single network topology change is inevitable. However if the topology keeps changing many times, then DetNet may not deliver on its SLA. The topology changes alone may not be sufficient on a traditional IP network to raise any alarms, but the mechanism's self-monitoring should detect this, and keep working in flapping network topologies.

3.4. Requirement 7: Must be scalable

Further to the requirement to work on inter-continental links, the deterministic latency forwarding mechanisms must scale to networks of significant size with numerous network devices and massive traffic flows.

3.4.1. Sub-requirement 7.1: Must be scalable to numerous network devices

A simple use case to understand is ultra-low-latency (public) 5G transport networks, which would require DetNet extend to every 5G base station. For some network operators, their network may need to connect to ~100 K base stations (serving multiple MNOs'), and this number will only increase with 5G.

3.4.2. Sub-requirement 7.2: Must be scalable to massive traffic flows

If each ultra-low-latency slice or MNO is treated as a separate deterministic latency traffic flow (or tunnel), then even if each base station has a limited number of ultra-low latency slices or MNOs (e.g. ~10), there will still be a lot of, ~1M, deterministic latency traffic flows on one network simultaneously.

4. IANA Considerations

This document makes no request of IANA.

5. Security Considerations

This document will not introduce new security problems.

6. Acknowledgements

The Authors would like to thank David Black, Stewart Bryant for their review, suggestion and comments to this document.

7. Normative References

- [draft-ietf-detnet-architecture]
"DetNet Architecture", <<https://datatracker.ietf.org/doc/draft-ietf-detnet-architecture/>>.
- [draft-ietf-detnet-problem-statement]
G.8262 : Timing characteristics of a synchronous Ethernet equipment slave clock
G.8262 : Timing characteristics of a synchronous Ethernet equipment slave clock, "DetNet Problem Statement", <<https://datatracker.ietf.org/doc/draft-ietf-detnet-problem-statement/>>.
- [Fast-Ethernet-MII-clock]
"Fast Ethernet MII clock",
<<http://www.ti.com/lit/ds/symlink/dp83865.pdf>>.
- [G.8262] "G.8262 : Timing characteristics of a synchronous Ethernet equipment slave clock",
<<https://www.itu.int/rec/T-REC-G.8262>>.
- [G.8273] "G.8273: Framework of phase and time clocks",
<<https://www.itu.int/rec/T-REC-G.8273/en>>.
- [IEEE-TSN]
"IEEE TSN Task Group",
<<http://www.ieee802.org/1/pages/tsn.html>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Liang Geng
China Mobile
Beijing
China

Email: gengliang@chinamobile.com

Peter Willis
BT
BT Adastral Park
Ipswich IP5 3RE
UK

Email: peter.j.willis@bt.com

Shunsuke Homma
NTT
Tokyo
Japan

Email: shunsuke.homma.fp@hco.ntt.co.jp

Li Qiang
Huawei
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: qiangli3@huawei.com

DetNet
Internet-Draft
Intended status: Informational
Expires: January 25, 2020

N. Finn
Huawei Technologies Co. Ltd
J-Y. Le Boudec
E. Mohammadpour
EPFL
J. Zhang
Huawei Technologies Co. Ltd
B. Varga
J. Farkas
Ericsson
July 24, 2019

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

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 25, 2020.

Copyright Notice

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

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

Table of Contents

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

1. Introduction

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

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

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

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

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

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

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

2. Terminology and Definitions

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

3. DetNet bounded latency model

3.1. Flow creation

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

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

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

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

3.1.1. Static flow latency calculation

The static problem:

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

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

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

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

3.1.2. Dynamic flow latency calculation

The dynamic problem:

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

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

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

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

3.2. Relay node model

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

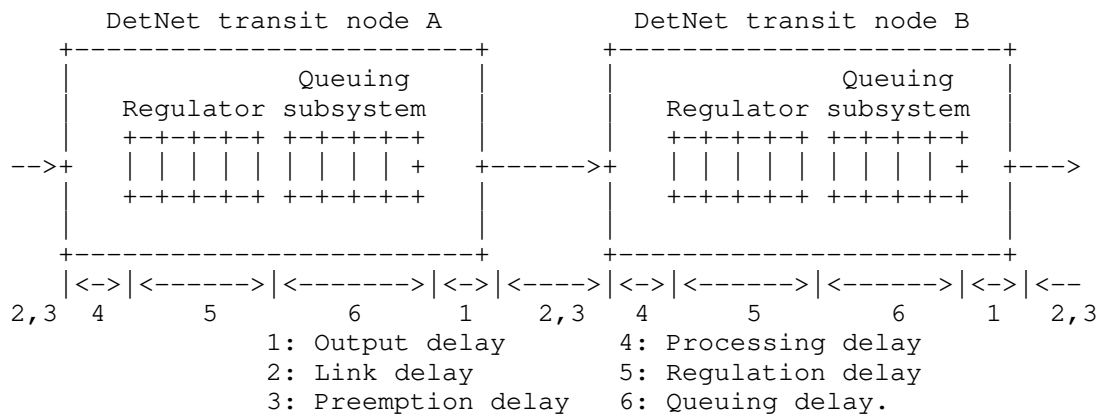


Figure 1: Timing model for DetNet or TSN

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

1. Output delay

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

2. Link delay

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

3. Preemption delay

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

4. Processing delay

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

5. Regulator delay

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

6. Queuing subsystem delay

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

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

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

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

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

4. Computing End-to-end Latency Bounds

4.1. Non-queueing delay bound

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

$$\text{end_to_end_latency_bound} = \text{non_queueing_latency} + \text{queueing_latency}$$

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

4.2. Queueing delay bound

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

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

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

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

4.2.1. Per-flow queuing mechanisms

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

4.2.2. Per-class queuing mechanisms

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

A solution to deal with this issue is to reshape the flows at every hop. This can be done with per-flow regulators (e.g. leaky bucket shapers), but this requires per-flow queuing and defeats the purpose of per-class queuing. An alternative is the interleaved regulator, which reshapes individual flows without per-flow queuing ([Specht2016UBS], [IEEE8021Qcr]). With an interleaved regulator, the packet at the head of the queue is regulated based on its (flow) regulation constraints; it is released at the earliest time at which this is possible without violating the constraint. One key feature of per-flow or interleaved regulator is that, it does not increase worst-case latency bounds [le_boudec_theory_2018]. Specifically, when an interleaved regulator is appended to a FIFO subsystem, it does not increase the worst-case delay of the latter.

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

$$\text{end_to_end_latency_bound_of_flow_f} = C_{12} + C_{23} + C_{34} + S_4$$

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

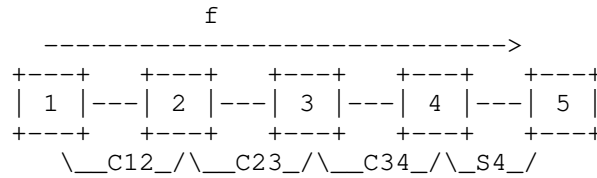


Figure 2: End-to-end latency computation example

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

4.3. Ingress considerations

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

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

4.4. Interspersed non-DetNet transit nodes

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

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

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

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

5. Achieving zero congestion loss

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

5.1. A General Formula

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

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

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

Let

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

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

$$\text{backlog_bound} = (\text{nb_classes} + \text{nb_input_ports}) * \text{max_packet_length} + \text{total_in_rate} * \text{max_delay45}$$

6. Queuing techniques

6.1. Queuing data model

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

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

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

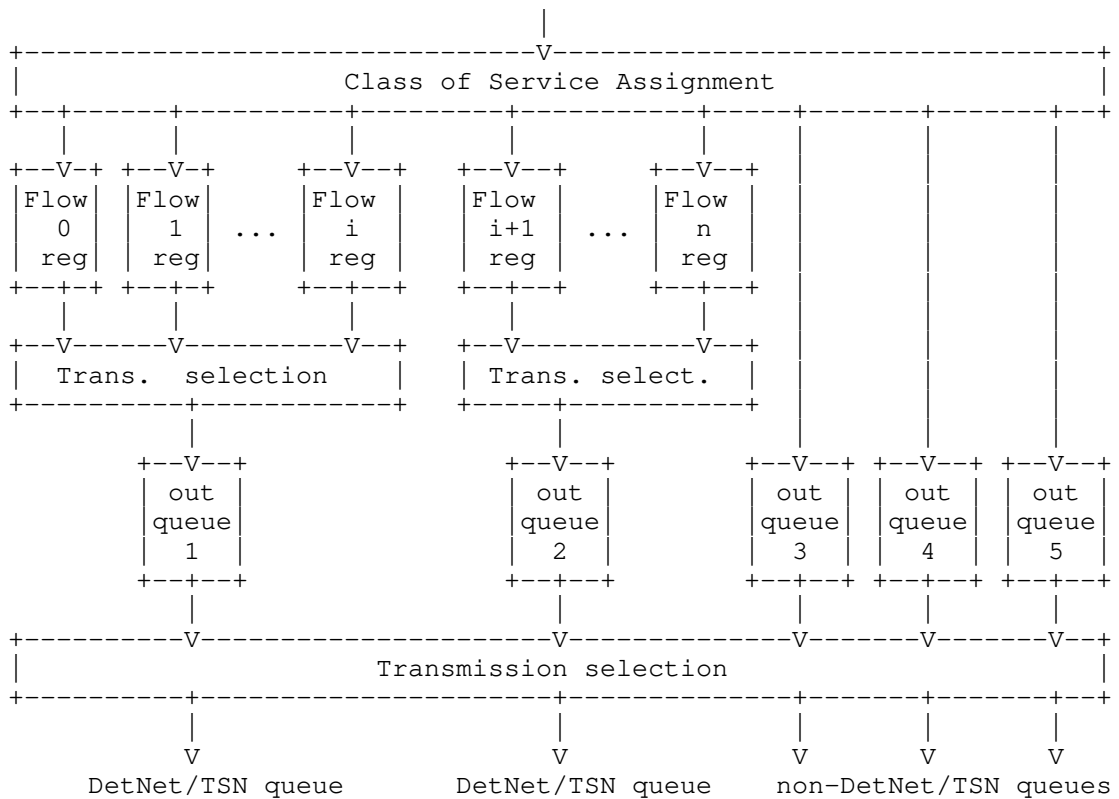


Figure 3: IEEE 802.1Q Queuing Model: Data flow

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

- o Discarding packets because a queue is full.

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

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

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

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

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

6.2. Preemption

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

6.3. Time-scheduled queuing

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

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

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

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

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

6.4. Credit-Based Shaper with Asynchronous Traffic Shaping

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

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

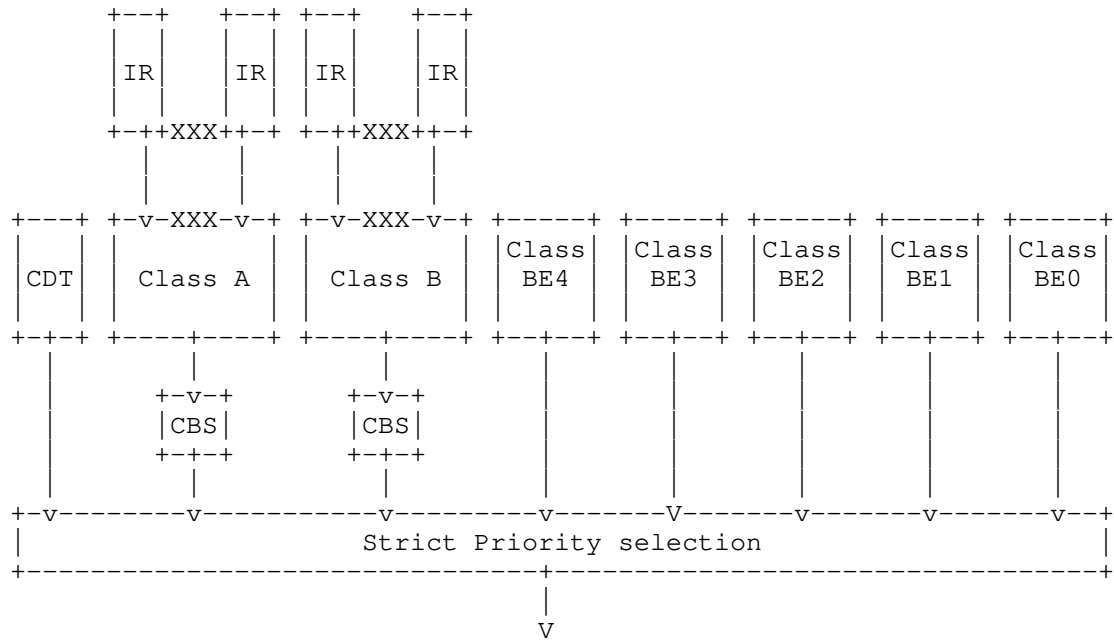


Figure 4: Architecture of a TSN node output port with interleaved regulators (IRs)

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

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

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

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

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

If the condition holds, the delay bound is:

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

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

If f is of class A:

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

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

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

If f is of class B:

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

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

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

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

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

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

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

6.4.1. Flow Admission

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

The static admission control:

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

The dynamic admission control:

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

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

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

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

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

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

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

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

6.5. IntServ

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

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

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

The traffic specification can be described as an arrival curve:

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

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

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

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

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

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

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

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

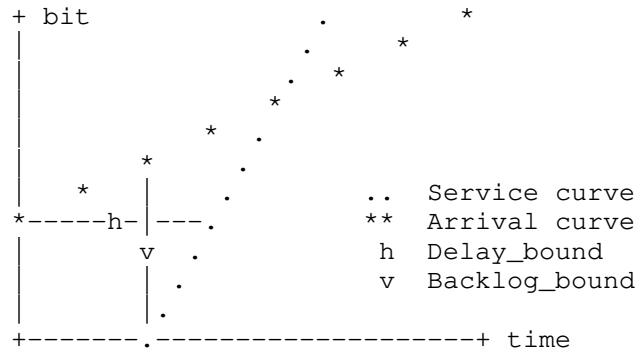


Figure 5: Computation of backlog bound and delay bound. Note that arrival and service curves are not necessary to be linear.

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

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

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

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

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

where

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

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

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

6.6. Cyclic Queuing and Forwarding

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

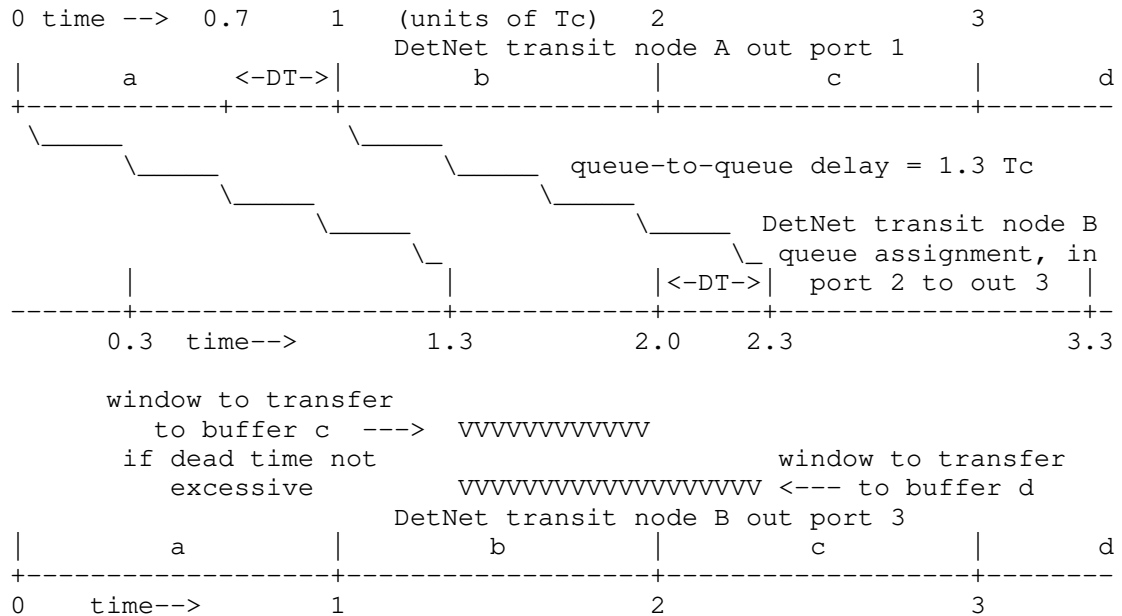


Figure 6: CQF timing diagram

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

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

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

6.6.1. CQF timing sequence

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

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

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

6.6.2. CQF latency calculation

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

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

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

7. References

7.1. Normative References

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

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

7.2. Informative References

- [bennett2002delay]
J.C.R. Bennett, K. Benson, A. Charny, W.F. Courtney, and J.-Y. Le Boudec, "Delay Jitter Bounds and Packet Scale Rate Guarantee for Expedited Forwarding", <<https://dl.acm.org/citation.cfm?id=581870>>.
- [charny2000delay]
A. Charny and J.-Y. Le Boudec, "Delay Bounds in a Network with Aggregate Scheduling", <https://link.springer.com/chapter/10.1007/3-540-39939-9_1>.
- [IEEE8021Q]
IEEE 802.1, "IEEE Std 802.1Q-2018: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks", 2018, <<http://ieeexplore.ieee.org/document/8403927>>.
- [IEEE8021Qcr]
IEEE 802.1, "IEEE P802.1Qcr: IEEE Draft Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment: Asynchronous Traffic Shaping", 2017, <<http://www.ieee802.org/1/files/private/cr-drafts/>>.
- [IEEE8021TSN]
IEEE 802.1, "IEEE 802.1 Time-Sensitive Networking (TSN) Task Group", <<http://www.ieee802.org/1/>>.
- [IEEE8023]
IEEE 802.3, "IEEE Std 802.3-2018: IEEE Standard for Ethernet", 2018, <<http://ieeexplore.ieee.org/document/8457469>>.
- [le_boudec_theory_2018]
J.-Y. Le Boudec, "A Theory of Traffic Regulators for Deterministic Networks with Application to Interleaved Regulators", <<http://arxiv.org/abs/1801.08477>>.

[NetCalBook]

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

[Specht2016UBS]

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

[TSNwithATS]

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

Authors' Addresses

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

Phone: +1 925 980 6430
Email: nfinn@nfinnconsulting.com

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

Email: jean-yves.leboudec@epfl.ch

Ehsan Mohammadpour
EPFL
IC Station 14
Lausanne EPFL 1015
Switzerland

Email: ehsan.mohammadpour@epfl.ch

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

Email: zhangjiayi11@huawei.com

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

Email: balazs.a.varga@ericsson.com

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

Email: janos.farkas@ericsson.com

DetNet
Internet-Draft
Intended status: Informational
Expires: January 2, 2020

B. Varga, Ed.
J. Farkas
Ericsson
L. Berger
D. Fedyk
LabN Consulting, L.L.C.
A. Malis
S. Bryant
Futurewei Technologies
J. Korhonen
July 1, 2019

DetNet Data Plane Framework
draft-ietf-detnet-data-plane-framework-01

Abstract

This document provides an overall framework for the Deterministic Networking data plane. It covers concepts and considerations that are generally common to any Deterministic Networking data plane specification.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 2, 2020.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Terms Used in This Document	3
2.2. Abbreviations	4
3. DetNet Data Plane Overview	4
3.1. Data Plane Characteristics	6
3.2. Encapsulation	6
3.3. DetNet Specific Metadata	7
3.4. DetNet IP Data Plane	8
3.5. DetNet MPLS Data Plane	8
3.6. Further DetNet Data Plane Considerations	9
3.6.1. Service Protection	11
3.6.2. Aggregation Considerations	13
3.6.3. End-System Specific Considerations	14
3.6.4. Sub-Network Considerations	14
4. Controller Plane (Management and Control) Considerations	15
4.1. DetNet Controller Plane Requirements	15
4.2. Generic Controller Plane Considerations	16
4.2.1. Flow Aggregation Control	17
4.2.2. Explicit Routes	18
4.2.3. Contention Loss and Jitter Reduction	19
4.2.4. Bidirectional Traffic	19
4.3. Packet Replication, Elimination, and Ordering (PREOF)	20
5. Security Considerations	20
6. IANA Considerations	21
7. Acknowledgements	21
8. References	21
8.1. Normative References	21
8.2. Informative References	22
Authors' Addresses	24

1. Introduction

Deterministic Networking (DetNet) provides a capability to carry specified unicast or multicast data flows for real-time applications with extremely low packet loss rates and assured maximum end-to-end delivery latency. A description of the general background and concepts of DetNet can be found in [I-D.ietf-detnet-architecture].

This document describes the concepts needed by any DetNet data plane specification and provides considerations for any corresponding implementation. It covers the building blocks that provide the DetNet service, the DetNet service sub-layer and the DetNet forwarding sub-layer functions as described in the DetNet Architecture.

The DetNet Architecture models the DetNet related data plane functions decomposed into two sub-layers: a service sub-layer and a forwarding sub-layer. The service sub-layer is used to provide DetNet service protection and reordering. The forwarding sub-layer is used to provide congestion protection (low loss, assured latency, and limited out-of-order delivery) and leverages Traffic Engineering mechanisms.

As part of the service sub-layer functions, this document describes typical DetNet node data plane operation. It describes the function and operation of the Packet Replication (PRF) Packet Elimination (PEF) and the Packet Ordering (POF) functions within the service sub-layer. It also describes the forwarding sub-layer that is used to eliminate (or reduce) contention loss and provide bounded latency for DetNet flows.

DetNet flows may be carried over network technologies that can provide the DetNet required service characteristics. For example, DetNet MPLS flows can be carried over IEEE 802.1 Time Sensitive Network (TSN) [IEEE802.1TSNTG] sub-networks. However, IEEE 802.1 TSN support is not required and some of the DetNet benefits can be gained by running over a data link layer that has not been specifically enhanced to support TSN.

Different traffic types, or application flows, can be mapped on top of DetNet. DetNet can optionally reuse header information provided by, or shared with, applications. An example of shared header fields can be found in [I-D.ietf-detnet-ip].

This document also covers concepts related to the controller plane and Operations, Administration, and Maintenance (OAM) functions related to the control plane. Data plane OAM specifics are out of scope for this document.

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 reader is assumed to be familiar with that document and its terminology.

2.2. Abbreviations

The following abbreviations are used in this document:

CW	Control Word.
DetNet	Deterministic Networking.
GRE	Generic Routing Encapsulation.
IPSec	IP Security.
L2	Layer 2.
LSR	Label Switching Router.
MPLS	Multiprotocol Label Switching.
MPLS-TE	Multiprotocol Label Switching - Traffic Engineering.
OAM	Operations, Administration, and Maintenance.
PEF	Packet Elimination Function.
PRF	Packet Replication Function.
PREOF	Packet Replication, Elimination and Ordering Functions.
POF	Packet Ordering Function.
PSN	Packet Switched Network.
PW	PseudoWire.
QoS	Quality of Service.
TSN	Time-Sensitive Network.

3. DetNet Data Plane Overview

This document describes how application flows, or app-flows, are carried over DetNet networks. The DetNet Architecture, [I-D.ietf-detnet-architecture], models the DetNet related data plane functions decomposed into two sub-layers: a service sub-layer and a forwarding sub-layer.

Figure 1 reproduced from the [I-D.ietf-detnet-architecture], shows a logical DetNet service with the two sub-layers.

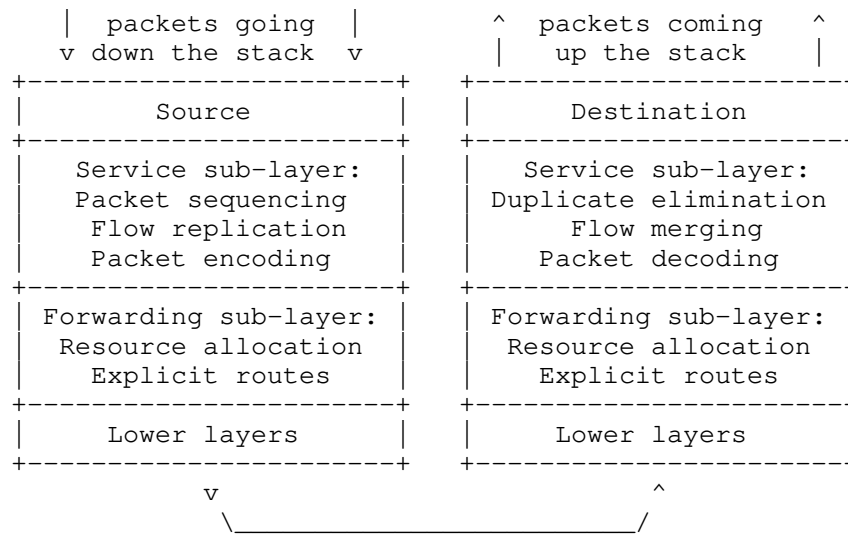


Figure 1: DetNet data plane protocol stack

The DetNet forwarding sub-layer may be directly provided by the DetNet service sub-layer, for example by IP tunnels or MPLS. Alternatively, an overlay approach may be used in which the packet is natively carried between key nodes within the DetNet network (say between PREOF nodes) and a sub-layer is used to provide the information needed to reach the next hop in the overlay.

The forwarding sub-layer provides the quality underpin needed by the DetNet flow. It may do this directly through the use of queuing techniques and traffic engineering methods, or it may do this through the assistance of its underlying connectivity. For example it may call upon Ethernet TSN capabilities defined in IEEE 802.1 TSN [IEEE802.1TSNTG].

The service sub-layer provides additional support beyond the connectivity function of the forwarding sub-layer. An example of this is Packet Replication, Elimination, and Ordering (PREOF) functions see Section 4.3.

The method of instantiating each of the layers is specific to the particular DetNet data plane method. There may be more than one approach that is applicable to a given bearer network type.

3.1. Data Plane Characteristics

There are two major characteristics to the data plane:

1. **Data plane technology:** The DetNet data plane is provided by the DetNet service and forwarding sub layers. The DetNet service sub-layer generally provides its functions for the DetNet application flows by using or applying existing standardized headers and/or encapsulations. The Detnet forwarding sub-layer may provide capabilities leveraging that same header or encapsulation technology e.g. Figure 2 or it may be achieved by other technologies e.g. Figure 3. DetNet is currently defined for operation over packet switched (IP) networks or label switched (MPLS) networks.
2. **Encapsulation format:** DetNet encodes specific flow attributes (namely flow identity and sequence number) in packets. For example, in DetNet IP, zero encapsulation may be used and no sequence number is available, and in DetNet MPLS, DetNet specific information may be added explicitly to the packets in the format of S-label and d-CW.

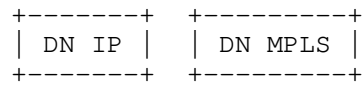


Figure 2: DetNet Services

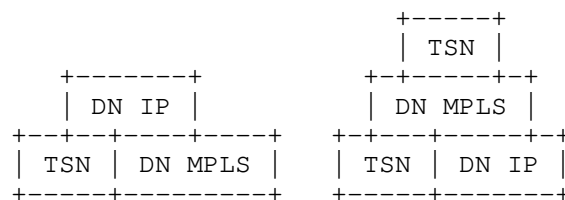


Figure 3: DetNet Service Examples

3.2. Encapsulation

The encapsulation of the DetNet flows allows them to be sent over a data plane technology other than their native type. Encapsulation is essential if, for example, it is required to send Ethernet TSN stream

as a DetNet Application over a data plane such as MPLS. Figure 3 illustrates some relationships between the components.

The use of encapsulation is also required if additional information (meta-data) is needed by the DetNet data plane and there is either no ability to include it in the client data packet, or the specification of the client data plane does not permit the modification of the packet to include additional data. An example of such meta-data is the inclusion of a sequence number required by the PREOF function.

Encapsulation may also be used to carry or aggregate flows for equipment with limited DetNet capability.

3.3. DetNet Specific Metadata

The DetNet data plane can provide or carry meta-data:

1. Flow-ID
2. Sequence Number

Both of these metadata are required for DetNet service sub-layer specific functions (e.g., PREOF). DetNet forwarding sub-layer related functions require only Flow-ID.

Metadata can be a useful way of identifying packets that need to be treated as a flow or flow aggregate. It is also useful as a way of including a sequence number the packet for use by the PREOF function or as a place to carry OAM indications or OAM information to instrument DetNet data plane operation.

Explicit inclusion of metadata is possible through the use of IP options or IP extension headers. New IP options are almost impossible to get standardized or to deploy in an operational network and will not be discussed further in this text. IPv6 extensions headers are finding popularity in current IPv6 development work, particularly in connection with Segment Routing of IPv6 (SRv6) and IP OAM. The design of a new IPv6 extension header or the modification of an existing one is a technique available in the tool box of the DetNet IP data plane designer.

Explicit inclusion of metadata in an IP packet is also possible through the inclusion of an MPLS label stack and the MPLS DetNet Control Word using one of the methods for carrying MPLS over IP [I-D.ietf-detnet-mpls-over-udp-ip]. This is described in more detail in Section 3.6.4.

Implicit metadata in IP can be included through the use of the network programming paradigm [I-D.ietf-spring-srv6-network-programming] in which the suffix of an IPv6 address is used to encode additional information for use by the network of the receiving host.

Some MPLS examples of implicit metadata include the sequence number for use by the PREOF function, or even all the essential information being included into the DetNet over MPLS label stack (the DetNet Control Word and the DetNet Service label).

3.4. DetNet IP Data Plane

An IP data plane may operate natively or through the use of an encapsulation. Many types of IP encapsulation can satisfy DetNet requirements and it is anticipated that more than one encapsulation may be deployed for example GRE, IPSec etc.

One method of operating an IP DetNet data plane without encapsulation is to use "6-tuple" based flow identification, where "6-tuple" refers to information carried in IP and higher layer protocol headers. General background on the use of IP headers, and "6-tuples", to identify flows and support Quality of Service (QoS) can be found in [RFC3670]. [RFC7657] also provides useful background on the delivery differentiated services (DiffServ) and "tuple" based flow identification. DetNet flow aggregation may be enabled via the use of wildcards, masks, prefixes and ranges. The operation of this method is described in detail in [I-D.ietf-detnet-ip].

The DetNet forwarding plane may use explicit route capabilities and traffic engineering capabilities to provide a forwarding sub-layer that is responsible for providing resource allocation and explicit routes. It is possible to include such information in a native IP packet explicitly, or implicitly.

3.5. DetNet MPLS Data Plane

MPLS provides the ability to forward traffic over implicit and explicit paths to the point in the network where the next DetNet service sub-layer action needs to take place. It does this through the use of a stack of one or more labels with various forwarding semantics.

MPLS also provides the ability to identify a service instance that is used to process the packet through the use of a label that maps the packet to a service instance.

In cases where metadata is needed to process an MPLS encapsulated packet at the service sub-layer, a shim layer also called a control word (CW) [RFC4385] can be used. Although such CWs are frequently 32 bits long, there is no architectural constraint on its size of this structure, only the requirement that it is fully understood by all parties operating on it in the DetNet service sub-layer. The operation of this method is described in detail in [I-D.ietf-detnet-mpls].

3.6. Further DetNet Data Plane Considerations

This section provides informative considerations related to providing DetNet service to flows which are identified based on their header information. At a high level, the following are provided on a per flow basis:

Reservation and Allocation of resources:

Reservation of resources can allocate resources to specific DetNet flows. This can eliminate packet contention and loss for DetNet traffic. This also can reduce jitter for the DetNet traffic. DetNet flows are assumed to behave with respect to the reserved traffic profile. If other traffic shares the link resources, the use of (queuing, policing, shaping) policies can be used to ensure that the allocation of resources reserved for DetNet is met. Queuing and shaping of DetNet traffic could be required to ensure that DetNet traffic does not exceed its reserved profile but this would impact the DetNet service characteristics.

Explicit routes:

Use of a specific path for a flow. This allows control of the network delay by steering the packet with the ability to influence the physical path. Explicit routes complement reservation by ensuring that a consistent path can be associated with its resources for the duration of that path. Coupled with the traffic mechanism, this limits misordering and bounds latency. Explicit route computation can encompass a wide set of constraints and optimize the path for a certain characteristic e.g. highest bandwidth or lowest jitter. In these cases the "best" path for any set of characteristics may not be a shortest path. The selection of path can take into account multiple network metrics. Some of these metrics are measured and distributed by the routing system as traffic engineering metrics.

Service protection:

Use of multiple packet streams using multiple paths, for example 1+1 or 1:1 linear protection. For DetNet this primarily relates to packet replication and elimination capabilities. MPLS offers a number of protection schemes. MPLS hitless protection can be used to switch traffic to an already established path in order to restore delivery rapidly after a failure. Path changes, even in the case of failure recovery, can lead to the out of order delivery of data requiring packet ordering functions either within the DetNet service or at a high layer in the application traffic. Establishment of new paths after a failure is out of scope for DetNet services.

Network Coding:

Network Coding, not to be confused with network programming, comprises several techniques where multiple data flows are encoded. These resulting flows can then be sent on different paths. The encoding operation can combine flows and error recovery information. When the encoded flows are decoded and recombined the original flows can be recovered. Note that Network coding uses an alternative to packet by packet PREOF. Therefore, for certain network topologies and traffic loads, Network Coding can be used to improve a network's throughput, efficiency, latency, and scalability, as well as resilience to partition, attacks, and eavesdropping, as compared to traditional methods. DetNet could utilize Network coding as an alternative to other protection means. Network coding is often applied in wireless networks and is being explored for other network types.

Load sharing:

Use of packet by packet distribution of the same DetNet flow over multiple paths is not recommended except for the cases listed above where PREOF is utilized to improve protection of traffic and maintain order. Packet by packet load sharing, e.g., via ECMP or UCMP, impacts ordering and possibly jitter.

Troubleshooting:

Since Detnet leverages many different forwarding sub-layers, those technologies also support a number of tools to troubleshoot connectivity for example, to support identification of misbehaving flows. At the service layer again there are existing mechanisms to troubleshoot or monitor flows. Many of these mechanisms exist for IP and MPLS networks. A client of a DetNet service can introduce any monitoring applications which can detect and monitor delay and loss.

Recognize flow(s) for analytics:

To a large degree this follows the logic in the previous section. Analytics can be inherited from the two sub-layers. At the DetNet service edge packet and bit counters e.g. sent, received, dropped, and out of sequence are maintained.

Correlate events with flows:

The provider of a DetNet service may allow other capabilities to monitor flows such as more detail loss statistics and time stamping of events. The details of these capabilities are currently out of scope for this document.

Several of these capabilities are expanded upon in more detail below.

3.6.1. Service Protection

Service protection allow DetNet services to increase reliability and maintain a DetNet Service Assurance in the case of network congestion or some failures. Detnet relies on the underlying technology capabilities for various protection schemes. Protection schemes enable partial or complete coverage of the network paths and active protection with combinations of PRF, PRE, and POF.

3.6.1.1. Linear Service Protection

An example DetNet MPLS network fragment and packet flow is illustrated in Figure 4.

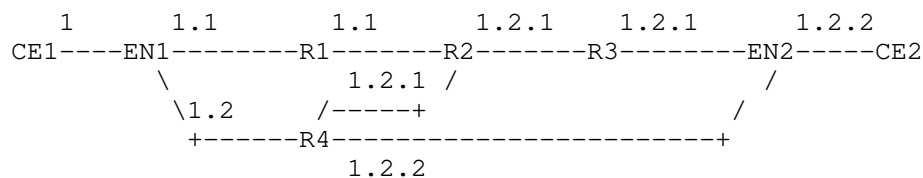


Figure 4: Example Packet Flow in DetNet protected Network

In Figure 4 the numbers are used to identify the instance of a packet. Packet 1 is the original packet, and packets 1.1, and 1.2 are two first generation copies of packet 1. Packet 1.2.1 is a second generation copy of packet 1.2 etc. Note that these numbers never appear in the packet, and are not to be confused with sequence numbers, labels or any other identifier that appears in the packet. They simply indicate the generation number of the original packet so

that its passage through the network fragment can be identified to the reader.

Customer Equipment CE1 sends a packet into the DetNet enabled network. This is packet (1). Edge Node EN1 encapsulates the packet as a DetNet Packet and sends it to Relay node R1 (packet 1.1). EN1 makes a copy of the packet (1.2), encapsulates it and sends this copy to Relay node R4.

Note that along the path from EN1 to R1 there may be zero or more nodes which, for clarity, are not shown. The same is true for any other path between two DetNet entities shown in Figure 4 .

Relay node R4 has been configured to send one copy of the packet to Relay Node R2 (packet 1.2.1) and one copy to Edge Node EN2 (packet 1.2.2).

R2 receives packet copy 1.2.1 before packet copy 1.1 arrives, and, having been configured to perform packet elimination on this DetNet flow, forwards packet 1.2.1 to Relay Node R3. Packet copy 1.1 is of no further use and so is discarded by R2.

Edge Node EN2 receives packet copy 1.2.2 from R4 before it receives packet copy 1.2.1 from R2 via relay Node R3. EN2 therefore strips any DetNet encapsulation from packet copy 1.2.2 and forwards the packet to CE2. When EN2 receives the later packet copy 1.2.1 this is discarded.

The above is of course illustrative of many network scenarios that can be configured.

This example also illustrates 1:1 protection scheme meaning there is traffic and path for each segment of the end to end path. Local DetNet relay nodes determine which packets are eliminated and which packets are forwarded. A 1+1 scheme where only one path is used for traffic at a time, could use the same topology. In this case there is no PRF function and traffic is switched upon detection of failure. An OAM scheme that monitors the paths detects the loss of path or traffic is required to initiate the switch. A POF may still be used in this case to prevent misordering of packets. In both cases the protection paths are established and maintained for the duration of the DetNet service.

3.6.1.2. Ring Service Protection

Ring protection may also be supported if the underlying technology supports it. Many of the same concepts apply however Rings are

normally 1+1 protection for data efficiency reasons. [RFC8227] is an example of MPLS-TP data plane that supports Ring protection.

3.6.2. Aggregation Considerations

The DetNet data plane also allows for the aggregation of DetNet flows, to improved scaling by reducing the state per hop. How this is accomplished is data plane or control plane dependent. When DetNet flows are aggregated, transit nodes provide service to the aggregate and not on a per-DetNet flow basis. When aggregating DetNet flows the flows should be compatible i.e. the same or very similar QoS and CoS characteristics. In this case, nodes performing aggregation will ensure that per-flow service requirements are achieved.

If bandwidth reservations are used, the sum of the reservations should be the sum of all the individual reservations, in other words, the reservations should not create an over subscription of bandwidth reservation. If maximum delay bounds are used the system should ensure that the aggregate does not exceed the delay bounds of the individual flows.

DetNet encapsulation is a data plane mechanism that can be used to aggregate traffic. Encapsulation can either be in the same service type or in a different service type see Figure 3 for example. When an encapsulation is used the choice of reserving a maximum resource level and then tracking the services in the aggregated service or adjusting the aggregated resources as the services are added is implementation and technology specific.

DetNet flows at edges must be able to handle rejection to an aggregation group due to lack of resources as well as conditions where general requirements are not satisfied.

3.6.2.1. IP Aggregation

IP aggregation has both data plane and controller plane aspects. For the data plane flows may be aggregated for treatment based on shared characteristics such as 6-tuple. Alternatively, an IP encapsulation may be used to tunnel an aggregate number of DetNet Flows between relay nodes.

3.6.2.2. MPLS Aggregation

MPLS aggregation similarly has data plane and controller plane aspects. In the case of MPLS flows are often tunneled in a forwarding sub-layer and reservation is associated with that MPLS tunnel.

3.6.3. End-System Specific Considerations

Data-flows requiring DetNet service are generated and terminated on end-systems. Encapsulation depends on the application and its preferences. For example, a DetNet MPLS domain the DN functions use the d-CWs, S-Labels and F-Labels to provide DetNet services. However, an application may exchange further flow related parameters (e.g., time-stamp), which are not provided by DN functions.

As a general rule, DetNet domains are capable of forwarding any DetNet flows and the DetNet domain does not mandate the end-system or edge system encapsulation format. Unless there is a proxy of some form present, end-systems peer with similar end-systems using the same application encapsulation format. For example, as shown in Figure 5, IP applications peer with IP applications and Ethernet applications peer with Ethernet applications.

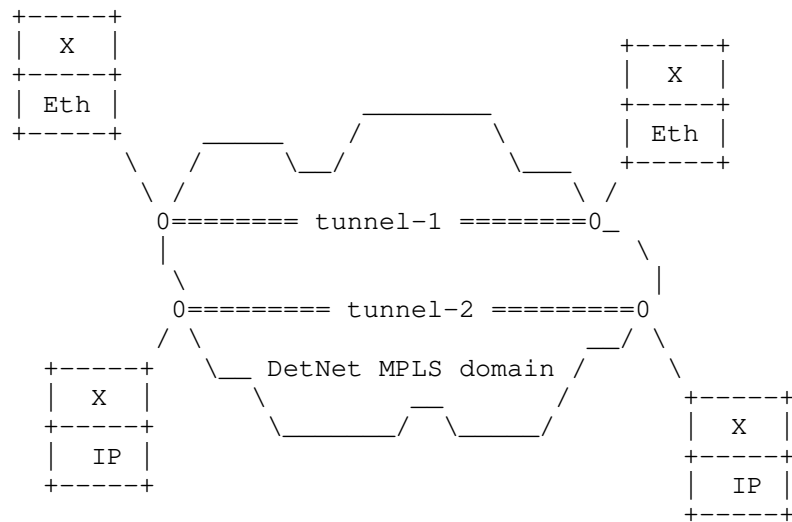


Figure 5: End-Systems and The DetNet MPLS Domain

3.6.4. Sub-Network Considerations

Any of the DetNet service types may be transported by another DetNet service. MPLS nodes may interconnected by different sub-network technologies, which may include point-to-point links. Each of these sub-network technologies need to provide appropriate service to DetNet flows. In some cases, e.g., on dedicated point-to-point links or TDM technologies, all that is required is for a DetNet node to appropriately queue its output traffic. In other cases, DetNet nodes

will need to map DetNet flows to the flow semantics (i.e., identifiers) and mechanisms used by an underlying sub-network technology. Figure 6 shows several examples of header formats that can be used to carry DetNet MPLS flows over different sub-network technologies. L2 represent a generic layer-2 encapsulation that might be used on a point-to-point link. TSN represents the encapsulation used on an IEEE 802.1 TSN network, as described in [I-D.ietf-detnet-mpls-over-tsn]. UDP/IP represents the encapsulation used on a DetNet IP PSN, as described in [I-D.ietf-detnet-mpls-over-udp-ip].

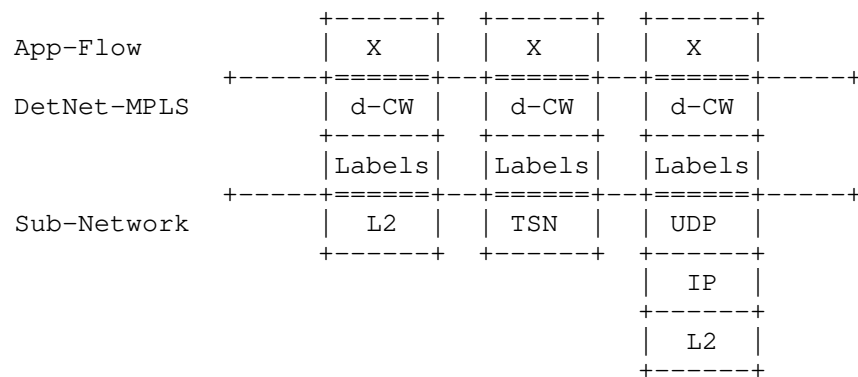


Figure 6: Example DetNet MPLS Sub-Network Formats

4. Controller Plane (Management and Control) Considerations

4.1. DetNet Controller Plane Requirements

While the definition of controller plane for DetNet is out of the scope of this document, there are particular considerations and requirements for such that result from the unique characteristics of the DetNet architecture [I-D.ietf-detnet-architecture] and data plane as defined herein.

The primary requirements of the DetNet controller plane are that it must be able to:

- o Instantiate DetNet flows in a DetNet domain (which may include some or all of explicit path determination, link bandwidth reservations, restricting flows to IEEE 802.1 TSN links, node buffer and other resource reservations, specification of required queuing disciplines along the path, ability to manage bidirectional flows, etc.) as needed for a flow.

- o In the case of MPLS, manage DetNet S-Label and F-Label allocation and distribution, where the DetNet MPLS encapsulation is in use see [I-D.ietf-detnet-mpls].
- o Support DetNet flow aggregation.
- o Advertise static and dynamic node and link resources such as capabilities and adjacencies to other network nodes (for dynamic signaling approaches) or to network controllers (for centralized approaches).
- o Scale to handle the number of DetNet flows expected in a domain (which may require per-flow signaling or provisioning).
- o Provision flow identification information at each of the nodes along the path. Flow identification may differ depending on the location in the network and the DetNet functionality (e.g. transit node vs. relay node).

These requirements, as stated earlier, could be satisfied using distributed control protocol signaling (such as RSVP-TE), centralized network management provisioning mechanisms (such as BGP, PCEP, YANG [I-D.ietf-detnet-flow-information-model], etc.) or hybrid combinations of the two, and could also make use of MPLS-based segment routing.

In the abstract, the results of either distributed signaling or centralized provisioning are equivalent from a DetNet data plane perspective – flows are instantiated, explicit routes are determined, resources are reserved, and packets are forwarded through the domain using the DetNet data plane.

However, from a practical and implementation standpoint, they are not equivalent at all. Some approaches are more scalable than others in terms of signaling load on the network. Some can take advantage of global tracking of resources in the DetNet domain for better overall network resource optimization. Some are more resilient than others if link, node, or management equipment failures occur. While a detailed analysis of the control plane alternatives is out of the scope of this document, the requirements from this document can be used as the basis of a later analysis of the alternatives.

4.2. Generic Controller Plane Considerations

This section covers control plane considerations that are independent of the data plane technology used for DetNet service delivery.

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, and the DetNet architecture [I-D.ietf-detnet-architecture] refers to these collectively as the 'Controller Plane'. This document therefore does not distinguish between information provided by distributed control plane protocols, e.g., RSVP-TE [RFC3209] and [RFC3473], or by centralized network management mechanisms, e.g., RestConf [RFC8040], YANG [RFC7950], and the Path Computation Element Communication Protocol (PCEP) [I-D.ietf-pce-pcep-extension-for-pce-controller] or any combination thereof. Specific considerations and requirements for the DetNet Controller Plane are discussed in Section 4.1.

Each respective data plane document also covers the control plane considerations for that technology. For example [I-D.ietf-detnet-ip] covers IP control plane normative considerations and [I-D.ietf-detnet-mpls] covers MPLS control plane normative considerations.

4.2.1. Flow Aggregation Control

Flow aggregation includes aggregation accomplished through the use of hierarchical LSPs in MPLS and tunnels, in the case of IP, MPLS and TSN, all of which aggregate multiple DetNet flows into a single new DetNet flow. Aggregation can also be grouping of IP flows that share 6-tuple attributes or flow identifiers at the DetNet sub-layer.

Control of aggregation involves a set of procedures listed here. Aggregation may use some or all of these capabilities and the order may vary:

- o Traffic engineering resource collection and distribution:

- Available resources are tracked through control plane or management plane databases and distributed amongst controllers or nodes that can manage resources.

- o Path computation and resource allocation:

- When DetNet services are provisioned or requested one or more paths meeting the requirements are selected and the resources verified and recorded.

- o Resource assignment and data plane co-ordination:

- The assignment of resources along the path depends on the technology and it includes assignment of specific links and

coordination of the queuing and other traffic management capabilities such as policing and shaping.

- o Assigned Resource recording and updating:

Depending on the specific technology the assigned resources are updated and distributed in the databases preventing over subscription.

4.2.2. Explicit Routes

Explicit routes are used to ensure that packets are routed through the resources that have been reserved for them, and hence provide the DetNet application with the required service. A requirement for the DetNet Controller Plane will be the ability to assign a particular identified DetNet IP flow to a path through the DetNet domain that has been assigned the required nodal resources. This provides the appropriate traffic treatment for the flow and also includes particular links as a part of the path that are able to support the DetNet flow. For example, by using IEEE 802.1 TSN links (as discussed in [I-D.ietf-detnet-mpls-over-tsn]) DetNet parameters can be maintained. Further considerations and requirements for the DetNet Controller Plane are discussed in Section 4.1.

Whether configuring, calculating and instantiating these routes is a single-stage or multi-stage process, or in a centralized or distributed manner, is out of scope of this document.

There are several approaches that could be used to provide explicit routes and resource allocation in the DetNet forwarding sub-layer. For example:

- o The path could be explicitly set up by a controller which calculates the path and explicitly configures each node along that path with the appropriate forwarding and resource allocation information.
- o The path could use a distributed control plane such as RSVP [RFC2205] or RSVP-TE [RFC3473] extended to support DetNet IP flows.
- o The path could be implemented using IPv6-based segment routing when extended to support resource allocation.

See Section 4.1 for further discussion of these alternatives. In addition, [RFC2386] contains useful background information on QoS-based routing, and [RFC5575] discusses a specific mechanism used by BGP for traffic flow specification and policy-based routing.

4.2.3. Contention Loss and Jitter Reduction

As discussed in Section 1, this document does not specify the mechanisms needed to eliminate packet contention, packet loss or reduce jitter for DetNet flows at the DetNet forwarding sub-layer. The ability to manage node and link resources to be able to provide these functions is a necessary part of the DetNet controller plane. It is also necessary to be able to control the required queuing mechanisms used to provide these functions along a flow's path through the network. See [I-D.ietf-detnet-ip] and Section 4.1 for further discussion of these requirements.

4.2.4. Bidirectional Traffic

DetNet applications typically generate bidirectional traffic. IP and MPLS typically treat each direction separately and do not force interdependence of each direction. MPLS has considered bidirectional traffic requirements and the MPLS definitions from [RFC5654] are useful to illustrate terms such as 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 forwarding path. This is analogous to standard IP routing. 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 reservations may differ in each direction. The concepts of associated bidirectional flows and co-routed bidirectional flows can also be applied to DetNet IP flows.

While the DetNet IP data plane must support bidirectional DetNet flows, there are no special bidirectional features with respect to the data plane other than the need for the two directions of a co-routed bidirectional flow to take the same path. That is to say that bidirectional DetNet flows are solely represented at the management and control plane levels, without specific support or knowledge within the DetNet data plane. Fate sharing and associated or co-routed bidirectional flows, can be managed at the control level.

DetNet's use of PREOF may increase the complexity of using co-routing bidirectional flows, since if PREOF is used, then the replication points in one direction would have to match the elimination points in the other direction, and vice versa, and the optimal points for these functions in one direction may not match the optimal points in the

other subsequent to the network and traffic constraints. Furthermore, due to the per packet service protection nature, bidirectional forwarding per packet may not be ensured. The first packet of received member flows is selected by the elimination function independently of which path it has taken through the network.

Control and management mechanisms need to support bidirectional flows, but the specification of such mechanisms are out of scope of this document. An example control plane solution for MPLS can be found in [RFC3473] , [RFC6387] and [RFC7551]. These requirements are included in Section 4.1.

4.3. Packet Replication, Elimination, and Ordering (PREOF)

The controller plane protocol solution required for managing the PREOF processing is outside the scope of this document. That said, it should be noted that the ability to determine, for a particular flow, optimal packet replication and elimination points in the DetNet domain requires explicit support. There may be capabilities that can be used, or extended, for example GMPLS end-to-end recovery [RFC4872] and GMPLS segment recovery [RFC4873].

5. Security Considerations

Security considerations for DetNet are described in detail in [I-D.ietf-detnet-security]. General security considerations are described in [I-D.ietf-detnet-architecture]. This section considers general security considerations applicable to all data planes.

Security aspects which are unique to DetNet are those whose aim is to provide the specific quality of service aspects of DetNet, which are primarily to deliver data flows with extremely low packet loss rates and bounded end-to-end delivery latency.

The primary considerations for the data plane is to maintain integrity of data and delivery of the associated DetNet service traversing the DetNet network. Application flows can be protected through whatever means is provided by the underlying technology. For example, encryption may be used, such as that provided by IPsec [RFC4301] for IP flows and/or by an underlying sub-net using MACSec [IEEE802.1AE-2018] for Ethernet (Layer-2) flows.

From a data plane perspective DetNet does not add or modify any header information.

At the management and control level DetNet flows are identified on a per-flow basis, which may provide controller plane attackers with

additional information about the data flows (when compared to controller planes that do not include per-flow identification). This is an inherent property of DetNet which has security implications that should be considered when determining if DetNet is a suitable technology for any given use case.

To provide uninterrupted availability of the DetNet service, provisions can be made against DOS attacks and delay attacks. To protect against DOS attacks, excess traffic due to malicious or malfunctioning devices can be prevented or mitigated, for example through the use of existing mechanism such as policing and shaping applied at the input of a DetNet domain. To prevent DetNet packets from being delayed by an entity external to a DetNet domain, DetNet technology definition can allow for the mitigation of Man-In-The-Middle attacks, for example through use of authentication and authorization of devices within the DetNet domain.

6. IANA Considerations

This document makes no IANA requests.

7. Acknowledgements

The authors wish to thank Pat Thaler, Norman Finn, Loa Anderson, David Black, Rodney Cummings, Ethan Grossman, Tal Mizrahi, David Mozes, Craig Gunther, George Swallow, Yuanlong Jiang and Carlos J. Bernardos for their various contributions to this work.

8. References

8.1. Normative References

- [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>>.
- [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>>.
- [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>>.

8.2. Informative References

- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", draft-ietf-
detnet-architecture-13 (work in progress), May 2019.
- [I-D.ietf-detnet-flow-information-model]
Farkas, J., Varga, B., Cummings, R., and Y. Jiang, "DetNet
Flow Information Model", draft-ietf-detnet-flow-
information-model-03 (work in progress), March 2019.
- [I-D.ietf-detnet-ip]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane: IP",
draft-ietf-detnet-ip-00 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS",
draft-ietf-detnet-mpls-00 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls-over-tsn]
Varga, B., Farkas, J., Malis, A., Bryant, S., and J.
Korhonen, "DetNet Data Plane: MPLS over IEEE 802.1 Time
Sensitive Networking (TSN)", draft-ietf-detnet-mpls-over-
tsn-00 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls-over-udp-ip]
Varga, B., Farkas, J., Berger, L., Malis, A., Bryant, S.,
and J. Korhonen, "DetNet Data Plane: MPLS over IP", draft-
ietf-detnet-mpls-over-udp-ip-00 (work in progress), May
2019.
- [I-D.ietf-detnet-security]
Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell,
J., Austad, H., Stanton, K., and N. Finn, "Deterministic
Networking (DetNet) Security Considerations", draft-ietf-
detnet-security-04 (work in progress), March 2019.
- [I-D.ietf-pce-pcep-extension-for-pce-controller]
Zhao, Q., Li, Z., Negi, M., and C. Zhou, "PCEP Procedures
and Protocol Extensions for Using PCE as a Central
Controller (PCECC) of LSPs", draft-ietf-pce-pcep-
extension-for-pce-controller-01 (work in progress),
February 2019.

- [I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J.,
daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6
Network Programming", draft-ietf-spring-srv6-network-
programming-00 (work in progress), April 2019.
- [IEEE802.1AE-2018]
IEEE Standards Association, "IEEE Std 802.1AE-2018 MAC
Security (MACsec)", 2018,
<<https://ieeexplore.ieee.org/document/8585421>>.
- [IEEE802.1TSNTG]
IEEE Standards Association, "IEEE 802.1 Time-Sensitive
Networking Task Group", <<http://www.ieee802.org/1/tsn>>.
- [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>>.
- [RFC2386] Crawley, E., Nair, R., Rajagopalan, B., and H. Sandick, "A
Framework for QoS-based Routing in the Internet",
RFC 2386, DOI 10.17487/RFC2386, August 1998,
<<https://www.rfc-editor.org/info/rfc2386>>.
- [RFC3670] Moore, B., Durham, D., Strassner, J., Westerinen, A., and
W. Weiss, "Information Model for Describing Network Device
QoS Datapath Mechanisms", RFC 3670, DOI 10.17487/RFC3670,
January 2004, <<https://www.rfc-editor.org/info/rfc3670>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the
Internet Protocol", RFC 4301, DOI 10.17487/RFC4301,
December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4872] Lang, J., Ed., Rekhter, Y., Ed., and D. Papadimitriou,
Ed., "RSVP-TE Extensions in Support of End-to-End
Generalized Multi-Protocol Label Switching (GMPLS)
Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007,
<<https://www.rfc-editor.org/info/rfc4872>>.
- [RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel,
"GMPLS Segment Recovery", RFC 4873, DOI 10.17487/RFC4873,
May 2007, <<https://www.rfc-editor.org/info/rfc4873>>.
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J.,
and D. McPherson, "Dissemination of Flow Specification
Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009,
<<https://www.rfc-editor.org/info/rfc5575>>.

- [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>>.
- [RFC6387] Takacs, A., Berger, L., Caviglia, D., Fedyk, D., and J. Meuric, "GMPLS Asymmetric Bandwidth Bidirectional Label Switched Paths (LSPs)", RFC 6387, DOI 10.17487/RFC6387, September 2011, <<https://www.rfc-editor.org/info/rfc6387>>.
- [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>>.
- [RFC7657] Black, D., Ed. and P. Jones, "Differentiated Services (Diffserv) and Real-Time Communication", RFC 7657, DOI 10.17487/RFC7657, November 2015, <<https://www.rfc-editor.org/info/rfc7657>>.
- [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>>.
- [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>>.

Authors' Addresses

Balazs Varga (editor)
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

Janos Farkas
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: janos.farkas@ericsson.com

Lou Berger
LabN Consulting, L.L.C.

Email: lberger@labn.net

Don Fedyk
LabN Consulting, L.L.C.

Email: dfedyk@labn.net

Andrew G. Malis
Futurewei Technologies

Email: agmalis@gmail.com

Stewart Bryant
Futurewei Technologies

Email: stewart.bryant@gmail.com

Jouni Korhonen

Email: jouni.nospam@gmail.com

DetNet
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

J. Farkas
B. Varga
Ericsson
R. Cummings
National Instruments
Y. Jiang
Huawei Technologies Co., Ltd.
July 08, 2019

DetNet Flow Information Model
draft-ietf-detnet-flow-information-model-04

Abstract

This document describes flow and service information model for Deterministic Networking (DetNet). These models are defined for IP and MPLS DetNet data planes

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	3
1.1. Goals	5
1.2. Non Goals	5
2. Terminology	5
2.1. Terms Used in This Document	5
2.2. Abbreviations	6
2.3. Naming Conventions	6
2.4. Requirements Language	7
3. DetNet Domain and its Modeling	7
3.1. DetNet Service Overview	7
3.2. Reference Points Used in Modeling	7
3.3. Information Elements	8
4. App-flow Related Parameters	8
4.1. App-flow Characteristics	8
4.2. App-flow Requirements	9
5. DetNet Flow Related Parameters	9
5.1. Management ID of the DetNet Flow	10
5.2. Payload type of the DetNet Flow	10
5.3. Format of the DetNet Flow	10
5.4. Identification and Specification of DetNet Flows	10
5.4.1. DetNet MPLS Flow Identification and Specification	11
5.4.2. DetNet IP Flow Identification and Specification	11
5.5. Traffic Specification of the DetNet Flow	11
5.6. Endpoints of the DetNet Flow	12
5.7. Rank of the DetNet Flow	12
5.8. Status of the DetNet Flow	12
5.9. Requirements of the DetNet Flow	13
5.9.1. Minimum Bandwidth of the DetNet Flow	14
5.9.2. Maximum Latency of the DetNet Flow	14
5.9.3. Maximum Latency Variation of the DetNet Flow	14
5.9.4. Maximum Loss of the DetNet Flow	14
5.9.5. Maximum Consecutive Loss of the DetNet Flow	14
5.9.6. Maximum Misordering Tolerance of the DetNet Flow	14
5.10. BiDir requirement of the DetNet Flow	14
6. DetNet Service Related Parameters	15
6.1. Management ID of the DetNet service	15
6.2. Delivery Type of the DetNet service	15
6.3. Delivery Profile of the DetNet Service	15
6.3.1. Minimum Bandwidth of the DetNet Service	16
6.3.2. Maximum Latency of the DetNet Service	16
6.3.3. Maximum Latency Variation of the DetNet Service	16
6.3.4. Maximum Loss of the DetNet Service	16
6.3.5. Maximum Consecutive Loss of the DetNet Service	16

6.3.6. Maximum Misordering Tolerance of the DetNet Service .	16
6.4. Connectivity Type of the DetNet Service	16
6.5. BiDir requirement of the DetNet Service	17
6.6. Rank of the DetNet Service	17
6.7. Status of the DetNet Service	17
7. Flow Specific Operations	18
7.1. Join Operation	18
7.2. Leave Operation	19
7.3. Modify Operation	19
8. Summary	19
9. IANA Considerations	19
10. Security Considerations	19
11. References	19
11.1. Normative References	19
11.2. Informative References	20
Authors' Addresses	21

1. Introduction

A Deterministic Networking (DetNet) service provides a capability to carry a unicast or a multicast data flow for an application with constrained requirements on network performance, e.g., low packet loss rate and/or latency. DetNet and TSN have common architecture as expressed in [IETFDetNet] and [I-D.ietf-detnet-architecture]. The DetNet service is provided for DetNet flows via the DetNet service and forwarding sub-layers.

DetNet service is IP or MPLS and DetNet is currently defined for IP and MPLS networks as shown in Figure 1 based on Figure 2 and Figure 3 of [I-D.ietf-detnet-data-plane-framework]. A DetNet flow includes one or more App-flow(s) as payload. App-flows can be Ethernet, MPLS, or IP flows, which impacts what header fields are use in order to identify a flow. DetNet flows are created by DetNet encapsulation of App-flow(s) (e.g., with added MPLS labels, etc.). In some scenarios App-flow and DetNet flow look similar on the wire (e.g., L3 App-flow over a DetNet IP network).

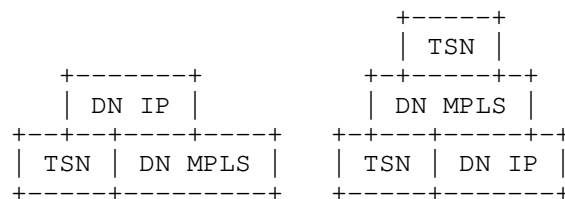


Figure 1: DetNet Service Examples as per Data Plane Framework

As shown in Figure 1 as per [I-D.ietf-detnet-data-plane-framework] a DetNet flow can be treated as an application level flow (App-flow) e.g., at DetNet flow aggregation or in a sub-network that interconnects DetNet nodes.

The DetNet flow and service information model provided by this document contains both DetNet flow and App-flow specific information in an integrated fashion.

In a given network scenario three information models can distinguished:

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

Service and flow information models are used between the user and the network operator. Configuration information models are used between the management/control plane entity of the network and the network nodes. They are shown in Figure 2.

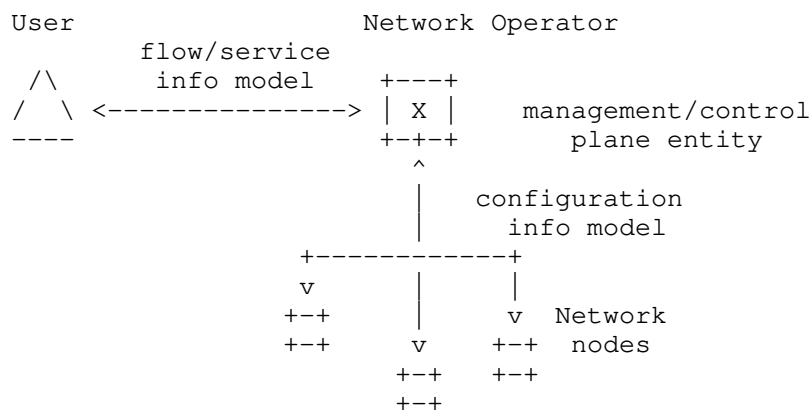


Figure 2: Usage of Information models (flow, service and configuration)

DetNet flow and service information model is based on [I-D.ietf-detnet-architecture] and on the concept of data model specified by [IEEE8021Qcc]. Furthermore, the starting point of the DetNet flow information model was the flow identification possibilities described in [IEEE8021CB], which is used by [IEEE8021Qcc] as well. In addition to TSN data model, [IEEE8021Qcc] also specifies configuration of TSN features (e.g., traffic scheduling specified by [IEEE8021Qbv]). Due to the common architecture and flow model, configuration features can be leveraged in certain deployment scenarios, e.g., when the network that provides the DetNet service includes both L3 and L2 network segments.

1.1. Goals

As it is expressed in the Charter [IETFDetNet], the DetNet WG collaborates with IEEE 802.1 TSN in order to define a common architecture for both Layer 2 and Layer 3, which is beneficial for various reasons, e.g., in order to simplify implementations. The flow and service information models should be also aligned along those lines. Therefore, the DetNet flow and service information models described in this document are based on [IEEE8021Qcc], which is an amendment to [IEEE8021Q].

This document intends to specify flow and service information models only.

1.2. Non Goals

This document (this revision) does not intend to specify either flow data model or DetNet configuration. From these aspects, the goals of this document differ from the goals of [IEEE8021Qcc], which also specifies data model and configuration of certain TSN features.

2. 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 the DetNet Data Plane Framework [I-D.ietf-detnet-data-plane-framework]. The reader is assumed to be familiar with these documents and any terminology defined therein. The DetNet <=> TSN dictionary of [I-D.ietf-detnet-architecture] is used to perform translation from [IEEE8021Qcc] to this document.

The following terminology is used according to [I-D.ietf-detnet-architecture]:

App-flow	The payload (data) carried over a DetNet service.
DetNet flow	A DetNet flow is a sequence of packets which conform uniquely to a flow identifier, and to which the DetNet service is to be provided. It includes any DetNet headers added to support the DetNet service and forwarding sub-layers.

The following terminology is introduced in this document:

Source	Reference point for an App-flow, where the flow starts.
Destination	Reference point for an App-flow, where the flow terminates.
DN Ingress	Reference point for DetNet flow, where it starts. Networking technology specific encapsulation may be added here to the served App-flow(s).
DN Egress	Reference point for DetNet flow, where it terminates. Networking technology specific encapsulation may be removed here from the served App-flow(s).

2.2. Abbreviations

The following abbreviations are used in this document:

DetNet	Deterministic Networking.
DN	DetNet.
MPLS	Multiprotocol Label Switching.
PSN	Packet Switched Network.
TSN	Time-Sensitive Networking.

2.3. Naming Conventions

The following naming conventions were used for naming information model components in this document. It is recommended that extensions of the model use the same conventions.

- o Names SHOULD be descriptive.
- o Names MUST start with uppercase letters.

- o Composed names MUST use capital letters for the first letter of each component. All other letters are lowercase, even for acronyms. Exceptions are made for acronyms containing a mixture of lowercase and capital letters, such as IPv6. Examples are SourceMacAddress and DestinationIPv6Address.

2.4. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. DetNet Domain and its Modeling

3.1. DetNet Service Overview

The DetNet service can be defined as a service that provides a capability to carry a unicast or a multicast data flow for an application with constrained requirements on network performance, e.g., low packet loss rate and/or latency.

Figure 5. and Figure 8. in [I-D.ietf-detnet-architecture] show the DetNet service related reference points and main components.

3.2. Reference Points Used in Modeling

From service design perspective a fundamental question is the location of the service/flow endpoints, i.e., where the service/flow starts and ends.

App-flow specific reference points are the Source (where it starts) and the Destination (where it terminates). Similarly a DetNet flow have reference points named as DN Ingress (where it starts) and DN Egress (where it ends). These reference points may coexist in the same node (e.g., in a DetNet IP end system). DN Ingress and DN Egress reference points are intermediate reference points for a served App-flow.

All reference points are assumed in this document to be packet-based reference points. A DN Ingress may add and a DN Egress may remove networking technology specific encapsulation to/from the served App-flow(s) (e.g., MPLS label(s), UDP and IP headers).

3.3. Information Elements

The DetNet flow information model and the service model relies on three groups of information elements:

- o App-flow related parameters: they describe the App-flow characteristics (e.g., identification, encapsulation, traffic specification, endpoints, status, etc.) and the App-flow requirements (e.g., delay, loss, etc.).
- o DetNet flow related parameters: they describe the DetNet flow characteristics (e.g., identification, format, traffic specification, endpoints, rank, etc.).
- o DetNet service related parameters: they describe the expected service characteristics (e.g., delivery type, connectivity delay/loss, status, rank, etc.).

In the information model a DetNet flow contains one or more App-flows (N:1 mapping). During DetNet aggregation the aggregated DetNet flows are treated as App-flows and the aggregate is the DetNet flow, which provides N:1 mapping. Similarly, there is a M:1 relationship of DetNet flow(s) and a DetNet Service.

4. App-flow Related Parameters

Deterministic service is required by time/loss sensitive application(s) running on an end system during communication with its peer(s). Such a data exchange has various requirements on delay and/or loss parameters.

4.1. App-flow Characteristics

App-flow characteristics are described with the following parameters:

- o FlowID: it is a unique (management) identifier of the App-flow. It can be used to define the N:1 mapping of App-flows to a DetNet flow.
- o FlowType: it is set according to the encapsulation format of the flow. It can be Ethernet (TSN), MPLS, or IP.
- o DataFlowSpecification: it is a flow descriptor, defining which packets belongs to a flow using, e.g., FlowType specific packet header fields like src-addr, dst-addr, label, VLAN-ID, etc.

- o TrafficSpecification: it is a flow descriptor, defining traffic parameters like packet size, interval, and max. packets per interval.
- o FlowEndpoints: it defines the start and termination reference points of the App-flow by pointing to the source interface/node and destination interface(s)/node(s).
- o FlowStatus: it provides the status of the App-flow with respect to the establishment of the flow by the network, e.g., ready, failed, etc.
- o FlowRank: it provides the rank of this flow relative to other flows in the network.

4.2. App-flow Requirements

App-flow requirements are described with the following parameters:

- o FlowRequirements: it defines the requirement of the App-flow regarding bandwidth, latency, latency variation, loss, and misorder tolerance.
- o FlowBiDir: it defines the requirement of the App-flow whether it has to be routed together with other App-flow(s) through the network, e.g., to provide congruent paths in the two directions.

5. DetNet Flow Related Parameters

Data model specified by [IEEE8021Qcc] describes data flows using TSN service as periodic flows with fix packet size (i.e., Constant Bit Rate (CBR) flows) or with variable packet size. The same concept is applied for flows using DetNet service.

Latency and loss parameters are correlated because the effect of late delivery can result data loss for an application. However, not all applications require hard limits on both parameters (latency and loss). For example, some real-time applications allow graceful degradation if loss happens (e.g., sample-based processing, media distribution). Some others may require high-bandwidth connections that make the usage of techniques like packet replication economically challenging or even impossible. Some applications may not tolerate loss, but are not latency sensitive (e.g., bufferless sensors). Time/loss sensitive applications may have somewhat special requirements especially for loss (e.g., no loss in two consecutive communication cycles; very low outage time, etc.).

DetNet flows have the following attributes:

- a. DnFlowID (Section 5.1)
- b. DnPayloadType (Section 5.2)
- c. DnFlowFormat (Section 5.3)
- d. DnFlowSpecification (Section 5.4)
- e. DnTrafficSpecification (Section 5.5)
- f. DnFlowEndpoints (Section 5.6)
- g. DnFlowRank (Section 5.7)
- h. DnFlowStatus (Section 5.8)

DetNet flows have the following requirement attributes:

- o DnFlowRequirements (Section 5.9)
- o DnFlowBiDir (Section 5.10)

Flow attributes are described in the following sections.

5.1. Management ID of the DetNet Flow

A unique (management) identifier is needed for each DetNet flow within the DetNet domain. It is specified in DnFlowID. It can be used to define the M:1 mapping of DetNet flows to a DetNet service.

5.2. Payload type of the DetNet Flow

DnPayloadType attribute is set according to encapsulated App-flow format. The attribute can be Ethernet, MPLS, or IP.

5.3. Format of the DetNet Flow

DnFlowFormat attribute is set according to DetNet PSN technology. The attribute can be MPLS or IP.

5.4. Identification and Specification of DetNet Flows

Identification options for DetNet flows at the Ingress/Egress and within the DetNet domain are specified as follows; see Section 5.4.1 for DetNet MPLS flows and Section 5.4.2 for DetNetw IP flows.

5.4.1. DetNet MPLS Flow Identification and Specification

Identification of DetNet MPLS flows within the DetNet domain are used in the service information model. The attributes are specific to the MPLS forwarding paradigm within the DetNet domain [I-D.ietf-detnet-mpls]. DetNetwork MPLS flows can be identified and specified by the following attributes:

- a. SLabel
- b. FLabelStack

5.4.2. DetNet IP Flow Identification and Specification

DetNet IP flows can be identified and specified by the following attributes (6-tuple) [I-D.ietf-detnet-ip]:

- a. SourceIpAddress
- b. DestinationIpAddress
- c. IPv6FlowLabel
- d. Dscp
- e. Protocol
- f. SourcePort
- g. DestinationPort

5.5. Traffic Specification of the DetNet Flow

DnTrafficSpecification attributes specify how the DN Ingress transmits packets for the DetNet flow. This is effectively the promise/request of the DN Ingress to the network. The network uses this traffic specification to allocate resources and adjust queue parameters in network nodes.

TrafficSpecification has the following attributes:

- a. Interval: the period of time in which the traffic specification cannot be exceeded.
- b. MaxPacketsPerInterval: the maximum number of packets that the Ingress will transmit in one Interval.

- c. MaxPayloadSize: the maximum payload size that the Ingress will transmit.

These attributes can be used to describe any type of traffic (e.g., CBR, VBR, etc.) and can be used during resource allocation to represent worst case scenarios.

[[Editor's note (to be removed from a future revision): Further optional attributes can be considered to achieve more efficient resource allocation. Such optional attributes might be worth for flows with soft requirements (i.e., the flow is only loss sensitive or only delay sensitive, but not both delay-and-loss sensitive). Possible options how to extend DnTrafficSpecification attributes is for further discussion.]]

5.6. Endpoints of the DetNet Flow

DnFlowEndpoints attribute defines the starting and termination reference points of the DetNet flow by pointing to the ingress interface/node and egress interface(s)/node(s). Depending on the network scenario it defines an interface or a node. Interface can be defined for example if the App-flow is a TSN Stream and it is received over a well defined UNI interface. For example for App-flows with MPLS encapsulation defining an ingress node is more common when per platform label space is used.

5.7. Rank of the DetNet Flow

DnFlowRank provides the rank of this flow relative to other flows in the DetNet domain. Rank (range: 0-255) is used by the DetNet domain to decide which flows can and cannot exist when network resources reach their limit. Rank is used to help to determine which flows can be dropped (i.e., removed from node configuration) if for example a port of a node becomes oversubscribed (e.g., due to network re-configuration).

5.8. Status of the DetNet Flow

DnFlowStatus provides the status of the DetNet flow with respect to the establishment of the flow by the DetNet domain.

The DnFlowStatus SHALL include the following attributes:

- a. DnIngressStatus is an enumeration for the status of the flow's Ingress reference point:
 - * None: no Ingress.

- * Ready: Ingress is ready.
 - * Failed: Ingress failed.
 - * OutOfService: Administratively blocked.
- b. DnEgressStatus is an enumeration for the status of the flow's Egress reference points:
- * None: no Egress.
 - * Ready: all Egresses are ready.
 - * PartialFailed: One or more Egress ready, and one or more Egress failed. The DetNet flow can be used if the Ingress is Ready.
 - * Failed: All Egresses failed.
 - * OutOfService: Administratively blocked.
- c. FailureCode: A non-zero code that specifies the problem if the DetNet flow encounters a failure (e.g., packet replication and elimination is requested but not possible, or DnIngressStatus is Failed, or DnEgressStatus is Failed, or DnEgressStatus is PartialFailed).

[[Editor's note (to be removed from a future revision): FailureCodes to be defined for DetNet. Table 46-1 of [IEEE8021Qcc] describes TSN failure codes.]]

5.9. Requirements of the DetNet Flow

DnFlowRequirements specifies requirements to ensure proper serving of the DetNet flow.

The DnFlowRequirements includes the following attributes:

- a. MinBandwidth
- b. MaxLatency
- c. MaxLatencyVariation
- d. MaxLoss
- e. MaxConsecutiveLossTolerance

f. MaxMisordering

5.9.1. Minimum Bandwidth of the DetNet Flow

MinBandwidth is the minimum bandwidth that has to be guaranteed for the DetNet flow.

5.9.2. Maximum Latency of the DetNet Flow

MaxLatency is the maximum latency from Ingress to Egress(es) for a single packet of the DetNet flow. MaxLatency is specified as an integer number of nanoseconds.

5.9.3. Maximum Latency Variation of the DetNet Flow

MaxLatencyVariation is the difference between the minimum and the maximum end-to-end one-way latency.

5.9.4. Maximum Loss of the DetNet Flow

MaxLoss defines the maximum Packet Loss Ratio (PLR) requirement for the DetNet flow between the Ingress and Egress(es).

5.9.5. Maximum Consecutive Loss of the DetNet Flow

Some applications have special loss requirement, like MaxConsecutiveLossTolerance. The maximum consecutive loss tolerance parameter describes the maximum number of consecutive packets whose loss can be tolerated. The maximum consecutive loss tolerance can be measured for example based on sequence number.

5.9.6. Maximum Misordering Tolerance of the DetNet Flow

MaxMisordering describes the tolerable maximum number of packets that can be received out of order. The maximum allowed misordering can be measured for example based on sequence number. The value zero for the maximum allowed misordering indicates that in order delivery is required, misordering cannot be tolerated.

5.10. BiDir requirement of the DetNet Flow

DnFlowBiDir attribute defines the requirement whether the served packets have to be routed together with packets of other flows through the DetNet domain, e.g., to provide congruent paths in the two directions.

6. DetNet Service Related Parameters

DetNet service have the following attributes:

- a. DnServiceID (Section 6.1)
- b. DnServiceDeliveryType (Section 6.2)
- c. DnServiceDeliveryProfile (Section 6.3)
- d. DnServiceConnectivity (Section 6.4)
- e. DnServiceBiDir (Section 6.5)
- f. DnServiceRank (Section 6.6)
- g. DnServiceStatus (Section 6.7)

Service attributes are described in the following sections.

6.1. Management ID of the DetNet service

A unique (management) identifier is needed for each DetNet service within the DetNet domain. It is specified in DnServiceID. It can be used to define the M:1 mapping of DetNet flows to a DetNet service.

6.2. Delivery Type of the DetNet service

DnServiceDeliveryType attribute is set according to the payload of the served DetNet flow (i.e., the encapsulated App-flow format). The attribute can be Ethernet, MPLS, or IP.

6.3. Delivery Profile of the DetNet Service

DnServiceDeliveryProfile specifies delivery profile to ensure proper serving of the DetNet flow.

The DnServiceDeliveryProfile includes the following attributes:

- a. MinBandwidth
- b. MaxLatency
- c. MaxLatencyVariation
- d. MaxLoss
- e. MaxConsecutiveLossTolerance

f. MaxMisordering

6.3.1. Minimum Bandwidth of the DetNet Service

MinBandwidth is the minimum bandwidth that has to be guaranteed for the DetNet service.

6.3.2. Maximum Latency of the DetNet Service

MaxLatency is the maximum latency from Ingress to Egress(es) for a single packet of the DetNet flow. MaxLatency is specified as an integer number of nanoseconds.

6.3.3. Maximum Latency Variation of the DetNet Service

MaxLatencyVariation is the difference between the minimum and the maximum end-to-end one-way latency.

6.3.4. Maximum Loss of the DetNet Service

MaxLoss defines the maximum Packet Loss Ratio (PLR) parameter for the DetNet service between the Ingress and Egress(es) of the DetNet domain.

6.3.5. Maximum Consecutive Loss of the DetNet Service

Some applications have special loss requirement, like MaxConsecutiveLossTolerance. The maximum consecutive loss tolerance parameter describes the maximum number of consecutive packets whose loss can be tolerated. The maximum consecutive loss tolerance can be measured for example based on sequence number.

6.3.6. Maximum Misordering Tolerance of the DetNet Service

MaxMisordering describes the tolerable maximum number of packets that can be received out of order. The maximum allowed misordering can be measured for example based on sequence number. The value zero for the maximum allowed misordering indicates that in order delivery is required, misordering cannot be tolerated.

6.4. Connectivity Type of the DetNet Service

Two connectivity types are distinguished: point-to-point (p2p) and point-to-multipoint (p2mp). Connectivity type p2mp is created by a transport layer function (e.g., p2mp LSP). (Note: mp2mp connectivity is a superposition of p2mp connections.)

6.5. BiDir requirement of the DetNet Service

DnServiceBiDir attribute defines the requirement whether the served packets have to be routed together with packets of other service instances through the DetNet domain, e.g., to provide congruent paths in the two directions.

6.6. Rank of the DetNet Service

DnServiceRank attribute provides the rank of a service instance relative to other services in the DetNet domain. DnServiceRank (range: 0-255) is used by the network in case of network resource limitation scenarios.

6.7. Status of the DetNet Service

DnServiceStatus information group includes elements that specify the status of the service specific state of the DetNet domain. This information group informs the user whether or not the service is ready for use.

The DnServiceStatus SHALL include the following attributes:

- a. DnServiceIngressStatus is an enumeration for the status of the service's Ingress:
 - * None: no Ingress.
 - * Ready: Ingress is ready.
 - * Failed: Ingress failed.
 - * OutOfService: Administratively blocked.
- b. DnServiceEgressStatus is an enumeration for the status of the service's Egress:
 - * None: no Egress.
 - * Ready: all Egresses are ready.
 - * PartialFailed: One or more Egress ready, and one or more Egress failed. The DetNet flow can be used if the Ingress is Ready.
 - * Failed: All Egresses failed.
 - * OutOfService: Administratively blocked.

- c. DnServiceFailureCode: A non-zero code that specifies the problem if the DetNet service encounters a failure (e.g., packet replication and elimination is requested but not possible, or DnServiceIngressStatus is Failed, or DnServiceEgressStatus is Failed, or DnServiceEgressStatus is PartialFailed).

[[Editor's note (to be removed from a future revision):
DnServiceFailureCodes to be defined for DetNet service. Table 46-1 of [IEEE8021Qcc] describes TSN failure codes.]]

7. Flow Specific Operations

The DetNet flow information model relies on three high level information groups:

- o DnIngress: The DnIngress information group includes elements that specify the source for a single DetNet flow. This information group is applied from the user of the DetNet service to the network.
- o DnEgress: The DnEgress information group includes elements that specify the destination for a single DetNet flow. This information group is applied from the user of the DetNet service to the network.
- o DnFlowStatus: The status information group includes elements that specify the status of the flow in the network. This information group is applied from the network to the user of the DetNet service. This information group informs the user whether or not the DetNet flow is ready for use.

There are three possible operations for each DetNet flow with respect to its DetNet service at a DN Ingress or a DN Egress (similarly to App-flows at a Source or a Destination):

- o Join: DN Ingress/DN Egress intends to join the flow.
- o Leave: DN Ingress/DN Egress intends to leave the flow.
- o Modify: DN Ingress/DN Egress intends to change the flow.

7.1. Join Operation

For the join operation, the DnFlowSpecification, DnFlowRank, DnFlowEndpoint, and DnTrafficSpecification SHALL be included within the DnIngress or DnEgress information group. For the join operation, the DnServiceRequirements groups MAY be included.

7.2. Leave Operation

For the leave operation, the DnFlowSpecification and DnFlowEndpoint SHALL be included within the DnIngress or DnEgress information group.

7.3. Modify Operation

For the modify operation, the DnFlowSpecification, DnFlowRank, DnFlowEndpoint, and DnTrafficSpecification SHALL be included within the DnIngress or DnEgress information group. For the join operation, the DnServiceRequirements groups MAY be included.

Modify operation can be considered to address cases when a flow is slightly changed, e.g., only MaxPayloadSize (Section 5.5) has been changed. The advantage of having a Modify is that it allows to initiate a change of flow spec while leaving the current flow is operating until the change is accepted. If there is no linkage between the Join and the Leave, then in figuring out whether the new flow spec can be supported, the controller entity has to assume that the resources committed to the current flow are in use. Via Modify the controller entity knows that the resources supporting the current flow can be available for supporting the altered flow. Modify is considered to be an optional operation due to possible controller plane limitations.

8. Summary

This document describes DetNet flow information model and service information model for DetNet IP networks and DetNet MPLS networks.

9. IANA Considerations

N/A.

10. Security Considerations

N/A.

11. References

11.1. Normative References

[I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", draft-ietf-
detnet-architecture-13 (work in progress), May 2019.

- [I-D.ietf-detnet-ip]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP", draft-ietf-detnet-ip-01 (work in progress), July 2019.
- [I-D.ietf-detnet-mpls]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS", draft-ietf-detnet-mpls-01 (work in progress), July 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6003] Papadimitriou, D., "Ethernet Traffic Parameters", RFC 6003, DOI 10.17487/RFC6003, October 2010, <<https://www.rfc-editor.org/info/rfc6003>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

11.2. Informative References

- [I-D.ietf-detnet-data-plane-framework]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane Framework", draft-ietf-detnet-data-plane-framework-01 (work in progress), July 2019.
- [IEEE8021CB]
IEEE Standards Association, "IEEE Std 802.1CB-2017 IEEE Standard for Local and metropolitan area networks - Frame Replication and Elimination for Reliability", 2017, <<https://ieeexplore.ieee.org/document/8091139/>>.
- [IEEE8021Q]
IEEE Standards Association, "IEEE Std 802.1Q-2018 IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks", 2018, <<https://ieeexplore.ieee.org/document/8403927>>.

[IEEE8021Qbv]

IEEE Standards Association, "IEEE Std 802.1Qbv-2015 IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", 2015,
<<https://ieeexplore.ieee.org/document/7572858/>>.

[IEEE8021Qcc]

IEEE Standards Association, "IEEE Std 802.1Qcc-2018: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks -- Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements", 2018,
<<https://ieeexplore.ieee.org/document/8514112/>>.

[IEEE8021TSN]

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

[IETFDetNet]

IETF, "IETF Deterministic Networking (DetNet) Working Group", <<https://datatracker.ietf.org/wg/detnet/charter/>>.

Authors' Addresses

Janos Farkas
Ericsson
Magyar tudosok korutja 11
Budapest 1117
Hungary

Email: janos.farkas@ericsson.com

Balazs Varga
Ericsson
Magyar tudosok korutja 11
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

Rodney Cummings
National Instruments
11500 N. Mopac Expwy
Bldg. C
Austin, TX 78759-3504
USA

Email: rodney.cummings@ni.com

Yuanlong Jiang
Huawei Technologies Co., Ltd.
Bantian, Longgang district
Shenzhen 518129
China

Email: jiangyuanlong@huawei.com

DetNet
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2020

B. Varga, Ed.
J. Farkas
Ericsson
L. Berger
D. Fedyk
LabN Consulting, L.L.C.
A. Malis
S. Bryant
Futurewei Technologies
J. Korhonen
July 1, 2019

DetNet Data Plane: IP
draft-ietf-detnet-ip-01

Abstract

This document specifies the Deterministic Networking data plane when operating in an IP packet switched network.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 2, 2020.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Terms Used In This Document	3
2.2. Abbreviations	3
2.3. Requirements Language	4
3. DetNet IP Data Plane Overview	4
4. DetNet IP Data Plane Considerations	6
4.1. End-System Specific Considerations	7
4.2. DetNet Domain-Specific Considerations	7
4.3. Forwarding Sub-Layer Considerations	9
4.3.1. Class of Service	9
4.3.2. Quality of Service	10
4.4. DetNet Flow Aggregation	10
4.5. Bidirectional Traffic	11
5. DetNet IP Data Plane Procedures	11
5.1. DetNet IP Flow Identification Procedures	12
5.1.1. IP Header Information	12
5.1.2. Other Protocol Header Information	13
5.2. Forwarding Procedures	14
5.3. DetNet IP Traffic Treatment Procedures	15
6. Management and Control Information Summary	15
7. Security Considerations	16
8. IANA Considerations	17
9. Acknowledgements	17
10. References	17
10.1. Normative references	17
10.2. Informative references	19
Authors' Addresses	21

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 the DetNet Architecture [I-D.ietf-detnet-architecture].

This document specifies the DetNet data plane operation for IP hosts and routers that provide DetNet service to IP encapsulated data. No DetNet specific encapsulation is defined to support IP flows, instead the existing IP and higher layer protocol header information is used

to support flow identification and DetNet service delivery. Common data plane procedures and control information for all DetNet data planes can be found in the [I-D.ietf-detnet-data-plane-framework].

The DetNet Architecture models the DetNet related data plane functions decomposed into two sub-layers: functions into two sub-layers: a service sub-layer and a forwarding sub-layer. The service sub-layer is used to provide DetNet service protection and reordering. The forwarding sub-layer is used to provides congestion protection (low loss, assured latency, and limited out-of-order delivery). Since no DetNet specific headers are added to support DetNet IP flows, only the forwarding sub-layer functions are supported using the DetNet IP defined by this document. Service protection can be provided on a per sub-net basis using technologies such as MPLS [I-D.ietf-detnet-dp-sol-mpls] and Ethernet as specified in the IEEE 802.1 TSN task group (referred to in this document simply as IEEE802.1 TSN).

This document provides an overview of the DetNet IP data plane in Section 3, considerations that apply to providing DetNet services via the DetNet IP data plane in Section 4. Section 5 provides the procedures for hosts and routers that support IP-based DetNet services. Section 6 summarizes the set of information that is needed to identify an individual DetNet flow.

2. Terminology

2.1. Terms Used In This Document

This document uses the terminology and concepts established in the DetNet architecture [I-D.ietf-detnet-architecture], and the reader is assumed to be familiar with that document and its terminology.

2.2. Abbreviations

The following abbreviations used in this document:

CoS	Class of Service.
DetNet	Deterministic Networking.
DN	DetNet.
DiffServ	Differentiated Services
DSCP	Differentiated Services Code Point
L2	Layer-2.

L3	Layer-3.
LSP	Label-switched path.
MPLS	Multiprotocol Label Switching.
PREOF	Packet Replication, Ordering and Elimination Function.
QoS	Quality of Service.
TSN	Time-Sensitive Networking, TSN is a Task Group of the IEEE 802.1 Working Group.

2.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. DetNet IP Data Plane Overview

This document describes how IP is used by DetNet nodes, i.e., hosts and routers, identify DetNet flows and provide a DetNet service using an IP data plane. From a data plane perspective, an end-to-end IP model is followed. As mentioned above, existing IP and higher layer protocol header information is used to support flow identification and DetNet service delivery. Common data plane procedures and control information for all DetNet data planes can be found in the [I-D.ietf-detnet-data-plane-framework].

The DetNet IP data plane uses "6-tuple" based flow identification, where 6-tuple refers to information carried in IP and higher layer protocol headers. The 6-tuple referred to in this document is the same as that defined in [RFC3290]. Specifically 6-tuple is (destination address, source address, IP protocol, source port, destination port, and differentiated services (DiffServ) code point (DSCP). General background on the use of IP headers, and 5-tuples, to identify flows and support Quality of Service (QoS) can be found in [RFC3670]. [RFC7657] also provides useful background on the delivery of DiffServ and "tuple" based flow identification. Referring to a 6-tuple allows DetNet nodes to forward packets with the 6-tuple as is or remap the DSCP where required by the DetNet service.

DetNet flow aggregation may be enabled via the use of wildcards, masks, prefixes and ranges. IP tunnels may also be used to support

flow aggregation. In these cases, it is expected that DetNet aware intermediate nodes will provide DetNet service assurance on the aggregate through resource allocation and congestion control mechanisms.

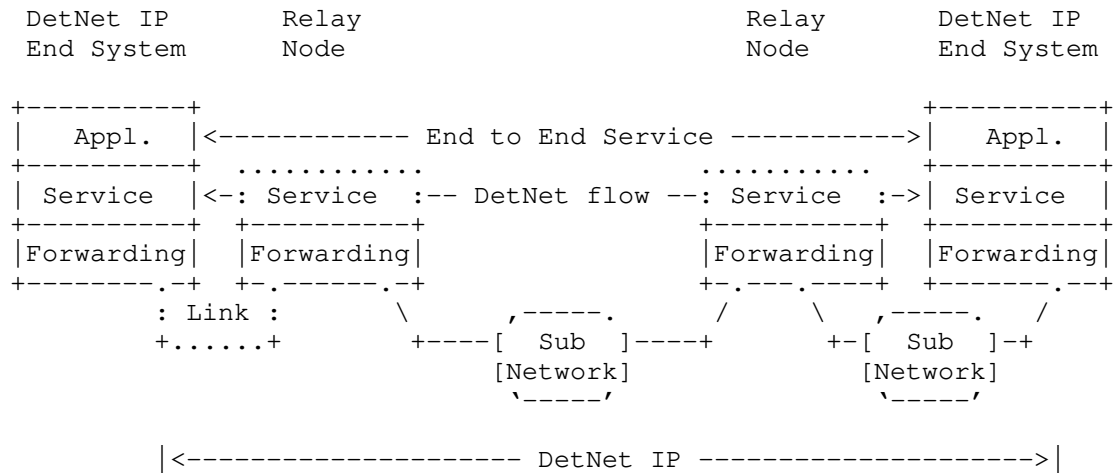


Figure 1: A Simple DetNet (DN) Enabled IP Network

Figure 1 illustrates a DetNet enabled IP network. The DetNet enabled end systems originate IP encapsulated traffic that is identified as DetNet flows, relay nodes understand the forwarding requirements of the DetNet flow and ensure that node, interface and sub-network resources are allocated to ensure DetNet service requirements. The dotted line around the Service component of the Relay Nodes indicates that the transit routers are DetNet service aware but do not perform any DetNet service sub-layer function, e.g., PREOF. IEEE 802.1 TSN is an example sub-network type which can provide support for DetNet flows and service.

Note: The sub-network can represent a TSN, MPLS or IP network segment.

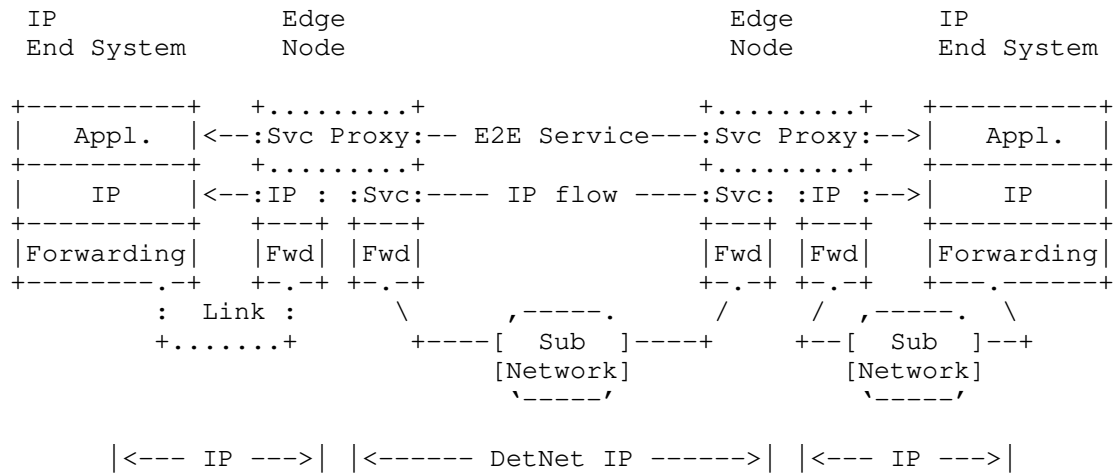


Figure 2: Non-DetNet aware IP end systems with DetNet IP Domain

Figure 2 illustrates a variant of Figure 1 where the end systems are not DetNet aware. In this case, edge nodes sit at the boundary of the DetNet domain and provide DetNet service proxies for the end applications by initiating and terminating DetNet service for the application's IP flows. The existing header information or an approach such as described in Section 4.4 can be used to support DetNet flow identification.

Note, that Figure 1 and Figure 2 can be combined, so IP DetNet End Systems can communicate over DetNet IP network with IP End System.

Non-DetNet and DetNet IP packets are identical on the wire. From data plane perspective, the only difference is that there is flow-associated DetNet information on each DetNet node that defines the flow related characteristics and required forwarding behavior. As shown above, edge nodes provide a Service Proxy function that "associates" one or more IP flows with the appropriate DetNet flow-specific information and ensures that the receives the proper traffic treatment within the domain.

Note: The operation of IEEE802.1 TSN end systems over DetNet enabled IP networks is not described in this document. TSN over MPLS is discribed in [I-D.ietf-detnet-tsn-vpn-over-mpls].

4. DetNet IP Data Plane Considerations

This section provides informative considerations related to providing DetNet service to flows which are identified based on their header information.

4.1. End-System Specific Considerations

Data-flows requiring DetNet service are generated and terminated on end systems. This document deals only with IP end systems. The protocols used by an IP end system are specific to an application and end systems peer with end systems using the same application encapsulation format. This said, DetNet's use of 6-tuple IP flow identification means that DetNet must be aware of not only the format of the IP header, but also of the next protocol carried within an IP packet.

When IP end systems are DetNet aware, no application-level or service-level proxy functions are needed inside the DetNet domain. For DetNet unaware IP end systems service-level proxy functions are needed inside the DetNet domain.

End systems need to ensure that DetNet service requirements are met when processing packets associated with a DetNet flow. When forwarding packets, this means that packets are appropriately shaped on transmission and received appropriate traffic treatment on the connected sub-network, see Section 4.3.2 and Section 4.2 for more details. When receiving packets, this means that there are appropriate local node resources, e.g., buffers, to receive and process a DetNet flow packets.

4.2. DetNet Domain-Specific Considerations

As a general rule, DetNet IP domains need to be able to forward any DetNet flow identified by the IP 6-tuple. Doing otherwise would limit end system encapsulation format. From a practical standpoint this means that all nodes along the end-to-end path of DetNet flows need to agree on what fields are used for flow identification, and the transport protocols (e.g., TCP/UDP/IPsec) which can be used to identify 6-tuple protocol ports.

From a connection type perspective two scenarios are identified:

1. DN attached: end system is directly connected to an edge node or end system is behind a sub-network. (See ES1 and ES2 in figure below)
2. DN integrated: end system is part of the DetNet domain. (See ES3 in figure below)

L3 (IP) end systems may use any of these connection types. A DetNet domain allows communication between any end-systems using the same encapsulation format, independent of their connection type and DetNet capability. DN attached end systems have no knowledge about the

DetNet domain and its encapsulation format. See Figure 3 for L3 end system connection examples.

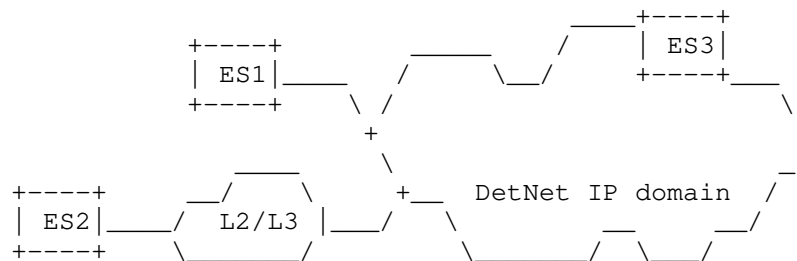


Figure 3: Connection types of L3 end systems

Within a DetNet domain, the DetNet enabled IP Routers are interconnected by links and sub-networks to support end-to-end delivery of DetNet flows. From a DetNet architecture perspective, these routers are DetNet relays, as they must be DetNet service aware. Such routers identify DetNet flows based on the IP 6-tuple, and ensure that the DetNet service required traffic treatment is provided both on the node and on any attached sub-network.

This solution provides DetNet functions end to end, but does so on a per link and sub-network basis. Congestion protection and latency control and the resource allocation (queuing, policing, shaping) are supported using the underlying link / sub net specific mechanisms. However, service protections (packet replication and packet elimination functions) are not provided at the DetNet layer end to end. Instead service protection can be provided on a per underlying L2 link and sub-network basis.

The DetNet Service Flow is mapped to the link / sub-network specific resources using an underlying system specific means. This implies each DetNet aware node on path looks into the forwarded DetNet Service Flow packet and utilize e.g., a 6-tuple to find out the required mapping within a node.

As noted earlier, the Service Protection is done within each link / sub-network independently using the domain specific mechanisms (due the lack of a unified end to end sequencing information that would be available for intermediate nodes). Therefore, service protection (if enabled) cannot be provided end-to-end, only within sub-networks. This is shown for a three sub-network scenario in Figure 4, where each sub-network can provide service protection between its borders.

"R" and "E" denotes replication and elimination points within the sub-network.

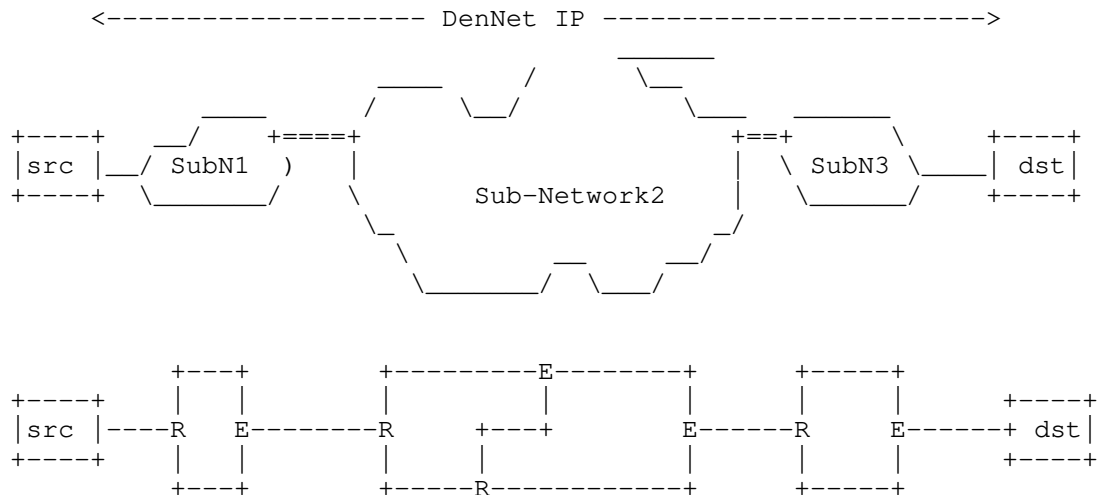


Figure 4: Replication and elimination in sub-networks for DetNet IP networks

If end to end service protection is desired, it can be implemented, for example, by the DetNet end systems using Layer-4 (L4) transport protocols or application protocols. However, these protocols are out of scope of this document.

4.3. Forwarding Sub-Layer Considerations

4.3.1. Class of Service

Class of Service (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].

4.3.2. Quality of Service

Quality of Service (QoS) for DetNet service flows carried in IP MUST be provided locally by the DetNet-aware hosts and routers supporting DetNet flows. Such support leverages the underlying network layer such as 802.1 TSN. 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, the combination of the 6-tuple i.e., the typical 5-tuple enhanced with the DSCP code, uniquely identifies a DetNet service flow.

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

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

4.4. DetNet Flow Aggregation

As described in [I-D.ietf-detnet-data-plane-framework], the ability to aggregate individual flows, and their associated resource control, into a larger aggregate is an important technique for improving scaling by reducing the state per hop. DetNet IP data plane aggregation can take place within a single node, when that node maintains state about both the aggregated and individual flows. It can also take place between nodes, where one node maintains state about only flow aggregates while the other node maintains state on all or a portion of the component flows. In either case, the management or control function that provisions the aggregate flows must ensure that adequate resources are allocated and configured to provide combined service requirements of the individual flows. As DetNet is concerned about latency and jitter, more than just bandwidth needs to be considered.

From a single node perspective, the aggregation of IP flows impacts DetNet IP data plane flow identification and resource allocation. As discussed above, IP flow identification uses the IP "6-tuple" for flow identification. DetNet IP flows can be aggregated using any of the 6-tuple fields defined in Section 5.1. The use of prefixes, wildcards, bitmasks, and value ranges allows a DetNet node to

identify aggregate DetNet flows. From a resource allocation perspective, DetNet nodes must provide service to a aggregate and not on a component flow basis.

It is the responsibility of the DetNet controller plane to properly provision the use of these aggregation mechanisms. This includes ensuring that aggregated flows have compatible e.g., the same or very similar QoS and/or CoS characteristics, see Section 4.3.2. It also includes ensuring that per component-flow service requirements are satisfied by the aggregate, see Section 5.3.

4.5. Bidirectional Traffic

While the DetNet IP data plane must support bidirectional DetNet flows, there are no special bidirectional features with respect to the data plane other than the need for the two directions of a co-routed bidirectional flow to take the same path. That is to say that bidirectional DetNet flows are solely represented at the management and control plane levels, without specific support or knowledge within the DetNet data plane. Fate sharing and associated or co-routed bidirectional flows can be managed at the control level.

Control and management mechanisms need to support bidirectional flows, but the specification of such mechanisms are out of scope of this document. An example control plane solution for MPLS can be found in [RFC7551].

5. DetNet IP Data Plane Procedures

This section provides DetNet IP data plane procedures. These procedures have been divided into the following areas: flow identification, forwarding and traffic treatment. Flow identification includes those procedures related to matching IP and higher layer protocol header information to DetNet flow (state) information and service requirements. Flow identification is also sometimes called Traffic classification, for example see [RFC5777]. Forwarding includes those procedures related to next hop selection and delivery. Traffic treatment includes those procedures related to providing an identified flow with the required DetNet service.

DetNet IP data plane establishment and operational procedures also have requirements on the control and management systems for DetNet flows and these are referred in this section. Specifically this section identifies a number of information elements that require support via the management and control interfaces supported by a DetNet node. The specific mechanism used for such support is out of the scope of this document. A summary of the requirements for management and control related information is included. Conformance

language is not used in the summary since applies to future mechanisms such as those that may be provided in YANG models [I-D.ietf-detnet-yang].

5.1. DetNet IP Flow Identification Procedures

IP and higher layer protocol header information is used to identify DetNet flows. All DetNet implementations that support this document MUST identify individual DetNet flows based on the set of information identified in this section. Note, that additional flow identification requirements, e.g., to support other higher layer protocols, may be defined in future.

The configuration and control information used to identify an individual DetNet flow MUST be ordered by an implementation. Implementations MUST support a fixed order when identifying flows, and MUST identify a DetNet flow by the first set of matching flow information.

Implementations of this document MUST support DetNet flow identification when the implementation is acting as a DetNet end systems, a relay node or as an edge node.

5.1.1. IP Header Information

Implementations of this document MUST support DetNet flow identification based on IP header information. The IPv4 header is defined in [RFC0791] and the IPv6 is defined in [RFC8200].

5.1.1.1. Source Address Field

Implementations of this document MUST support DetNet flow identification based on the Source Address field of an IP packet. Implementations SHOULD support longest prefix matching for this field, see [RFC1812] and [RFC7608]. Note that a prefix length of zero (0) effectively means that the field is ignored.

5.1.1.2. Destination Address Field

Implementations of this document MUST support DetNet flow identification based on the Destination Address field of an IP packet. Implementations SHOULD support longest prefix matching for this field, see [RFC1812] and [RFC7608]. Note that a prefix length of zero (0) effectively means that the field is ignored.

Note: any IP address value is allowed, including an IP multicast destination address.

5.1.1.3. IPv4 Protocol and IPv6 Next Header Fields

Implementations of this document MUST support DetNet flow identification based on the IPv4 Protocol field when processing IPv4 packets, and the IPv6 Next Header Field when processing IPv6 packets. An implementation MUST support flow identification based on the next protocol values defined in Section 5.1.2. Other, non-zero values, MUST be used for flow identification. Implementations SHOULD allow for these fields to be ignored for a specific DetNet flow.

5.1.1.4. IPv4 Type of Service and IPv6 Traffic Class Fields

These fields are used to support Differentiated Services [RFC2474] and Explicit Congestion Notification [RFC3168]. Implementations of this document MUST support DetNet flow identification based on the IPv4 Type of Service field when processing IPv4 packets, and the IPv6 Traffic Class Field when processing IPv6 packets. Implementations MUST support bitmask based matching, where bits set to one (1) in the bitmask indicate which subset of the bits in the field are to be used in determining a match. Note that all bits set to zero (0) value as a bitmask effectively means that these fields are ignored.

5.1.1.5. IPv6 Flow Label Field

Implementations of this document SHOULD support identification of DetNet flows based on the IPv6 Flow Label field. Implementations that support matching based on this field MUST allow for this field to be ignored for a specific DetNet flow. When this field is used to identify a specific DetNet flow, implementations MAY exclude the IPv6 Next Header field and next header information as part of DetNet flow identification.

5.1.2. Other Protocol Header Information

Implementations of this document MUST support DetNet flow identification based on header information identified in this section. Support for TCP, UDP and IPsec flows is defined. Future documents are expected to define support for other protocols.

5.1.2.1. TCP and UDP

DetNet flow identification for TCP [RFC0793] and UDP [RFC0768] is achieved based on the Source and Destination Port fields carried in each protocol's header. These fields share a common format and common DetNet flow identification procedures.

5.1.2.1.1. Source Port Field

Implementations of this document MUST support DetNet flow identification based on the Source Port field of a TCP or UDP packet. Implementations MUST support flow identification based on a particular value carried in the field, i.e., an exact value. Implementations SHOULD support range-based port matching. Implementation MUST also allow for the field to be ignored for a specific DetNet flow.

5.1.2.1.2. Destination Port Field

Implementations of this document MUST support DetNet flow identification based on the Destination Port field of a TCP or UDP packet. Implementations MUST support flow identification based on a particular value carried in the field, i.e., an exact value. Implementations SHOULD support range-based port matching. Implementation MUST also allow for the field to be ignored for a specific DetNet flow.

5.1.2.2. IPsec AH and ESP

IPsec Authentication Header (AH) [RFC4302] and Encapsulating Security Payload (ESP) [RFC4303] share a common format for the Security Parameters Index (SPI) field. Implementations MUST support flow identification based on a particular value carried in the field, i.e., an exact value. Implementation SHOULD also allow for the field to be ignored for a specific DetNet flow.

5.2. Forwarding Procedures

General requirements for IP nodes are defined in [RFC1122], [RFC1812] and [RFC6434], and are not modified by this document. The typical next-hop selection process is impacted by DetNet. Specifically, implementations of this document SHALL use management and control information to select the one or more outgoing interfaces and next hops to be used for a packet belonging to a DetNet flow.

The use of multiple paths or links, e.g., ECMP, to support a single DetNet flow is NOT RECOMMENDED. ECMP MAY be used for non-DetNet flows within a DetNet domain.

The above implies that management and control functions will be defined to support this requirement, e.g., see [I-D.ietf-detnet-yang].

5.3. DetNet IP Traffic Treatment Procedures

Implementations of this document MUST ensure that a DetNet flow receives the traffic treatment that is provisioned for it via configuration or the controller plane, e.g., via [I-D.ietf-detnet-yang]. General information on DetNet service can be found in [I-D.ietf-detnet-flow-information-model]. Typical mechanisms used to provide different treatment to different flows includes the allocation of system resources (such as queues and buffers) and provisioning of related parameters (such as shaping, and policing). Support can also be provided via an underlying network technology such as MPLS [I-D.ietf-detnet-ip-over-mpls] and IEEE802.1 TSN [I-D.ietf-detnet-ip-over-tsn]. Other than in the TSN case, the specific mechanisms used by a DetNet node to ensure DetNet service delivery requirements are met for supported DetNet flows is outside the scope of this document.

6. Management and Control Information Summary

The following summarizes the set of information that is needed to identify individual and aggregated DetNet flows:

- o IPv4 and IPv6 source address field.
- o IPv4 and IPv6 source address prefix length, where a zero (0) value effectively means that the address field is ignored.
- o IPv4 and IPv6 destination address field.
- o IPv4 and IPv6 destination address prefix length, where a zero (0) effectively means that the address field is ignored.
- o IPv4 protocol field. A limited set of values is allowed, and the ability to ignore this field, e.g., via configuration of the value zero (0), is desirable.
- o IPv6 next header field. A limited set of values is allowed, and the ability to ignore this field, e.g., via configuration of the value zero (0), is desirable.
- o IPv4 Type of Service and IPv6 Traffic Class Fields.
- o IPv4 Type of Service and IPv6 Traffic Class Field Bitmask, where a zero (0) effectively means that these fields are ignored.
- o IPv6 flow label field. This field can be optionally used for matching. When used, can be exclusive of matching against the next header field.

- o TCP and UDP Source Port. Exact and wildcard matching is required. Port ranges can optionally be used.
- o TCP and UDP Destination Port. Exact and wildcard matching is required. Port ranges can optionally be used.
- o IPsec Header SPI field. Exact matching is required.

This information MUST be provisioned per DetNet flow via configuration, e.g., via the controller or management plane.

Information identifying a DetNet flow is ordered and implementations use the first match. This can, for example, be used to provide a DetNet service for a specific UDP flow, with unique Source and Destination Port field values, while providing a different service for the aggregate of all other flows with that same UDP Destination Port value.

It is the responsibility of the DetNet controller plane to properly provision both flow identification information and the flow specific resources needed to provide the traffic treatment needed to meet each flow's service requirements. This applies for aggregated and individual flows.

7. Security Considerations

Security considerations for DetNet are described in detail in [I-D.ietf-detnet-security]. General security considerations are described in [I-D.ietf-detnet-architecture]. This section considers exclusively security considerations which are specific to the DetNet IP data plane.

Security aspects which are unique to DetNet are those whose aim is to provide the specific quality of service aspects of DetNet, which are primarily to deliver data flows with extremely low packet loss rates and bounded end-to-end delivery latency.

The primary considerations for the data plane is to maintain integrity of data and delivery of the associated DetNet service traversing the DetNet network. Application flows can be protected through whatever means is provided by the underlying technology. For example, encryption may be used, such as that provided by IPsec [RFC4301] for IP flows and/or by an underlying sub-net using MACSec [IEEE802.1AE-2018] for IP over Ethernet (Layer-2) flows.

From a data plane perspective this document does not add or modify any header information.

At the management and control level DetNet flows are identified on a per-flow basis, which may provide controller plane attackers with additional information about the data flows (when compared to controller planes that do not include per-flow identification). This is an inherent property of DetNet which has security implications that should be considered when determining if DetNet is a suitable technology for any given use case.

To provide uninterrupted availability of the DetNet service, provisions can be made against DOS attacks and delay attacks. To protect against DOS attacks, excess traffic due to malicious or malfunctioning devices can be prevented or mitigated, for example through the use of existing mechanism such as policing and shaping applied at the input of a DetNet domain. To prevent DetNet packets from being delayed by an entity external to a DetNet domain, DetNet technology definition can allow for the mitigation of Man-In-The-Middle attacks, for example through use of authentication and authorization of devices within the DetNet domain.

8. IANA Considerations

This document does not require an action from IANA.

9. Acknowledgements

The authors wish to thank Pat Thaler, Norman Finn, Loa Anderson, David Black, Rodney Cummings, Ethan Grossman, Tal Mizrahi, David Mozes, Craig Gunther, George Swallow, Yuanlong Jiang and Carlos J. Bernardos for their various contributions to this work.

10. References

10.1. Normative references

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.

- [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <<https://www.rfc-editor.org/info/rfc1812>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC7608] Boucadair, M., Petrescu, A., and F. Baker, "IPv6 Prefix Length Recommendation for Forwarding", BCP 198, RFC 7608, DOI 10.17487/RFC7608, July 2015, <<https://www.rfc-editor.org/info/rfc7608>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [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>>.

10.2. Informative references

- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-13 (work in progress), May 2019.
- [I-D.ietf-detnet-data-plane-framework]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane Framework", draft-ietf-detnet-data-plane-framework-00 (work in progress), May 2019.
- [I-D.ietf-detnet-dp-sol-mpls]
Korhonen, J. and B. Varga, "DetNet MPLS Data Plane Encapsulation", draft-ietf-detnet-dp-sol-mpls-02 (work in progress), March 2019.
- [I-D.ietf-detnet-flow-information-model]
Farkas, J., Varga, B., Cummings, R., and Y. Jiang, "DetNet Flow Information Model", draft-ietf-detnet-flow-information-model-03 (work in progress), March 2019.
- [I-D.ietf-detnet-ip-over-mpls]
Varga, B., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP over MPLS", draft-ietf-detnet-ip-over-mpls-00 (work in progress), May 2019.
- [I-D.ietf-detnet-ip-over-tsn]
Varga, B., Farkas, J., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP over IEEE 802.1 Time Sensitive Networking (TSN)", draft-ietf-detnet-ip-over-tsn-00 (work in progress), May 2019.
- [I-D.ietf-detnet-security]
Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell, J., Austad, H., Stanton, K., and N. Finn, "Deterministic Networking (DetNet) Security Considerations", draft-ietf-detnet-security-04 (work in progress), March 2019.

- [I-D.ietf-detnet-tsn-vpn-over-mpls]
Varga, B., Farkas, J., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IEEE 802.1 Time Sensitive Networking over MPLS", draft-ietf-detnet-tsn-vpn-over-mpls-00 (work in progress), May 2019.
- [I-D.ietf-detnet-yang]
Geng, X., Chen, M., Li, Z., and R. Rahman, "Deterministic Networking (DetNet) Configuration YANG Model", draft-ietf-detnet-yang-02 (work in progress), March 2019.
- [IEEE802.1AE-2018]
IEEE Standards Association, "IEEE Std 802.1AE-2018 MAC Security (MACsec)", 2018,
<<https://ieeexplore.ieee.org/document/8585421>>.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989,
<<https://www.rfc-editor.org/info/rfc1122>>.
- [RFC3290] Bernet, Y., Blake, S., Grossman, D., and A. Smith, "An Informal Management Model for Diffserv Routers", RFC 3290, DOI 10.17487/RFC3290, May 2002,
<<https://www.rfc-editor.org/info/rfc3290>>.
- [RFC3670] Moore, B., Durham, D., Strassner, J., Westerinen, A., and W. Weiss, "Information Model for Describing Network Device QoS Datapath Mechanisms", RFC 3670, DOI 10.17487/RFC3670, January 2004, <<https://www.rfc-editor.org/info/rfc3670>>.
- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., Ed., and A. Lior, "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, DOI 10.17487/RFC5777, February 2010,
<<https://www.rfc-editor.org/info/rfc5777>>.
- [RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, DOI 10.17487/RFC6434, December 2011, <<https://www.rfc-editor.org/info/rfc6434>>.
- [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>>.

[RFC7657] Black, D., Ed. and P. Jones, "Differentiated Services (Diffserv) and Real-Time Communication", RFC 7657, DOI 10.17487/RFC7657, November 2015, <<https://www.rfc-editor.org/info/rfc7657>>.

Authors' Addresses

Balazs Varga (editor)
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

Janos Farkas
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: janos.farkas@ericsson.com

Lou Berger
LabN Consulting, L.L.C.

Email: lberger@labn.net

Don Fedyk
LabN Consulting, L.L.C.

Email: dfedyk@labn.net

Andrew G. Malis
Futurewei Technologies

Email: agmalis@gmail.com

Stewart Bryant
Futurewei Technologies

Email: stewart.bryant@gmail.com

Jouni Korhonen

Email: jouni.nospam@gmail.com

DetNet
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2020

B. Varga, Ed.
J. Farkas
Ericsson
L. Berger
D. Fedyk
LabN Consulting, L.L.C.
A. Malis
S. Bryant
Futurewei Technologies
J. Korhonen
July 1, 2019

DetNet Data Plane: IP over MPLS
draft-ietf-detnet-ip-over-mpls-01

Abstract

This document specifies the Deterministic Networking data plane when operating in an IP over MPLS packet switched network.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 2, 2020.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
2. Terminology	2
2.1. Terms Used In This Document	3
2.2. Abbreviations	3
2.3. Requirements Language	3
3. DetNet IP Data Plane Overview	4
4. IP over DetNet MPLS	4
4.1. IP Over DetNet MPLS Data Plane Scenarios	5
4.2. DetNet IP over DetNet MPLS Encapsulation	6
5. IP over DetNet MPLS Procedures	8
5.1. DetNet IP over DetNet MPLS Flow Identification Procedures	8
5.2. DetNet IP over DetNet MPLS Traffic Treatment Procedures .	8
6. Management and Control Information Summary	9
7. Security Considerations	9
8. IANA Considerations	10
9. Acknowledgements	10
10. References	10
10.1. Normative references	10
10.2. Informative references	11
Authors' Addresses	12

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 the DetNet Architecture [I-D.ietf-detnet-architecture].

This document specifies use of the IP DetNet encapsulation over an MPLS network. It maps the IP data plane encapsulation described in [I-D.ietf-detnet-ip] to the DetNet MPLS data plane defined in [I-D.ietf-detnet-mpls].

2. Terminology

2.1. Terms Used In This Document

This document uses the terminology and concepts established in the DetNet architecture [I-D.ietf-detnet-architecture] and [I-D.ietf-detnet-data-plane-framework], and the reader is assumed to be familiar with these documents and their terminology.

2.2. Abbreviations

This document uses the abbreviations defined in the DetNet architecture [I-D.ietf-detnet-architecture] and [I-D.ietf-detnet-data-plane-framework]. This document uses the following abbreviations:

CE	Customer Edge equipment.
DetNet	Deterministic Networking.
DF	DetNet Flow.
DN	DetNet.
L2	Layer-2.
L3	Layer-3.
LSP	Label-switched path.
MPLS	Multiprotocol Label Switching.
PE	Provider Edge.
PREOF	Packet Replication, Ordering and Elimination Function.
PSN	Packet Switched Network.
PW	Pseudowire.
TE	Traffic Engineering.
TSN	Time-Sensitive Networking, TSN is a Task Group of the IEEE 802.1 Working Group.

2.3. Requirements Language

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

14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. DetNet IP Data Plane Overview

Figure 1 illustrates an IP DetNet, with an MPLS based DetNet network as a sub-network between the relay nodes. It shows a more complex DetNet enabled IP network where an IP flow is mapped to one or more PWs and MPLS (TE) LSPs. The end systems still originate IP encapsulated traffic that are identified as DetNet flows. The relay nodes follow procedures defined in Section 4 to map each DetNet flow to MPLS LSPs. While not shown, relay nodes can provide service sub-layer functions such as PREOF using DetNet over MPLS, and this is indicated by the solid line for the MPLS facing portion of the Service component. Note that the Transit node is MPLS (TE) LSP aware and performs switching based on MPLS labels, and need not have any specific knowledge of the DetNet service or the corresponding DetNet flow identification. See Section 4 for details on the mapping of IP flows to MPLS, and [I-D.ietf-detnet-mpls] for general support of DetNet services using MPLS.

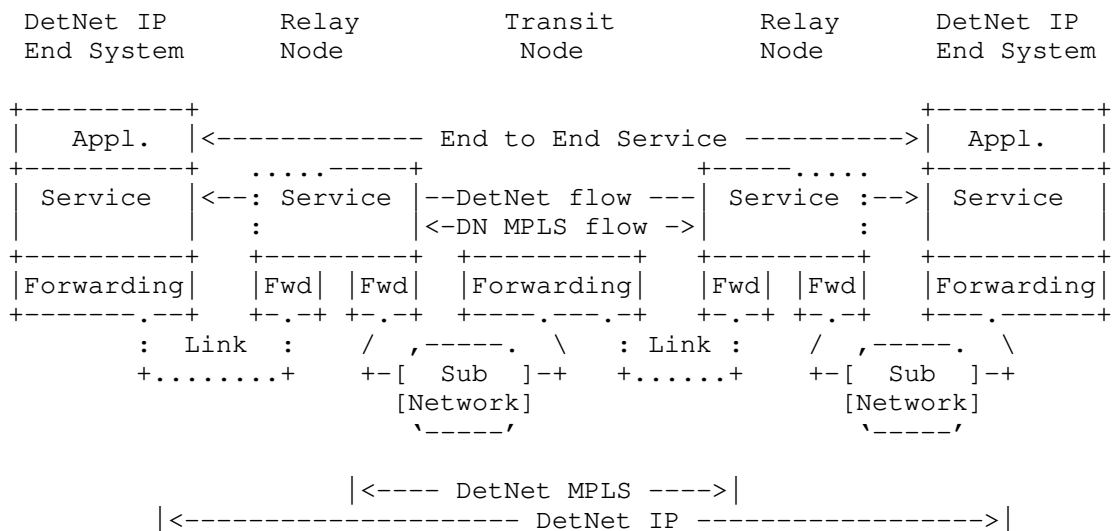


Figure 1: DetNet IP Over DetNet MPLS Network

4. IP over DetNet MPLS

This section defines how IP encapsulated flows are carried over a DetNet MPLS data plane as defined in [I-D.ietf-detnet-mpls]. Since both Non-DetNet and DetNet IP packet are identical on the wire, this

section is applicable to any node that supports IP over DetNet MPLS, and this section refers to both cases as DetNet IP over DetNet MPLS.

4.1. IP Over DetNet MPLS Data Plane Scenarios

An example use of DetNet IP over DetNet MPLS is presented here.

Figure 1 illustrated DetNet enabled End Systems (hosts), connected to DetNet (DN) enabled IP networks, operating over a DetNet aware MPLS network. iUsing this figure we can have a case where the Relay nodes act as T-PEs and sit at the boundary of the MPLS domain since the non-MPLS domain is DetNet aware. This case is very similar to the DetNet MPLS Network figure 2 in [I-D.ietf-detnet-mpls]. However in [I-D.ietf-detnet-mpls] figure 2 the T-PEs are located at the end syetem and MPLS spans the whole DetNet service. The primary difference in this document is that the Relay nodes are at the edges of the MPLS domain and therefore function as T-PEs, and that iMPLS service sub-layer functions are not provided over the DetNet IP network. The transit node functions show above are identical to those described in [I-D.ietf-detnet-mpls].

Figure 2 illustrates how relay nodes can provide service protection over an MPLS domain. In this case, CE1 and CE2 are IP DetNet end systems which are interconnected via a MPLS domain such as described in [I-D.ietf-detnet-mpls]. Note that R1 and R3 sit at the edges of an MPLS domain and therefore are similar to T-PEs, while R2 sits in the middle of the domain and is therefore similar to an S-PE.

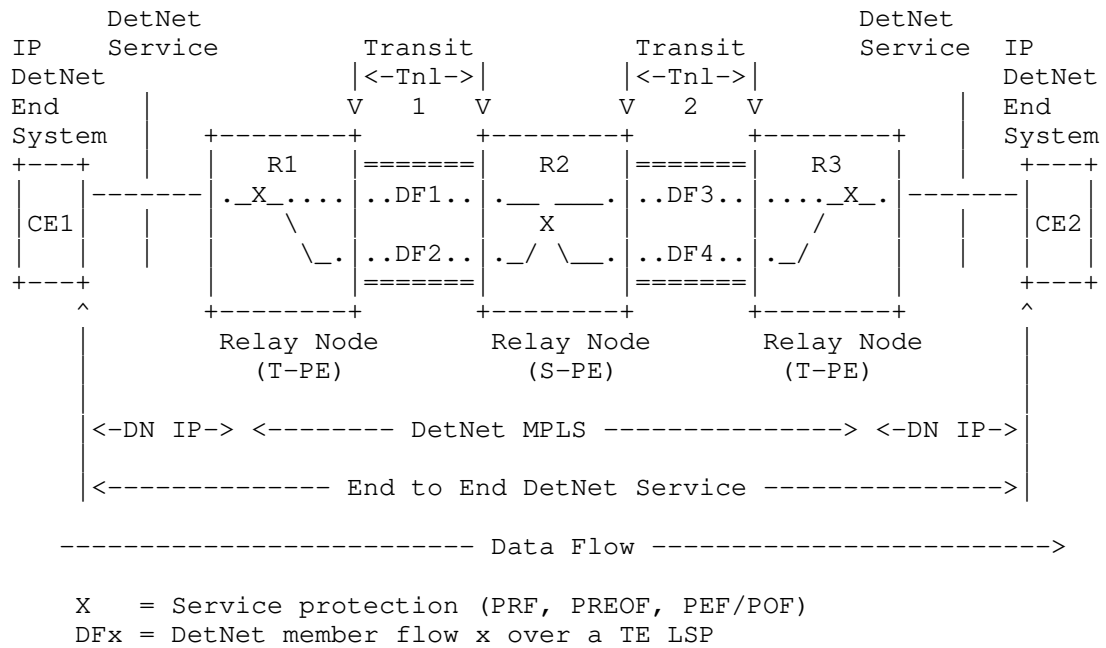


Figure 2: DetNet IP Over DetNet MPLS Network

Figure 1 illustrates DetNet enabled End Systems (hosts), connected to DetNet (DN) enabled MPLS network. A similar situation occurs when end systems are not DetNet aware. In this case, edge nodes sit at the boundary of the MPLS domain since it is also a DetNet domain boundary. The edge nodes provide DetNet service proxies for the end applications by initiating and terminating DetNet service for the application's IP flows. While the node types differ, there is essentially no difference in data plane processing between relay and edges. There are likely to be differences in controller plane operation, particularly when distributed control plane protocols are used.

It is still possible to provide DetNet service protection for non-DetNet aware end systems. The case is basically the same as Figure 2, with the exception that CE1 and CE2 are non-DetNet aware end systems and R1 and R3 become edge nodes.

4.2. DetNet IP over DetNet MPLS Encapsulation

The basic encapsulation approach is to treat a DetNet IP flow as an app-flow from the DetNet MPLS perspective. The corresponding example DetNet Sub-Network format is shown in Figure 3.

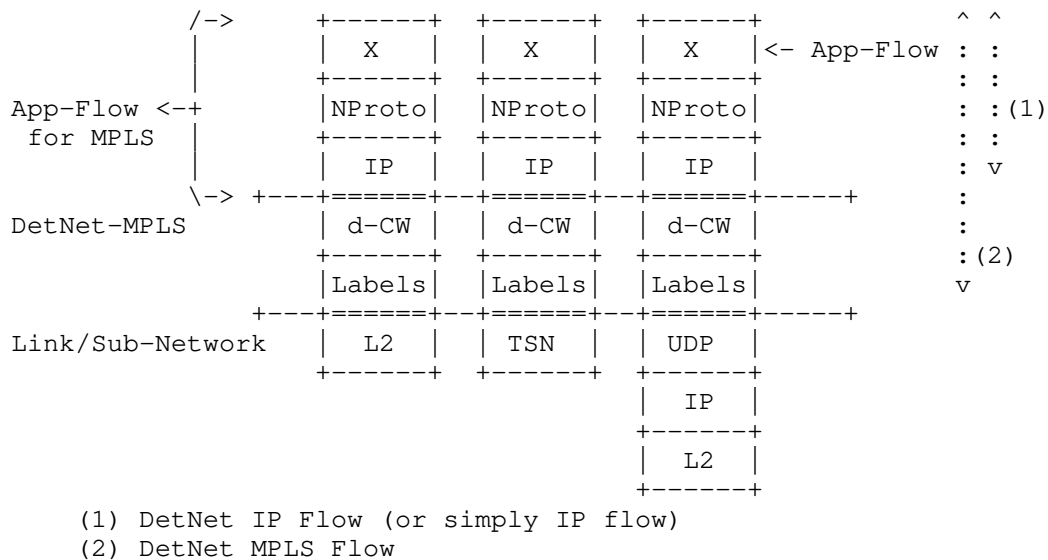


Figure 3: Example DetNet IP over MPLS Sub-Network Formats

In the figure, "App-Flow" indicates the payload carried by the DetNet IP data plane. "IP" and "NProto" indicate the fields described in Section 7.1.1. IP Header Information and Section 7.1.2. Other Protocol Header Information in [I-D.ietf-detnet-ip], respectively. "MPLS App-Flow" indicates that an individual DetNet IP flow is the payload from the perspective of the DetNet MPLS data plane defined in [I-D.ietf-detnet-mpls].

Per [I-D.ietf-detnet-mpls], the DetNet MPLS data plane uses a single S-Label to support a single app flow. Section 7.1. DetNet IP Flow Identification Procedures in [I-D.ietf-detnet-ip] states that a single DetNet flow is identified based on IP, and next level protocol, header information. Section 7.4. Aggregation Considerations in [I-D.ietf-detnet-ip] defines that aggregation is supported through the use of prefixes, wildcards, bitmasks, and port ranges. Collectively, this results in the fairly straight forward procedures defined in this section.

As shown in Figure 2, DetNet relay nodes are responsible for the mapping of a DetNet flow, at the service sub-layer, from the IP to MPLS DetNet data planes and back again. Their related DetNet IP over DetNet MPLS data plane operation is comprised of two sets of procedures: the mapping of flow identifiers; and ensuring proper traffic treatment.

Mapping of IP to the MPLS Detnet is similar for IP Detnet flows and IP flows. The six-tuple of IP is mapped to the S-Label in both cases. The various fields may be mapped or ignored when going from IP to MPLS.

5. IP over DetNet MPLS Procedures

5.1. DetNet IP over DetNet MPLS Flow Identification Procedures

A DetNet relay node (ingress T-PE) that sends a DetNet IP flow over a DetNet MPLS network MUST map a DetNet IP flow, as identified in [I-D.ietf-detnet-ip] into a single MPLS DetNet flow and MUST process it in accordance to the procedures defined in [I-D.ietf-detnet-mpls] Section 6.1. PRF MAY be supported at the MPLS level for DetNet IP flows sent over an DetNet MPLS network. Aggregation MAY be supported as defined in [I-D.ietf-detnet-mpls] Section 5.4. Aggregation considerations in [I-D.ietf-detnet-ip] MAY be used to identify an individual DetNet IP flow. The provisioning of the mapping of DetNet IP flows to DetNet MPLS flows MUST be supported via configuration, e.g., via the controller plane.

A DetNet relay node (egress T-PE) MAY be provisioned to handle packets received via the DetNet MPLS data plane as DetNet IP flows. A single incoming DetNet MPLS flow MAY be treated as a single DetNet IP flow, without examination of IP headers. Alternatively, packets received via the DetNet MPLS data plane MAY follow the normal DetNet IP flow identification procedures defined in [I-D.ietf-detnet-ip] Section 7.1.

An implementation MUST support the provisioning for handling any received DetNet MPLS data plane as DetNet IP flows via configuration. Note that such configuration MAY include support from PEOF on the incoming DetNet MPLS flow.

5.2. DetNet IP over DetNet MPLS Traffic Treatment Procedures

The traffic treatment required for a particular DetNet IP flow is provisioned via configuration or the controller plane. When an DetNet IP flow is sent over DetNet MPLS, a DetNet relay node MUST ensure that the provisioned DetNet IP traffic treatment is provided at the forwarding sub-layer as described in [I-D.ietf-detnet-mpls] Section 5.2. Note that the PRF function MAY be utilized when sending IP over MPLS.

Traffic treatment for DetNet IP flows received over the DetNet MPLS data plane MUST follow Section 7.3 DetNet IP Traffic Treatment Procedures in [I-D.ietf-detnet-ip].

6. Management and Control Information Summary

The following summarizes the set of information that is needed to support DetNet IP over DetNet MPLS at the MPLS ingress node:

- o Each MPLS App-Flow is identified using the IP flow identification information as defined in [I-D.ietf-detnet-ip]. The information is summarized in Section 6 of that document, and includes all wildcards, port ranges and ability to ignore specific IP fields.
- o The DetNet MPLS service that is to be used to send the matching IP traffic. Logically this is a pointer to the information provided in [I-D.ietf-detnet-mpls] Section 5.1, and includes both service and traffic delivery information.

The following summarizes the set of information that is needed to support DetNet IP over DetNet MPLS at the MPLS egress node:

- o S-Label values that are carrying MPLS over IP encapsulated traffic.
- o For each S-Label, how the received traffic is to be handled. The traffic may be processed according as any other DetNet IP traffic as defined in this document or in [I-D.ietf-detnet-ip], or the traffic may be directly treated as an MPLS App-flow for additional processing according to [I-D.ietf-detnet-mpls].

It is the responsibility of the DetNet controller plane to properly provision both flow identification information and the flow specific resources needed to provide the traffic treatment needed to meet each flow's service requirements. This applies for aggregated and individual flows.

7. Security Considerations

This draft does not have additional security considerations. Security considerations for DetNet are described in detail in [I-D.ietf-detnet-security]. General security considerations are described in [I-D.ietf-detnet-architecture]. MPLS and IP specific considerations are described in [I-D.ietf-detnet-mpls] and [I-D.ietf-detnet-ip].

Security aspects which are unique to DetNet are those whose aim is to provide the specific quality of service aspects of DetNet, which are primarily to deliver data flows with extremely low packet loss rates and bounded end-to-end delivery latency.

The primary considerations for the data plane is to maintain integrity of data and delivery of the associated DetNet service traversing the DetNet network. Application flows can be protected through whatever means is provided by the underlying technology. For example, encryption may be used, such as that provided by IPSec [RFC4301] for IP flows and/or by an underlying sub-net using MACSec [IEEE802.1AE-2018] for IP over Ethernet (Layer-2) flows.

From a data plane perspective this document does not add or modify any header information.

At the management and control level DetNet flows are identified on a per-flow basis, which may provide controller plane attackers with additional information about the data flows (when compared to controller planes that do not include per-flow identification). This is an inherent property of DetNet which has security implications that should be considered when determining if DetNet is a suitable technology for any given use case.

To provide uninterrupted availability of the DetNet service, provisions can be made against DOS attacks and delay attacks. To protect against DOS attacks, excess traffic due to malicious or malfunctioning devices can be prevented or mitigated, for example through the use of existing mechanism such as policing and shaping applied at the input of a DetNet domain. To prevent DetNet packets from being delayed by an entity external to a DetNet domain, DetNet technology definition can allow for the mitigation of Man-In-The-Middle attacks, for example through use of authentication and authorization of devices within the DetNet domain.

8. IANA Considerations

This document makes no IANA requests.

9. Acknowledgements

The authors wish to thank Pat Thaler, Norman Finn, Loa Anderson, David Black, Rodney Cummings, Ethan Grossman, Tal Mizrahi, David Mozes, Craig Gunther, George Swallow, Yuanlong Jiang and Carlos J. Bernardos for their various contributions to this work.

10. References

10.1. Normative references

- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", draft-ietf-
detnet-architecture-13 (work in progress), May 2019.
- [I-D.ietf-detnet-ip]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane: IP",
draft-ietf-detnet-ip-00 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS",
draft-ietf-detnet-mpls-00 (work in progress), May 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

10.2. Informative references

- [I-D.ietf-detnet-data-plane-framework]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane
Framework", draft-ietf-detnet-data-plane-framework-00
(work in progress), May 2019.
- [I-D.ietf-detnet-security]
Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell,
J., Austad, H., Stanton, K., and N. Finn, "Deterministic
Networking (DetNet) Security Considerations", draft-ietf-
detnet-security-04 (work in progress), March 2019.
- [IEEE802.1AE-2018]
IEEE Standards Association, "IEEE Std 802.1AE-2018 MAC
Security (MACsec)", 2018,
<<https://ieeexplore.ieee.org/document/8585421>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the
Internet Protocol", RFC 4301, DOI 10.17487/RFC4301,
December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.

Authors' Addresses

Balazs Varga (editor)
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

Janos Farkas
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: janos.farkas@ericsson.com

Lou Berger
LabN Consulting, L.L.C.

Email: lberger@labn.net

Don Fedyk
LabN Consulting, L.L.C.

Email: dfedyk@labn.net

Andrew G. Malis
Futurewei Technologies

Email: agmalis@gmail.com

Stewart Bryant
Futurewei Technologies

Email: stewart.bryant@gmail.com

Jouni Korhonen

Email: jouni.nospam@gmail.com

DetNet
Internet-Draft
Intended status: Standards Track
Expires: November 6, 2019

B. Varga, Ed.
J. Farkas
Ericsson
A. Malis
S. Bryant
Huawei Technologies
J. Korhonen
May 5, 2019

DetNet Data Plane: IP over IEEE 802.1 Time Sensitive Networking (TSN)
draft-ietf-detnet-ip-over-tsn-00

Abstract

This document specifies the Deterministic Networking IP data plane when operating over a TSN network.

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 November 6, 2019.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Terms Used In This Document	3
2.2. Abbreviations	3
3. Requirements Language	4
4. DetNet IP Data Plane Overview	4
5. DetNet IP Data Plane Considerations	7
5.1. DetNet Routers	8
5.2. Networks With Multiple Technology Segments	9
6. Mapping DetNet IP Flows to IEEE 802.1 TSN	10
6.1. TSN Stream ID Mapping	11
6.2. TSN Usage of FRER	13
6.3. Procedures	14
7. Management and Control Implications	14
8. Security Considerations	16
9. IANA Considerations	16
10. Acknowledgements	16
11. References	16
11.1. Normative references	16
11.2. Informative references	18
Authors' Addresses	21

1. Introduction

[Editor's note: Introduction to be made specific to DetNet IP over TSN scenario. May be similar to intro of DetNet MPLS over TSN.].

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 the DetNet Architecture [I-D.ietf-detnet-architecture].

This document specifies the DetNet data plane operation for IP hosts and routers that provide DetNet service to IP encapsulated data. No DetNet specific encapsulation is defined to support IP flows, rather existing IP and higher layer protocol header information is used to support flow identification and DetNet service delivery.

The DetNet Architecture decomposes the DetNet related data plane functions into two sub-layers: a service sub-layer and a forwarding sub-layer. The service sub-layer is used to provide DetNet service protection and reordering. The forwarding sub-layer is used to

provides congestion protection (low loss, assured latency, and limited reordering). As no DetNet specific headers are added to support DetNet IP flows, only the forwarding sub-layer functions are supported using the DetNet IP defined by this document. Service protection can be provided on a per sub-net basis using technologies such as MPLS [I-D.ietf-detnet-dp-sol-mpls] and IEEE802.1 TSN.

2. Terminology

[Editor's note: Needs clean up.].

2.1. Terms Used In This Document

This document uses the terminology and concepts established in the DetNet architecture [I-D.ietf-detnet-architecture], and the reader is assumed to be familiar with that document and its terminology.

2.2. Abbreviations

The following abbreviations used in this document:

CE	Customer Edge equipment.
CoS	Class of Service.
DetNet	Deterministic Networking.
DF	DetNet Flow.
L2	Layer-2.
L3	Layer-3.
LSP	Label-switched path.
MPLS	Multiprotocol Label Switching.
OAM	Operations, Administration, and Maintenance.
PE	Provider Edge.
PREOF	Packet Replication, Ordering and Elimination Function.
PSN	Packet Switched Network.
PW	Pseudowire.
QoS	Quality of Service.

TE Traffic Engineering.

TSN Time-Sensitive Networking, TSN is a Task Group of the
IEEE 802.1 Working Group.

3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. DetNet IP Data Plane Overview

[Editor's note: simplify this section and highlight DetNet IP over subnets scenario being the focus in the remaining part of the document.].

This document describes how IP is used by DetNet nodes, i.e., hosts and routers, to identify DetNet flows and provide a DetNet service. From a data plane perspective, an end-to-end IP model is followed. As mentioned above, existing IP and higher layer protocol header information is used to support flow identification and DetNet service delivery.

DetNet uses "6-tuple" based flow identification, where "6-tuple" refers to information carried in IP and higher layer protocol headers. General background on the use of IP headers, and "5-tuples", to identify flows and support Quality of Service (QoS) can be found in [RFC3670]. [RFC7657] also provides useful background on the delivery differentiated services (DiffServ) and "6-tuple" based flow identification.

DetNet flow aggregation may be enabled via the use of wildcards, masks, prefixes and ranges. IP tunnels may also be used to support flow aggregation. In these cases, it is expected that DetNet aware intermediate nodes will provide DetNet service assurance on the aggregate through resource allocation and congestion control mechanisms.

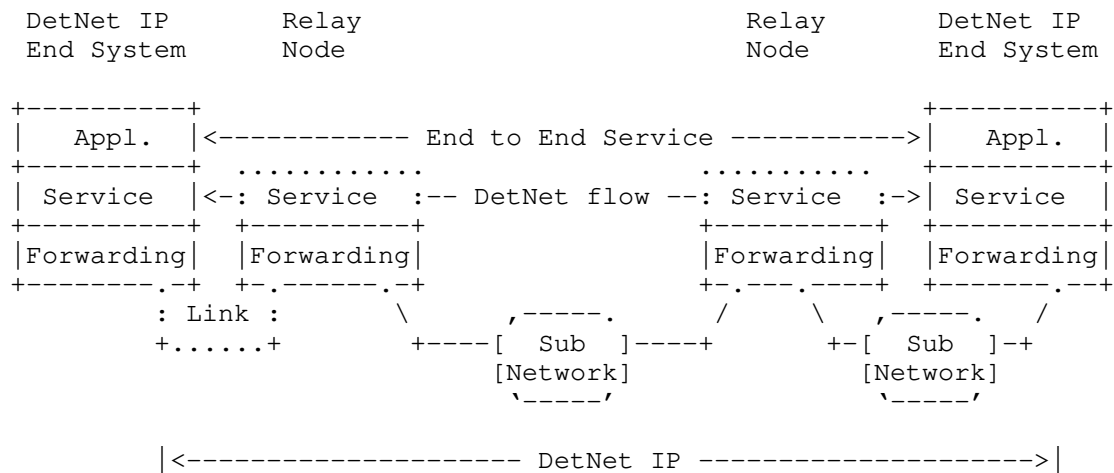


Figure 1: A Simple DetNet (DN) Enabled IP Network

Figure 1 illustrates a DetNet enabled IP network. The DetNet enabled end systems originate IP encapsulated traffic that is identified as DetNet flows, relay nodes understand the forwarding requirements of the DetNet flow and ensure that node, interface and sub-network resources are allocated to ensure DetNet service requirements. The dotted line around the Service component of the Relay Nodes indicates that the transit routers are DetNet service aware but do not perform any DetNet service sub-layer function, e.g., PREOF. IEEE 802.1 TSN is an example sub-network type which can provide support for DetNet flows and service. The mapping of DetNet IP flows to TSN streams and TSN protection mechanisms is covered in Section 6.

Note: The sub-network can represent a TSN, MPLS or IP network segment.

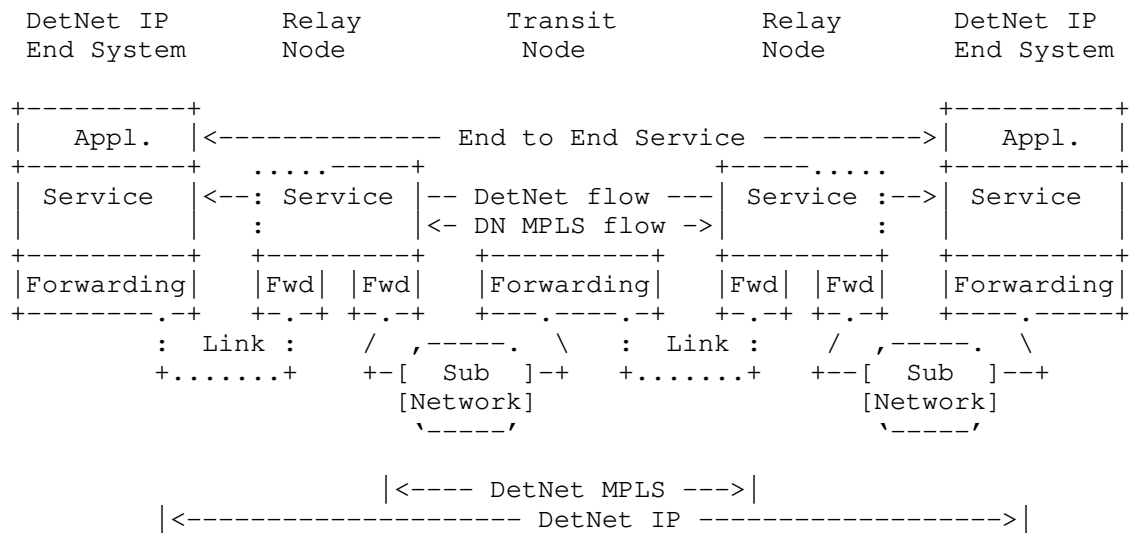


Figure 2: DetNet IP Over DetNet MPLS Network

Figure 2 illustrates a variant of Figure 1, with an MPLS based DetNet network as a sub-network between the relay nodes. It shows a more complex DetNet enabled IP network where an IP flow is mapped to one or more PWs and MPLS (TE) LSPs. The end systems still originate IP encapsulated traffic that is identified as DetNet flows. The relay nodes follow procedures defined in RRR to map each DetNet flow to MPLS LSPs. While not shown, relay nodes can provide service sub-layer functions such as PREOF using DetNet over MPLS, and this is indicated by the solid line for the MPLS facing portion of the Service component. Note that the Transit node is MPLS (TE) LSP aware and performs switching based on MPLS labels, and need not have any specific knowledge of the DetNet service or the corresponding DetNet flow identification. See RRR for details on the mapping of IP flows to MPLS, and [I-D.ietf-detnet-dp-sol-mpls] for general support of DetNet services using MPLS.

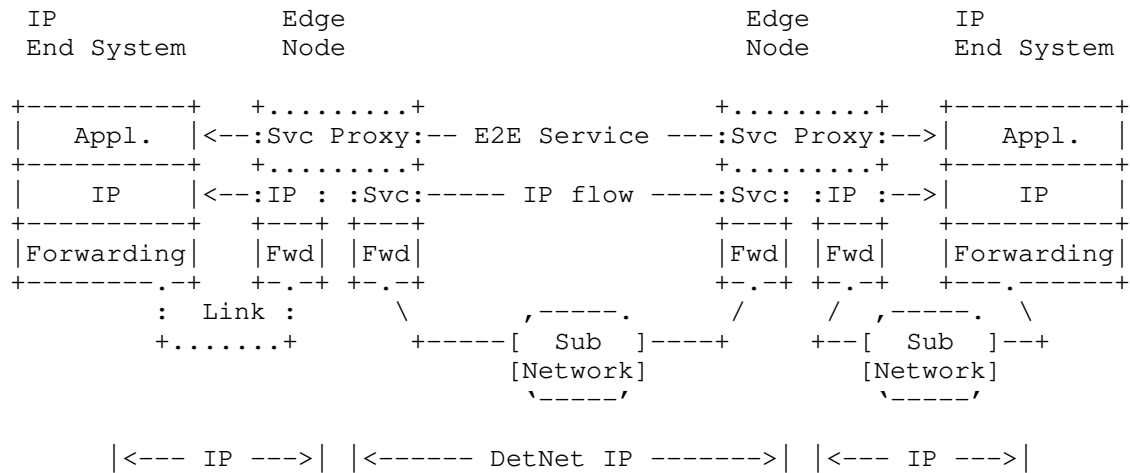


Figure 3: Non-DetNet aware IP end systems with DetNet IP Domain

Figure 3 illustrates another variant of Figure 1 where the end systems are not DetNet aware. In this case, edge nodes sit at the boundary of the DetNet domain and provide DetNet service proxies for the end applications by initiating and terminating DetNet service for the application's IP flows. The existing header information or an approach used for aggregation can be used to support DetNet flow identification.

Non-DetNet and DetNet IP packets are identical on the wire. From data plane perspective, the only difference is that there is flow-associated DetNet information on each DetNet node that defines the flow related characteristics and required forwarding behavior. As shown above, edge nodes provide a Service Proxy function that "associates" one or more IP flows with the appropriate DetNet flow-specific information and ensures that the receives the proper traffic treatment within the domain.

Note: The operation of IEEE802.1 TSN end systems over DetNet enabled IP networks is not described in this document. While TSN flows could be encapsulated in IP packets by an IP End System or DetNet Edge Node in order to produce DetNet IP flows, the details of such are out of scope of this document.

5. DetNet IP Data Plane Considerations

[Editor's note: Sort out what data plane considerations are relevant for sub-net scenarios.]

5.1. DetNet Routers

Within a DetNet domain, the DetNet enabled IP Routers interconnect links and sub-networks to support end-to-end delivery of DetNet flows. From a DetNet architecture perspective, these routers are DetNet relays, as they must be DetNet service aware. Such routers identify DetNet flows based on the IP 6-tuple, and ensure that the DetNet service required traffic treatment is provided both on the node and on any attached sub-network.

This solution provides DetNet functions end to end, but does so on a per link and sub-network basis. Congestion protection and latency control and the resource allocation (queuing, policing, shaping) are supported using the underlying link / sub net specific mechanisms. However, service protections (packet replication and packet elimination functions) are not provided at the DetNet layer end to end. But such service protection can be provided on a per underlying L2 link and sub-network basis.

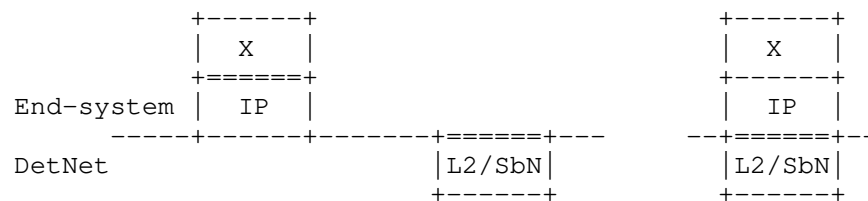


Figure 4: Encapsulation of DetNet Routing in simplified IP service L3 end-systems

The DetNet Service Flow is mapped to the link / sub-network specific resources using an underlying system specific means. This implies each DetNet aware node on path looks into the forwarded DetNet Service Flow packet and utilize e.g., a 5- (or 6-) tuple to find out the required mapping within a node.

As noted earlier, the Service Protection is done within each link / sub-network independently using the domain specific mechanisms (due the lack of a unified end to end sequencing information that would be available for intermediate nodes). Therefore, service protection (if any) cannot be provided end-to-end, only within sub-networks. This is shown for a three sub-network scenario in Figure 5, where each sub-network can provide service protection between its borders.

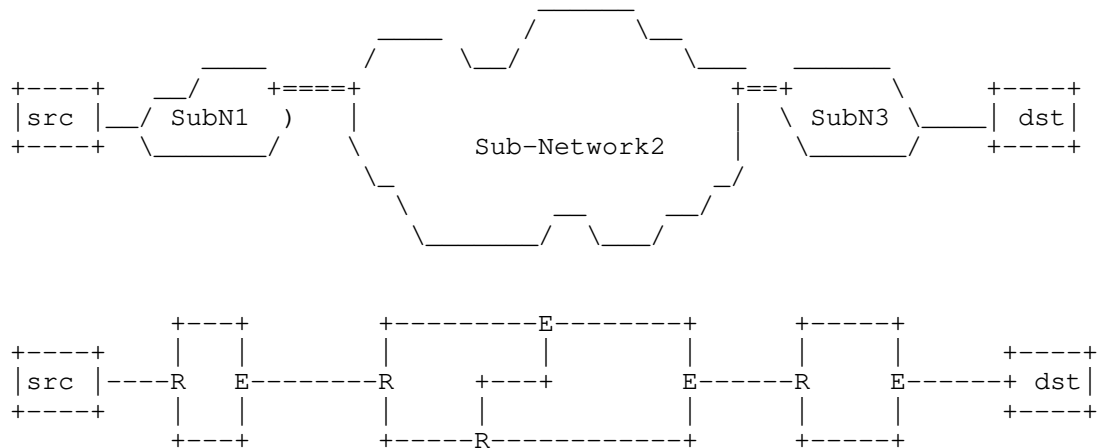


Figure 5: Replication and elimination in sub-networks for DetNet IP networks

If end to end service protection is desired that can be implemented, for example, by the DetNet end systems using Layer-4 (L4) transport protocols or application protocols. However, these are out of scope of this document.

5.2. Networks With Multiple Technology Segments

There are network scenarios, where the DetNet domain contains multiple technology segments (IEEE 802.1 TSN, MPLS) and all those segments are under the same administrative control (see Figure 6). Furthermore, DetNet nodes may be interconnected via TSN segments.

DetNet routers ensure that detnet service requirements are met per hop by allocating local resources, both receive and transmit, and by mapping the service requirements of each flow to appropriate sub-network mechanisms. Such mapping is sub-network technology specific. The mapping of DetNet IP Flows to MPLS is covered RRR . The mapping of IP DetNet Flows to IEEE 802.1 TSN is covered in Section 6.

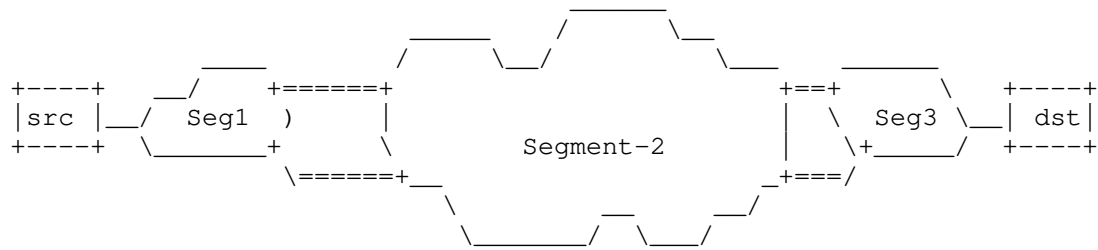


Figure 6: DetNet domains and multiple technology segments

6. Mapping DetNet IP Flows to IEEE 802.1 TSN

[Authors note: how do we handle control protocols such as ICMP, IPsec, etc.]

This section covers how DetNet IP flows operate over an IEEE 802.1 TSN sub-network. Figure 7 illustrates such a scenario, where two IP (DetNet) nodes are interconnected by a TSN sub-network. Node-1 is single homed and Node-2 is dual-homed. IP nodes can be (1) DetNet IP End System, (2) DetNet IP Edge or Relay node or (3) IP End System.

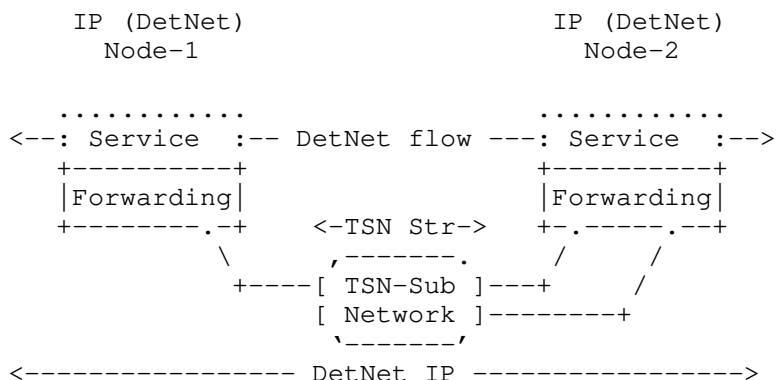


Figure 7: DetNet (DN) Enabled IP Network over a TSN sub-network

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. Furthermore IEEE 802.1CB [IEEE8021CB] defines frame replication and elimination functions for reliability that should prove both compatible with and useful to, DetNet networks. All these functions have to identify flows those require TSN treatment.

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 may need additional marking, such as the priority field of an IEEE Std 802.1Q VLAN tag, to give the packet proper service.

TSN capabilities of the TSN sub-network are made available for IP (DetNet) flows via the protocol interworking function defined in IEEE 802.1CB [IEEE8021CB]. For example, applied on the TSN edge port connected to the IP (DetNet) node it can convert an ingress unicast IP (DetNet) flow to use a specific multicast destination MAC address and VLAN, in order to direct the packet through a specific path inside the bridged network. A similar interworking pair at the other end of the TSN sub-network would restore the packet to its original destination MAC address and VLAN.

Placement of TSN functions depends on the TSN capabilities of nodes. IP (DetNet) Nodes may or may not support TSN functions. For a given TSN Stream (i.e., DetNet flow) an IP (DetNet) node is treated as a Talker or a Listener inside the TSN sub-network.

6.1. TSN Stream ID Mapping

DetNet IP Flow and TSN Stream mapping is based on the active Stream Identification function, that operates at the frame level. IEEE 802.1CB [IEEE8021CB] defines an Active Destination MAC and VLAN Stream identification function, what can replace some Ethernet header fields namely (1) the destination MAC-address, (2) the VLAN-ID and (3) priority parameters with alternate values. Replacement is provided for the frame passed down the stack from the upper layers or up the stack from the lower layers.

Active Destination MAC and VLAN Stream identification can be used within a Talker to set flow identity or a Listener to recover the original addressing information. It can be used also in a TSN bridge that is providing translation as a proxy service for an End System. As a result IP (DetNet) flows can be mapped to use a particular {MAC-address, VLAN} pair to match the Stream in the TSN sub-network.

From the TSN sub-network perspective DetNet IP nodes without any TSN functions can be treated as TSN-unaware Talker or Listener. In such cases relay nodes in the TSN sub-network MUST modify the Ethernet encapsulation of the DetNet IP flow (e.g., MAC translation, VLAN-ID setting, Sequence number addition, etc.) to allow proper TSN specific handling of the flow inside the sub-network. This is illustrated in Figure 8.

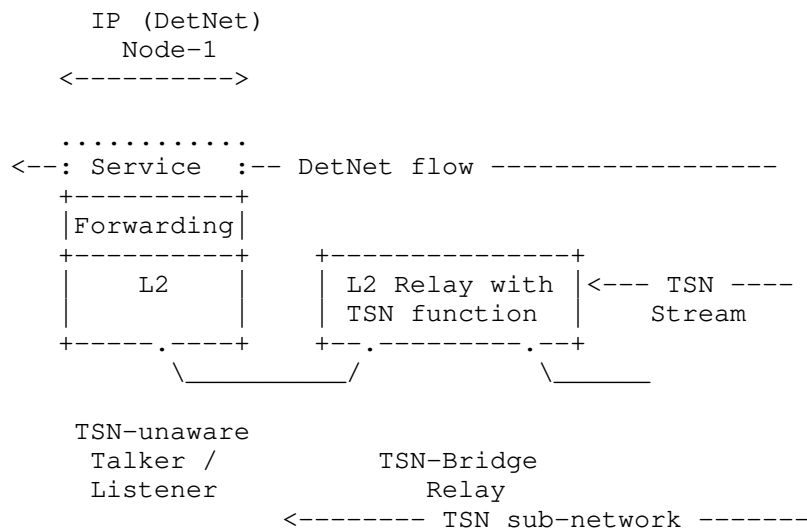


Figure 8: IP (DetNet) node without TSN functions

IP (DetNet) nodes being TSN-aware can be treated as a combination of a TSN-unaware Talker/Listener and a TSN-Relay, as shown in Figure 9. In such cases the IP (DetNet) node MUST provide the TSN sub-network specific Ethernet encapsulation over the link(s) towards the sub-network. An TSN-aware IP (DetNet) node MUST support the following TSN components:

1. For recognizing flows:
 - * Stream Identification
2. For FRER used inside the TSN domain, additionally:
 - * Sequencing function
 - * Sequence encode/decode function
3. For FRER when the node is a replication or elimination point, additionally:
 - * Stream splitting function
 - * Individual recovery function

[Editor's note: Should we added here requirements regarding IEEE 802.1Q C-VLAN component?]

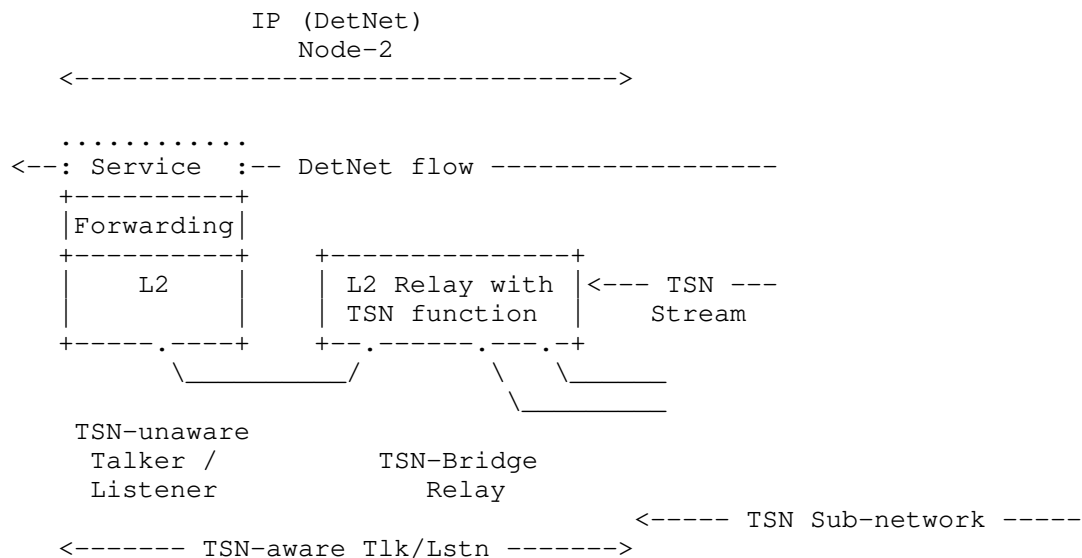


Figure 9: IP (DetNet) node with TSN functions

A Stream identification component MUST be able to instantiate the following functions (1) Active Destination MAC and VLAN Stream identification function, (2) IP Stream identification function and (3) the related managed objects in Clause 9 of IEEE 802.1CB [IEEE8021CB]. IP Stream identification function provides a 6-tuple match.

The Sequence encode/decode function MUST support the Redundancy tag (R-TAG) format as per Clause 7.8 of IEEE 802.1CB [IEEE8021CB].

6.2. TSN Usage of FRER

TSN Streams supporting DetNet flows may use Frame Replication and Elimination for Redundancy (FRER) [802.1CB] based on the loss service requirements of the TSN Stream, which is derived from the DetNet service requirements of the DetNet mapped flow. The specific operation of FRER is not modified by the use of DetNet and follows IEEE 802.1CB [IEEE8021CB].

FRER function and the provided service recovery is available only within the TSN sub-network (as shown in Figure 5) as the Stream-ID and the TSN sequence number are not valid outside the sub-network. An IP (DetNet) node represents a L3 border and as such it terminates all related information elements encoded in the L2 frames.

6.3. Procedures

[Editor's note: This section is TBD - covers required behavior of a TSN-aware DetNet node using a TSN underlay.]

This section provides DetNet IP data plane procedures to interwork with a TSN underlay sub-network when the IP (DetNet) node acts as a TSN-aware Talker or Listener (see Figure 9). These procedures have been divided into the following areas: flow identification, mapping of a DetNet flow to a TSN Stream and ensure proper TSN encapsulation.

Flow identification procedures are described in RRR . A TSN-aware IP (DetNet) node SHALL support the Stream Identification TSN components as per IEEE 802.1CB [IEEE8021CB].

Implementations of this document SHALL use management and control information to map a DetNet flow to a TSN Stream. N:1 mapping (aggregating DetNet flows in a single TSN Stream) SHALL be supported. The management or control function that provisions flow mapping SHALL ensure that adequate resources are allocated and configured to provide proper service requirements of the mapped flows.

For proper TSN encapsulation implementations of this document SHALL support active Stream Identification function as defined in chapter 6.6 in IEEE 802.1CB [IEEE8021CB].

A TSN-aware IP (DetNet) node SHALL support Ethernet encapsulation with Redundancy tag (R-TAG) as per chapter 7.8 in IEEE 802.1CB [IEEE8021CB].

Depending whether FRER functions are used in the TSN sub-network to serve the mapped TSN Stream, a TSN-aware IP (DetNet) node SHALL support Sequencing function and Sequence encode/decode function as per chapter 7.4 and 7.6 in IEEE 802.1CB [IEEE8021CB]. Furthermore when a TSN-aware IP (DetNet) node acting as a replication or elimination point for FRER it SHALL implement the Stream splitting function and the Individual recovery function as per chapter 7.7 and 7.5 in IEEE 802.1CB [IEEE8021CB].

7. Management and Control Implications

[Editor's note: This section is TBD Covers Creation, mapping, removal of TSN Stream IDs, related parameters and, when needed, configuration of FRER. Supported by management/control plane.]

DetNet flow and TSN Stream mapping related information are required only for TSN-aware IP (DetNet) nodes. From the Data Plane

perspective there is no practical difference based on the origin of flow mapping related information (management plane or control plane).

TSN-aware DetNet IP nodes are member of both the DetNet domain and the TSN sub-network. Within the TSN sub-network the TSN-aware IP (DetNet) node has a TSM-aware Talker/Listener role, so TSN specific management and control plane functionalities must be implemented. There are many similarities in the management plane techniques used in DetNet and TSN, but that is not the case for the control plane protocols. For example, RSVP-TE and MSRP behaves differently. Therefore management and control plane design is an important aspect of scenarios, where mapping between DetNet and TSN is required.

In order to use a TSN sub-network between DetNet nodes, DetNet specific information must be converted to TSN sub-network specific ones. DetNet flow ID and flow related parameters/requirements must be converted to a TSN Stream ID and stream related parameters/requirements. Note that, as the TSN sub-network is just a portion of the end2end DetNet path (i.e., single hop from IP perspective), some parameters (e.g., delay) may differ significantly. Other parameters (like bandwidth) also may have to be tuned due to the L2 encapsulation used in the TSN sub-network.

In some case it may be challenging to determine some TSN Stream related information. For example on a TSN-aware IP (DetNet) node that acts as a Talker, it is quite obvious which DetNet node is the Listener of the mapped TSN stream (i.e., the IP Next-Hop). However it may be not trivial to locate the point/interface where that Listener is connected to the TSN sub-network. Such attributes may require interaction between control and management plane functions and between DetNet and TSN domains.

Mapping between DetNet flow identifiers and TSN Stream identifiers, if not provided explicitly, can be done by a TSN-aware IP (DetNet) node locally based on information provided for configuration of the TSN Stream identification functions (IP Stream identification and active Stream identification function).

Triggering the setup/modification of a TSN Stream in the TSN sub-network is an example where management and/or control plane interactions are required between the DetNet and TSN sub-network. TSN-unaware IP (DetNet) nodes make such a triggering even more complicated as they are fully unaware of the sub-network and run independently.

Configuration of TSN specific functions (e.g., FRER) inside the TSN sub-network is a TSN specific decision and may not be visible in the DetNet domain.

8. Security Considerations

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

9. IANA Considerations

TBD.

10. Acknowledgements

Thanks for Norman Finn and Lou Berger for their comments and contributions.

11. References

11.1. Normative references

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <<https://www.rfc-editor.org/info/rfc1812>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [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>>.
- [RFC7608] Boucadair, M., Petrescu, A., and F. Baker, "IPv6 Prefix Length Recommendation for Forwarding", BCP 198, RFC 7608, DOI 10.17487/RFC7608, July 2015, <<https://www.rfc-editor.org/info/rfc7608>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.

11.2. Informative references

- [G.8275.1] International Telecommunication Union, "Precision time protocol telecom profile for phase/time synchronization with full timing support from the network", ITU-T G.8275.1/Y.1369.1 G.8275.1, June 2016, <<https://www.itu.int/rec/T-REC-G.8275.1/en>>.
- [G.8275.2] International Telecommunication Union, "Precision time protocol telecom profile for phase/time synchronization with partial timing support from the network", ITU-T G.8275.2/Y.1369.2 G.8275.2, June 2016, <<https://www.itu.int/rec/T-REC-G.8275.2/en>>.
- [I-D.ietf-detnet-architecture] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-12 (work in progress), March 2019.
- [I-D.ietf-detnet-dp-sol-mpls] Korhonen, J. and B. Varga, "DetNet MPLS Data Plane Encapsulation", draft-ietf-detnet-dp-sol-mpls-02 (work in progress), March 2019.
- [I-D.ietf-detnet-flow-information-model] Farkas, J., Varga, B., Cummings, R., and Y. Jiang, "DetNet Flow Information Model", draft-ietf-detnet-flow-information-model-03 (work in progress), March 2019.

- [I-D.ietf-detnet-security]
Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell, J., Austad, H., Stanton, K., and N. Finn, "Deterministic Networking (DetNet) Security Considerations", draft-ietf-detnet-security-04 (work in progress), March 2019.
- [I-D.ietf-teas-pce-native-ip]
Wang, A., Zhao, Q., Khasanov, B., Chen, H., and R. Mallya, "PCE in Native IP Network", draft-ietf-teas-pce-native-ip-03 (work in progress), April 2019.
- [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/>>.
- [RFC1122]
Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [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>>.
- [RFC2386]
Crawley, E., Nair, R., Rajagopalan, B., and H. Sandick, "A Framework for QoS-based Routing in the Internet", RFC 2386, DOI 10.17487/RFC2386, August 1998, <<https://www.rfc-editor.org/info/rfc2386>>.
- [RFC3670]
Moore, B., Durham, D., Strassner, J., Westerinen, A., and W. Weiss, "Information Model for Describing Network Device QoS Datapath Mechanisms", RFC 3670, DOI 10.17487/RFC3670, January 2004, <<https://www.rfc-editor.org/info/rfc3670>>.

- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<https://www.rfc-editor.org/info/rfc5575>>.
- [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>>.
- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., Ed., and A. Lior, "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, DOI 10.17487/RFC5777, February 2010, <<https://www.rfc-editor.org/info/rfc5777>>.
- [RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, DOI 10.17487/RFC6434, December 2011, <<https://www.rfc-editor.org/info/rfc6434>>.
- [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>>.
- [RFC7657] Black, D., Ed. and P. Jones, "Differentiated Services (Diffserv) and Real-Time Communication", RFC 7657, DOI 10.17487/RFC7657, November 2015, <<https://www.rfc-editor.org/info/rfc7657>>.
- [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>>.
- [RFC8169] Mirsky, G., Ruffini, S., Gray, E., Drake, J., Bryant, S., and A. Vainshtein, "Residence Time Measurement in MPLS Networks", RFC 8169, DOI 10.17487/RFC8169, May 2017, <<https://www.rfc-editor.org/info/rfc8169>>.
- [RFC8283] Farrel, A., Ed., Zhao, Q., Ed., Li, Z., and C. Zhou, "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", RFC 8283, DOI 10.17487/RFC8283, December 2017, <<https://www.rfc-editor.org/info/rfc8283>>.

Authors' Addresses

Balazs Varga (editor)
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

Janos Farkas
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: janos.farkas@ericsson.com

Andrew G. Malis
Huawei Technologies

Email: agmalis@gmail.com

Stewart Bryant
Huawei Technologies

Email: stewart.bryant@gmail.com

Jouni Korhonen

Email: jouni.nospam@gmail.com

DetNet
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2020

B. Varga, Ed.
J. Farkas
Ericsson
L. Berger
D. Fedyk
LabN Consulting, L.L.C.
A. Malis
S. Bryant
Futurewei Technologies
J. Korhonen
July 1, 2019

DetNet Data Plane: MPLS
draft-ietf-detnet-mpls-01

Abstract

This document specifies the Deterministic Networking data plane when operating over an 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 January 2, 2020.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Terms Used in This Document	3
2.2. Abbreviations	4
2.3. Requirements Language	5
3. DetNet MPLS Data Plane Overview	5
3.1. Layers of DetNet Data Plane	5
3.2. DetNet MPLS Data Plane Scenarios	6
4. MPLS-Based DetNet Data Plane Solution	8
4.1. DetNet Over MPLS Encapsulation Components	8
4.2. MPLS Data Plane Encapsulation	9
4.2.1. DetNet Control Word and the DetNet Sequence Number	10
4.2.2. S-Labels	11
4.2.3. F-Labels	14
4.3. OAM Indication	16
4.4. Flow Aggregation	17
4.4.1. Aggregation Via LSP Hierarchy	17
4.4.2. Aggregating DetNet Flows as a new DetNet flow	17
4.5. Service Sub-Layer Considerations	19
4.5.1. Edge Node Processing	19
4.5.2. Relay Node Processing	19
4.6. Forwarding Sub-Layer Considerations	20
4.6.1. Class of Service	20
4.6.2. Quality of Service	20
5. Management and Control Information Summary	21
5.1. Service Sub-Layer Information Summary	21
5.1.1. Service Aggregation Information Summary	22
5.2. Forwarding Sub-Layer Information Summary	23
6. Security Considerations	24
7. IANA Considerations	24
8. Acknowledgements	25
9. References	25
9.1. Normative References	25
9.2. Informative References	26
Authors' Addresses	29

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].

The DetNet Architecture models the DetNet related data plane functions decomposed into two sub-layers: a service sub-layer and a forwarding sub-layer. The service sub-layer is used to provide DetNet service functions such as protection and reordering. The forwarding sub-layer is used to provide forwarding assurance (low loss, assured latency, and limited reordering).

This document specifies the DetNet data plane operation and the on-wire encapsulation of DetNet flows over an MPLS-based Packet Switched Network (PSN) using the service sub-layer reference model. MPLS encapsulation already provides a solid foundation of building blocks to enable the DetNet service and forwarding sub-layer functions. MPLS encapsulated DetNet can be carried over a variety of different network technologies that can provide the DetNet required level of service. However, the specific details of how DetNet MPLS is carried over different network technologies is out of scope of this document.

MPLS encapsulated DetNet flows can carry different types of traffic. The details of the types of traffic that are carried in DetNet are also out of scope of this document. An example of IP using DetNet MPLS sub-networks can be found in [I-D.ietf-detnet-ip]. DetNet MPLS may use an associated controller and Operations, Administration, and Maintenance (OAM) functions that are defined outside of this document.

Important background information common to all data planes for DetNet can be found in the DetNet Data Plane Framework [I-D.ietf-detnet-data-plane-framework].

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 the DetNet Data Plane Framework [I-D.ietf-detnet-data-plane-framework]. The reader is assumed to be familiar with these documents and any terminology defined therein.

The following terminology is introduced in this document:

F-Label	A Detnet "forwarding" label that identifies the LSP used to forward a DetNet flow across an MPLS PSN, e.g., a hop-by-hop label used between label switching routers (LSR).
---------	--

S-Label	A DetNet "service" label that is used between DetNet nodes that implement also the DetNet service sub-layer functions. An S-Label is also used to identify a DetNet flow at DetNet service sub-layer.
A-Label	A special case of an S-Label, whose aggregation properties are known only at the aggregation and deaggregation end-points.
d-CW	A DetNet Control Word (d-CW) is used for sequencing information of a DetNet flow at the DetNet service sub-layer.

2.2. Abbreviations

The following abbreviations are used in this document:

CoS	Class of Service.
CW	Control Word.
DetNet	Deterministic Networking.
LSR	Label Switching Router.
MPLS	Multiprotocol Label Switching.
MPLS-TE	Multiprotocol Label Switching - Traffic Engineering.
MPLS-TP	Multiprotocol Label Switching - Transport Profile.
OAM	Operations, Administration, and Maintenance.
PE	Provider Edge.
PEF	Packet Elimination Function.
PRF	Packet Replication Function.
PREOF	Packet Replication, Elimination and Ordering Functions.
POF	Packet Ordering Function.
PSN	Packet Switched Network.
PW	PseudoWire.
QoS	Quality of Service.

S-PE Switching Provider Edge.
 T-PE Terminating Provider Edge.
 TSN Time-Sensitive Network.

2.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. DetNet MPLS Data Plane Overview

3.1. Layers of DetNet Data Plane

MPLS provides a wide range of capabilities that can be utilised by DetNet. A straight forward approach utilizing MPLS for a DetNet service sub-layer is based on the existing pseudowire (PW) encapsulations and by utilizing existing MPLS Traffic Engineering encapsulations and mechanisms. Background on PWs can be found in [RFC3985] and [RFC3031]. Background on MPLS Traffic Engineering can be found in [RFC3272] and [RFC3209].

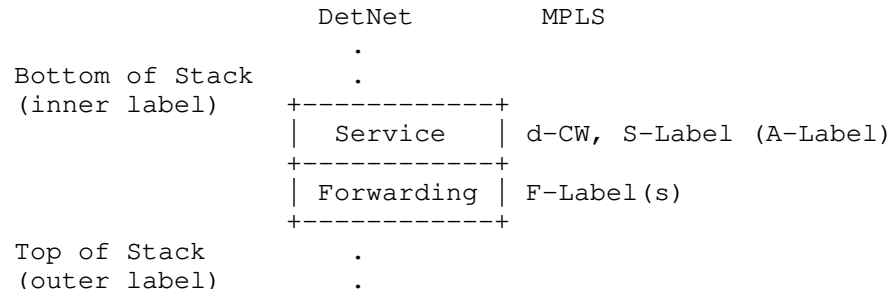


Figure 1: DetNet Adaptation to MPLS Data Plane

The DetNet MPLS data plane representation is illustrated in Figure 1. The service sub-layer includes a DetNet control word (d-CW) and a identifying service label (S-Label). The DetNet control word (d-CW) conforms to the Generic PW MPLS Control Word (PWMCW) defined in [RFC4385]. An aggregation label (A-Label) is a special case of S-Label used for aggregation.

A node operating on a DetNet flow in the DetNet service sub-layer, uses the local context associated with that S-Label, provided by a received F-Label, to determine what local DetNet operation(s) are applied to that packet. An S-Label may be taken from the platform label space [RFC3031], making it unique, enabling DetNet flow identification regardless of which input interface or LSP the packet arrives on.

The DetNet forwarding sub-layer is supported by zero or more forwarding labels (F-Labels). MPLS Traffic Engineering encapsulations and mechanisms can be utilized to provide a forwarding sub-layer that is responsible for providing resource allocation and explicit routes.

3.2. DetNet MPLS Data Plane Scenarios

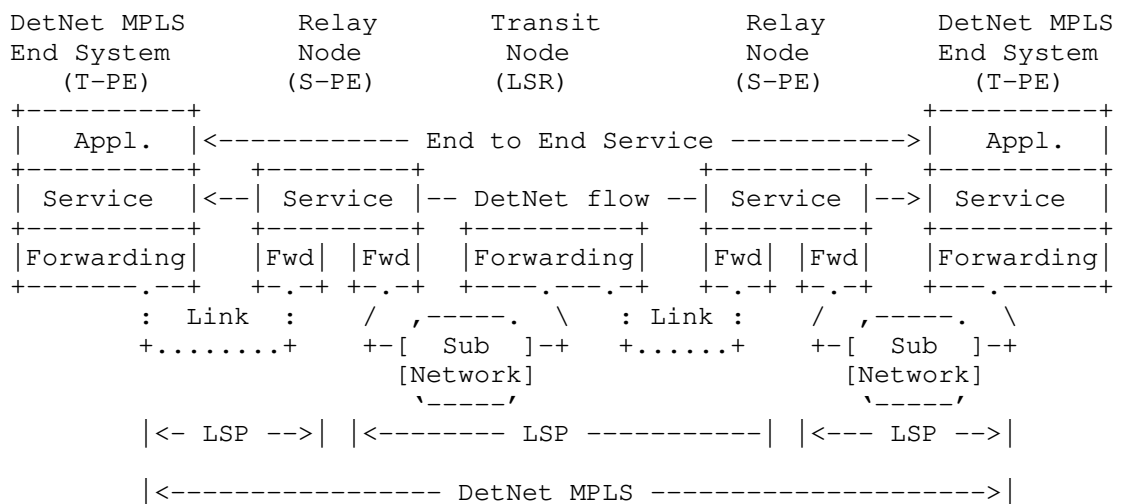


Figure 2: A DetNet MPLS Network

Figure 2 illustrates a hypothetical DetNet MPLS-only network composed of DetNet aware MPLS enabled end systems, operating over a DetNet aware MPLS network. In this figure, the relay nodes are PE devices that define the MPLS LSP boundaries and the transit nodes are LSRs.

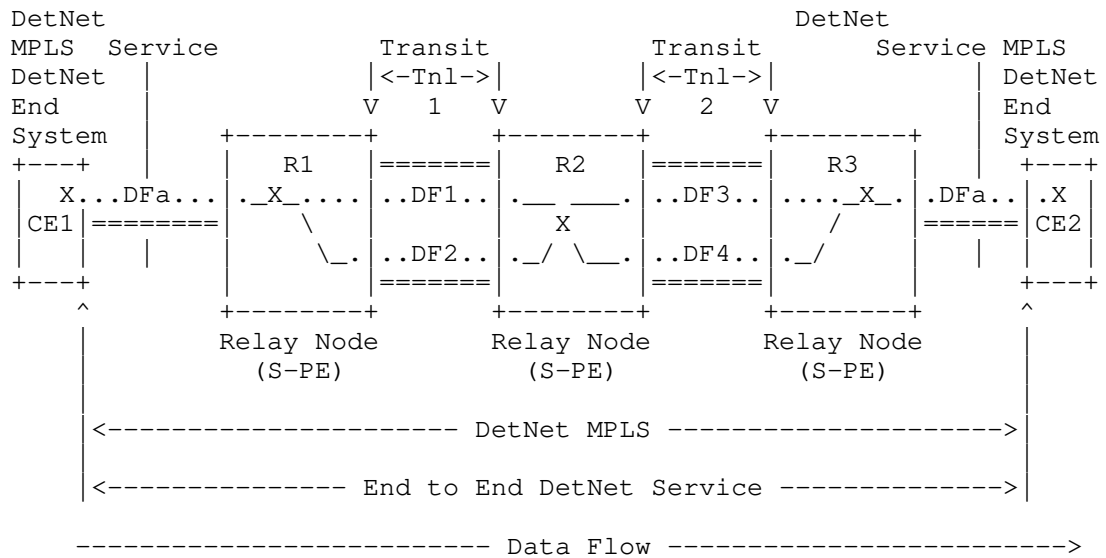
DetNet end system and relay nodes understand the particular needs of DetNet flows and provide both DetNet service and forwarding sub-layer functions. In the case of MPLS, DetNet service-aware nodes add, remove and process d-CWs, S-Labels and F-labels as needed. DetNet MPLS nodes provide functionality analogous to T-PEs when they sit at

the edge of an MPLS domain, and S-PEs when they are in the middle of an MPLS domain, see [RFC6073].

In a DetNet MPLS network, transit nodes may be DetNet service aware or may be DetNet unaware MPLS Label Switching Routers (LSRs). In this latter case, such LSRs would be unaware of the special requirements of the DetNet service sub-layer, but would still provide traffic engineering functions and the QoS capabilities needed to ensure that the (TE) LSPs meet the service requirements of the carried DetNet flows.

The application of DetNet using MPLS supports a number of control plane/management plane types. These types support certain MPLS data plane deployments. For example RSVP-TE, MPLS-TP, or MPLS Segment Routing (when extended to support resource allocation) are all valid MPLS deployment cases.

Figure 3 illustrates how an end-to-end MPLS-based DetNet service is provided in a more detail. In this figure, the customer end systems, CE1 and CE2, are able to send and receive MPLS encapsulated DetNet flows, and R1, R2 and R3 are relay nodes in the middle of a DetNet network. The 'X' in the end systems, and relay nodes represents potential DetNet compound flow packet replication and elimination points. In this example, service protection is supported utilizing at least two DetNet member flows and TE LSPs. For a unidirectional flow, R1 supports PRF and R3 supports PEF and POF. Note that the relay nodes may change the underlying forwarding sub-layer, for example tunneling MPLS over IEEE 802.1 TSN [I-D.ietf-detnet-mpls-over-tsn], or simply over interconnect network links.



X = Optional service protection (none, PRF, PREOF, PEF/POF)
 DFx = DetNet member flow x over a TE LSP

Figure 3: MPLS-Based Native DetNet

4. MPLS-Based DetNet Data Plane Solution

4.1. DetNet Over MPLS Encapsulation Components

To carry DetNet over MPLS the following is required:

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

In this design an MPLS service label (the S-Label), similar to a pseudowire (PW) label [RFC3985], is used to identify both the DetNet flow identity and the payload MPLS payload type satisfying (1) and

(2) in the list above. OAM traffic discrimination happens through the use of the Associated Channel method described in [RFC4385]. The DetNet sequence number is carried in the DetNet Control word which carries the Data/OAM discriminator. To simplify implementation and to maximize interoperability two sequence number sizes are supported: a 16 bit sequence number and a 28 bit sequence number. The 16 bit sequence number is needed to support some types of legacy clients. The 28 bit sequence number is used in situations where it is necessary ensure that in high speed networks the sequence number space does not wrap whilst packets are in flight.

The LSP used to forward the DetNet packet may be of any type (MPLS-LDP, MPLS-TE, MPLS-TP [RFC5921], or MPLS-SR [I-D.ietf-spring-segment-routing-mpls]). The LSP (F-Label) label and/or the S-Label may be used to indicate the queue processing as well as the forwarding parameters. Note that the possible use of Penultimate Hop Popping (PHP) means that the S-Label may be the only label received at the terminating DetNet service.

4.2. MPLS Data Plane Encapsulation

Figure 4 illustrates a DetNet data plane MPLS encapsulation. The MPLS-based encapsulation of the DetNet flows is well suited for the scenarios described in [I-D.ietf-detnet-data-plane-framework]. Furthermore, an end to end DetNet service i.e., native DetNet deployment (see Section 3.2) is also possible if DetNet end systems are capable of initiating and termination MPLS encapsulated packets.

The MPLS-based DetNet data plane encapsulation consists of:

- o DetNet control word (d-CW) containing sequencing information for packet replication and duplicate elimination purposes, and the OAM indicator.
- o DetNet service Label (S-Label) that identifies a DetNet flow at the receiving DetNet service sub-layer processing node.
- o Zero or more Detnet MPLS Forwarding label(s) (F-Label) used to direct the packet along the label switched path (LSP) to the next service sub-layer processing node along the path. When Penultimate Hop Popping is in use there may be no label F-Label in the protocol stack on the final hop.
- o The necessary data-link encapsulation is then applied prior to transmission over the physical media.

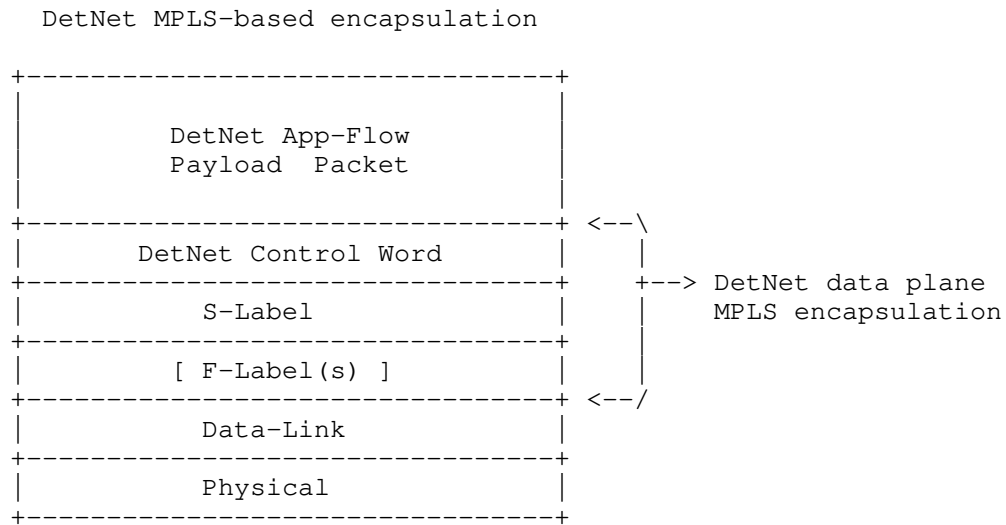


Figure 4: Encapsulation of a DetNet App-Flow in an MPLS PSN

4.2.1. DetNet Control Word and the DetNet Sequence Number

A DetNet control word (d-CW) conforms to the Generic PW MPLS Control Word (PVMCW) defined in [RFC4385]. The d-CW formatted as shown in Figure 5 MUST be present in all DetNet packets containing app-flow data.

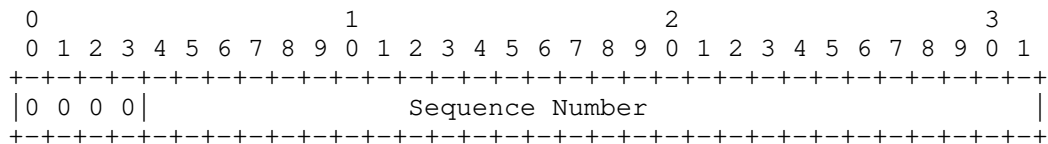


Figure 5: DetNet Control Word

(bits 0 to 3)

Per [RFC4385], MUST be set to zero (0).

Sequence Number (bits 4 to 31)

An unsigned value implementing the DetNet sequence number.

A separate sequence number space MUST be maintained by the node that adds the d-CW for each DetNet app-flow. The following sequence number field lengths MUST be supported:

0 bits

16 bits

28 bits

The sequence number length MUST be provisioned on a per app-flow basis via configuration, i.e., via the controller plane described in [I-D.ietf-detnet-data-plane-framework].

A 0 bit sequence number field length indicates that there is no DetNet sequence number used for the flow. When the length is zero, the sequence number field MUST be set to zero (0) on all packets sent for the flow.

When the sequence number field length is 16 or 28 bits for a flow, the sequence number MUST be incremented by one for each new app-flow packet sent. When the field length is 16 bits, d-CW bits 4 to 15 MUST be set to zero (0). The values carried in this field can wrap and it is important to note that zero (0) is a valid field value. For example, were the sequence number size is 16 bits, the sequence will contain: 65535, 0, 1, where zero (0) is an ordinary sequence number.

It is important to note that this document differs from [RFC4448] where a sequence number of zero (0) is used to indicate that the sequence number check algorithm is not used.

The sequence number is optionally used during receive processing as described below in Section 4.2.2.1 and Section 4.2.2.2.

4.2.2. S-Labels

App-flow identification at a DetNet service sub-layer is realized by an S-Label. MPLS-aware DetNet end systems and edge nodes, which are by definition MPLS ingress and egress nodes, MUST add and remove an app-flow specific d-CW and S-Label. Relay nodes MAY swap S-Label values when processing an app-flow.

The S-Label value MUST be provisioned per app-flow via configuration, e.g., via the controller plane described in [I-D.ietf-detnet-data-plane-framework]. Note that S-Labels provide app-flow identification at the downstream DetNet service sub-layer receiver, not the sender. As such, S-Labels MUST be allocated by the

entity that controls the service sub-layer receiving node's label space, and MAY be allocated from the platform label space [RFC3031]. Because S-Labels are local to each node rather than being a global identifier within a domain, they must be advertised to their upstream DetNet service-aware peer nodes (e.g., a DetNet MPLS End System or a DetNet Relay or Edge Node and interpreted in the context of their received F-Label.

The S-Label will normally be at the bottom of the label stack once the last F-Label is removed, immediately preceding the d-CW. To support service sub-layer level OAM, an OAM Associated Channel Header (ACH) [RFC4385] together with a Generic Associated Channel Label (GAL) [RFC5586] MAY be used in place of a d-CW.

Similarly, an Entropy Label Indicator/Entropy Label (ELI/EL) [RFC6790] MAY be carried below the S-Label in the label stack in networks where DetNet flows would otherwise received ECMP treatment. When ELs are used, the same EL value SHOULD be used for all of the packets sent using a specific S-Label to force the flow to follow the same path. However, as outlines in [I-D.ietf-detnet-data-plane-framework] the use of ECMP for DetNet flows is NOT RECOMMENDED. ECMP MAY be used for non-DetNet flows within a DetNet domain.

When receiving a DetNet MPLS flow, an implementation MUST identify the app-flow associated with the incoming packet based on the S-Label. When a node is using platform labels for S-Labels, no additional information is needed as the S-label uniquely identifies the app-flow. In the case where platform labels are not used, zero or more F-Labels and optionally, the incoming interface, proceeding the S-Label MUST be used together with the S-Label to uniquely identify the app-flows associated with a received packet. The incoming interface MAY also be used to together with any present F-Label(s) and the S-Label to uniquely identify an incoming app-flows, for example, to in the case where PHP is used. Note that choice to use platform label space for S-Label or S-Label plus one or more F-Labels to identify app flows is a local implementation choice, with one caveat. When one or more F-labels, or incoming interface, is needed together with an S-Label to uniquely identify, the controller plane MUST ensure that incoming DetNet MPLS packets arrive with the needed information (F-label(s) and/or incoming interface); the details of such are outside the scope of this document.

The use of platform labels for S-Labels matches other pseudowire encapsulations for consistency but there is no hard requirement in this regard.

4.2.2.1. Packet Elimination Function Processing

Implementations MAY support the Packet Elimination Function (PEF) for received DetNet MPLS flows. When supported, use of the PEF for a particular app-flow MUST be provisioned via configuration, e.g., via the controller plane described in [I-D.ietf-detnet-data-plane-framework].

After an app-flow is identified for a received DetNet MPLS packet, as described above, an implementation MUST check if PEF is configured for that app-flow. When configured, the implementation MUST track the sequence number contained in received d-CWs and MUST ensure that duplicate (replicated) instances of a particular sequence number are discarded. The specific mechanisms used for an implementation to identify which received packets are duplicates and which are new is an implementation choice. Note that per Section 4.2.1 the sequence number field length may be 16 or 28 bits, and the field value can wrap.

Note that an implementation MAY wish to constrain the maximum number sequence numbers that are tracked, on platform-wide or per flow basis. Some implementations MAY support the provisioning of the maximum number sequence numbers that are tracked number on either a platform-wide or per flow basis.

4.2.2.2. Packet Ordering Function Processing

A function that is related to in-order delivery is the Packet Ordering Function (POF). Implementations MAY support POF. When supported, use of the POF for a particular app-flow MUST be provisioned via configuration, e.g., via the controller plane described by [I-D.ietf-detnet-data-plane-framework]. Implementations MAY required that PEF and POF be used in combination. There is no requirement related to the order of execution of the Packet Elimination and Ordering Functions in an implementation.

After an app-flow is identified for a received DetNet MPLS packet, as described above, an implementation MUST check if POF is configured for that app-flow. When configured, the implementation MUST track the sequence number contained in received d-CWs and MUST ensure that packets are processed in the order indicated in the received d-CW sequence number field, which may not be in the order the packets are received. As defined in Section 4.2.1 the sequence number field length may be 16 or 28 bits, is incremented by one (1) for each new app-flow packet sent, and the field value can wrap. The specific mechanisms used for an implementation to identify the order of received packets is an implementation choice.

Note that an implementation MAY wish to constrain the maximum number of out of order packets that can be processed, on platform-wide or per flow basis. Some implementations MAY support the provisioning of this number on either a platform-wide or per flow basis. The number of out of order packets that can be processed also impacts the latency of a flow.

4.2.3. F-Labels

F-Labels are supported the DetNet forwarding sub-layer. F-Labels are used to provide LSP-based connectivity between DetNet service sub-layer processing nodes.

4.2.3.1. Service Sub-Layer and Packet Replication Function Processing

DetNet MPLS end systems, edge nodes and relay nodes may operate at the DetNet service sub-layer with understand of app-flows and their requirements. As mentioned earlier, when operating at this layer such nodes can push, pop or swap (pop then push) S-Labels. In all cases, the F-Labels used for the app-flow are always replaced and the following procedures apply.

When sending a DetNet flow, zero or more F-Labels MAY be pushed on top of an S-Label by the node pushing an S-Label. The F-Labels to be pushed when sending a particular app-flow MUST be provisioned per app-flow via configuration, e.g., via the controller plane discussed in [I-D.ietf-detnet-data-plane-framework]. F-Labels can also provide context for an S-Label. To allow for the omission of F-Labels, an implementation SHOULD also allow an outgoing interface to be used.

The Packet Replication Function (PRF) function MAY be supported by an implementation for outgoing DetNet flows. When replication is supported, the same app-flow data will be sent over multiple outgoing forwarding sub-layer LSPs. To support PRF an implementation MUST support the setting of different sets of F-Labels. To allow for the omission of F-Labels, an implementation SHOULD also allow multiple outgoing interfaces to be provisioned. PRF MUST NOT be used with app-flows configured with a d-CW sequence number field length of 0 bits.

When a single set of F-Labels is provisioned for a particular outgoing app-flow, that set of F-labels MUST be pushed after the S-Label is pushed. The outgoing packet is then forwarded as described below in Section 4.2.3.2. When a single outgoing interface is provisioned, the outgoing packet is then forwarded as described below in Section 4.2.3.2.

When multiple sets of F-Labels or interfaces are provisioned for a particular outgoing app-flow, a copy of the outgoing packet, including the pushed S-Label, MUST be made per F-label set and outgoing interface. Each set of provisioned F-Labels are then pushed onto a copy of the packet. Each copy is then forwarded as described below in Section 4.2.3.2.

As described in the previous section, when receiving a DetNet MPLS flow, an implementation identifies the app-flow associated with the incoming packet based on the S-Label. When a node is using platform labels for S-Labels, any F-Labels can be popped and the S-label uniquely identifies the app-flow. In the case where platform labels are not used, F-Label(s) and, optionally, the incoming interface MUST also be provisioned for incoming app-flows. The provisioned information MUST then be used to identify incoming app-flows based on the combination of S-Label and F-Label(s) or incoming interface.

4.2.3.2. Common F-Label Processing

All DetNet aware MPLS nodes process F-Labels as needed to meet the service requirements of the DetNet flow or flows carried in the LSPs represented by the F-Labels. This includes normal push, pop and swap operations. Such processing is essentially the same type of processing provided for TE LSPs, although the specific service parameters, or traffic specification, can differ. When the DetNet service parameters of the app-flow or flows carried in an LSP represented by an F-Label can be met by an existing TE mechanism, the forwarding sub-layer processing node MAY be a DetNet unaware, i.e., standard, MPLS LSR. Such TE LSPs may provide LSP forwarding service as defined in, but not limited to, [RFC3209], [RFC3270], [RFC3272], [RFC3473], [RFC4875], [RFC5440], and [RFC6006].

More specifically, as mentioned above, the DetNet forwarding sub-layer provides explicit routes and allocated resources, and F-Labels are used to map to each. Explicit routes are supported based on the topmost (outermost) F-Label that is pushed or swapped and the LSP that corresponds to this label. This topmost (outgoing) label MUST be associated with a provisioned outgoing interface and, for non-point-to-point outgoing interfaces, a next hop LSR. Note that this information MUST be provisioned via configuration or the controller plane. In the previously mentioned special case where there are no added F-labels and the outgoing interface is not a point-to-point interface, the outgoing interface MUST also be associated with a next hop LSR.

Resources may be allocated in a hierarchical fashion per LSP that is represented by each F-Label. Each LSP MAY be provisioned with a service parameters that dictates the specific traffic treatment to be

received by the traffic carried over that LSP. Implementations of this document MUST ensure that traffic carried over each LSP represented by one or more F-Labels receives the traffic treatment provisioned for that LSP. Typical mechanisms used to provide different treatment to different flows includes the allocation of system resources (such as queues and buffers) and provisioning or related parameters (such as shaping, and policing). Support can also be provided via an underlying network technology such IEEE802.1 TSN [I-D.ietf-detnet-mpls-over-tsn]. The specific mechanisms used by a DetNet node to ensure DetNet service delivery requirements are met for supported DetNet flows is outside the scope of this document.

Packets that are marked in a way that do not correspond to allocated resources, e.g., lack a provisioned F-Label, 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:

- o MUST defend the DetNet QoS by discarding or remarking (to an allocated DetNet flow or non-competing non-DetNet flow) packets received that are not associated with a completed resource allocation.
- o MUST NOT use a DetNet allocated resource, e.g. a queue or shaper reserved for DetNet flows, for any packet that does match the corresponding DetNet flow.
- o MUST ensure a QoS flow does not exceed its allocated resources or provisioned service level,
- o MUST ensure a CoS flow or service class does not impact the service delivered to other flows. This requirement is similar to requirement for MPLS LSRs to that CoS LSPs do not impact the resources allocated to TE LSPs, e.g., via [RFC3473].

Subsequent sections provide additional considerations related to CoS (Section 4.6.1), QoS (Section 4.6.2) and aggregation (Section 4.4).

4.3. OAM Indication

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

As shown in Figure 3 of [RFC5085] when the first nibble of the d-CW is 0x0 the payload following the d-CW is normal user data. However, when the first nibble of the d-CW is 0x1, the payload that follows

the d-DW is an OAM payload with the OAM type indicated by the value in the d-CW Channel Type field.

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

4.4. Flow Aggregation

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

The resource control and management aspects of aggregation (including the configuration of queuing, shaping, and policing) are the responsibility of the DetNet controller plane and is out of scope of this documents. It is also the responsibility of the controller plane to ensure that consistent aggregation methods are used.

4.4.1. Aggregation Via LSP Hierarchy

DetNet flows forwarded 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 individual LSPs and H-LSPs receive the traffic treatment required to ensure the required DetNet service is preserved.

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

4.4.2. Aggregating DetNet Flows as a new DetNet flow

An aggregate can be built by layering DetNet flows, including both their S-Label and, when present, F-Labels as shown below:

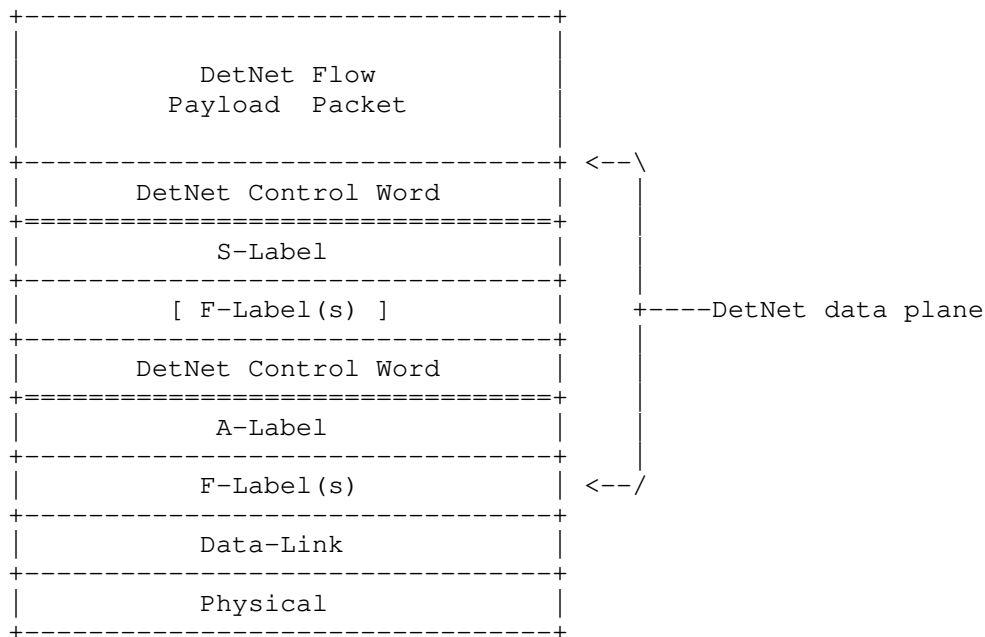


Figure 6: DetNet Aggregation as a new DetNet Flow

Both the aggregation label, which is referred to as an A-Label, and the individual flow's S-Label have their MPLS S bit set indicating bottom of stack, and the d-CW allows the PREOF to work. An A-Label is a special case of an S-Label, whose properties are known only at the aggregation and deaggregation end-points.

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

As in the previous section, nodes supporting this type of aggregation will need to ensure that individual and aggregated flows receive the traffic treatment required to ensure the required DetNet service is preserved. Also, it is the controller plane's responsibility to ensure that the service required on the aggregate flow are properly provisioned.

4.5. Service Sub-Layer Considerations

The edge and relay node internal procedures related to PREOF are implementation specific. The order of a packet elimination or replication is out of scope in this specification.

It is important that the DetNet layer is configured such that a DetNet node never receives its own replicated packets. If it were to receive such packets the replication function would make the loop more destructive of bandwidth than a conventional unicast loop. Ultimately the TTL in the S-Label will cause the packet to die during a transient loop, but given the sensitivity of applications to packet latency the impact on the DetNet application would be severe. To avoid the problem of a transient forwarding loop, changes to an LSP supporting DetNet MUST be loop-free.

4.5.1. Edge Node Processing

An edge node is responsible for matching ingress packets to the service they require and encapsulating them accordingly. An edge node may participate in the packet replication and duplicate packet elimination.

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.

The internal design of a relay node is out of scope of this document. However the reader's attention is drawn to the need to make any PREOF state available to the packet processor(s) dealing with packets to which the PREOF functions must be applied, and to maintain that state is such as way that it is available to the packet processor operation on the next packet in the DetNet flow (which may be a duplicate, a late packet, or the next packet in sequence).

4.5.2. Relay Node Processing

A DetNet Relay node operates in the DetNet forwarding sub-layer . For DetNet using MPLS this processing is performed on the F-Label. 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. Some relay nodes may be DetNet service aware, while others may be unmodified LSRs that only understand how to switch MPLS-TE LSPs.

It is also possible to treat the relay node as a transit node, see Section 4.4. Again, this is entirely up to how the forwarding has been programmed.

4.6. Forwarding Sub-Layer Considerations

4.6.1. Class of Service

Class and quality of service, i.e., CoS and QoS, are terms that are often used interchangeably and confused with each other. 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].

4.6.2. Quality of Service

In addition to explicit routes, and packet replication and elimination, described in Section 4 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. This includes Quality of Service (QoS) mechanisms at the MPLS layer, that may be combined with the mechanisms defined by the underlying network layer such as 802.1TSN.

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.

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

5. Management and Control Information Summary

The specific information needed for the processing of each DetNet service depends on the DetNet node type and the functions being provided on the node. This section summarizes based on the DetNet sub-layer and if the DetNet traffic is being sent or received. All DetNet node types are DetNet forwarding sub-layer aware, while all but transit nodes are service sub-layer aware. This is shown in Figure 2.

Much like other MPLS labels, there are a number of alternatives available for DetNet S-Label and F-Label advertisement to an upstream peer node. These include distributed signaling protocols such as RSVP-TE, centralized label distribution via a controller that manages both the sender and the receiver using NETCONF/YANG, BGP, PCEP, etc., and hybrid combinations of the two. The details of the controller plane solution required for the label distribution and the management of the label number space are out of scope of this document. There are particular DetNet considerations and requirements that are discussed in [I-D.ietf-detnet-data-plane-framework].

5.1. Service Sub-Layer Information Summary

The following summarizes the information that is needed on service sub-layer aware nodes that send DetNet MPLS traffic, on a per service basis:

- o App-Flow identification information, e.g., an incoming service on a relay node or IP information as defined in [I-D.ietf-detnet-ip-over-mpls].
- o The sequence number length to be used for the service. Valid values included 0, 16 and 28 bits. 0 bits cannot be used when PRF is configured for the service.
- o The S-Label for the service.
- o If PRF is to be provided for the service.
- o The forwarding sub-layer information associated with the output of the service sub-layer. Note that when the PRF function is provisioned, this information is per DetNet member flow. Logically this is a pointer to details provided below for transmission of Detnet flows at the forwarding sub-layer.

The following summarizes the information that is needed on service sub-layer aware nodes that receives DetNet MPLS traffic, on a per service basis:

- o The forwarding sub-layer information associated with the input of the service sub-layer. Note that when the PEF function is provisioned, this information is per DetNet member flow. Logically this is a pointer to details provided below related to the reception of Detnet flows at the forwarding sub-layer or A-Label.
- o The S-Label for the received service.
- o If PEF or POF is to be provided for the service.
- o The sequence number length to be used for the service. Valid values included 0, 16 and 28 bits. 0 bits cannot be used when PEF or POF are configured for the service.

5.1.1. Service Aggregation Information Summary

Nodes performing aggregation using A-Labels, per Section 4.4.2, require the additional information summarized in this section.

The following summarizes the information that is needed on a node that sends aggregated flows using A-Labels:

- o The S-Labels or F-Labels that are to be carried over each aggregated service.

- o The A-Label associated with each aggregated service.
- o The other S-Label information summarized above.

On the receiving node, the A-Label provides the forwarding context of an incoming interface or an F-Label and is used in subsequent service or forwarding sub-layer receive processing, as appropriated. The related addition configuration that may be provided discussed elsewhere in this section.

5.2. Forwarding Sub-Layer Information Summary

The following summarizes the information that is needed on forwarding sub-layer aware nodes that send DetNet MPLS traffic, on a per forwarding sub-layer flow basis:

- o The outgoing F-Label stack to be pushed. The stack may include H-LSP labels.
- o The traffic parameters associated with a specific label in the stack. Note that there may be multiple sets of traffic parameters associated with specific labels in the stack, e.g., when H-LSPs are used.
- o Outgoing interface and, for unicast traffic, the next hop information.
- o Sub-network specific parameters on a technology specific basis. For example, see [I-D.ietf-detnet-mpls-over-tsn].

The following summarizes the information that is needed on forwarding sub-layer aware nodes that receive DetNet MPLS traffic, on a per forwarding sub-layer flow basis:

- o The incoming interface. For some implementations and deployment scenarios, this information may not be needed.
- o The incoming F-Label stack to be popped. The stack may include H-LSP labels.
- o How the incoming forwarding sub-layer flow is to be handled, i.e., forwarded as a transit node, or provided to the service sub-layer.

It is the responsibility of the DetNet controller plane to properly provision both flow identification information and the flow specific resources needed to provide the traffic treatment needed to meet each flow's service requirements. This applies for aggregated and individual flows.

6. Security Considerations

Security considerations for DetNet are described in detail in [I-D.ietf-detnet-security]. General security considerations are described in [I-D.ietf-detnet-architecture]. This section considers exclusively security considerations which are specific to the DetNet MPLS data plane.

Security aspects which are unique to DetNet are those whose aim is to provide the specific quality of service aspects of DetNet, which are primarily to deliver data flows with extremely low packet loss rates and bounded end-to-end delivery latency.

The primary considerations for the data plane is to maintain integrity of data and delivery of the associated DetNet service traversing the DetNet network. Application flows can be protected through whatever means is provided by the underlying technology. For example, encryption may be used, such as that provided by IPSec [RFC4301] for IP flows and/or by an underlying sub-net using MACSec [IEEE802.1AE-2018] for IP over Ethernet (Layer-2) flows.

From a data plane perspective this document does not add or modify any header information.

At the management and control level DetNet flows are identified on a per-flow basis, which may provide controller plane attackers with additional information about the data flows (when compared to controller planes that do not include per-flow identification). This is an inherent property of DetNet which has security implications that should be considered when determining if DetNet is a suitable technology for any given use case.

To provide uninterrupted availability of the DetNet service, provisions can be made against DOS attacks and delay attacks. To protect against DOS attacks, excess traffic due to malicious or malfunctioning devices can be prevented or mitigated, for example through the use of existing mechanism such as policing and shaping applied at the input of a DetNet domain. To prevent DetNet packets from being delayed by an entity external to a DetNet domain, DetNet technology definition can allow for the mitigation of Man-In-The-Middle attacks, for example through use of authentication and authorization of devices within the DetNet domain.

7. IANA Considerations

This document makes no IANA requests.

8. Acknowledgements

The authors wish to thank Pat Thaler, Norman Finn, Loa Anderson, David Black, Rodney Cummings, Ethan Grossman, Tal Mizrahi, David Mozes, Craig Gunther, George Swallow, Yuanlong Jiang and Carlos J. Bernardos for their various contributions to this work.

9. References

9.1. 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>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <<https://www.rfc-editor.org/info/rfc3270>>.

- [RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <<https://www.rfc-editor.org/info/rfc3443>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [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>>.
- [RFC5085] Nadeau, T., Ed. and C. Pignataro, Ed., "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, DOI 10.17487/RFC5085, December 2007, <<https://www.rfc-editor.org/info/rfc5085>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

9.2. Informative References

- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", draft-ietf-detnet-architecture-13 (work in progress), May 2019.

- [I-D.ietf-detnet-data-plane-framework]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane Framework", draft-ietf-detnet-data-plane-framework-00 (work in progress), May 2019.
- [I-D.ietf-detnet-ip-over-mpls]
Varga, B., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP over MPLS", draft-ietf-detnet-ip-over-mpls-00 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls-over-tsn]
Varga, B., Farkas, J., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS over IEEE 802.1 Time Sensitive Networking (TSN)", draft-ietf-detnet-mpls-over-tsn-00 (work in progress), May 2019.
- [I-D.ietf-detnet-security]
Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell, J., Austad, H., Stanton, K., and N. Finn, "Deterministic Networking (DetNet) Security Considerations", draft-ietf-detnet-security-04 (work in progress), March 2019.
- [I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-22 (work in progress), May 2019.
- [IEEE802.1AE-2018]
IEEE Standards Association, "IEEE Std 802.1AE-2018 MAC Security (MACsec)", 2018,
<<https://ieeexplore.ieee.org/document/8585421>>.
- [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>>.
- [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>>.

- [RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002, <<https://www.rfc-editor.org/info/rfc3272>>.
- [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>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <<https://www.rfc-editor.org/info/rfc4448>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <<https://www.rfc-editor.org/info/rfc5586>>.
- [RFC5921] Bocci, M., Ed., Bryant, S., Ed., Frost, D., Ed., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, DOI 10.17487/RFC5921, July 2010, <<https://www.rfc-editor.org/info/rfc5921>>.
- [RFC6003] Papadimitriou, D., "Ethernet Traffic Parameters", RFC 6003, DOI 10.17487/RFC6003, October 2010, <<https://www.rfc-editor.org/info/rfc6003>>.

- [RFC6006] Zhao, Q., Ed., King, D., Ed., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 6006, DOI 10.17487/RFC6006, September 2010, <<https://www.rfc-editor.org/info/rfc6006>>.
- [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>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<https://www.rfc-editor.org/info/rfc6790>>.

Authors' Addresses

Balazs Varga (editor)
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

Janos Farkas
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: janos.farkas@ericsson.com

Lou Berger
LabN Consulting, L.L.C.

Email: lberger@labn.net

Don Fedyk
LabN Consulting, L.L.C.

Email: dfedyk@labn.net

Andrew G. Malis
Futurewei Technologies

Email: agmalis@gmail.com

Stewart Bryant
Futurewei Technologies

Email: stewart.bryant@gmail.com

Jouni Korhonen

Email: jouni.nospam@gmail.com

DetNet
Internet-Draft
Intended status: Standards Track
Expires: November 6, 2019

B. Varga, Ed.
J. Farkas
Ericsson
A. Malis
S. Bryant
Huawei Technologies
J. Korhonen
May 5, 2019

DetNet Data Plane: MPLS over IEEE 802.1 Time Sensitive Networking (TSN)
draft-ietf-detnet-mpls-over-tsn-00

Abstract

This document specifies the Deterministic Networking MPLS data plane when operating over a TSN network.

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 November 6, 2019.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Terms Used in This Document	3
2.2. Abbreviations	4
3. Requirements Language	5
4. DetNet MPLS Data Plane Overview	5
4.1. Layers of DetNet Data Plane	5
4.2. DetNet MPLS Data Plane Scenarios	6
4.3. Packet Flow Example with Service Protection	9
5. DetNet MPLS Data Plane Considerations	11
5.1. Sub-Network Considerations	12
6. DetNet MPLS Operation Over IEEE 802.1 TSN Sub-Networks	12
6.1. Mapping of TSN Stream ID and Sequence Number	14
6.2. TSN Usage of FRER	15
6.3. Procedures	16
6.4. Layer 2 Addressing and QoS Considerations	16
7. Management and Control Considerations	16
8. Security Considerations	17
9. IANA Considerations	17
10. Acknowledgements	17
11. References	17
11.1. Normative References	17
11.2. Informative References	19
Appendix A. Example of DetNet Data Plane Operation	23
Authors' Addresses	23

1. Introduction

[Editor's note: Introduction to be made specific to DetNet MPLS over TSN scenario. May be similar to intro of DetNet IP over TSN.].

Deterministic Networking (DetNet) is a service that can be offered by a network to DetNet flows. DetNet provides these flows with a 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].

The DetNet Architecture decomposes the DetNet related data plane functions into two sub-layers: a service sub-layer and a forwarding sub-layer. The service sub-layer is used to provide DetNet service protection and reordering. The forwarding sub-layer is used to provides congestion protection (low loss, assured latency, and limited reordering) leveraging MPLS Traffic Engineering mechanisms.

This document specifies the DetNet data plane operation and the on-wire encapsulation of DetNet flows over an MPLS-based Packet Switched Network (PSN). The specified encapsulation provides the building blocks to enable the DetNet service and forwarding sub-layer functions and supports flow identification as described in the DetNet Architecture. As part of the service sub-layer functions, this document describes DetNet node data plane operation. It also describes the function and operation of the Packet Replication (PRF) Packet Elimination (PEF) and Packet Ordering (POF) functions with an MPLS data plane. It also describes an MPLS-based DetNet forwarding sub-layer that eliminates (or reduces) contention loss and provides bounded latency for DetNet flows.

MPLS encapsulated DetNet flows can be carried over network technologies that can provide the DetNet required level of service. This document defines examples of such, specifically carrying DetNet MPLS flows over IEEE 802.1 TSN sub-networks, and over DetNet IP PSN.

The intent is for this document to support different traffic types being mapped over DetNet MPLS, but this is out side the scope of this document. An example of such can be found in [I-D.ietf-detnet-dp-sol-ip]. This document also allows for, but does not define, associated controller plane and Operations, Administration, and Maintenance (OAM) functions.

2. Terminology

[Editor's note: Needs clean up.].

2.1. Terms Used in This Document

This document uses the terminology established in the DetNet architecture [I-D.ietf-detnet-architecture], and the reader is assumed to be familiar with that document and its terminology.

The following terminology is introduced in this document:

F-Label	A Detnet "forwarding" label that identifies the LSP used to forward a DetNet flow across an MPLS PSN, e.g., a hop-by-hop label used between label switching routers (LSR).
S-Label	A DetNet "service" label that is used between DetNet nodes that implement also the DetNet service sub-layer functions. An S-Label is also used to identify a DetNet flow at DetNet service sub-layer.

d-CW A DetNet Control Word (d-CW) is used for sequencing and identifying duplicate packets of a DetNet flow at the DetNet service sub-layer.

2.2. Abbreviations

The following abbreviations are used in this document:

AC	Attachment Circuit.
CE	Customer Edge equipment.
CoS	Class of Service.
CW	Control Word.
DetNet	Deterministic Networking.
DF	DetNet Flow.
DN-IWF	DetNet Inter-Working Function.
L2	Layer 2.
L2VPN	Layer 2 Virtual Private Network.
L3	Layer 3.
LSR	Label Switching Router.
MPLS	Multiprotocol Label Switching.
MPLS-TE	Multiprotocol Label Switching - Traffic Engineering.
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.
PEF	Packet Elimination Function.
PRF	Packet Replication Function.

PREOF	Packet Replication, Elimination and Ordering Functions.
POF	Packet Ordering Function.
PSN	Packet Switched Network.
PW	PseudoWire.
QoS	Quality of Service.
S-PE	Switching Provider Edge.
T-PE	Terminating Provider Edge.
TSN	Time-Sensitive Network.

3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. DetNet MPLS Data Plane Overview

[Editor's note: simplify this section and highlight DetNet MPLS over subnets scenario being the focus in the remaining part of the document.].

4.1. Layers of DetNet Data Plane

This document describes how DetNet flows are carried over MPLS networks. The DetNet Architecture, [I-D.ietf-detnet-architecture], decomposes the DetNet data plane into two sub-layers: a service sub-layer and a forwarding sub-layer. The basic approach defined in this document supports the DetNet service sub-layer based on existing pseudowire (PW) encapsulations and mechanisms, and supports the DetNet forwarding sub-layer based on existing MPLS Traffic Engineering encapsulations and mechanisms. Background on PWs can be found in [RFC3985] and [RFC3031]. Background on MPLS Traffic Engineering can be found in [RFC3272] and [RFC3209].

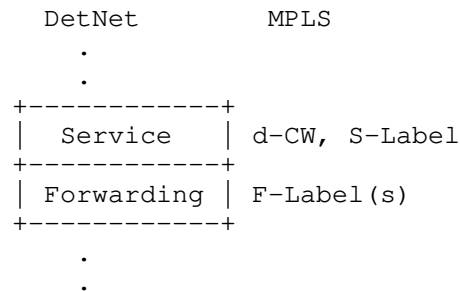


Figure 1: DetNet Adaptation to MPLS Data Plane

The DetNet MPLS data plane approach defined in this document is shown in Figure 1. The service sub-layer is supported by a DetNet control word (d-CW) which conforms to the Generic PW MPLS Control Word (PWMCW) defined in [RFC4385]. A d-CW identifying service label (S-Label) is also used.

A node operating on a DetNet flow in the Detnet service sub-layer, i.e. a node processing a DetNet packet which has the S-Label as top of stack uses the local context associated with that S-Label, for example a received F-Label, to determine what local DetNet operation(s) are applied to that packet. An S-Label may be unique when taken from the platform label space [RFC3031], which would enable correct DetNet flow identification regardless of which input interface or LSP the packet arrives on.

The DetNet MPLS data plane builds on MPLS Traffic Engineering encapsulations and mechanisms to provide a forwarding sub-layer that is responsible for providing resource allocation and explicit routes. The forwarding sub-layer is supported by one or more forwarding labels (F-Labels).

4.2. DetNet MPLS Data Plane Scenarios

[Editor's note: simplify this section and highlight DetNet MPLS over subnets scenario being the focus in the remaining part of the document.].

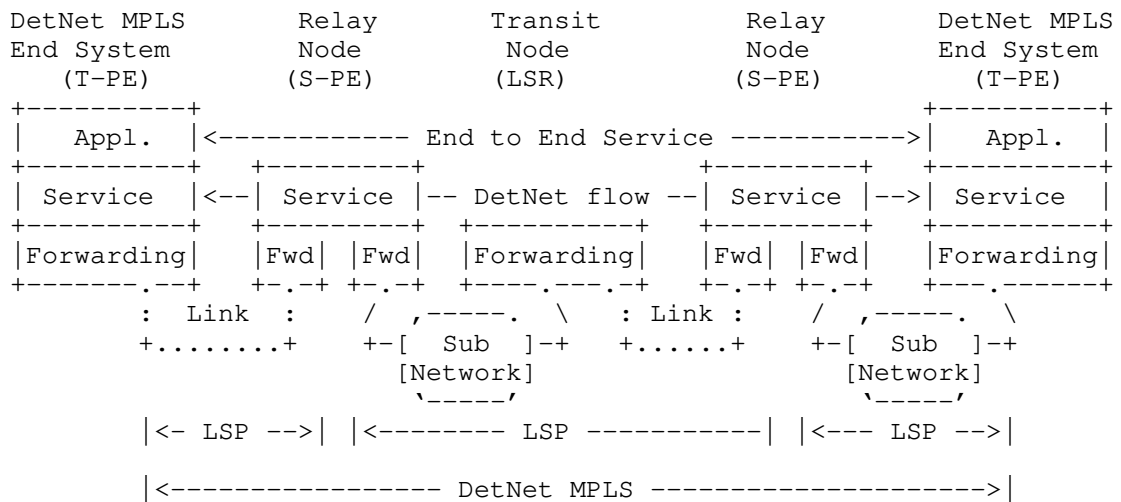


Figure 2: A DetNet MPLS Network

Figure 2 illustrates a hypothetical DetNet MPLS-only network composed of DetNet aware MPLS enabled end systems, operating over a DetNet aware MPLS network. In this figure, relay nodes sit at MPLS LSP boundaries and transit nodes are LSRs.

DetNet end system and relay nodes are DetNet service sub-layer aware, understand the particular needs of DetNet flows and provide both DetNet service and forwarding sub-layer functions. They add, remove and process d-CWs, S-Labels and F-labels as needed. MPLS enabled end system and relay nodes can enhance the reliability of delivery by enabling the replication of packets where multiple copies, possibly over multiple paths, are forwarded through the DetNet domain. They can also eliminate surplus previously replicated copies of DetNet packets. DetNet MPLS nodes provide functionality similar to T-PEs when they sit at the edge of an MPLS domain, and functionality similar to S-PEs when they are in the middle of an MPLS domain, see [RFC6073]. End system and relay nodes also include DetNet forwarding sub-layer functions, support for notably explicit routes, and resources allocation to eliminate (or reduce) congestion loss and jitter.

DetNet transit nodes reside wholly within a DetNet domain, and also provide DetNet forwarding sub-layer functions in accordance with the performance required by a DetNet flow carried over an LSP. Unlike other DetNet node types, transit nodes provide no service sub-layer processing. In a DetNet MPLS network, transit nodes may be DetNet service aware or may be DetNet unaware MPLS Label Switching Routers

(LSRs). In this latter case, such LSRs would be unaware of the special requirements of the DetNet service sub-layer, but would still provide traffic engineering services and the QoS need to ensure that the (TE) LSPs meet the service requirements of the carried DetNet flows.

The LSPs may be provided by any MPLS controller method. For example they may be provisioned via a management plane, RSVP-TE, MPLS-TP, or MPLS Segment Routing (when extended to support resource allocation).

[Editor's note: Figure 3. and surrounding text are candidates to delete from this document.].

Figure 3 illustrates how an end to end MPLS-based DetNet service is provided in a more detail. In this figure, the end systems, CE1 and CE2, are able to send and receive MPLS encapsulated DetNet flows, and R1, R2 and R3 are relay nodes as they sit in the middle of a DetNet network. The 'X' in the end systems, and relay nodes represents potential DetNet compound flow packet replication and elimination points. In this example, service protection is supported over four DetNet member flows and TE LSPs. For a unidirectional flow, R1 supports PRF, R2 supports PREOF and R3 supports PEF and POF. Note that the relay nodes may change the underlying forwarding sub-layer, for example tunneling MPLS over IEEE 802.1 TSN Section 6, or simply over interconnect network links.

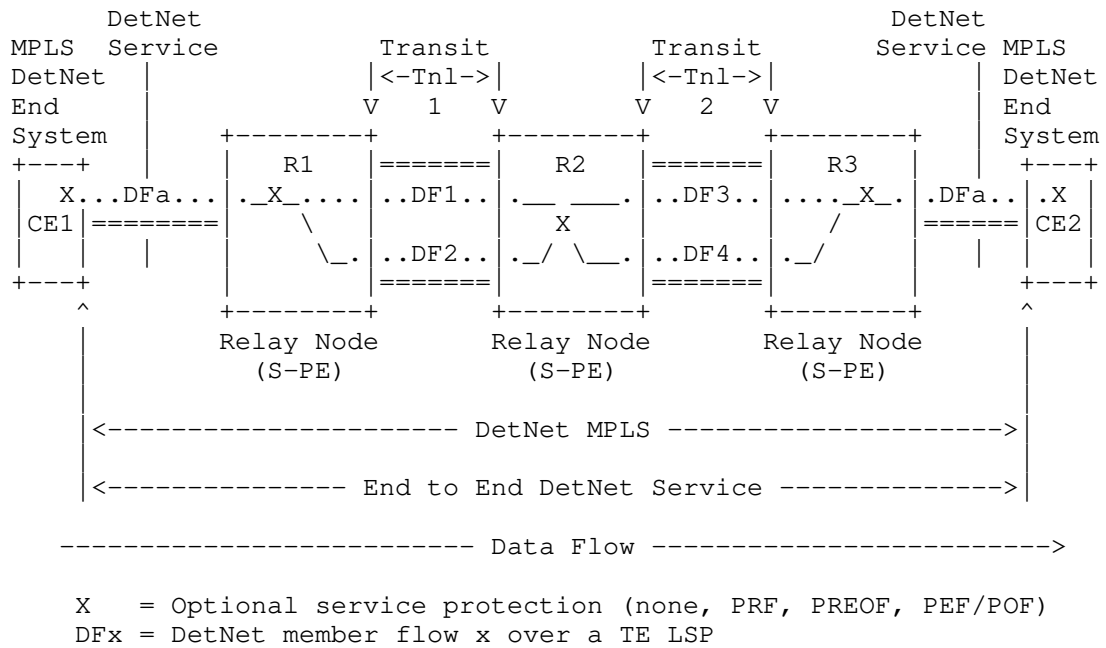


Figure 3: MPLS-Based Native DetNet

As previously mentioned, this document specifies how MPLS is used to support DetNet flows using an MPLS data plane as well as how such can be mapped to IEEE 802.1 TSN and IP DetNet PSNs. An equally important scenario is when IP is supported over DetNet MPLS and this is covered in [I-D.ietf-detnet-dp-sol-ip]. Another important scenario is where an Ethernet Layer 2 service is supported over DetNet MPLS and this is covered in [TBD-TSN-OVER-DETNET].

4.3. Packet Flow Example with Service Protection

[Editor's note: this text might be relevant for the discussion of FRER within the TSN sub-network. Needs revision.].

An example DetNet MPLS network fragment and packet flow is illustrated in Figure 4.

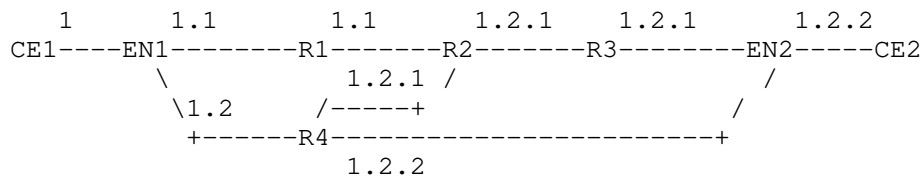


Figure 4: Example Packet Flow in DetNet Enabled MPLS Network

In Figure 4 the numbers are used to identify the instance of a packet. Packet 1 is the original packet, and packets 1.1, and 1.2 are two first generation copies of packet 1. Packet 1.2.1 is a second generation copy of packet 1.2 etc. Note that these numbers never appear in the packet, and are not to be confused with sequence numbers, labels or any other identifier that appears in the packet. They simply indicate the generation number of the original packet so that its passage through the network fragment can be identified to the reader.

Customer Equipment CE1 sends a packet into the DetNet enabled MPLS network. This is packet (1). Edge Node EN1 encapsulates the packet as a DetNet Packet and sends it to Relay node R1 (packet 1.1). EN1 makes a copy of the packet (1.2), encapsulates it and sends this copy to Relay node R4.

Note that along the MPLS path from EN1 to R1 there may be zero or more LSRs which, for clarity, are not shown. The same is true for any other path between two DetNet entities shown in Figure 4.

Relay node R4 has been configured to send one copy of the packet to Relay Node R2 (packet 1.2.1) and one copy to Edge Node EN2 (packet 1.2.2).

R2 receives packet copy 1.2.1 before packet copy 1.1 arrives, and, having been configured to perform packet elimination on this DetNet flow, forwards packet 1.2.1 to Relay Node R3. Packet copy 1.1 is of no further use and so is discarded by R2.

Edge Node EN2 receives packet copy 1.2.2 from R4 before it receives packet copy 1.2.1 from R2 via relay Node R3. EN2 therefore strips any DetNet encapsulation from packet copy 1.2.2 and forwards the packet to CE2. When EN2 receives the later packet copy 1.2.1 this is discarded.

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

5. DetNet MPLS Data Plane Considerations

[Editor's note: Sort out what data plane considerations are relevant for sub-net scenarios.].

This section provides informative considerations related to providing DetNet service to flows which are identified based on their header information. At a high level, the following are provided on a per flow basis:

Eliminating contention loss and jitter reduction:

Use of allocated resources (queuing, policing, shaping) to ensure that the congestion-related loss and latency/jitter requirements of a DetNet flow are met.

Explicit routes:

Use of a specific path for a flow. This limits misordering and bounds latency.

Service protection:

Which in the case of this document primarily relates to replication and elimination. Changing the explicit path after a failure is detected in order to restore delivery of the required DetNet service characteristics is also possible. Path changes, even in the case of failure recovery, can lead to the out of order delivery of data.

Load sharing:

Generally, distributing packets of the same DetNet flow over multiple paths is not recommended. Such load sharing, e.g., via ECMP or UCMP, impacts ordering and possibly jitter.

Troubleshooting:

For example, to support identification of misbehaving flows.

Recognize flow(s) for analytics:

For example, increase counters.

Correlate events with flows:

For example, unexpected loss.

The DetNet data plane also allows for the aggregation of DetNet flows, e.g., via MPLS hierarchical LSPs, to improved scaling. When DetNet flows are aggregated, transit nodes provide service to the aggregate and not on a per-DetNet flow basis. In this case, nodes performing aggregation will ensure that per-flow service requirements are achieved.

5.1. Sub-Network Considerations

As shown in Figure 2, MPLS nodes are interconnected by different sub-network technologies, which may include point-to-point links. Each of these need to provide appropriate service to DetNet flows. In some cases, e.g., on dedicated point-to-point links or TDM technologies, all that is required is for a DetNet node to appropriately queue its output traffic. In other cases, DetNet nodes will need to map DetNet flows to the flow semantics (i.e., identifiers) and mechanisms used by an underlying sub-network technology. Figure 5 shows several examples of header formats that can be used to carry DetNet MPLS flows over different sub-network technologies. L2 represent a generic layer-2 encapsulation that might be used on a point-to-point link. TSN represents the encapsulation used on an IEEE 802.1 TSN network, as described in Section 6. UDP/IP represents the encapsulation used on a DetNet IP PSN.

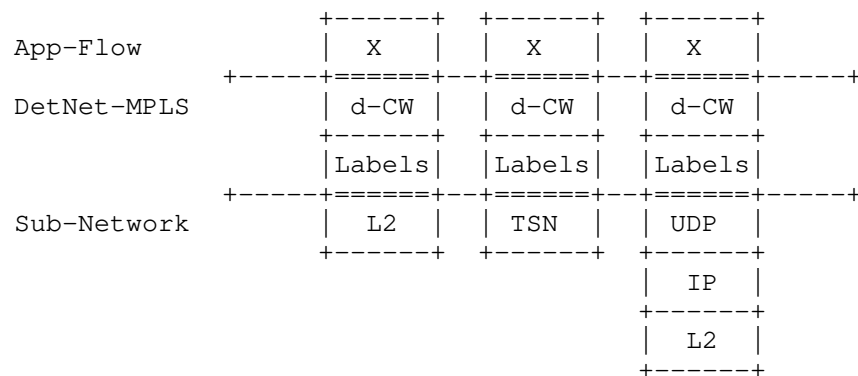


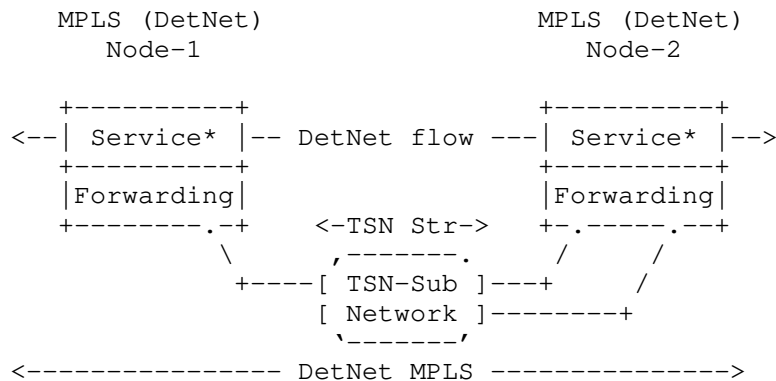
Figure 5: Example DetNet MPLS Sub-Network Formats

6. DetNet MPLS Operation Over IEEE 802.1 TSN Sub-Networks

[Editor's note: this is a place holder section. A standalone section on MPLS over IEEE 802.1 TSN. Includes RFC2119 Language.]

This section covers how DetNet MPLS flows operate over an IEEE 802.1 TSN sub-network. Figure 6 illustrates such a scenario, where two MPLS (DetNet) nodes are interconnected by a TSN sub-network. Node-1 is single homed and Node-2 is dual-homed. MPLS nodes can be (1) DetNet MPLS End System, (2) DetNet MPLS Edge or Relay node or (3) MPLS Transit node.

Note: in case of MPLS Transit node there is no DetNet Service sub-layer processing.



Note: * no service sub-layer required for transit nodes

Figure 6: DetNet Enabled MPLS Network Over a TSN Sub-Network

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. Furthermore IEEE 802.1CB [IEEE8021CB] defines frame replication and elimination functions for reliability that should prove both compatible with and useful to, DetNet networks. All these functions have to identify flows those require TSN treatment.

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 may need a CoS marking, such as the priority field of an IEEE Std 802.1Q VLAN tag, to give the packet proper service.

The challenge for MPLS DeNet flows is that the protocol interworking function defined in IEEE 802.1CB [IEEE8021CB] works only for IP flows. The aim of the protocol interworking function is to convert an ingress flow to use a specific multicast destination MAC address and VLAN, for example to direct the packets through a specific path

inside the bridged network. A similar interworking pair at the other end of the TSN sub-network would restore the packet to its original destination MAC address and VLAN.

As protocol interworking function defined in [IEEE8021CB] does not work for MPLS labeled flows, the DetNet MPLS nodes MUST ensure proper TSN sub-network specific Ethernet encapsulation of the DetNet MPLS packets. For a given TSN Stream (i.e., DetNet flow) an MPLS (DetNet) node MUST behave as a TSN-aware Talker or a Listener inside the TSN sub-network.

6.1. Mapping of TSN Stream ID and Sequence Number

TSN capable MPLS (DetNet) nodes are TSN-aware Talker/Listener as shown in Figure 7. MPLS (DetNet) node MUST provide the TSN sub-network specific Ethernet encapsulation over the link(s) towards the sub-network. An TSN-aware MPLS (DetNet) node MUST support the following TSN components:

1. For recognizing flows:
 - * Stream Identification (MPLS-flow-aware)
2. For FRER used inside the TSN domain, additionally:
 - * Sequencing function (MPLS-flow-aware)
 - * Sequence encode/decode function
3. For FRER when the node is a TSN replication or elimination point, additionally:
 - * Stream splitting function
 - * Individual recovery function

[Editor's note: Should we added here requirements regarding IEEE 802.1Q C-VLAN component?]

The Stream Identification and The Sequencing functions are slightly modified for frames passed down the protocol stack from the upper layers.

Stream Identification MUST pair MPLS flows and TSN Streams and encode that in data plane formats as well. The packet's stream_handle subparameter (see IEEE 802.1CB [IEEE8021CB]) inside the Talker/Listener is defined based on the Flow-ID used in the upper DetNet MPLS layer. Stream Identification function MUST encode Ethernet

header fields namely (1) the destination MAC-address, (2) the VLAN-ID and (3) priority parameters with TSN sub-network specific values. Encoding is provided for the frame passed down the stack from the upper layers.

The sequence generation function resides in the Sequencing function. It generates a sequence_number subparameter for each packet of a Stream passed down to the lower layers. Sequencing function MUST copy sequence information from the MPLS d-CW of the packet to the sequence_number subparameter for the frame passed down the stack from the upper layers.

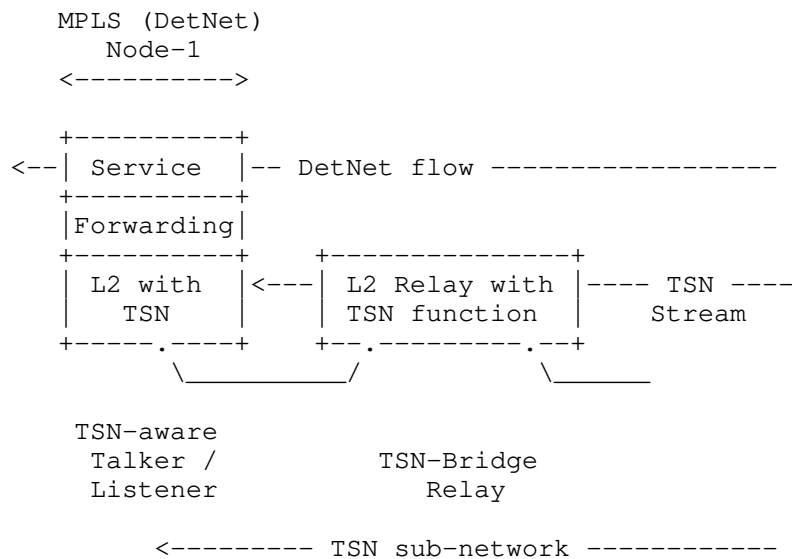


Figure 7: MPLS (DetNet) Node with TSN Functions

The Sequence encode/decode function MUST support the Redundancy tag (R-TAG) format as per Clause 7.8 of IEEE 802.1CB [IEEE8021CB].

6.2. TSN Usage of FRER

TSN Streams supporting DetNet flows may use Frame Replication and Elimination for Redundancy (FRER) [802.1CB] based on the loss service requirements of the TSN Stream, which is derived from the DetNet service requirements of the DetNet mapped flow. The specific operation of FRER is not modified by the use of DetNet and follows IEEE 802.1CB [IEEE8021CB].

FRER function and the provided service recovery is available only within the TSN sub-network however as the Stream-ID and the TSN

sequence number are paired with the MPLS flow parameters they can be combined with PREOF functions.

6.3. Procedures

[Editor's note: This section is TBD - covers required behavior of a TSN-aware DetNet node using a TSN underlay.]

6.4. Layer 2 Addressing and QoS Considerations

[Editor's NOTE: review and simplify this section. May overlap with previous sections.]

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 IP encapsulations.

7. Management and Control Considerations

[Editor's note: This section is TBD Covers Creation, mapping, removal of TSN Stream IDs, related parameters and, when needed, configuration of FRER. Supported by management/control plane. SEE sections in removed text file.]

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, and the DetNet architecture [I-D.ietf-detnet-architecture] refers to these collectively as the 'Controller Plane'. This document therefore does not distinguish between information provided by distributed control plane protocols, e.g., RSVP-TE [RFC3209] and [RFC3473], or by centralized network management mechanisms, e.g., RestConf [RFC8040], YANG [RFC7950], and the Path Computation Element Communication Protocol (PCEP) [I-D.ietf-pce-pcep-extension-for-pce-controller] or any combination thereof. Specific considerations and requirements for the DetNet Controller Plane are discussed below.

8. 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.

9. IANA Considerations

This document makes no IANA requests.

10. Acknowledgements

Thanks for Norman Finn and Lou Berger for their comments and contributions.

11. References

11.1. Normative References

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

- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <<https://www.rfc-editor.org/info/rfc3270>>.
- [RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <<https://www.rfc-editor.org/info/rfc3443>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [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>>.
- [RFC5085] Nadeau, T., Ed. and C. Pignataro, Ed., "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, DOI 10.17487/RFC5085, December 2007, <<https://www.rfc-editor.org/info/rfc5085>>.

- [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>>.
- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <<https://www.rfc-editor.org/info/rfc7510>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

11.2. Informative References

- [G.8275.1] International Telecommunication Union, "Precision time protocol telecom profile for phase/time synchronization with full timing support from the network", ITU-T G.8275.1/Y.1369.1 G.8275.1, June 2016, <<https://www.itu.int/rec/T-REC-G.8275.1/en>>.
- [G.8275.2] International Telecommunication Union, "Precision time protocol telecom profile for phase/time synchronization with partial timing support from the network", ITU-T G.8275.2/Y.1369.2 G.8275.2, June 2016, <<https://www.itu.int/rec/T-REC-G.8275.2/en>>.
- [I-D.ietf-detnet-architecture] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-12 (work in progress), March 2019.
- [I-D.ietf-detnet-dp-sol-ip] Korhonen, J., Varga, B., "DetNet IP Data Plane Encapsulation", 2018.
- [I-D.ietf-detnet-flow-information-model] Farkas, J., Varga, B., Cummings, R., and Y. Jiang, "DetNet Flow Information Model", draft-ietf-detnet-flow-information-model-03 (work in progress), March 2019.

- [I-D.ietf-pce-pcep-extension-for-pce-controller]
Zhao, Q., Li, Z., Negi, M., and C. Zhou, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) of LSPs", draft-ietf-pce-pcep-extension-for-pce-controller-01 (work in progress), February 2019.
- [I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-22 (work in progress), May 2019.
- [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>>.
- [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>>.

- [RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002, <<https://www.rfc-editor.org/info/rfc3272>>.
- [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>>.
- [RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <<https://www.rfc-editor.org/info/rfc4448>>.
- [RFC4872] Lang, J., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007, <<https://www.rfc-editor.org/info/rfc4872>>.
- [RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, DOI 10.17487/RFC4873, May 2007, <<https://www.rfc-editor.org/info/rfc4873>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <<https://www.rfc-editor.org/info/rfc5586>>.
- [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>>.

- [RFC5921] Bocci, M., Ed., Bryant, S., Ed., Frost, D., Ed., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, DOI 10.17487/RFC5921, July 2010, <<https://www.rfc-editor.org/info/rfc5921>>.
- [RFC6003] Papadimitriou, D., "Ethernet Traffic Parameters", RFC 6003, DOI 10.17487/RFC6003, October 2010, <<https://www.rfc-editor.org/info/rfc6003>>.
- [RFC6006] Zhao, Q., Ed., King, D., Ed., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 6006, DOI 10.17487/RFC6006, September 2010, <<https://www.rfc-editor.org/info/rfc6006>>.
- [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>>.
- [RFC6387] Takacs, A., Berger, L., Caviglia, D., Fedyk, D., and J. Meuric, "GMPLS Asymmetric Bandwidth Bidirectional Label Switched Paths (LSPs)", RFC 6387, DOI 10.17487/RFC6387, September 2011, <<https://www.rfc-editor.org/info/rfc6387>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<https://www.rfc-editor.org/info/rfc6790>>.
- [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>>.
- [RFC8169] Mirsky, G., Ruffini, S., Gray, E., Drake, J., Bryant, S., and A. Vainshtein, "Residence Time Measurement in MPLS Networks", RFC 8169, DOI 10.17487/RFC8169, May 2017, <<https://www.rfc-editor.org/info/rfc8169>>.

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 PREOF. The figure is subject to change depending on the further DT decisions on the label handling..]

Authors' Addresses

Balazs Varga (editor)
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

Janos Farkas
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: janos.farkas@ericsson.com

Andrew G. Malis
Huawei Technologies

Email: agmalis@gmail.com

Stewart Bryant
Huawei Technologies

Email: stewart.bryant@gmail.com

Jouni Korhonen

Email: jouni.nospam@gmail.com

DetNet
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2020

B. Varga, Ed.
J. Farkas
Ericsson
L. Berger
LabN Consulting, L.L.C.
A. Malis
S. Bryant
Futurewei Technologies
J. Korhonen
July 1, 2019

DetNet Data Plane: MPLS over UDP/IP
draft-ietf-detnet-mpls-over-udp-ip-01

Abstract

This document specifies the MPLS Deterministic Networking data plane operation and encapsulation over an IP network. The approach is modeled on the operation of MPLS and over UDP/IP 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 January 2, 2020.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Terms Used in This Document	3
2.2. Abbreviations	3
2.3. Requirements Language	4
3. DetNet MPLS Operation over DetNet	
IP PSNs	4
4. DetNet Data Plane Procedures	5
5. Management and Control Information Summary	6
6. Security Considerations	6
7. IANA Considerations	6
8. Acknowledgements	6
9. References	7
9.1. Normative References	7
9.2. Informative References	7
Authors' Addresses	8

1. Introduction

Deterministic Networking (DetNet) is a service that can be offered by a network to DetNet flows. DetNet provides these flows with a 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 use of the MPLS DetNet encapsulation over an IP network. The approach is modeled on the operation of MPLS over an IP Packet Switched Network (PSN) [RFC7510]. It maps the MPLS data plane encapsulation described in [I-D.ietf-detnet-mpls] to the DetNet IP data plane defined in [I-D.ietf-detnet-ip].

To carry DetNet flows with full functionality at the DetNet layer over an IP network, the following components are required (these are a subset of the requirements for MPLS encapsulation listed in [I-D.ietf-detnet-mpls]):

1. A method of identifying the DetNet flow group to the processing element.
2. A method of carrying the DetNet sequence number.

3. A method of distinguishing DetNet OAM packets from DetNet data packets.
4. A method of carrying queuing and forwarding indication.

These requirements are satisfied by the DetNet over MPLS Encapsulation described in [I-D.ietf-detnet-mpls] and they are partly satisfied by the DetNet IP data plane defined in [I-D.ietf-detnet-ip]

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 reader is assumed to be familiar with that document and its terminology.

2.2. Abbreviations

The following abbreviations are used in this document:

d-CW	A DetNet Control Word (d-CW) is used for sequencing and identifying duplicate packets of a DetNet flow at the DetNet service sub-layer.
DetNet	Deterministic Networking.
A-Label	A special case of an S-Label, whose properties are known only at the aggregation and deaggregation end-points.
F-Label	A Detnet "forwarding" label that identifies the LSP used to forward a DetNet flow across an MPLS PSN, e.g., a hop-by-hop label used between label switching routers.
MPLS	Multiprotocol Label Switching.
OAM	Operations, Administration, and Maintenance.
PEF	Packet Elimination Function.
POF	Packet Ordering Function.
PRF	Packet Replication Function.
PSN	Packet Switched Network.

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

2.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. DetNet MPLS Operation over DetNet IP PSNs

This document builds on the specification of MPLS over UDP defined in [RFC7510]. It may replace partly or entirely the F-Label(s) used in [I-D.ietf-detnet-mpls] with UDP and IP headers. The UDP and IP header information is used to identify DetNet flows, including member flows, per [I-D.ietf-detnet-ip]. The resulting encapsulation is shown in Figure 1. There may be zero or more F-label(s) between the S-label and the UDP header.

Note that this encapsulation works equally well with IPv4, IPv6, and IPv6-based Segment Routing [I-D.ietf-6man-segment-routing-header].

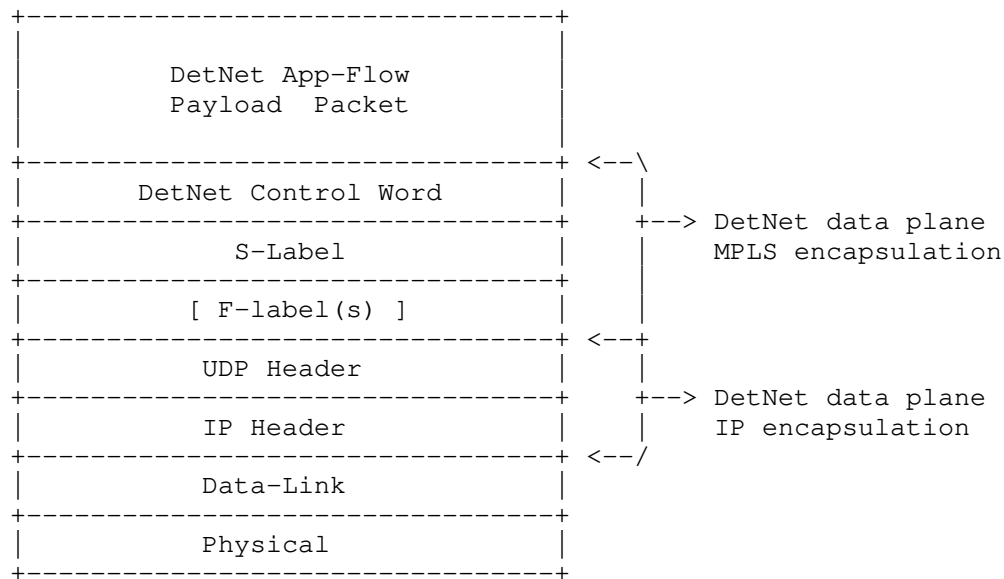


Figure 1: UDP/IP Encapsulation of DetNet MPLS

d-CW, S-Labels and zero or more F-Labels are used as defined in [I-D.ietf-detnet-mpls] and are not modified by this document. In case of aggregates the A-Label is treated as an S-Label and it too is not modified.

4. DetNet Data Plane Procedures

To support outgoing DetNet MPLS over UDP/IP encapsulation, an implementation MUST support the provisioning of UDP and IP header information in addition or in place of F-Label(s). Note, when PRF is performed at the MPLS service sub-layer, there will be multiple member flows, and each member flow will require the provisioning of their own UDP and IP header information. The headers for each outgoing packet MUST be formatted on the configuration information and as defined in [RFC7510], with one exception. Note that the UDP Source Port value MUST be set to uniquely identify the DetNet flow. The packet MUST then be handed as a DetNet IP packet, per [I-D.ietf-detnet-ip]. This includes QoS related traffic treatment.

To support receive processing an implementation MUST also support the provisioning of received UDP and IP header information. The provisioned information MUST be used to identify incoming app-flows based on the combination of S-Label and incoming encapsulation header

information. Normal receive processing as defined in [I-D.ietf-detnet-mpls], including PEF and POF, can then take place.

5. Management and Control Information Summary

The following summarizes the set of information that is needed to configure DetNet MPLS over UDP/IP:

- o Label information (S-label or F-label) to be mapped to UDP/IP flow. Note that a single S-Label can map to multiple sets of UDP/IP information when PREOF is used.
- o IPv4 and IPv6 source address field.
- o IPv4 and IPv6 destination address field.
- o IPv4 Type of Service and IPv6 Traffic Class Fields.
- o UDP Source Port.
- o UDP Destination Port.

This information MUST be provisioned per DetNet flow via configuration, e.g., via the controller or management plane.

It is the responsibility of the DetNet controller plane to properly provision both flow identification information and the flow specific resources needed to provide the traffic treatment needed to meet each flow's service requirements. This applies for aggregated and individual flows.

6. Security Considerations

The security considerations of DetNet in general are discussed in [I-D.ietf-detnet-architecture] and [I-D.ietf-detnet-security]. MPLS and IP specific security considerations are described in [I-D.ietf-detnet-mpls] and [I-D.ietf-detnet-ip]. This draft does not have additional security considerations.

7. IANA Considerations

This document makes no IANA requests.

8. Acknowledgements

The authors wish to thank Pat Thaler, Norman Finn, Loa Anderson, David Black, Rodney Cummings, Ethan Grossman, Tal Mizrahi, David

Mozes, Craig Gunther, George Swallow, Yuanlong Jiang and Carlos J. Bernardos for their various contributions to this work.

9. References

9.1. Normative References

- [I-D.ietf-detnet-ip]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP", draft-ietf-detnet-ip-00 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS", draft-ietf-detnet-mpls-00 (work in progress), May 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <<https://www.rfc-editor.org/info/rfc7510>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

9.2. Informative References

- [I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-21 (work in progress), June 2019.
- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-13 (work in progress), May 2019.

[I-D.ietf-detnet-security]

Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell, J., Austad, H., Stanton, K., and N. Finn, "Deterministic Networking (DetNet) Security Considerations", draft-ietf-detnet-security-04 (work in progress), March 2019.

Authors' Addresses

Balazs Varga (editor)
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

Janos Farkas
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: janos.farkas@ericsson.com

Lou Berger
LabN Consulting, L.L.C.

Email: lberger@labn.net

Andrew G. Malis
Futurewei Technologies

Email: agmalis@gmail.com

Stewart Bryant
Futurewei Technologies

Email: stewart.bryant@gmail.com

Jouni Korhonen

Email: jouni.nospam@gmail.com

DetNet
Internet-Draft
Intended status: Standards Track
Expires: November 6, 2019

B. Varga, Ed.
J. Farkas
Ericsson
A. Malis
S. Bryant
Huawei Technologies
J. Korhonen
May 5, 2019

DetNet Data Plane: IEEE 802.1 Time Sensitive Networking over MPLS
draft-ietf-detnet-tsn-vpn-over-mpls-00

Abstract

This document specifies the Deterministic Networking data plane when TSN networks interconnected over an 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 November 6, 2019.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
2. Terminology	2
2.1. Terms Used in This Document	2
2.2. Abbreviations	3
3. Requirements Language	4
4. IEEE 802.1 TSN Over DetNet MPLS Data Plane Scenario	4
5. DetNet MPLS Data Plane Considerations	6
5.1. End-System Specific Considerations	7
6. MPLS-Based DetNet Data Plane Solution	8
6.1. DetNet Over MPLS Encapsulation Components	8
6.2. TSN over MPLS Data Plane Encapsulation	9
6.2.1. Edge Node Processing	9
6.2.2. Layer 2 Addressing and QoS Considerations	10
7. Controller Plane (Management and Control) Considerations	11
8. Security Considerations	11
9. IANA Considerations	11
10. Acknowledgements	11
11. References	11
11.1. Normative References	11
11.2. Informative References	13
Appendix A. Example of TSN over DetNet Data Plane Operation . .	17
Authors' Addresses	17

1. Introduction

[Editor's note: Introduction to be made specific to TSN over DetNet scenario. Do we intend to cover both TSN over DetNet IP and TSN over DetNet MPLS? Or this document is limited to MPLS scenarios?].

2. Terminology

[Editor's note: text to be review what is really needed here.].

2.1. Terms Used in This Document

This document uses the terminology established in the DetNet architecture [I-D.ietf-detnet-architecture], and the reader is assumed to be familiar with that document and its terminology.

The following terminology is introduced in this document:

F-Label	A Detnet "forwarding" label that identifies the LSP used to forward a DetNet flow across an MPLS PSN, e.g., a hop-by-hop label used between label switching routers (LSR).
S-Label	A DetNet "service" label that is used between DetNet nodes that implement also the DetNet service sub-layer functions. An S-Label is also used to identify a DetNet flow at DetNet service sub-layer.
d-CW	A DetNet Control Word (d-CW) is used for sequencing and identifying duplicate packets of a DetNet flow at the DetNet service sub-layer.

2.2. Abbreviations

[Editor's note: text to be cleaned up].

The following abbreviations are used in this document:

AC	Attachment Circuit.
CE	Customer Edge equipment.
CoS	Class of Service.
CW	Control Word.
DetNet	Deterministic Networking.
DF	DetNet Flow.
DN-IWF	DetNet Inter-Working Function.
L2	Layer 2.
L2VPN	Layer 2 Virtual Private Network.
L3	Layer 3.
LSR	Label Switching Router.
MPLS	Multiprotocol Label Switching.
MPLS-TE	Multiprotocol Label Switching - Traffic Engineering.
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.
PEF	Packet Elimination Function.
PRF	Packet Replication Function.
PREOF	Packet Replication, Elimination and Ordering Functions.
POF	Packet Ordering Function.
PSN	Packet Switched Network.
PW	PseudoWire.
QoS	Quality of Service.
S-PE	Switching Provider Edge.
T-PE	Terminating Provider Edge.
TSN	Time-Sensitive Network.

3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. IEEE 802.1 TSN Over DetNet MPLS Data Plane Scenario

[Author's note: review required on his part.]

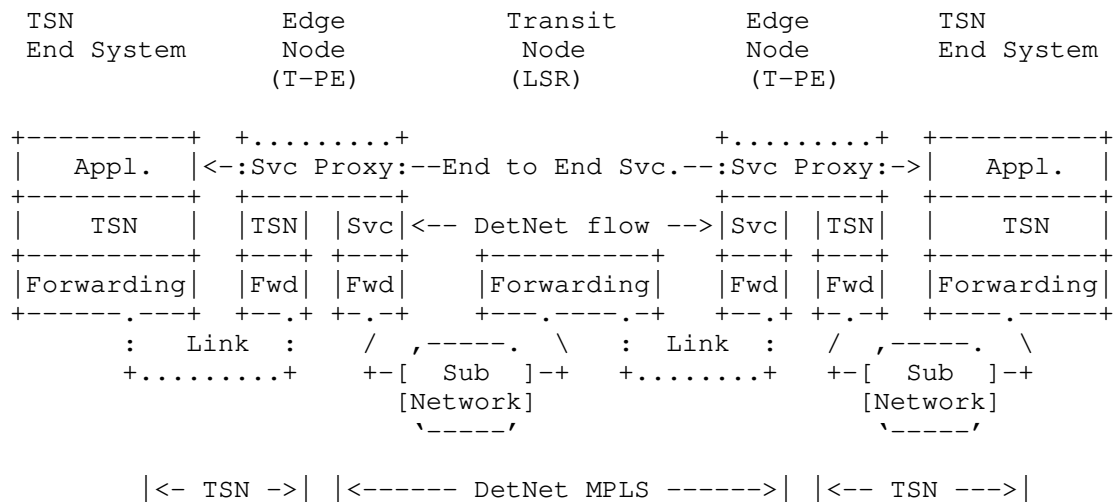


Figure 1: A TSN over DetNet MPLS Enabled Network

Figure 1 shows IEEE 802.1 TSN end stations operating over a TSN aware DetNet service running over an MPLS network. DetNet Edge Nodes sit at the boundary of a DetNet domain. They are responsible for mapping non-DetNet aware L2 traffic to DetNet services. They also support the imposition and disposition of the required DetNet encapsulation. These are functionally similar to pseudowire (PW) Terminating Provider Edge (T-PE) nodes which use MPLS-TE LSPs. In this example they understand and support IEEE 802.1 TSN and are able to map TSN flows into DetNet flows. The specifics of this operation are discussed later in this document.

Native TSN flow and DetNet MPLS flow differ not only by the additional MPLS specific encapsulation, but DetNet MPLS flows have on each DetNet node an associated DetNet specific data structure, what defines flow related characteristics and required forwarding functions. In this example, edge nodes provide a service proxy function that "associates" the DetNet flows and native flows at the edge of the DetNet domain. This ensures that the DN Flow is properly served at the Edge node (and inside the domain).

Figure 2 illustrates how DetNet can provide services for IEEE 802.1 TSN end systems, CE1 and CE2, over a DetNet enabled MPLS network. Edge nodes, E1 and E2, insert and remove required DetNet data plane encapsulation. The 'X' in the edge nodes and relay node, R1, represent a potential DetNet compound flow packet replication and elimination point. This conceptually parallels L2VPN services, and could leverage existing related solutions as discussed below.

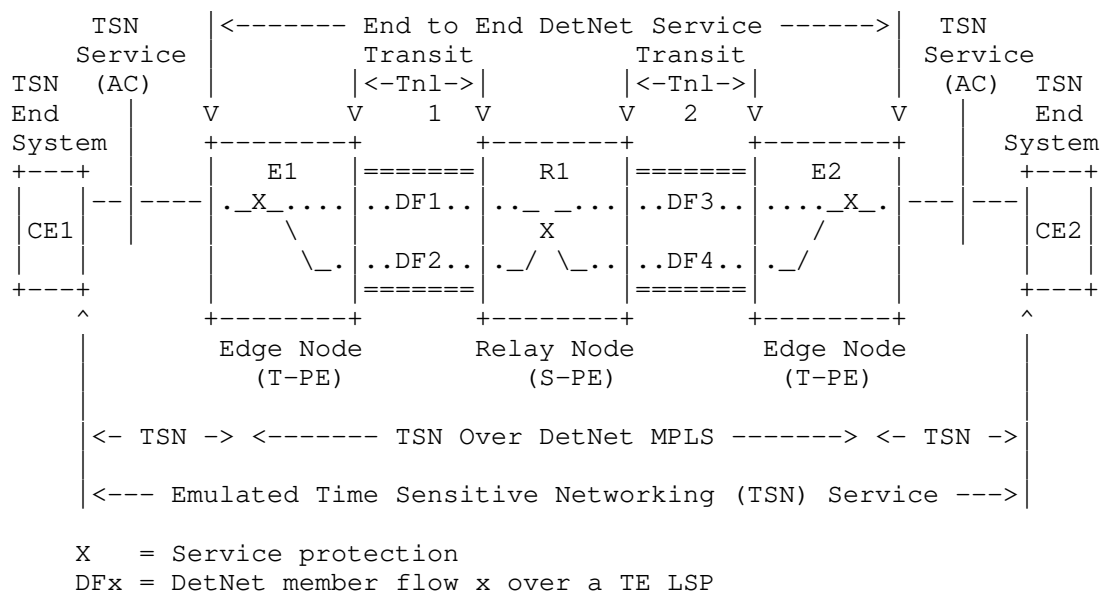


Figure 2: IEEE 802.1TSN Over DetNet

5. DetNet MPLS Data Plane Considerations

[Editor's note: Needs clean up, what is relevant for TSN over DetNet scenarios.]

This section provides informative considerations related to providing DetNet service to flows which are identified based on their header information. At a high level, the following are provided on a per flow basis:

Eliminating contention loss and jitter reduction:

Use of allocated resources (queuing, policing, shaping) to ensure that the congestion-related loss and latency/jitter requirements of a DetNet flow are met.

Explicit routes:

Use of a specific path for a flow. This limits misordering and bounds latency.

Service protection:

Which in the case of this document primarily relates to replication and elimination. Changing the explicit path after a failure is detected in order to restore delivery of the required DetNet service characteristics is also possible. Path changes, even in the case of failure recovery, can lead to the out of order delivery of data.

Load sharing:

Generally, distributing packets of the same DetNet flow over multiple paths is not recommended. Such load sharing, e.g., via ECMP or UCMP, impacts ordering and possibly jitter.

Troubleshooting:

For example, to support identification of misbehaving flows.

Recognize flow(s) for analytics:

For example, increase counters.

Correlate events with flows:

For example, unexpected loss.

The DetNet data plane also allows for the aggregation of DetNet flows, e.g., via MPLS hierarchical LSPs, to improved scaling. When DetNet flows are aggregated, transit nodes provide service to the aggregate and not on a per-DetNet flow basis. In this case, nodes performing aggregation will ensure that per-flow service requirements are achieved.

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 MPLS domain the DN functions use the d-CWs, S-Labels and F-Labels to provide DetNet services. However, an application may exchange further flow related parameters (e.g., time-stamp), which are not provided by DN functions.

Specifics related to non-MPLS DetNet end station behavior are outside the scope of this document. For example, details on support for DetNet IP data flows can be found in [I-D.ietf-detnet-dp-sol-ip]. This document is also useful for end stations that map IP flows to DetNet flows.

As a general rule, DetNet MPLS domains are capable of forwarding any DetNet MPLS flows and the DetNet domain does not mandate the end-system or edge system encapsulation format. Unless there is a proxy of some form present, end-systems peer with similar end-systems using the same application encapsulation format. For example, as shown in Figure 3, IP applications peer with IP applications and Ethernet L2VPN applications peer with Ethernet L2VPN applications.

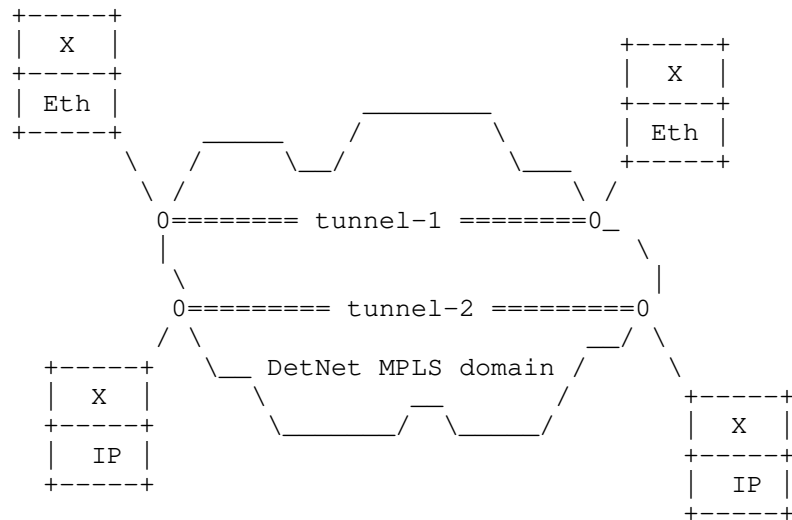


Figure 3: End-Systems and The DetNet MPLS Domain

6. MPLS-Based DetNet Data Plane Solution

[Editor's note: Needs clean up. Text should focus on Edge node related topics.].

6.1. DetNet Over MPLS Encapsulation Components

To carry DetNet over MPLS the following is required:

1. A method of identifying the MPLS payload type.
2. A method of identifying the DetNet flow group to the processing element.
3. A method of distinguishing DetNet OAM packets from DetNet data packets.
4. A method of carrying the DetNet sequence number.

5. A suitable LSP to deliver the packet to the egress PE.
6. A method of carrying queuing and forwarding indication.

In this design an MPLS service label (the S-Label), similar to a pseudowire (PW) label [RFC3985], is used to identify both the DetNet flow identity and the payload MPLS payload type satisfying (1) and (2) in the list above. OAM traffic discrimination happens through the use of the Associated Channel method described in [RFC4385]. The DetNet sequence number is carried in the DetNet Control word which carries the Data/OAM discriminator. To simplify implementation and to maximize interoperability two sequence number sizes are supported: a 16 bit sequence number and a 28 bit sequence number. The 16 bit sequence number is needed to support some types of legacy clients. The 28 bit sequence number is used in situations where it is necessary ensure that in high speed networks the sequence number space does not wrap whilst packets are in flight.

The LSP used to forward the DetNet packet may be of any type (MPLS-LDP, MPLS-TE, MPLS-TP [RFC5921], or MPLS-SR [I-D.ietf-spring-segment-routing-mpls]). The LSP (F-Label) label and/or the S-Label may be used to indicate the queue processing as well as the forwarding parameters. Note that the possible use of Penultimate Hop Popping (PHP) means that the only label in a received label stack may be the S-Label.

6.2. TSN over MPLS Data Plane Encapsulation

6.2.1. Edge Node Processing

An edge node is responsible for matching ingress packets to the service they require and encapsulating them accordingly. An edge node may participate in the packet replication and duplication elimination.

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.

The internal design of a relay node is out of scope of this document. However the reader's attention is drawn to the need to make any PREOF

state available to the packet processor(s) dealing with packets to which the PREOF functions must be applied, and to maintain that state is such as way that it is available to the packet processor operation on the next packet in the DetNet flow (which may be a duplicate, a late packet, or the next packet in sequence).

[Editor's note: I think the rest of this section belongs in a new "802.1 TSN (island Interconnect) over DetNet MPLS" section.]

This may be done in the DetNet layer, or where the native service processing (NSP) [RFC3985] is IEEE 802.1CB [IEEE8021CB] capable, the packet replication and duplicate elimination MAY entirely be done in the NSP, bypassing the DetNet flow encapsulation and logic entirely. This enables operating over unmodified implementations and deployments. The NSP approach works only between edge nodes and cannot make use of relay nodes.

The NSP approach is useful end to end tunnel and for for "island interconnect" scenarios. However, when there is a need to do PREOF in a middle of the network, such plain edge to edge operation is not sufficient.

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).

6.2.2. Layer 2 Addressing and QoS Considerations

[Editor's NOTE: review and simplify this section if possible.]

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 IP encapsulations.

7. Controller Plane (Management and Control) Considerations

[Editor's note: requires considerations related to TSN over DetNet.].

8. 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.

9. IANA Considerations

This document makes no IANA requests.

10. Acknowledgements

Thanks for Norman Finn and Lou Berger for their comments and contributions.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <<https://www.rfc-editor.org/info/rfc3270>>.
- [RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <<https://www.rfc-editor.org/info/rfc3443>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [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>>.

- [RFC5085] Nadeau, T., Ed. and C. Pignataro, Ed., "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, DOI 10.17487/RFC5085, December 2007, <<https://www.rfc-editor.org/info/rfc5085>>.
- [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>>.
- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <<https://www.rfc-editor.org/info/rfc7510>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

11.2. Informative References

- [G.8275.1] International Telecommunication Union, "Precision time protocol telecom profile for phase/time synchronization with full timing support from the network", ITU-T G.8275.1/Y.1369.1 G.8275.1, June 2016, <<https://www.itu.int/rec/T-REC-G.8275.1/en>>.
- [G.8275.2] International Telecommunication Union, "Precision time protocol telecom profile for phase/time synchronization with partial timing support from the network", ITU-T G.8275.2/Y.1369.2 G.8275.2, June 2016, <<https://www.itu.int/rec/T-REC-G.8275.2/en>>.
- [I-D.ietf-detnet-architecture] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-12 (work in progress), March 2019.
- [I-D.ietf-detnet-dp-sol-ip] Korhonen, J., Varga, B., "DetNet IP Data Plane Encapsulation", 2018.

- [I-D.ietf-detnet-flow-information-model]
Farkas, J., Varga, B., Cummings, R., and Y. Jiang, "DetNet Flow Information Model", draft-ietf-detnet-flow-information-model-03 (work in progress), March 2019.
- [I-D.ietf-pce-pcep-extension-for-pce-controller]
Zhao, Q., Li, Z., Negi, M., and C. Zhou, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) of LSPs", draft-ietf-pce-pcep-extension-for-pce-controller-01 (work in progress), February 2019.
- [I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-22 (work in progress), May 2019.
- [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>>.

- [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>>.
- [RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002, <<https://www.rfc-editor.org/info/rfc3272>>.
- [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>>.
- [RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <<https://www.rfc-editor.org/info/rfc4448>>.
- [RFC4872] Lang, J., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007, <<https://www.rfc-editor.org/info/rfc4872>>.
- [RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, DOI 10.17487/RFC4873, May 2007, <<https://www.rfc-editor.org/info/rfc4873>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <<https://www.rfc-editor.org/info/rfc5586>>.

- [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>>.
- [RFC5921] Bocci, M., Ed., Bryant, S., Ed., Frost, D., Ed., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, DOI 10.17487/RFC5921, July 2010, <<https://www.rfc-editor.org/info/rfc5921>>.
- [RFC6003] Papadimitriou, D., "Ethernet Traffic Parameters", RFC 6003, DOI 10.17487/RFC6003, October 2010, <<https://www.rfc-editor.org/info/rfc6003>>.
- [RFC6006] Zhao, Q., Ed., King, D., Ed., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 6006, DOI 10.17487/RFC6006, September 2010, <<https://www.rfc-editor.org/info/rfc6006>>.
- [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>>.
- [RFC6387] Takacs, A., Berger, L., Caviglia, D., Fedyk, D., and J. Meuric, "GMPLS Asymmetric Bandwidth Bidirectional Label Switched Paths (LSPs)", RFC 6387, DOI 10.17487/RFC6387, September 2011, <<https://www.rfc-editor.org/info/rfc6387>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<https://www.rfc-editor.org/info/rfc6790>>.
- [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>>.

[RFC8169] Mirsky, G., Ruffini, S., Gray, E., Drake, J., Bryant, S., and A. Vainshtein, "Residence Time Measurement in MPLS Networks", RFC 8169, DOI 10.17487/RFC8169, May 2017, <<https://www.rfc-editor.org/info/rfc8169>>.

Appendix A. Example of TSN over DetNet Data Plane Operation

[Editor's note: Add a simplified example of DetNet data plane and how labels etc work in the case of TSN over DetNet MPLS and utilizing e.g., PREOF.]

Authors' Addresses

Balazs Varga (editor)
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

Janos Farkas
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: janos.farkas@ericsson.com

Andrew G. Malis
Huawei Technologies

Email: agmalis@gmail.com

Stewart Bryant
Huawei Technologies

Email: stewart.bryant@gmail.com

Jouni Korhonen

Email: jouni.nospam@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

X. Geng
M. Chen
Huawei Technologies
Y. Ryoo
ETRI
Z. Li
China Mobile
R. Rahman
Cisco Systems
July 08, 2019

Deterministic Networking (DetNet) Configuration YANG Model
draft-ietf-detnet-yang-03

Abstract

This document contains the specification for Deterministic Networking flow configuration YANG Model. The model allows for provisioning of end-to-end DetNet service along the path without dependency on any signaling protocol.

The YANG module defined in this document conforms to the Network Management Datastore Architecture (NMDA).

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Terminologies	3
3. DetNet Configuration Model	3
3.1. DetNet Application Flow Configuration Attributes	3
3.2. DetNet Service Sub-layer Configuration Attributes	4
3.3. DetNet Forwarding Sub-layer Configuration Attributes	4
3.4. DetNet Sub-network Configurations Attributes	5
4. Overview of DetNet YANG Structure	5
4.1. DetNet YANG Structure Considerations	5
4.2. DetNet YANG Structure	6
4.2.1. YANG Structure of Application Flow	6
4.2.2. YANG Structure of DetNet Service Sub-layer	6
4.2.3. YANG Structure of DetNet Forwarding Sub-layer	8
4.2.4. YANG Structure of DetNet sub-network	9
5. DetNet Configuration YANG Model	10
6. Open Issues	34
7. IANA Considerations	34
8. Security Considerations	34
9. Acknowledgements	34
10. References	34
10.1. Normative References	34
10.2. Informative References	35
Authors' Addresses	37

1. Introduction

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

Information models for DetNet are categorized as flow models, service models and configuration models, which is defined in [I-D.ietf-detnet-flow-information-model].

Configuration models are used for DetNet topology discovery and DetNet flow configuration. This document defines a YANG model for DetNet flow configurations based on YANG data types and modeling language defined in [RFC6991] and [RFC7950]. A YANG model for topology discovery is defined in [I-D.ietf-detnet-topology-yang]. The DetNet configuration YANG model is designed for DetNet flow path establishment, flow status reporting, and DetNet functions configuration in order to achieve end-to-end bounded latency and zero congestion loss.

2. Terminologies

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

3. DetNet Configuration Model

DetNet flow configuration includes DetNet App-flow configuration, DetNet Service Sub-layer configuration, and DetNet Forwarding Sub-layer configuration and DetNet sub-network. The corresponding attributes used in different sub-layers are defined in Section 3.1, 3.2, 3.3, 3.4 respectively.

3.1. DetNet Application Flow Configuration Attributes

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

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

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

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

3.2. DetNet Service Sub-layer Configuration Attributes

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

- o DetNet flow identification, refers to Section 8.1.3 of [I-D.ietf-detnet-flow-information-model].
- o Service function indication, indicates which service function will be invoked at a DetNet edge, relay node or end station. (DetNet tunnel initialization or termination are default functions in DetNet service layer, so there is no need for explicit indication.)
- o Flow Rank, refers to Section 8.3 of [I-D.ietf-detnet-flow-information-model].
- o Service Rank, refers to Section 16 of [I-D.ietf-detnet-flow-information-model].
- o Service Sub-layer, refers to Section 4.5 and Section 4.6 of [I-D.ietf-detnet-mpls]
- o Forwarding Sub-layer, refers to Section 4.3 of [I-D.ietf-detnet-ip] and Section 4.5 and Section 4.6 of [I-D.ietf-detnet-mpls]

3.3. DetNet Forwarding Sub-layer Configuration Attributes

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

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

- o Traffic Specification, refers to Section 7.2 of [I-D.ietf-detnet-flow-information-model]. It may be used for bandwidth reservation, flow shaping, filtering and policing.
- o Explicit path, existing explicit route mechanisms can be reused. For example, if Segment Routing (SR) tunnel is used as the transport tunnel, the configuration is mainly at the ingress node

of the transport layer; if the static MPLS tunnel is used as the transport tunnel, the configurations need to be at every transit node along the path; for pure IP based transport tunnel, it's similar to the static MPLS case.

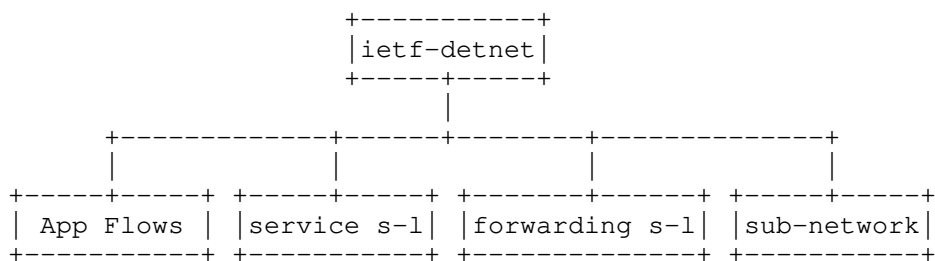
3.4. DetNet Sub-network Configurations Attributes

TBD

4. Overview of DetNet YANG Structure

4.1. DetNet YANG Structure Considerations

The picture shows that the general structure of the DetNet YANG Model:



There are four instances in DetNet YANG Model: App-flow instance, service sub-layer instance, forwarding sub-layer instance and sub-network instance, respectively corresponding to four parts of DetNet functions defined in section 3. In each instance, there are four elements: name, in-segments, out-segments and operations, which means:

- o Name: indicates the key value of the instance identification.
- o In-segments: indicates the key value of identification, e.g., Layer 2 App flow identification, Layer 3 App flow identification and DetNet flow identification.
- o Out-segments: indicates the information of DetNet processing(e.g., DetNet forwarding, DetNet header Encapsulation) and the mapping relationship to the lower sub-layer/sub-network.
- o Operations: indicates DetNet functions, e.g., DetNet forwarding functions, DetNet Service functions, DetNet Resource Reservation.

These elements are different when the technologies used for the specific instance is different. So this structure is abstract, which

allows for different technology specifics as defined in different data plane drafts.

4.2. DetNet YANG Structure

4.2.1. YANG Structure of Application Flow

The picture below shows that the general YANG structure of DetNet App-flow:

```

+--rw app-flow
|   +--rw operations
|   |   +--rw sequence-number
|   |   |   +--rw sequence-number-generation-type?   sequence-number-gener
ation-type
|   |   |   |   +--rw sequence-number-length?           uint8
|   |   +--rw in-segments
|   |   |   +--rw app-flow-type?           flow-type-ref
|   |   |   +--rw source-mac-address?      yang:mac-address
|   |   |   +--rw destination-mac-address? yang:mac-address
|   |   |   +--rw ethertype?               eth:ethertype
|   |   |   +--rw vlan-id?                 uint16
|   |   |   +--rw pcp?                     uint8
|   |   |   +--rw src-ipv4-prefix           inet:ipv4-prefix
|   |   |   +--rw dest-ipv4-prefix         inet:ipv4-prefix
|   |   |   +--rw protocol                 uint8
|   |   |   +--rw dscp?                    uint8
|   |   |   +--rw dscp-bitmask?            uint8
|   |   |   +--rw src-ipv6-prefix           inet:ipv6-prefix
|   |   |   +--rw dest-ipv6-prefix         inet:ipv6-prefix
|   |   |   +--rw next-header              uint8
|   |   |   +--rw traffic-class?           uint8
|   |   |   +--rw traffic-class-bitmask?   uint8
|   |   |   +--rw flow-label?              inet:ipv6-flow-label
|   |   |   +--rw flow-label-flag?         boolean
|   |   |   +--rw lower-source-port?       inet:port-number
|   |   |   +--rw upper-source-port?       inet:port-number
|   |   |   +--rw lower-destination-port?  inet:port-number
|   |   |   +--rw upper-destination-port?  inet:port-number
|   |   +--rw out-segments
|   |   |   +--rw detnet-service-sub-layer? lower-layer-ref

```

4.2.2. YANG Structure of DetNet Service Sub-layer

The picture shows that the general YANG structure of DetNet Service Sub-layer:

```

+--rw service-sub-layer
|   +--rw operations

```



```

    +--rw service-operation
    |   +--rw service-operation-type?    service-operation-ref
    +--rw service-protection
    |   +--rw service-protection-type?    service-protection-type
+--rw in-segments
    +--rw detnet-service-type?    flow-type-ref
    +--rw detnet-service-list* [detnet-service-index]
    |   +--rw detnet-service-index        uint8
    |   +--rw src-ipv4-prefix             inet:ipv4-prefix
    |   +--rw dest-ipv4-prefix            inet:ipv4-prefix
    |   +--rw protocol                    uint8
    |   +--rw dscp?                       uint8
    |   +--rw dscp-bitmask?               uint8
    |   +--rw src-ipv6-prefix             inet:ipv6-prefix
    |   +--rw dest-ipv6-prefix            inet:ipv6-prefix
    |   +--rw next-header                  uint8
    |   +--rw traffic-class?              uint8
    |   +--rw traffic-class-bitmask?      uint8
    |   +--rw flow-label?                  inet:ipv6-flow-label
    |   +--rw flow-label-flag?            boolean
    |   +--rw mpls-flow-identification
    |   |   +--rw platform-label-flag?    boolean
    |   |   +--rw non-platform-label-space
    |   |   |   +--rw incoming-interface? if:interface-ref
    |   |   |   +--rw non-platform-label-stack* [index]
    |   |   |   |   +--rw index        uint8
    |   |   |   |   +--rw label?      rt-type:mpls-label
    |   |   |   |   +--rw tc?        uint8
    |   |   +--rw platform-label-space
    |   |   |   +--rw label?      rt-type:mpls-label
    |   |   |   +--rw tc?        uint8
+--rw out-segments
    +--rw detnet-service-processing-type?    flow-type-ref
    +--rw detnet-service-encapsulation
    |   +--rw detnet-service-processing-list* [detnet-service-processi
ng-index]
    |   |   +--rw detnet-service-processing-index    uint32
    |   +--rw ip-flow
    |   |   +--rw ipv4-flow
    |   |   |   +--rw src-ipv4-address    inet:ipv4-address
    |   |   |   +--rw dest-ipv4-address    inet:ipv4-address
    |   |   |   +--rw protocol            uint8
    |   |   |   +--rw dscp?               uint8
    |   |   +--rw ipv6-flow
    |   |   |   +--rw src-ipv6-address    inet:ipv6-address
    |   |   |   +--rw dest-ipv6-address    inet:ipv6-address
    |   |   |   +--rw next-header          uint8
    |   |   |   +--rw traffic-class?      uint8
    |   |   |   +--rw flow-label?          inet:ipv6-flow-label

```

```

    +--rw l4-port-header
      +--rw source-port?      inet:port-number
      +--rw destination-port? inet:port-number
+--rw mpls-flow
  +--rw detnet-mpls-label-stack* [index]
    +--rw index                uint8
    +--rw label?               rt-type:mpls-label
    +--rw tc?                  uint8
    +--rw s-bit?               boolean
    +--rw d-cw-encapsulate-flag? boolean
+--rw detnet-forwarding-sub-layer-info
  +--rw detnet-forwarding-sub-layer? lower-layer-ref

```

4.2.3. YANG Structure of DetNet Forwarding Sub-layer

The picture shows that the general YANG structure of DetNet Forwarding Sub-layer:

```

+--rw forwarding-sub-layer
  +--rw operations
    +--rw forwarding-operation
      +--rw forwarding-operation-type? forwarding-operation-ref
    +--rw resource-allocate
      +--rw interval?                uint32
      +--rw max-packets-per-interval? uint32
      +--rw max-payload-size?        uint32
      +--rw average-packets-per-interval? uint32
      +--rw average-payload-size?    uint32
    +--rw qos
  +--rw in-segments
    +--rw detnet-forwarding-type?    flow-type-ref
    +--rw src-ipv4-prefix             inet:ipv4-prefix
    +--rw dest-ipv4-prefix            inet:ipv4-prefix
    +--rw protocol                    uint8
    +--rw dscp?                      uint8
    +--rw dscp-bitmask?              uint8
    +--rw src-ipv6-prefix             inet:ipv6-prefix
    +--rw dest-ipv6-prefix            inet:ipv6-prefix
    +--rw next-header                 uint8
    +--rw traffic-class?              uint8
    +--rw traffic-class-bitmask?     uint8
    +--rw flow-label?                inet:ipv6-flow-label
    +--rw flow-label-flag?            boolean
    +--rw mpls-flow-identification
      +--rw platform-label-flag?      boolean
    +--rw non-platform-label-space
      +--rw incoming-interface?       if:interface-ref
      +--rw non-platform-label-stack* [index]

```

```

      +--rw index      uint8
      +--rw label?     rt-type:mpls-label
      +--rw tc?        uint8
+--rw platform-label-space
      +--rw label?     rt-type:mpls-label
      +--rw tc?        uint8
+--rw out-segments
      +--rw detnet-forwarding-processing-type?  flow-type-ref
+--rw natively-detnet-forwarding
      +--rw ipv4-flow
      |   +--rw ipv4-next-hop-address?  inet:ipv4-address
      +--rw ipv6-flow
      |   +--rw ipv6-next-hop-address?  inet:ipv6-address
+--rw detnet-forwarding-encapsulation
      +--rw ip-flow
      |   +--rw ipv4-flow
      |   |   +--rw src-ipv4-address      inet:ipv4-address
      |   |   +--rw dest-ipv4-address     inet:ipv4-address
      |   |   +--rw protocol              uint8
      |   |   +--rw dscp?                 uint8
      |   +--rw ipv6-flow
      |   |   +--rw src-ipv6-address      inet:ipv6-address
      |   |   +--rw dest-ipv6-address     inet:ipv6-address
      |   |   +--rw next-header           uint8
      |   |   +--rw traffic-class?        uint8
      |   |   +--rw flow-label?           inet:ipv6-flow-label
      |   +--rw l4-port-header
      |   |   +--rw source-port?          inet:port-number
      |   |   +--rw destination-port?     inet:port-number
      +--rw mpls-flow
      |   +--rw detnet-mpls-label-stack* [index]
      |   |   +--rw index                  uint8
      |   |   +--rw label?                 rt-type:mpls-label
      |   |   +--rw tc?                    uint8
      |   |   +--rw s-bit?                 boolean
      |   |   +--rw d-cw-encapsulate-flag? boolean
      +--rw lower-layer-info
      |   +--rw lower-layer-type?  flow-type-ref
      |   +--rw interface
      |   |   +--rw outgoing-interface?  if:interface-ref
      +--rw sub-layer
      |   +--rw sub-layer?  lower-layer-ref

```

4.2.4. YANG Structure of DetNet sub-network

TBD

5. DetNet Configuration YANG Model

```
<CODE BEGINS> file ietf-detnet-config@20190324.yang
module ietf-detnet-config {
  namespace "urn:ietf:params:xml:ns:yang:ietf-detnet-config";
  prefix "ietf-detnet";

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

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

  import ietf-ethertypes {
    prefix "eth";
  }

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

  import ietf-interfaces {
    prefix "if";
  }

  organization "IETF DetNet Working Group";

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

               Janos Farkas
               <mailto:janos.farkas@ericsson.com>

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

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

    Editor:    Zhenqiang Li
               <mailto:lizhenqiang@chinamobile.com>

    Editor:    Reshad Rahman
```

<mailto:rrahman@cisco.com>

Editor: Yeoncheol Ryoo
<mailto:dbduscjf@etri.re.kr>;

```
description
  "This YANG module describes the parameters needed
  for DetNet flow configuration and flow status reporting";

revision 2019-03-24 {
  description "initial revision";
  reference "RFC XXXX: draft-ietf-detnet-yang-02";
}

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

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

identity copy-to-inner {
  base "ttl-action";
  description
    "Copy the TTL of the outer header
    to the inner header";
}

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

identity config-type {
  description
    "Base identity from which all configuration instances are derived";
}

identity App-flow {
  base "config-type";
  description
```

```
    "App-flow configuration";
}

identity service-sub-layer {
    base "config-type";
    description
        "A DetNet MPLS or IP service sub-layer configuration";
}

identity forwarding-sub-layer {
    base "config-type";
    description
        "A DetNet MPLS or IP forwarding sub-layer configuration";
}

identity tsn-sub-network {
    base "config-type";
    description
        "A TSN sub-net configuration";
}

identity flow-type {
    description
        "Base identity from which all flow type are derived";
}

identity ipv4 {
    base "flow-type";
    description
        "An IPv4 flow";
}

identity ipv6 {
    base "flow-type";
    description
        "An IPv6 flow";
}

identity mpls {
    base "flow-type";
    description
        "An MPLS flow";
}

identity l2 {
    base "flow-type";
    description
        "An MPLS flow";
}
```

```
}

identity tsn {
  base "flow-type";
  description
    "An MPLS flow";
}

identity service-operation {
  description
    "Base identity from which all service operation are derived";
}

identity service-initiation {
  base "service-operation";
  description
    "A DetNet service encapsulates";
}

identity service-termination {
  base "service-operation";
  description
    "A DetNet service decapsulates";
}

identity service-relay {
  base "service-operation";
  description
    "A DetNet service swap";
}

identity forwarding-operation {
  description
    "Base identity from which all data plane operation are derived";
}

identity natively-forward {
  base "forwarding-operation";
  description
    "A packet natively forward to lower-layer";
}

identity impose-and-forward {
  base "forwarding-operation";
  description
    "Impose a header(MPLS/IP) and forward to lower-layer";
}
```

```
identity pop-and-forward {
  base "forwarding-operation";
  description
    "Pop an identified packet header and forward to lower-layer";
}

identity pop-impose-and-forward {
  base "forwarding-operation";
  description
    "Pop an identified packet header, impose a one or more outgoing
    header and forward to lower-layer ";
}

identity swap-and-forward {
  base "forwarding-operation";
  description
    "Swap an identified packet header with outgoing header and forward
    to lower-layer ";
}

identity pop-and-lookup {
  base "forwarding-operation";
  description
    "Pop an identified packet header and perform a lookup";
}

identity label-space {
  description
    "Base identity from which all label space are derived";
}

identity platform-label {
  base "label-space";
  description
    "label allocated from the platform label space";
}

identity non-platform-label {
  base "label-space";
  description
    "label allocated from the non-platform label space";
}

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



```
}

typedef config-type-ref {
  type identityref {
    base "config-type";
  }
  description
    "config-type-ref";
}

typedef flow-type-ref {
  type identityref {
    base "flow-type";
  }
  description
    "flow-type-ref";
}

typedef service-operation-ref{
  type identityref {
    base "service-operation";
  }
  description
    "service-operation-ref";
}

typedef forwarding-operation-ref {
  type identityref {
    base "forwarding-operation";
  }
  description
    "forwarding-operation-ref";
}

typedef label-space-ref {
  type identityref {
    base "label-space";
  }
  description
    "label-space-ref";
}

typedef lower-layer-ref {
  type leafref {
    path "/ietf-detnet:detnet-config/ietf-detnet:detnet-config-list"
    + "/ietf-detnet:name";
  }
  description
```

```
    "lower-layer-ref";
}

typedef service-protection-type {
    type enumeration {
        enum none {
            description
                "no service protection provide";
        }
        enum replication {
            description
                "A Packet Replication Function (PRF) replicates
                DetNet flow packets and forwards them to one or
                more next hops in the DetNet domain. The number of
                packet copies sent to each next hop is a
                DetNet flow specific parameter at the node doing
                the replication. PRF can be implemented by an
                edge node, a relay node, or an end system";
        }
        enum elimination {
            description
                "A Packet Elimination Function (PEF) eliminates
                duplicate copies of packets to prevent excess
                packets flooding the network or duplicate
                packets being sent out of the DetNet domain.
                PEF can be implemented by an edge node, a relay
                node, or an end system.";
        }
        enum ordering {
            description
                "A Packet Ordering Function (POF) re-orders
                packets within a DetNet flow that are received
                out of order. This function can be implemented
                by an edge node, a relay node, or an end system.";
        }
        enum elimination-ordering {
            description
                "A combination of PEF and POF that can be
                implemented by an edge node, a relay node, or
                an end system.";
        }
        enum elimination-replication {
            description
                "A combination of PEF and PRF that can be
                implemented by an edge node, a relay node, or
                an end system";
        }
    }
}
```

```
enum elimination-ordering-replicaiton {
  description
    "A combination of PEF, POF and PRF that can be
    implemented by an edge node, a relay node, or
    an end system";
}
}
description
  "service-protection-type";
}

typedef sequence-number-generation-type {
  type enumeration {
    enum none {
      description
        "No sequence number generation function provide";
    }
    enum copy-from-app-flow {
      description
        "Copy the app-flow sequence number to the DetNet-flow";
    }
    enum generate-by-detnet-flow {
      description
        "Generate the sequence number by DetNet flow";
    }
  }
}
description
  "sequence-number-generation-type";
}

grouping l4-port-header {
  description
    "The TCP/UDP port(source/destination) information";
  leaf source-port {
    type inet:port-number;
    description
      "The source port number";
  }
  leaf destination-port {
    type inet:port-number;
    description
      "The destination port number";
  }
}

grouping ipv4-header {
  description
    "The IPv4 packet header information";
```

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

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

```
        description
            "The traffic class value of the header";
    }
    leaf flow-label {
        type inet:ipv6-flow-label;
        description
            "The flow label value of the header";
    }
}

grouping mpls-header {
    description
        "The MPLS packet header information";
    leaf label {
        type rt-type:mpls-label;
        description
            "The label value of the MPLS header";
    }
    leaf tc {
        type uint8;
        description
            "The traffic class value of the MPLS header";
    }
    leaf s-bit {
        type boolean;
        description
            "The s-bit value of the MPLS header,
            which indicates the bottom of the label shack";
    }
    leaf d-cw-encapsulate-flag {
        type boolean;
        description
            "the indication of whether D-CW is encapsulated or not,
            when the D-CW is encapsulated, the sequence number is
            determined by sequence generation type";
    }
}

grouping l2-header {
    description
        "The Ethernet or TSN packet header information";
    leaf source-mac-address {
        type yang:mac-address;
        description
            "The source MAC address value of the ethernet header";
    }
    leaf destination-mac-address {
        type yang:mac-address;
    }
}
```

```
        description
            "The destination MAC address value of the ethernet header";
    }
    leaf ethertype {
        type eth:ethertype;
        description
            "The ethernet packet type value of the ethernet header";
    }
    leaf vlan-id {
        type uint16;
        description
            "The Vlan value of the ethernet header";
    }
    leaf pcp {
        type uint8;
        description
            "The priority value of the ethernet header";
    }
}

grouping l4-port-identification {
    description
        "The TCP/UDP port(source/destination) identification information";
    leaf lower-source-port {
        type inet:port-number;
        description
            "The lower source port number of the source port range";
    }
    leaf upper-source-port {
        type inet:port-number;
        description
            "The upper source port number of the source port range";
    }
    leaf lower-destination-port {
        type inet:port-number;
        description
            "The lower destination port number or the destination port range";
    }
    leaf upper-destination-port {
        type inet:port-number;
        description
            "The upper destination port number of the destination port range";
    }
}

grouping ipv4-flow-identification {
    description
        "The IPv4 packet header identification information";
```

```
leaf src-ipv4-prefix {
    type inet:ipv4-prefix;
    mandatory true;
    description
        "The source IP address of the header";
}
leaf dest-ipv4-prefix {
    type inet:ipv4-prefix;
    mandatory true;
    description
        "The destination IP address of the header";
}
leaf protocol {
    type uint8;
    mandatory true;
    description
        "The protocol of the header";
}
leaf dscp {
    type uint8;
    description
        "The DSCP field of the header";
}
leaf dscp-bitmask {
    type uint8;
    description
        "The bitmask value that determines whether to use
        the DSCP(IPv4) value for flow identification or not";
}
}

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

```
    mandatory true;
    description
        "The next header of the IPv6 header";
}
leaf traffic-class {
    type uint8;
    description
        "The traffic class value of the header";
}
leaf traffic-class-bitmask {
    type uint8;
    description
        "The bitmask value that determines whether to use
        the Traffic class(IPv6) value for flow identification or not";
}
leaf flow-label {
    type inet:ipv6-flow-label;
    description
        "The flow label value of the header";
}
leaf flow-label-flag {
    type boolean;
    description
        "The flag that determines whether to use
        the Flow Label value for flow identification or not";
}
}

grouping mpls-flow-identification {
    description
        "The MPLS packet header identification information";
    leaf label {
        type rt-type:mpls-label;
        description
            "The label value of the MPLS header";
    }
    leaf tc {
        type uint8;
        description
            "The traffic class value of the MPLS header";
    }
}

grouping l2-flow-identification {
    description
        "The Ethernet or TSN packet header identification information";
    leaf source-mac-address {
        type yang:mac-address;
    }
}
```



```
        description
            "The source MAC address value of the ethernet header";
    }
    leaf destination-mac-address {
        type yang:mac-address;
        description
            "The destination MAC address value of the ethernet header";
    }
    leaf ethertype {
        type eth:ethertype;
        description
            "The ethernet packet type value of the ethernet header";
    }
    leaf vlan-id {
        type uint16;
        description
            "The Vlan value of the ethernet header";
    }
    leaf pcp {
        type uint8;
        description
            "The priority value of the ethernet header";
    }
}

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

```
    type uint32;
    description
        "The maximum payload size that the source
        will transmit.";
}
leaf average-packets-per-interval {
    type uint32;
    description
        "The average number of packets that the
        source will transmit in one Interval";
}
leaf average-payload-size {
    type uint32;
    description
        "The average payload size that the
        source will transmit.";
}
}

container detnet-config {
    description
        "DetNet configurations";
    leaf node-id {
        type yang:dotted-quad;
        description
            "A 32-bit number in the form of a dotted quad that is used by
            identifying a DetNet node";
    }
    list detnet-config-list {
        key "name";
        description
            "list of the DetNet configurations";
        leaf name {
            type string;
            description
                "The name to identify the DetNet configuration";
        }
        leaf config-type {
            type config-type-ref;
            description
                "The DetNet configuration type such as a App-flow, service
                sub-layer, forwarding sub-layer, and TSN sub-network";
        }
        container App-flow {
            when "../config-type = 'ietf-detnet:App-flow'";
            description
                "The DetNet App-flow configuration";
            container operations {
```

```
    description "operations";
    container sequence-number {
        description "The DetNet sequence number operations grouping";
        leaf sequence-number-generation-type {
            type sequence-number-generation-type;
            description "The DetNet sequence number generation type";
        }
        leaf sequence-number-length {
            type uint8;
            description
                "The DetNet sequence number length";
        }
    }
}
container in-segments {
    description "The App-flow identification information";
    leaf app-flow-type {
        type flow-type-ref;
        description
            "The App-flow type such as a L2, IPv4, and IPv6";
    }
    uses l2-flow-identification {
        when "app-flow-type = 'ietf-detnet:tsn' or 'ietf-detnet:l2'";
    }
    uses ipv4-flow-identification {
        when "app-flow-type = 'ietf-detnet:ipv4'";
    }
    uses ipv6-flow-identification {
        when "app-flow-type = 'ietf-detnet:ipv6'";
    }
    uses l4-port-identification {
        when "app-flow-type = 'ietf-detnet:ipv6' or 'ietf-detnet:ipv4'";
        or 'ietf-detnet:ipv4'";
    }
}
container out-segments {
    description
        "The DetNet service information associated with this App-flow";
    leaf detnet-service-sub-layer {
        type lower-layer-ref;
        description "Specify associated service sub-layer";
    }
}
}
container service-sub-layer {
    when "../config-type = 'ietf-detnet:service-sub-layer'";
    description "The DetNet service sub-layer configuration";
    container operations {
```

```

description
  "The DetNet service sub-layer operations grouping";
container service-operation {
  description "The DetNet service operations grouping";
  leaf service-operation-type {
    type service-operation-ref;
    description
      "The DetNet service operations type such as DetNet
      service initiation, termination, and relay";
  }
}
container service-protection {
  description
    "The DetNet service protection operations grouping";
  leaf service-protection-type {
    type service-protection-type;
    description
      "The DetNet service protection type such as PRF, PEF, PEOF,
      PERF, and PEORF";
  }
}
}
container in-segments {
  when "../operations/service-operation"
  + "/service-operation-type != 'service-initiation'";
  description
    "DetNet service identification information";
  leaf detnet-service-type {
    type flow-type-ref;
    description
      "incoming DetNet service flow type";
  }
  list detnet-service-list {
    key "detnet-service-index";
    description
      "Incoming DetNet member flows or a compound flow";
    leaf detnet-service-index {
      type uint8;
      description
        "Incoming DetNet service index";
    }
    uses ipv4-flow-identification {
when "../detnet-service-type = 'ietf-detnet:ipv4'";
    }
    uses ipv6-flow-identification {
when "../detnet-service-type = 'ietf-detnet:ipv6'";
    }
    container mpls-flow-identification {

```

```

        when "../detnet-service-type = 'ietf-detnet:mpls'";
        description
            "MPLS type DetNet service identification";
    leaf label-space {
        type label-space-ref;
        description
            "Indicate the incoming MPLS label is associated with
            platform label space or not";
    }
    container non-platform-label-space {
when "../label-space = 'ietf-detnet:non-platform-label'";
        description
            "MPLS label is associated with non-platform label space,
            all of the F-labels and incoming interface information was
            used for identification";
        leaf incoming-interface {
            type if:interface-ref;
            description
                "DetNet service incoming interface information";
        }
        list non-platform-label-stack {
            key "index";
            description
                "All of the label information from the outer label
                to the current label";
            leaf index {
                type uint8;
                description
                    "Index of the labels stack";
            }
            uses mpls-flow-identification;
        }
    }
    container platform-label-space {
when "../label-space = 'ietf-detnet:platform-label'";
        description
            "MPLS label is associated with platform label space, only
            the F-label is used for identification";
        uses mpls-flow-identification;
    }
}

}

}
container out-segments {
    when "../operations/service-operation"
    + "/service-operation-type != 'service-termination'";
    description
        "DetNet Service outgoing processing grouping";
}

```

```
leaf detnet-service-processing-type {
    type flow-type-ref;
description
    "Outgoing DetNet service flow type";
}
container detnet-service-encapsulation {
description
    "DetNet service encapsulation information";
list detnet-service-processing-list {
    key "detnet-service-processing-index";
description
    "The list of single or multiple outgoing DetNet service(s)";
leaf detnet-service-processing-index {
    type uint32;
description "Outgoing segment entry";
}
container ip-flow {
    when "../.../detnet-service-processing-type ="
    + "'ietf-detnet:ipv4' or 'ietf-detnet:ipv6'";
description
    "IP type DetNet flow(s) encapsulation information";
container ipv4-flow {
    when "../.../detnet-service-processing-type ="
    + "'ietf-detnet:ipv4'";
description
    "IPv4 packet header encapsulation information";
    uses ipv4-header;
}
container ipv6-flow {
    when "../.../detnet-service-processing-type ="
    + "'ietf-detnet:ipv6'";
description
    "IPv6 packet header encapsulation information";
    uses ipv6-header;
}
container l4-port-header {
description
    "TCP/UDP source or destination port number";
    uses l4-port-header;
}
}
container mpls-flow {
    when "../.../detnet-service-processing-type ="
    + "'ietf-detnet:mpls'";
description
    "MPLS type DetNet flow(s) encapsulation information";
list detnet-mpls-label-stack {
    key "index";
```

```

        description
            "The list of MPLS labels stack for swap or encapsulation";
        leaf index {
            type uint8;
            description "Index of the labels stack";
        }
        uses mpls-header;
    }
}
container detnet-forwarding-sub-layer-info {
    description
        "The forwarding sub-layer information that associated with
        this DetNet service sub-layer";
    leaf detnet-forwarding-sub-layer {
        type lower-layer-ref;
        description
            "Specify associated forwarding sub-layer";
    }
}
}
}
}
}
}
container forwarding-sub-layer {
    when "../config-type = 'ietf-detnet:forwarding-sub-layer'";
    description
        "The DetNet forwarding sub-layer configuration";
    container operations {
        description
            "The DetNet forwarding sub-layer operations grouping";
        container forwarding-operation {
            description
                "DetNet forwarding function operations grouping";
            leaf forwarding-operation-type {
                type forwarding-operation-ref;
                description
                    "DetNet forwarding operation type such as
                    natively forward, impose and forward, pop and forward,
                    pop and impose and forward, swap and forward,
                    and pop and lookup";
            }
        }
    }
    container resource-allocate {
        description
            "resource-allocation function operations grouping";
        uses traffic-specification;
    }
    container qos {

```

```
        description
            "QoS function operations grouping";
    }
}
container in-segments {
    description
        "DetNet forwarding sub-layer packet identification information";
    leaf detnet-forwarding-type {
        type flow-type-ref;
        description
            "incoming DetNet forwarding packet type";
    }
    uses ipv4-flow-identification {
when "detnet-forwarding-type = 'ietf-detnet:ipv4'";
    }
    uses ipv6-flow-identification {
when "detnet-forwarding-type = 'ietf-detnet:ipv6'";
    }
    container mpls-flow-identification {
        when "../detnet-forwarding-type = 'ietf-detnet:mpls'";
        description
            "MPLS type identification information";
    }
    leaf label-space {
        type label-space-ref;
        description
            "Indicate the incoming MPLS label is associated with platform
            label space or not";
    }
    container non-platform-label-space {
when "../label-space = 'ietf-detnet:non-platform-label'";
        description
            "MPLS label is associated with non-platform label space,
            all of the F-labels and incoming interface information was
            used for identification";
        leaf incoming-interface {
            type if:interface-ref;
            description
                "The information of DetNet forwarding packet incoming
                interface";
        }
    }
    list non-platform-label-stack {
        key "index";
        description
            "All of the label information from the outer label to
            the current label";
        leaf index {
            type uint8;
            description
```



```

        "index number 0 indicate last inner label";
    }
    uses mpls-flow-identification;
}
}
container platform-label-space {
    when "../label-space = 'ietf-detnet:platform-label'";
    description
        "MPLS label is associated with platform label space, only
        the F-label is used for identification";
    uses mpls-flow-identification;
}
}
container out-segments {
    description
        "DetNet forwarding sub-layer packet processing information";
    leaf detnet-forwarding-processing-type {
        type flow-type-ref;
        description
            "outgoing DetNet forwarding packet type";
    }
    container natively-detnet-forwarding {
        when "../operations/forwarding-operation"
        + " /forwarding-operation-type = 'natively-forwarding'";
        description
            "Packet forwarding processing information";
        container ipv4-flow {
            when "../detnet-forwarding-processing-type ="
            + "'ietf-detnet:ipv4'";
            description
                "IPv4 type packet forwarding information";
            leaf ipv4-next-hop-address {
                type inet:ipv4-address;
                description
                    "IPv4 type Next hop IP address";
            }
        }
        container ipv6-flow {
            when "../detnet-forwarding-processing-type ="
            + "'ietf-detnet:ipv6'";
            description
                "IPv6 type packet forwarding information";
            leaf ipv6-next-hop-address {
                type inet:ipv6-address;
                description
                    "IPv6 type Next hop IP address";
            }
        }
    }
}

```

```
    }
  }
  container detnet-forwarding-encapsulation {
    when "../operations/forwarding-operation"
    + "/forwarding-operation-type != 'natively-forward'";
    description
      "Packet encapsulation information";
    container ip-flow {
      when "../detnet-forwarding-processing-type = "
      + "'ietf-detnet:ipv4' or 'ietf-detnet:ipv6'";
      description
        "The IP type DetNet flow(s) encapsulation information";
      container ipv4-flow {
        when "../../../../detnet-forwarding-processing-type = "
        + "'ietf-detnet:ipv4'";
        description
          "IPv4 packet header encapsulation information";
        uses ipv4-header;
      }
      container ipv6-flow {
        when "../../../../detnet-forwarding-processing-type = "
        + "'ietf-detnet:ipv6'";
        description
          "IPv6 packet header encapsulation information";
        uses ipv6-header;
      }
    }
    container l4-port-header {
      description
        "TCP/UDP source or destination port number";
      uses l4-port-header;
    }
  }
  container mpls-flow {
    when "../detnet-forwarding-processing-type = "
    + "'ietf-detnet:mpls'";
    description
      "MPLS label encapsulation information";
    list detnet-mpls-label-stack {
      key "index";
      description
        "The list of MPLS labels stack for swap or encapsulation";
      leaf index {
        type uint8;
        description
          "Index of the labels stack";
      }
      uses mpls-header;
    }
  }
```

```

    }
    container lower-layer-info {
      description
        "The lower-layer information associated with
         this forwarding sub-layer";
      leaf lower-layer-type {
        type flow-type-ref;
        description
          "indicate lower-layer type";
      }
      container interface {
        when "../lower-layer-type = 'ietf-detnet:l2'";
        description
          "indicate the lower-layer is the outgoing interface";
        leaf outgoing-interface {
          type if:interface-ref;
          description
            "Outgoing interface";
        }
      }
      container sub-layer {
        when "../lower-layer-type != 'ietf-detnet:l2'";
        description
          "indicate the lower-layer is some of the DetNet sub-layer
           or TSN sub-network";
        leaf sub-layer {
          type lower-layer-ref;
          description
            "Specify associated DetNet sub-layer or TSN sub-network";
        }
      }
    }
  }
}

container sub-network {
  when "../config-type = 'ietf-detnet:tsn-sub-network'";
  description
    "sub-network";
}
}
}
<CODE ENDS>
```

6. Open Issues

There are some open issues that are still under discussion:

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

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

7. IANA Considerations

This document makes no request of IANA.

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

8. Security Considerations

<TBD>

9. Acknowledgements

10. References

10.1. Normative References

- [I-D.finn-detnet-bounded-latency]
Finn, N., Boudec, J., Mohammadpour, E., Zhang, J., Varga, B., and J. Farkas, "DetNet Bounded Latency", draft-finn-detnet-bounded-latency-04 (work in progress), June 2019.
- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-13 (work in progress), May 2019.
- [I-D.ietf-detnet-flow-information-model]
Farkas, J., Varga, B., Cummings, R., and Y. Jiang, "DetNet Flow Information Model", draft-ietf-detnet-flow-information-model-03 (work in progress), March 2019.

- [I-D.ietf-detnet-ip]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP", draft-ietf-detnet-ip-01 (work in progress), July 2019.
- [I-D.ietf-detnet-mpls]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS", draft-ietf-detnet-mpls-00 (work in progress), May 2019.
- [I-D.ietf-detnet-topology-yang]
Geng, X., Chen, M., Li, Z., and R. Rahman, "Deterministic Networking (DetNet) Topology YANG Model", draft-ietf-detnet-topology-yang-00 (work in progress), January 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

10.2. Informative References

- [I-D.geng-detnet-info-distribution]
Geng, X., Chen, M., and Z. Li, "IGP-TE Extensions for DetNet Information Distribution", draft-geng-detnet-info-distribution-03 (work in progress), October 2018.
- [I-D.ietf-detnet-use-cases]
Grossman, E., "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-20 (work in progress), December 2018.
- [I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-21 (work in progress), April 2019.

- [I-D.ietf-teas-yang-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-22 (work in progress), June 2019.
- [I-D.thubert-tsvwg-detnet-transport]
Thubert, P., "A Transport Layer for Deterministic Networks", draft-thubert-tsvwg-detnet-transport-01 (work in progress), October 2017.
- [I-D.varga-detnet-service-model]
Varga, B. and J. Farkas, "DetNet Service Model", draft-varga-detnet-service-model-02 (work in progress), May 2017.
- [IEEE802.1CB]
IEEE, "IEEE, "Frame Replication and Elimination for Reliability (IEEE Draft P802.1CB)", 2017, <<http://www.ieee802.org/1/files/private/cb-drafts/>>.", 2016.
- [IEEE802.1Q-2014]
"IEEE, "IEEE Std 802.1Q Bridges and Bridged Networks", 2014, <<http://ieeexplore.ieee.org/document/6991462/>>.", 2014.
- [IEEE802.1Qbu]
IEEE, "IEEE, "IEEE Std 802.1Qbu Bridges and Bridged Networks - Amendment 26: Frame Preemption", 2016, <<http://ieeexplore.ieee.org/document/7553415/>>.", 2016.
- [IEEE802.1Qbv]
"IEEE, "IEEE Std 802.1Qbu Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", 2015, <<http://ieeexplore.ieee.org/document/7572858/>>.", 2016.
- [IEEE802.1Qcc]
IEEE, "IEEE, "Stream Reservation Protocol (SRP) Enhancements and Performance Improvements (IEEE Draft P802.1Qcc)", 2017, <<http://www.ieee802.org/1/files/private/cc-drafts/>>.".
- [IEEE802.1Qch]
IEEE, "IEEE, "Cyclic Queuing and Forwarding (IEEE Draft P802.1Qch)", 2017, <<http://www.ieee802.org/1/files/private/ch-drafts/>>.", 2016.

- [IEEE802.1Qci]
IEEE, "IEEE, "Per-Stream Filtering and Policing (IEEE Draft P802.1Qci)", 2016,
<<http://www.ieee802.org/1/files/private/ci-drafts/>>.", 2016.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

Authors' Addresses

Xuesong Geng
Huawei Technologies

Email: gengxuesong@huawei.com

Mach(Guoyi) Chen
Huawei Technologies

Email: mach.chen@huawei.com

Yeoncheol Ryoo
ETRI

Email: dbduscjf@etri.re.kr

Zhenqiang Li
China Mobile

Email: lizhenqiang@chinamobile.com

Reshad Rahman
Cisco Systems

Email: rrahman@cisco.com

DetNet Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 19, 2019

Y. Jiang
N. Finn
Huawei Technologies
J. Ryoo
ETRI
B. Varga
Ericsson
L. Geng
China Mobile
June 17, 2019

Deterministic Networking Application in Ring Topologies
draft-jiang-detnet-ring-04

Abstract

Deterministic Networking (DetNet) provides a capability to carry data flows for real-time applications with extremely low data loss rates and bounded latency. This document describes how DetNet can be used in ring topologies to support Point-to-Point (P2P) and Point-to-Multipoint (P2MP) real-time services.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 19, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions Used in This Document	3
3. Abbreviations	3
4. P2P DetNet Ring	4
4.1. DetNet applications on a single ring for P2P traffic . .	4
4.2. Implementation implications of a DetNet ring for P2P traffic	5
5. P2MP DetNet Ring	5
5.1. DetNet applications on a single ring for P2MP traffic . .	5
5.2. Section LSPs as underlay (service sub-layer replication)	6
5.3. P2MP LSP tunnels as underlay (forwarding sub-layer replication)	7
6. DetNet Ring Interconnections	8
6.1. Single node interconnection	8
6.2. Dual node interconnection	9
6.2.1. Dual node interconnection for P2P traffic	9
6.2.2. Dual node interconnection for P2MP traffic using section LSP	10
6.2.3. Dual node interconnection for P2MP traffic using P2MP LSP	11
7. Resource Reservation	11
8. IANA Considerations	11
9. Security Considerations	11
10. References	12
10.1. Normative References	12
10.2. Informative References	12
Authors' Addresses	13

1. Introduction

The overall architecture for Deterministic Networking (DetNet), which provides a capability to carry specified unicast or multicast data flows for real-time applications with extremely low data loss rates and bounded latency, is specified in [I-D.ietf-detnet-architecture], and the generic data plane framework, which is common to any DetNet data plane implementations, is provided at [I-D.ietf-detnet-data-plane-framework]. In addition to the DetNet architecture documents, RFC 8578 [RFC8578] outlines several DetNet use cases where multicast capability is needed. If a multicast

service replicates all of its packets from the source (as a traditional Virtual Private LAN Service (VPLS) does), the requirements of deterministic delay and high availability for all these replicated packets will pose a great challenge to the DetNet network.

Ring topologies have been very popular and widely deployed in network arrangements for various transport networks, such as Synchronous Digital Hierarchy, Synchronous Optical Network, Optical Transport Network, and Ethernet. For Multi-Protocol Label Switching - Transport Profile (MPLS-TP), the applicability of the MPLS-TP linear protection [RFC6378][RFC7271] for ring topologies and the ring-specific protection mechanism are specified in RFC 6974 [RFC6974] and RFC 8227 [RFC8227], respectively. All these works, except Ethernet ring protection, typically use swapping or steering as the protection mechanism. As ring topologies are widely deployed for transport networks, it is also necessary for the DetNet to support ring topologies.

This document demonstrates how the DetNet can be used in a ring topology. Specifically, DetNet ring supports for Point-to-Point (P2P) and Point-to-Multipoint (P2MP, for multicast services) are discussed in details. This document assumes that the Multi-Protocol Label Switching (MPLS) encapsulation for DetNet is supported as specified in [I-D.ietf-detnet-mpls] and all nodes in a ring network can support the MPLS functionalities. It should be noted that it is more convenient for the DetNet to support a ring topology with the intrinsic duplication and elimination mechanism, as there is no need of swapping or steering operations (consequently, its Operations, Administration and Maintenance (OAM) can also be simplified) for service protection.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Abbreviations

This document uses the following abbreviations:

DetNet Deterministic Networking
 LSP Label Switched Path
 MPLS Multi-Protocol Label Switching
 MPLS-TP Multi-Protocol Label Switching - Transport Profile
 P2MP Point-to-Multipoint
 P2P Point-to-Point
 PEF Packet Elimination Function
 POF Packet Ordering Function
 PRF Packet Replication Function
 PW Pseudowire

4. P2P DetNet Ring

This section describes how the DetNet can deliver P2P traffic over a single ring.

4.1. DetNet applications on a single ring for P2P traffic

Figure 1 shows an example of the DetNet ring for P2P real time traffic. Nodes A and C are DetNet aware devices, and P2P DetNet traffic is transported from node A to node C.

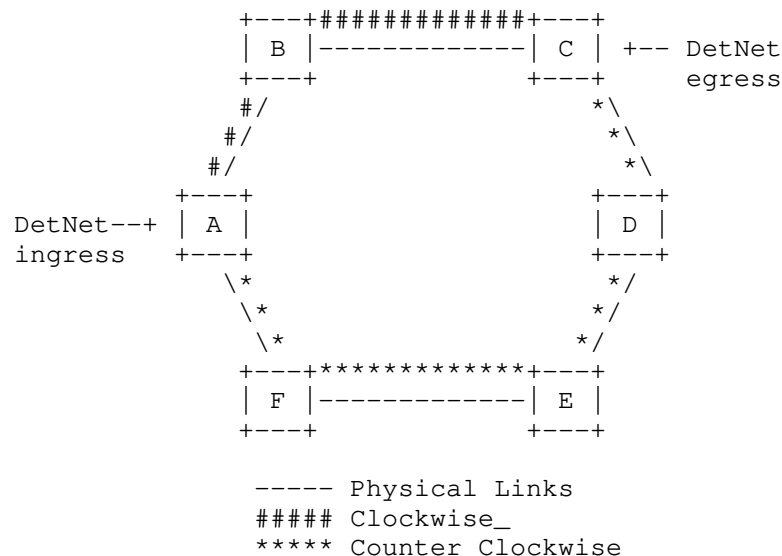


Figure 1: DetNet Ring for P2P traffic

A clockwise and a counter clockwise Label Switched Paths (LSPs) are configured from node A to node C using the DetNet forwarding labels

(F-Labels) are configured from node A to node C. The DetNet service sub-layer functions are provided at nodes A and C utilizing the DetNet service label(s) (S-Label) and DetNet control word (d-CW) as described in [I-D.ietf-detnet-mpls]. The P2P traffic is replicated by a Packet Replication Function (PRF) in node A, encapsulated with the d-CW and specific S-Label and F-Label(s), and transported on both LSP paths towards node C. Upon reception of the traffic, node C terminates the LSP and is aware of the DetNet traffic by inspection of the S-Label carried in each packet. A Packet Elimination Function (PEF) in node C guarantees that only one copy of the DetNet service exits on egress with the help of the DetNet sequence number. A Packet Ordering Function (POF) can further reorder packets in node C before transport of these packets to the destination.

4.2. Implementation implications of a DetNet ring for P2P traffic

In a DetNet ring for P2P traffic, one path may be far longer than the other path. The buffer for reordering at the egress needs to be large enough to accommodate for the sequence number difference between these two paths.

5. P2MP DetNet Ring

5.1. DetNet applications on a single ring for P2MP traffic

Figure 2 shows an example of the DetNet ring for P2MP real time traffic. Nodes A, B, C, E and F are DetNet aware devices, and P2MP DetNet traffic is transported from head-end node A to multiple tail-end nodes C, E and F.

Two approaches are described in Section 5.2 and Section 5.3 for P2MP traffic.

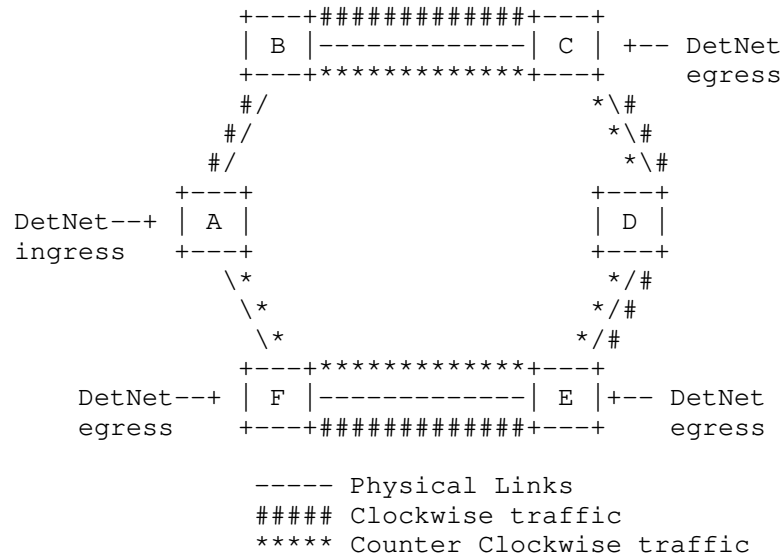


Figure 2: DetNet Ring for P2MP traffic

5.2. Section LSPs as underlay (service sub-layer replication)

If section LSPs are used as an underlay for DetNet services, a bidirectional section LSP tunnel is set up between each pair of neighboring nodes in the ring (e.g., node A and node B, ..., node F and node A). In this case, the DetNet sub-layer replicates the DetNet packets from one tail-end to another neighboring tail-end.

The DetNet head-end (i.e., node A) in the ring needs to support DetNet replication function. Upon reception on node A, the DetNet traffic is replicated with a d-CW, encapsulated with a S-Label and a section LSP label per DetNet member flow, and transported on both section LSPs (i.e., A-B and A-F).

All intermediate nodes (non tail-ends) on the ring MUST transparently forward the DetNet packet, which contains a d-CW and S-Label, to the next hop on the ring.

All DetNet tail-ends except the penultimate node (egress nodes such as nodes C and E in the clockwise, and nodes F, E and C in the counter clockwise) on the ring MUST support both DetNet PRF and PEF functions, and MAY further support a DetNet POF function. For the example of Figure 2, upon reception of the clockwise traffic, node C terminates the section LSP and recognizes the DetNet flow by inspection of the S-label in the packet. Firstly, node C needs to forward the DetNet packet to the next hop on the ring in the

clockwise direction. Secondly, the DetNet packet is also directed to a DetNet PEF associated with the DetNet flow, only one copy is egressed from the ring by inspection of the sequence number in the d-CW. Furthermore, if the DetNet POF function is enabled, the packets in the DetNet flow are reordered before exit to DetNet egress.

If multiple endpoints are attached to a tail-end node, a multicast module can be used to forward the traffic to all these endpoints.

To avoid a loop of DetNet service, the penultimate node in the ring (such as node B on the counter clock-wise LSP) MUST terminate the DetNet flow. For example, upon reception of the clockwise DetNet traffic, node F terminates the DetNet traffic by inspection of the S-Label in the packet. As an alternative, the last DetNet tail-end (such as node C on the counter clock-wise LSP) MAY terminate the DetNet flow, so that the bandwidth from this node to the penultimate node can be saved.

5.3. P2MP LSP tunnels as underlay (forwarding sub-layer replication)

If P2MP LSPs are used as an underlay for the DetNet service, a P2MP unidirectional LSP tunnel in clockwise is set up from head-end (ingress node A) to all the tail-ends (egress nodes C, E and F) for the ring, and another P2MP unidirectional LSP tunnel in counter clockwise is set up from head-end (ingress node A) to all the tail-ends (egress nodes F, E and C) for the ring. Thus, a PRF in LSP layer replicates the DetNet packets from one tail-end to another neighboring tail-end.

The DetNet head-end (i.e., node A) in the ring needs to support the DetNet PRF function. Upon reception on node A, the DetNet traffic is replicated with a d-CW, encapsulated with a S-Label per DetNet member flow, and transported on both P2MP LSP tunnels in the ring.

All DetNet tail-ends (egress nodes such as nodes C, E and F in Figure 2) on the ring need to support the DetNet PEF function. For example, upon reception of the traffic, node C pops the P2MP LSP label and is aware of the DetNet traffic by inspection of the S-Label label in the label stack. Two DetNet member flows are identified with their S-Labels and directed to the same PEF so that only one copy of the DetNet service is selected by inspection of the DetNet sequence number in the d-CW. Furthermore, if DetNet POF function is enabled, the packets in the DetNet flow are reordered before exit to DetNet egress.

If multiple endpoints are attached to a tail-end node, a multicast module can be used to forward the filtered DetNet traffic to all these endpoints

6. DetNet Ring Interconnections

Two DetNet rings can be connected via one or more interconnection nodes. Figure 3 shows the ring interconnection scenarios with a single node and dual nodes. In the interconnected rings, each ring operates in the same way as described in Section 4 and Section 5 except the node or nodes that are used to interconnect two rings.

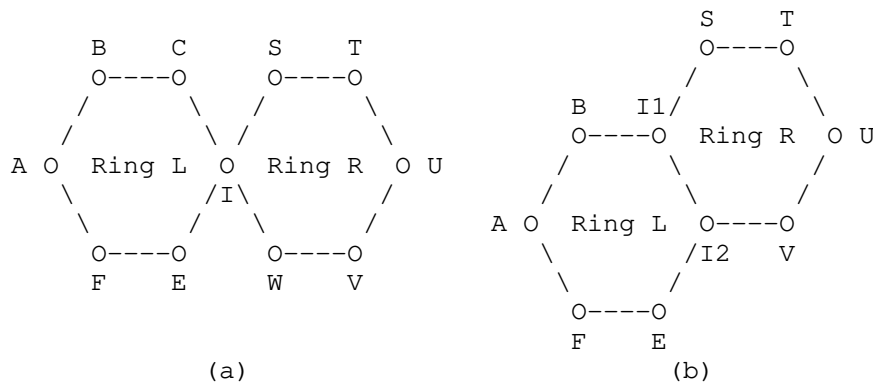


Figure 3: DetNet ring interconnection with: (a) single node (node I), and (b) dual nodes (nodes I1 and I2)

In this section, we describe the behavior of interconnection nodes with the traffic going from Ring L to Ring R. Symmetrical description is assumed for the traffic in the other direction (i.e., from Ring R to Ring L).

6.1. Single node interconnection

In the case of the single node interconnection, as shown in Figure 3(a), both P2P and P2MP DetNet traffic that needs to be transported between Ring L and Ring R use a single interconnection node between two rings. Thus, the interconnection node acts as a DetNet relay node, which provides both PRF and PEF functions.

For P2P DetNet traffic going from Ring L to Ring R, interconnection node I receives the same DetNet flow traffic from both node C and node E (i.e., clockwise and counter-clockwise), a PEF in node I performs packet elimination, and a PRF in node I replicates the packet, node I then sends one copy to node S and another copy to node W.

For P2MP DetNet traffic going from Ring L to Ring R, interconnection node I performs the same packet elimination and replication functions as described above. In addition, node I further transparently forwards the P2MP DetNet traffic on Ring L in the same direction if it is not the last tail-end node.

6.2. Dual node interconnection

In order to prevent a single point of failure, two interconnection nodes can be used as shown in Figure 3(b). To provide high availability for DetNet services, dual node interconnection is recommended. Two interconnection nodes act as DetNet relay nodes, each provides both packet replication and elimination functions.

6.2.1. Dual node interconnection for P2P traffic

For the P2P DetNet traffic that flows from Ring L to Ring R in Figure 3(b), the operations of interconnection nodes I1 and I2 are described below.

When interconnection node I1 receives clockwise traffic from node B, it replicates the traffic and sends one copy to interconnection node I2 and the other copy to a PEF in interconnection node I1.

When interconnection node I1 receives counter-clockwise traffic from interconnection node I2, it forwards the traffic to the PEF of interconnection node I1.

At the PEF of interconnection node I1, duplicate elimination is performed for the clockwise traffic from node B and the counter-clockwise traffic from interconnection node I2, and only one copy is sent to the clockwise direction of Ring R (i.e., sent towards node S). Furthermore, if DetNet POF function is enabled on interconnection node I1, the packets in the DetNet flow are reordered before being forwarded to Ring R.

When interconnection node I2 receives counter-clockwise traffic from node E, it replicates the traffic and sends one copy to interconnection node I1 and the other copy to a PEF in interconnection node I2.

When interconnection node I2 receives clockwise traffic from interconnection node I1, it forwards the traffic to the PEF of interconnection node I2.

At the PEF of interconnection node I2, duplicate elimination is performed for the counter-clockwise traffic from node E and the clockwise traffic from interconnection node I1, and only one copy is

sent to the counter-clockwise direction of Ring R (i.e., sent towards node V). Furthermore, if DetNet POF function is enabled on interconnection node I2, the packets in the DetNet flow are reordered before being forwarded to Ring R.

6.2.2. Dual node interconnection for P2MP traffic using section LSP

For the P2MP traffic that flows from Ring L to Ring R in Figure 3(b), each ring is configured and operated as described in Section 5.2 except the interconnection nodes, whose operations are described below.

When interconnection node I1 receives clockwise traffic from node B, its PRF replicates the traffic and sends one copy to interconnection node I2 and the other copy to interconnection node I1's PEF.

When interconnection node I1 receives the counter-clockwise traffic from interconnection node I2, its PRF replicates the traffic and sends one copy to node B and the other copy to interconnection node I1's PEF unless interconnection node I1 is the penultimate node for the counter-clockwise traffic on Ring L. In the case that interconnection node I1 is the penultimate node for the counter-clockwise traffic on Ring L, the counter-clockwise traffic from interconnection node I2 is only forwarded to interconnection node I1's PEF.

At interconnection node I1's PEF, duplicate elimination is performed for the clockwise traffic from node B and the counter-clockwise traffic from interconnection node I2, and only one copy is sent to the clockwise direction of Ring R (i.e., sent towards node S). Furthermore, if DetNet POF function is enabled on node I1, the packets in the DetNet flow are reordered before being forwarded to Ring R.

When interconnection node I2 receives the counter-clockwise traffic from node E, its PRF replicates the traffic and sends one copy to interconnection node I1 and the other copy to node I2's PEF.

When interconnection node I2 receives the clockwise traffic from interconnection node I1, its PRF replicates the traffic and sends one copy to node E and the other copy to interconnection node I2's PEF unless interconnection node I2 is the penultimate node for the clockwise traffic on Ring L. In the case that interconnection node I2 is the penultimate node for the clockwise traffic on Ring L, the clockwise traffic from interconnection node I1 is only forwarded to node I2's PEF.

At node I2's PEF, duplicate elimination is performed for the counter-clockwise traffic from node E and the clockwise traffic from interconnection node I1, and only one copy is sent to the counter-clockwise direction of Ring R (i.e., sent towards node V). Furthermore, if DetNet POF function is enabled on interconnection node I2, the packets in the DetNet flow are reordered before being forwarded to Ring R.

6.2.3. Dual node interconnection for P2MP traffic using P2MP LSP

If P2MP LSPs are used in the interconnected rings, two P2MP unidirectional LSP tunnels are used on each ring for the clockwise and counter-clockwise directions.

When the P2MP traffic is forwarded from one ring to another ring, for example from Ring L to Ring R in Figure 3(b), each P2MP LSP in Ring L MUST include interconnection nodes I1 and I2 as its tail-ends. For Ring R, one P2MP LSP is set up from interconnection node I1 to all the tail-ends in the clockwise direction on Ring R, and the other P2MP LSP is set up from interconnection node I2 to all the tail-ends in the counter-clockwise direction on Ring R. Therefore, an interconnection node acts as a tail-end for one ring and a head-end for another ring in one direction, and performs the same operation of tail-end and head-end as specified in Section 5.3.

7. Resource Reservation

In order to guarantee that DetNet flows do not suffer from network congestion, the DetNet data plane considerations on resource reservation and allocation as described in [I-D.ietf-detnet-data-plane-framework] apply here.

8. IANA Considerations

There are no IANA actions required by this document

9. Security Considerations

This document describes the application of DetNet MPLS on ring topologies. Thus, the security considerations described in [I-D.ietf-detnet-mpls] are also applied to this document. If any new security considerations specific to ring topologies are identified, they will be added in a future version of this draft.

10. References

10.1. Normative References

- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", draft-ietf-
detnet-architecture-13 (work in progress), May 2019.
- [I-D.ietf-detnet-data-plane-framework]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane
Framework", draft-ietf-detnet-data-plane-framework-00
(work in progress), May 2019.
- [I-D.ietf-detnet-mpls]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS",
draft-ietf-detnet-mpls-00 (work in progress), May 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

10.2. Informative References

- [RFC6378] Weingarten, Y., Ed., Bryant, S., Osborne, E., Sprecher,
N., and A. Fulignoli, Ed., "MPLS Transport Profile (MPLS-
TP) Linear Protection", RFC 6378, DOI 10.17487/RFC6378,
October 2011, <<https://www.rfc-editor.org/info/rfc6378>>.
- [RFC6974] Weingarten, Y., Bryant, S., Ceccarelli, D., Caviglia, D.,
Fondelli, F., Corsi, M., Wu, B., and X. Dai,
"Applicability of MPLS Transport Profile for Ring
Topologies", RFC 6974, DOI 10.17487/RFC6974, July 2013,
<<https://www.rfc-editor.org/info/rfc6974>>.

- [RFC7271] Ryoo, J., Ed., Gray, E., Ed., van Helvoort, H., D'Alessandro, A., Cheung, T., and E. Osborne, "MPLS Transport Profile (MPLS-TP) Linear Protection to Match the Operational Expectations of Synchronous Digital Hierarchy, Optical Transport Network, and Ethernet Transport Network Operators", RFC 7271, DOI 10.17487/RFC7271, June 2014, <<https://www.rfc-editor.org/info/rfc7271>>.
- [RFC8227] Cheng, W., Wang, L., Li, H., van Helvoort, H., and J. Dong, "MPLS-TP Shared-Ring Protection (MSRP) Mechanism for Ring Topology", RFC 8227, DOI 10.17487/RFC8227, August 2017, <<https://www.rfc-editor.org/info/rfc8227>>.
- [RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <<https://www.rfc-editor.org/info/rfc8578>>.

Authors' Addresses

Yuanlong Jiang
Huawei Technologies
Bantian, Longgang district
Shenzhen 518129
China

Phone: +86-18926415311
Email: jiangyuanlong@huawei.com

Norman Finn
Huawei Technologies
3755 Avocado Blvd
California 91941
USA

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

Jeong-dong Ryoo
ETRI
218 Gajeongno
Yuseong-gu, Daejeon 34129
South Korea

Phone: +82-42-860-5384
Email: ryoo@etri.re.kr

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

Email: balazs.a.varga@ericsson.com

Liang Geng
China Mobile
Beijing
China

Email: gengliang@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 6, 2020

A. Malis
Futurewei Technologies
X. Geng
M. Chen
Huawei
F. Qin
China Mobile
July 05, 2019

Deterministic Networking (DetNet) Controller Plane Framework
draft-malis-detnet-controller-plane-framework-01

Abstract

This document provides a framework overview for the Deterministic Networking (DetNet) controller plane. It discusses concepts and requirements that will be basis for Detnet controller plane solution documents.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 6, 2020.

Copyright Notice

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

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

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
2. DetNet Controller Plane Requirements	4
3. DetNet Control Plane Architecture	5
3.1. Distributed Control Plane and Signaling Protocols	5
3.2. SDN/Fully Centralized Control Plane	6
3.3. Hybrid Control Plane	7
4. DetNet Control Plane Additional Details and Issues	8
4.1. Explicit Paths	8
4.2. Resource Reservation	8
4.3. PREOF Support	9
4.4. DetNet in a Traditional MPLS Domain	9
4.5. IP	10
4.6. DetNet with Segment Routing (SR)	10
5. Management Plane Overview	12
5.1. Provisioning	12
5.2. DetNet Operations, Administration and Maintenance (OAM)	12
5.2.1. OAM for Performance Monitoring (PM)	12
5.2.2. OAM for Fault/Defect Management (FM)	13
6. IANA Considerations	13
7. Security Considerations	13
8. Acknowledgments	13
9. References	13
9.1. Normative References	14
9.2. Informative References	15
Authors' Addresses	19

1. Introduction

Deterministic Networking (DetNet) provides the capability to carry specified unicast and/or multicast data flows for real-time applications with extremely low data loss rates and bounded latency within a network domain. As discussed in the Deterministic Networking Architecture [I-D.ietf-detnet-architecture], techniques used to provide this capability include reserving data plane resources for individual (or aggregated) DetNet flows in some or all of the intermediate nodes along the path of the flow, providing explicit routes for DetNet flows that do not immediately change with the network topology, and 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.

The DetNet data plane is defined in a set of documents that are anchored by the DetNet Data Plane Framework [I-D.ietf-detnet-data-plane-framework] and the associated DetNet MPLS [I-D.ietf-detnet-mpls] and IP [I-D.ietf-detnet-ip] data plane specifications, with additional details and subnet mappings provided in [I-D.ietf-detnet-ip-over-mpls], [I-D.ietf-detnet-mpls-over-udp-ip], [I-D.ietf-detnet-mpls-over-tsn], [I-D.ietf-detnet-ip-over-tsn], and [I-D.ietf-detnet-tsn-vpn-over-mpls].

While the Detnet Architecture and Data Plane Framework documents are primarily concerned with data plane operations, they do contain some references and requirements for functions that would be required in order to automate DetNet service provisioning and monitoring via a DetNet controller plane. The purpose of this document is to gather these references and requirements into a single document and discuss how various possible DetNet controller plane architectures could be used to satisfy these requirements, while not providing the actual protocol details for a DetNet controller plane solution. Such controller plane protocol solutions will be the subject of subsequent documents.

Note that in the DetNet overall architecture, the controller plane includes what are more traditionally considered separate control and management planes. Traditionally, the management plane is primarily involved with node and network provisioning, operational OAM for performance monitoring, and troubleshooting network behaviors and outages, while the control plane is primarily responsible for the instantiation and maintenance of flows, MPLS label allocation and distribution, and active in-band or out-of-band signaling to support these functions. In the DetNet architecture, all of this functionality is combined into a single Controller Plane. See Section 4.4.2 of [I-D.ietf-detnet-architecture] and the aggregation of Control and Management planes in [RFC7426] for further details.

1.1. Terminology

This document uses the terminology established in the DetNet Architecture [I-D.ietf-detnet-architecture], and the reader is assumed to be familiar with that document and its terminology.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. DetNet Controller Plane Requirements

Other DetNet documents, including [I-D.ietf-detnet-architecture] and [I-D.ietf-detnet-data-plane-framework], contain requirements for the Controller Plane. For convenience, these requirements have been compiled here. The primary requirements of the DetNet Controller Plane are that it must be able to:

- o Support the dynamic creation, modification, and deletion of DetNet flows. This may include some or all of explicit path determination, link bandwidth reservations, restricting flows to IEEE 802.1 Time-Sensitive Networking (TSN) links, node buffer and other resource reservations, specification of required queuing disciplines along the path, ability to manage bidirectional flows, etc., as needed for a flow.
- o Support DetNet flow aggregation and de-aggregation via the ability to dynamically create and delete flow aggregates (FAs), and be able to modify existing FAs by adding or deleting members.
- o Operate in a converged network domain that contains both DetNet and non-DetNet flows.
- o Allow flow instantiation requests to originate in an end application (via an Application Programming Interface (API), via static provisioning, or via a dynamic control plane, such as a centralized SDN controller or distributed signaling protocols. See Section 3 for further discussion of these options.
- o In the case of the DetNet MPLS data plane, manage DetNet S-Label and F-Label allocation and distribution.
- o Also in the case of the DetNet MPLS data plane, support packet replication, duplicate elimination, and packet ordering functions (PREOF), and to be able to place these functions at appropriate places in the network.
- o Support applications that require the ability to synchronize the clocks in end systems to the extent supported by the DetNet data plane.
- o Support queue control techniques defined in Section 4.5 of [I-D.ietf-detnet-architecture] and [I-D.finn-detnet-bounded-latency] that require time synchronization among network nodes.
- o Advertise static and dynamic node and link resources such as capabilities and adjacencies to other network nodes (for dynamic

signaling approaches) or to network controllers (for centralized approaches).

- o Adapt to network topology changes such as links or nodes failures.
- o Scale to handle the number of DetNet flows expected in a domain (which may require per-flow signaling or provisioning). This is similar to scalability requirements associated with network slicing [I-D.dong-spring-sr-for-enhanced-vpn].
- o Provision flow identification information at each of the nodes along the path. Flow identification may differ depending on the location in the network and the DetNet functionality (e.g. transit node vs. relay node).
- o Monitor the performance of DetNet flows to ensure that they are meeting required objectives.

3. DetNet Control Plane Architecture

As noted in the Introduction, the DetNet control plane is responsible for the instantiation and maintenance of flows, MPLS label allocation and distribution, and active in-band or out-of-band signaling to support these functions.

The following sections define three possible classes of DetNet control plane architectures: a fully distributed control plane utilizing dynamic signaling protocols, a fully centralized SDN-like control plane, and a hybrid control plane. They discuss the various information exchanges between entities in the network in each of these architectures and the advantages and disadvantages of each option.

In each of the following sections, examples are used to illustrate possible mechanisms that could be used in each of the architectures. These are not meant to be exhaustive or to preclude any other possible mechanism that could be used in place of those used in the examples.

3.1. Distributed Control Plane and Signaling Protocols

In a fully distributed configuration model, User-to-Network Interface (UNI) information is transmitted over a (to-be-defined) DetNet UNI protocol from the user side to the network side, and then UNI and network configuration information propagate in the network via distributed control plane signaling protocols. Using an RSVP-TE traffic-engineered MPLS network as an example:

1. An IGP collects topology information and DetNet capabilities of the network [draft-geng-detnet-info-distribution];
2. The control plane of the ingress edge node receives a flow establishment request from the UNI and calculates one or more valid path(s);
3. Using RSVP-TE [RFC3209], the ingress edge node sends a PATH message with an explicit route. After receiving the PATH message, the egress edge node sends a RESV message with the distributed label and resource reservation request.

Current reservation-oriented distributed control plane protocols, e.g. RSVP-TE and Stream Reservation Protocol (SRP) [IEEE.802.1Qcc-2018], can only reserve bandwidth along the path, while the configuration of a fine-grained schedule, e.g., Time Aware Shaping (TAS) [IEEE.802.1QBV_2015], is not supported. If RSVP-TE or SRP were to be used for a DetNet application, it would require extensions in order to support queue and scheduler reservations in addition to bandwidth reservation.

As discussed in Section 4.9 of [I-D.ietf-detnet-architecture], scalability is a primary concern for DetNet, given the large number of expected flows in a DetNet domain. This could potentially be much larger than, for example, the number of MPLS traffic tunnels in a network using MPLS traffic engineering, which would typically be $N*(N-1)$ tunnels, where N is the number of edge routers in the domain.

Even when flow aggregation is used, DetNet domains can be expected to support a very large number of flows that will need particular queuing disciplines and/or resource allocation, depending on the requirements for each flow. This could require a large amount of dynamic signaling, such as an RSVP-TE session to establish and maintain each flow. Other RSVP-TE scalability concerns are further discussed in [RFC5439].

All of the above tends to argue against a purely distributed control plane for DetNet domains.

3.2. SDN/Fully Centralized Control Plane

In the fully SDN/centralized configuration model, UNI information is transmitted from a Centralized User Configuration (CUC) or from applications via an API or northbound interface to a Centralized Controller, which is the sole source of routing and forwarding information for the domain. Configurations of nodes for DetNet flows are performed by the controller using a protocol such as NETCONF [RFC6241]/YANG [RFC6020] or PCE-CC [RFC8283]. For example:

1. The controller collects topology information and DetNet capabilities of the network via NETCONF/YANG;
2. The controller receives a flow establishment request from a UNI and calculates one or more valid path(s) through the network;
3. The controller chooses the optimal path and configures the devices along that path for flow transmission via PCE-CC.

3.3. Hybrid Control Plane

In the hybrid model, a controller and control plane protocols work together to provide DetNet services, and there are a number of possible combinations. For example:

1. A Centralized Controller collects topology information and DetNet capabilities of the network via an IGP and/or BGP-LS [RFC7752];
2. The controller receives a flow establishment request from a UNI and calculates one or more valid path(s) through the network;
3. Based on the calculation result, the CNC distributes flow path information to the ingress edge node and other information (e.g. replication/duplicate elimination) to the relevant nodes.
4. Using RSVP-TE, the ingress edge node sends a PATH message with an explicit route. After receiving the PATH message, the egress edge node sends a RESV message with the distributed label and resource reservation request.

or

1. The controller collects topology information and DetNet capability of the network via an IGP or BGP-LS;
2. The control plane of the ingress edge node receives a flow establishment request via a UNI;
3. The Ingress edge node sends the path establishment request to the controller through PCEP [RFC5440];
4. After path calculation, the CNC sends the path information of the flow to the ingress edge node via PCEP;
5. Using RSVP-TE, the ingress edge node sends a PATH message with an explicit route. After receiving the PATH message, the egress edge node sends a RESV message with the distributed label and resource reservation request.

There are many other variations that could be included in a hybrid control plane. This document cannot discuss all the possible control plane mechanisms that could be used in hybrid configuration models. Every solution has its own mechanisms and corresponding parameters that are required for it to work.

4. DetNet Control Plane Additional Details and Issues

This section discusses some additional DetNet control plane details and issues.

4.1. Explicit Paths

Explicit paths are required in DetNet to provide a stable transport service and guarantee that DetNet service is not effected when the network topology changes. The following features are necessary to have explicit paths in DetNet:

- o Path computation: DetNet explicit paths need to meet the SLA (Service Level Agreement) requirements and/or resource guarantees from the application/client, which include bandwidth, maximum end-to-end delay, maximum end-to-end delay variation, maximum loss ratio, etc. In an distributed system with IGP-TE, CSPF (Constrained Shortest Path First) can be used to compute a set of feasible paths for a DetNet service. In a system with a network controller, a PCE (Path Computation Engine) can compute paths satisfying the requirements of DetNet with the network information collected from the DetNet domain.
- o Path establishment: Once the path has been computed, the options discussed in Section 3 can be used to establish the path. Also see Section 4.4 and Section 4.6 for some additional considerations depending on the details of the network infrastructure.
- o Strict or loose paths: An explicit path is strict when every intermediate hop is specified so that its route can't change. An explicit path is loose when any IGP route is allowed along the path. Generally, end-to-end SLA guarantees require a strict explicit path in DetNet. However, when the IGP route is known to be able to meet the SLA requirements, loose explicit paths are also acceptable.

4.2. Resource Reservation

Network congestion could cause uncontrolled delay and/or packet loss. DetNet flows are supposed to be protected from congestion, so sufficient resource reservation for DetNet service is necessary. Resources in the network are complex and hard to quantize, and may

include such entities as packet processing resources, packet buffering, port and link bandwidth, and so on. The resources a particular flow requires are determined by the flow's characteristics and SLA.

- o Resource Allocation: Port bandwidth is one of the basic attributes of a network device which is easy to obtain or calculate. In current traffic engineering implementations, network resource allocation is synonymous with bandwidth allocation. A DetNet flow is characterized with a traffic specification as defined in [I-D.ietf-detnet-flow-information-model], including attributes such as Interval, Maximum Packets Per Interval, and Maximum Payload Size. The traffic specification describes the worst case, rather than the average case, for the traffic, to ensure that sufficient bandwidth and buffering resources are reserved to satisfy the traffic specification.
- o Device configuration with or without flow discrimination: The resource allocation can be guaranteed by device configuration. For example, an output port bandwidth reservation can be configured as a parameter of queue management and the port scheduling algorithm. When DetNet flows are aggregated, a group of DetNet flows share the allocated resource in the network device. When the DetNet flows are treated independently, the device should maintain a mapping relationship between a DetNet flow and its corresponding resources.

4.3. PREOF Support

DetNet path redundancy is supported via packet replication and duplicate elimination (PREOF). A DetNet flow is replicated and goes through multiple networks paths to avoid packet loss caused by device or link failures. In general, current control plane mechanisms that can be used to establish an explicit path, whether distributed or centralized, support point-to-point (P2P) and point-to-multipoint (P2MP) path establishment. PREOF requires the ability to compute and establish a point-to-multipoint-to-point (P2MP2P) path. Protocol extensions will be required to support this new feature.

4.4. DetNet in a Traditional MPLS Domain

For the purposes of this document, "traditional MPLS" is defined as MPLS without the use of segment routing (see Section 4.6 for a discussion of MPLS with segment routing) or MPLS-TP [RFC5960].

In traditional MPLS domains, a dynamic control plane using distributed signaling protocols is typically used for the distribution of MPLS labels used for forwarding MPLS packets. The

dynamic signaling protocols most commonly used for label distribution are LDP [RFC5036], RSVP-TE, and BGP [RFC8277] (which enables BGP/MPLS-based Layer 3 VPNs [RFC4384] and Layer 2 VPNs [RFC7432]).

Any of these protocols could be used to distribute DetNet Service Labels (S-Labels) and Aggregation Labels (A-Labels) [I-D.ietf-detnet-mpls]. As discussed in [I-D.ietf-detnet-data-plane-framework], S-Labels are similar to other MPLS service labels, such as pseudowire, L3 VPN, and L2 VPN labels, and could be distributed in a similar manner, such as through the use of targeted LDP or BGP. If these were to be used for DetNet, they would require extensions to support DetNet-specific features such as PREOF, aggregation (A-Labels), node resource allocation, and queue placement.

However, as discussed in Section 3.1, distributed signaling protocols may have difficulty meeting DetNet's scalability requirements. MPLS also allows SDN-like centralized label management and distribution as an alternative to distributed signaling protocols, using protocols such as PCEP and OpenFlow [OPENFLOW].

PCEP, particularly when used as a part of PCE-CC, is a possible candidate protocol to use for centralized management of traditional MPLS-based DetNet domains. However, PCE path calculation algorithms would need to be extended to include the location determination for PREOF nodes in a path, and the means to signal the necessary resource reservation and PREOF function placement information to network nodes. See ((I-D.ietf-pce-pcep-extension-for-pce-controller)) for further discussion of PCE-CC and PCEP for centralized control of an MPLS domain.

4.5. IP

In a later revision of this document, this section will discuss necessary protocol extensions to existing IP routing protocols such as IS-IS and BGP. It should be noted that a DetNet IP domain is simpler than a DetNet MPLS domain, and doesn't support PREOF, so only one path per flow or flow aggregate is required, with no path merging.

4.6. DetNet with Segment Routing (SR)

Segment Routing [RFC8402] is a scalable approach to building network domains that utilizes a combination of source routing in packet headers and centralized network control to compute paths through the network and distribute those paths with associated policy to network edge nodes for use in packet headers. It greatly reduces the amount of network signaling associated with distributed signaling protocols

such as RSVP-TE, and also greatly reduces the amount of state in core nodes compared with that required for traditional MPLS and IP routing, as the state is now in the packets rather than in the routers. This is especially useful for DetNet, where a very large number of flows through a network domain are expected, which would otherwise require the instantiation of state for each flow traversing each node in the network.

The DetNet MPLS and IP data planes were specifically constructed to allow the use of DetNet with both types of segment routing, SR-MPLS [I-D.ietf-spring-segment-routing-mpls] and SRv6 [I-D.ietf-6man-segment-routing-header].

In the DetNet context, DetNet in an SR-MPLS or SRv6 data plane could be used in conjunction with centralized flow management and complete label stack distribution to Detnet domain entry nodes via a centralized controller. Extensions to PCEP to allow the use of PCE-CC with SR-MPLS

One possible architecture is PCE-CC combined with SR-MPLS or SRv6. Extensions to PCEP to allow the use of PCE-CC with SR-MPLS are described in [I-D.zhao-pce-pcep-extension-pce-controller-sr], with SRv6 in [I-D.dhody-pce-pcep-extension-pce-controller-srv6].

This approach would allow the details of packet or flow treatment to be encoded directly in the SIDs on each packet in a flow to reduce the amount of state in network nodes. This approach also allows the integration of DetNet domains with general SR-based backbone networks in a converged domain. In this approach, a new set of functions for DetNet queuing treatments available in the DetNet domain would need to be defined for inclusion in the SR stack.

This is not the only possible approach. There is ongoing work on a number of alternative signaling mechanisms for MPLS-SR and SRv6, including extensions to IGPs and BGP to support distributed signaling. In addition, BGP-LS and BGP route reflectors could be added for a hybrid solution.

A possible mostly centralized hybrid approach could be to use a PCE-CC to push paths represented by SID lists while using BGP-LS to collect network topology and link state information. An IGP is used for the usual link state flooding in order to establish adjacencies, but not for DetNet flow path calculations, only for best effort traffic as usual.

A similar approach for network slicing that could be leveraged for DetNet is described in [I-D.dong-spring-sr-for-enhanced-vpn].

Also, note that SR cannot currently support DetNet PREOF functionality without extensions. One possible approach could be to combine SR with BIER-TE, as discussed in [I-D.ietf-bier-te-arch]. Another possible approach specific to SRv6 is discussed in [I-D.geng-detnet-dp-sol-srv6].

5. Management Plane Overview

The Management Plane includes the ability to statically provision network nodes and to use OAM to monitor DetNet performance and detect outages or other issues at the DetNet layer.

5.1. Provisioning

Static provisioning in a Detnet network will be performed via the use of appropriate YANG models, including [I-D.ietf-detnet-yang] and [I-D.ietf-detnet-topology-yang].

5.2. DetNet Operations, Administration and Maintenance (OAM)

The overall framework and requirements for DetNet OAM are discussed in [I-D.mirsky-detnet-oam]. This document currently includes additional OAM details that may eventually be merged into that document.

5.2.1. OAM for Performance Monitoring (PM)

5.2.1.1. Active PM

Active PM is performed by injecting OAM packets into the network to estimate the performance of the network by measuring the performance of the OAM packets. Adding extra traffic can affect the delay and throughput performance of the network, and for this reason active PM is not recommended for use in operational DetNet domains. However, it is a useful test tool when commissioning a new network.

5.2.1.2. Passive PM

Passive PM monitors the actual service traffic in a network domain in order to measure its performance without having a detrimental affect on the network. As compared to Active PM, Passive PM is much preferred for use in DetNet domains.

A proposal for DetNet passive performance measurement is contained in [I-D.chen-detnet-loss-delay].

5.2.2. OAM for Fault/Defect Management (FM)

[I-D.mirsky-detnet-oam] contains requirements for fault/defect detection and management in a DetNet domain.

6. IANA Considerations

This document has no actions for IANA.

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

7. Security Considerations

The overall security considerations of DetNet are discussed in [I-D.ietf-detnet-architecture] and [I-D.ietf-detnet-security]. For DetNet networks that make use of Segment Routing (whether SR-MPLS or SRv6), the security considerations in [RFC8402] also apply.

DetNet networks that make use of a centralized controller plane may be threatened by the loss of connectivity (whether accidental or malicious) between the central controller and the network nodes, and/or the spoofing of control messages from the controller to the network nodes. This is important since such networks depend on centralized controllers to calculate flow paths and instantiate flow state in the network nodes. For networks that use both DetNet and Segment Routing with a centralized controller, this would also include the calculation of SID lists and their installation in edge/border routers.

In both cases, such threats may be mitigated through redundant controllers, the use of authentication between the controller(s) and the network nodes, and other mechanisms for protection against DOS attacks. A mechanism for supporting one or more alternative central controllers and the ability to fail over to such an alternative controller will be required.

8. Acknowledgments

Thanks to Jim Guichard, Donald Eastlake, and Stewart Bryant for their review comments.

9. References

9.1. Normative References

- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", draft-ietf-
detnet-architecture-13 (work in progress), May 2019.
- [I-D.ietf-detnet-data-plane-framework]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane
Framework", draft-ietf-detnet-data-plane-framework-01
(work in progress), July 2019.
- [I-D.ietf-detnet-flow-information-model]
Farkas, J., Varga, B., Cummings, R., and Y. Jiang, "DetNet
Flow Information Model", draft-ietf-detnet-flow-
information-model-03 (work in progress), March 2019.
- [I-D.ietf-detnet-ip]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane: IP",
draft-ietf-detnet-ip-01 (work in progress), July 2019.
- [I-D.ietf-detnet-mpls]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS",
draft-ietf-detnet-mpls-01 (work in progress), July 2019.
- [I-D.ietf-detnet-security]
Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell,
J., Austad, H., Stanton, K., and N. Finn, "Deterministic
Networking (DetNet) Security Considerations", draft-ietf-
detnet-security-04 (work in progress), March 2019.
- [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>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [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>>.

9.2. Informative References

- [I-D.chen-detnet-loss-delay]
Chen, M. and A. Malis, "DetNet Packet Loss and Delay Performance Measurement", draft-chen-detnet-loss-delay-01 (work in progress), October 2018.
- [I-D.dhody-pce-pcep-extension-pce-controller-srv6]
Negi, M., Li, Z., and X. Geng, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) for SRv6", draft-dhody-pce-pcep-extension-pce-controller-srv6-01 (work in progress), February 2019.
- [I-D.dong-spring-sr-for-enhanced-vpn]
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., and Z. Li, "Segment Routing for Enhanced VPN Service", draft-dong-spring-sr-for-enhanced-vpn-04 (work in progress), July 2019.
- [I-D.finn-detnet-bounded-latency]
Finn, N., Boudec, J., Mohammadpour, E., Zhang, J., Varga, B., and J. Farkas, "DetNet Bounded Latency", draft-finn-detnet-bounded-latency-04 (work in progress), June 2019.
- [I-D.geng-detnet-dp-sol-srv6]
Geng, X., Chen, M., and Y. Zhu, "DetNet SRv6 Data Plane Encapsulation", draft-geng-detnet-dp-sol-srv6-01 (work in progress), July 2019.
- [I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-21 (work in progress), June 2019.
- [I-D.ietf-bier-te-arch]
Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication (BIER-TE)", draft-ietf-bier-te-arch-02 (work in progress), May 2019.

- [I-D.ietf-detnet-ip-over-mpls]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP over MPLS", draft-ietf-detnet-ip-over-mpls-01 (work in progress), July 2019.
- [I-D.ietf-detnet-ip-over-tsn]
Varga, B., Farkas, J., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP over IEEE 802.1 Time Sensitive Networking (TSN)", draft-ietf-detnet-ip-over-tsn-00 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls-over-tsn]
Varga, B., Farkas, J., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS over IEEE 802.1 Time Sensitive Networking (TSN)", draft-ietf-detnet-mpls-over-tsn-00 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls-over-udp-ip]
Varga, B., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS over UDP/IP", draft-ietf-detnet-mpls-over-udp-ip-01 (work in progress), July 2019.
- [I-D.ietf-detnet-topology-yang]
Geng, X., Chen, M., Li, Z., and R. Rahman, "Deterministic Networking (DetNet) Topology YANG Model", draft-ietf-detnet-topology-yang-00 (work in progress), January 2019.
- [I-D.ietf-detnet-tsn-vpn-over-mpls]
Varga, B., Farkas, J., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IEEE 802.1 Time Sensitive Networking over MPLS", draft-ietf-detnet-tsn-vpn-over-mpls-00 (work in progress), May 2019.
- [I-D.ietf-detnet-yang]
Geng, X., Chen, M., Li, Z., and R. Rahman, "Deterministic Networking (DetNet) Configuration YANG Model", draft-ietf-detnet-yang-02 (work in progress), March 2019.
- [I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-22 (work in progress), May 2019.

- [I-D.mirsky-detnet-oam]
Mirsky, G. and M. Chen, "Operations, Administration and Maintenance (OAM) for Deterministic Networks (DetNet)", draft-mirsky-detnet-oam-03 (work in progress), May 2019.
- [I-D.zhao-pce-pcep-extension-pce-controller-sr]
Zhao, Q., Li, Z., Negi, M., and C. Zhou, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) of SR-LSPs", draft-zhao-pce-pcep-extension-pce-controller-sr-04 (work in progress), February 2019.
- [IEEE.802.1QBV_2015]
IEEE, "IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", IEEE 802.1Qbv-2015, DOI 10.1109/IEEESTD.2016.7572858, March 2016, <<http://ieeexplore.ieee.org/servlet/opac?punumber=7572858>>.
- [IEEE.802.1Qcc-2018]
IEEE, "IEEE Standard for Local and Metropolitan Area Networks -- Bridges and Bridged Networks -- Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements", IEEE 802.1Qcc-2018, DOI 10.1109/ieeestd.2018.8514112, October 2018, <<http://ieeexplore.ieee.org/servlet/opac?punumber=8514110>>.
- [OPENFLOW]
Open Networking Foundation, "OpenFlow Switch Specification, Version 1.5.1 (Protocol version 0x06)", ONF TS-025, March 2015, <<https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>>.
- [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>>.
- [RFC4384] Meyer, D., "BGP Communities for Data Collection", BCP 114, RFC 4384, DOI 10.17487/RFC4384, February 2006, <<https://www.rfc-editor.org/info/rfc4384>>.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <<https://www.rfc-editor.org/info/rfc5036>>.

- [RFC5439] Yasukawa, S., Farrel, A., and O. Komolafe, "An Analysis of Scaling Issues in MPLS-TE Core Networks", RFC 5439, DOI 10.17487/RFC5439, February 2009, <<https://www.rfc-editor.org/info/rfc5439>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC5960] Frost, D., Ed., Bryant, S., Ed., and M. Bocci, Ed., "MPLS Transport Profile Data Plane Architecture", RFC 5960, DOI 10.17487/RFC5960, August 2010, <<https://www.rfc-editor.org/info/rfc5960>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<https://www.rfc-editor.org/info/rfc7432>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <<https://www.rfc-editor.org/info/rfc8277>>.
- [RFC8283] Farrel, A., Ed., Zhao, Q., Ed., Li, Z., and C. Zhou, "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", RFC 8283, DOI 10.17487/RFC8283, December 2017, <<https://www.rfc-editor.org/info/rfc8283>>.

Authors' Addresses

Andrew G. Malis
Futurewei Technologies

Email: agmalis@gmail.com

Xuesong Geng
Huawei

Email: gengxuesong@huawei.com

Mach (Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Fengwei Qin
China Mobile

Email: qinfengwei@chinamobile.com

DetNet Working Group
Internet-Draft
Intended status: Informational
Expires: January 8, 2020

G. Mirsky
ZTE Corp.
M. Chen
Huawei
July 7, 2019

Operations, Administration and Maintenance (OAM) for Deterministic
Networks (DetNet) with IP Data Plane
draft-mirsky-detnet-ip-oam-00

Abstract

This document defines the principals for using Operations, Administration, and Maintenance protocols and mechanisms in the Deterministic Networking networks with IP data plane.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 8, 2020.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Conventions used in this document	2
2.1. Terminology	2
2.2. Keywords	3
3. Active OAM for DetNet Networks with IP Data Plane	3
4. Use of Hybrid OAM in DetNet	4
5. OAM of DetNet IP Interworking with OAM of DetNet MPLS	4
6. OAM of DetNet IP Interworking with OAM of TSN	4
7. IANA Considerations	4
8. Security Considerations	4
9. Acknowledgment	4
10. References	5
10.1. Normative References	5
10.2. Informational References	5
Authors' Addresses	6

1. Introduction

[I-D.ietf-detnet-architecture] introduces and explains Deterministic Networks (DetNet) architecture.

Operations, Administration and Maintenance (OAM) protocols are used to detect, localize defects in the network, and monitor network performance. Some OAM functions, e.g., failure detection, work in the network proactively, while others, e.g., defect localization, usually performed on-demand. These tasks achieved by a combination of active and hybrid, as defined in [RFC7799], OAM methods.

[I-D.mirsky-detnet-oam] lists the functional requirements toward OAM for DetNet domain. The list can further be used for gap analysis of available OAM tools to identify possible enhancements of existing or whether new OAM tools are required to support proactive and on-demand path monitoring and service validation. Also, the document defines the OAM use principals for the DetNet networks with IP data plane.

2. Conventions used in this document

2.1. Terminology

The term "DetNet OAM" used in this document interchangeably with longer version "set of OAM protocols, methods and tools for Deterministic Networks".

DetNet Deterministic Networks

DiffServ Differentiated Services

DSCP DiffServ Code Point

OAM: Operations, Administration and Maintenance

PREF Packet Replication and Elimination Function

POF Packet Ordering Function

RDI Remote Defect Indication

Underlay Network or Underlay Layer: The network that provides connectivity between the DetNet nodes. MPLS network providing LSP connectivity between DetNet nodes is an example of the underlay layer.

DetNet Node - a node that is an actor in the DetNet domain. DetNet domain edge node and node that performs PREF within the domain are examples of DetNet node.

2.2. Keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Active OAM for DetNet Networks with IP Data Plane

OAM protocols and mechanisms act within the data plane of the particular networking layer. And thus it is critical that the data plane encapsulation supports OAM mechanisms in such a way that DetNet OAM packets are in-band with a DetNet flow being monitored, i.e., DetNet OAM test packets follow precisely the same path as DetNet data plane traffic both for unidirectional and bi-directional DetNet paths.

The DetNet data plane encapsulation in a transport network with IP encapsulations specified in [I-D.ietf-detnet-ip]. For the IP underlay network, DetNet flows are identified by the 6-tuple that is the destination IP address, source IP address, IP protocol, source port number, destination port number, and differentiated services (DiffServ) code point (DSCP). Active IP OAM protocols like Bidirectional Forwarding Detection (BFD) [RFC5880] or STAMP [I-D.ietf-ippm-stamp], use UDP transport and the well-known UDP port numbers as the destination port. Thus a DetNet node should be able to associate an IP DetNet flow with the particular test session to

ensure that test packets experience the same treatment as the DetNet flow packets.

4. Use of Hybrid OAM in DetNet

Hybrid OAM methods are used in performance monitoring and defined in [RFC7799] as:

Hybrid Methods are Methods of Measurement that use a combination of Active Methods and Passive Methods.

A hybrid measurement method may produce metrics as close to passive, but it still alters something in a data packet even if that is the value of a designated field in the packet encapsulation. One example of such a hybrid measurement method is the Alternate Marking method (AMM) described in [RFC8321]. One of the advantages of the use of AMM in a DetNet domain with IP data plane is that the marking is applied to a data flow, thus ensuring that a measured metrics are directly applicable to the DetNet flow.

5. OAM of DetNet IP Interworking with OAM of DetNet MPLS

TBA

6. OAM of DetNet IP Interworking with OAM of TSN

TBA

7. IANA Considerations

This document does not have any requests for IANA allocation. This section can be deleted before the publication of the draft.

8. Security Considerations

This document describes the applicability of the existing Fault Management and Performance Monitoring IP OAM protocols, and does not raise any security concerns or issues in addition to ones common to networking or already documented for the referenced OAM protocols.

9. Acknowledgment

TBA

10. References

10.1. Normative References

- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", draft-ietf-
detnet-architecture-13 (work in progress), May 2019.
- [I-D.ietf-detnet-ip]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane: IP",
draft-ietf-detnet-ip-01 (work in progress), July 2019.
- [I-D.ietf-detnet-ip-over-mpls]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane: IP over
MPLS", draft-ietf-detnet-ip-over-mpls-01 (work in
progress), July 2019.
- [I-D.ietf-detnet-ip-over-tsn]
Varga, B., Farkas, J., Malis, A., Bryant, S., and J.
Korhonen, "DetNet Data Plane: IP over IEEE 802.1 Time
Sensitive Networking (TSN)", draft-ietf-detnet-ip-over-
tsn-00 (work in progress), May 2019.
- [I-D.mirsky-detnet-oam]
Mirsky, G. and M. Chen, "Operations, Administration and
Maintenance (OAM) for Deterministic Networks (DetNet)",
draft-mirsky-detnet-oam-03 (work in progress), May 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

10.2. Informational References

- [I-D.ietf-ippm-stamp]
Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple
Two-way Active Measurement Protocol", draft-ietf-ippm-
stamp-06 (work in progress), April 2019.

- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <<https://www.rfc-editor.org/info/rfc8321>>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

DetNet Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

G. Mirsky
ZTE Corp.
M. Chen
Huawei
July 8, 2019

Operations, Administration and Maintenance (OAM) for Deterministic
Networks (DetNet) with MPLS Data Plane
draft-mirsky-detnet-mpls-oam-00

Abstract

This document lists functional requirements for Operations, Administration, and Maintenance (OAM) toolset in Deterministic Networks (DetNet) and, using these requirements; defines format and use principals of the DetNet service Associated Channel over a DetNet network with the MPLS data plane..

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Keywords	4
3. Requirements	4
4. Active OAM for DetNet Networks with MPLS Data Plane	5
4.1. DetNet Active OAM Encapsulation	6
4.2. DetNet Replication, Elimination, and Ordering Sub- functions Interaction with Active OAM	9
5. Use of Hybrid OAM in DetNet	9
6. OAM of DetNet MPLS Interworking with OAM of DetNet IP	9
7. OAM of DetNet MPLS Interworking with OAM of TSN	9
8. IANA Considerations	9
9. Security Considerations	9
10. Acknowledgment	10
11. References	10
11.1. Normative References	10
11.2. Informational References	10
Authors' Addresses	11

1. Introduction

[I-D.ietf-detnet-architecture] introduces and explains Deterministic Networks (DetNet) architecture and how the Packet Replication and Elimination function (PREF) can be used to ensure low packet drop ratio in DetNet domain.

Operations, Administration and Maintenance (OAM) protocols are used to detect, localize defects in the network, and monitor network performance. Some OAM functions, e.g., failure detection, work in the network proactively, while others, e.g., defect localization, usually performed on-demand. These tasks achieved by a combination of active and hybrid, as defined in [RFC7799], OAM methods.

This document lists the functional requirements toward OAM for DetNet domain. The list can further be used for gap analysis of available OAM tools to identify possible enhancements of existing or whether new OAM tools are required to support proactive and on-demand path monitoring and service validation. Also, this document defines format and use principals of the DetNet service Associated Channel over a DetNet network with the MPLS data plane [I-D.ietf-detnet-mpls].

2. Conventions used in this document

2.1. Terminology

The term "DetNet OAM" used in this document interchangeably with longer version "set of OAM protocols, methods and tools for Deterministic Networks".

CW Control Word

DetNet Deterministic Networks

d-ACH DetNet Associated Channel Header

d-CW DetNet Control Word

DNH DetNet Header

GAL Generic Associated Channel Label

G-ACh Generic Associated Channel

OAM: Operations, Administration and Maintenance

PREF Packet Replication and Elimination Function

POF Packet Ordering Function

PW Pseudowire

RDI Remote Defect Indication

TSN Time-Sensitive Network

F-Label A Detnet "forwarding" label that identifies the LSP used to forward a DetNet flow across an MPLS PSN, e.g., a hop-by-hop label used between label switching routers (LSR).

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

Underlay Network or Underlay Layer: The network that provides connectivity between the DetNet nodes. MPLS network providing LSP connectivity between DetNet nodes is an example of the underlay layer.

DetNet Node - a node that is an actor in the DetNet domain. DetNet domain edge node and node that performs PREF within the domain are examples of DetNet node.

2.2. Keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Requirements

This section lists requirements for OAM in DetNet domain with MPLS data plane:

1. It MUST be possible to initiate DetNet OAM session from any DetNet node towards another DetNet node(s) within given domain.
2. It SHOULD be possible to initialize DetNet OAM session from a centralized controller.
3. DetNet OAM MUST support proactive and on-demand OAM monitoring and measurement methods.
4. DetNet OAM packets MUST be in-band, i.e., follow precisely the same path as DetNet data plane traffic.
5. DetNet OAM MUST support unidirectional OAM methods, continuity check, connectivity verification, and performance measurement.
6. DetNet OAM MUST support bi-directional OAM methods. Such OAM methods MAY combine in-band monitoring or measurement in the forward direction and out-of-bound notification in the reverse direction, i.e., from egress to ingress end point of the OAM test session.
7. DetNet OAM MUST support proactive monitoring of a DetNet node availability in the given DetNet domain.
8. DetNet OAM MUST support Path Maximum Transmission Unit discovery.
9. DetNet OAM MUST support Remote Defect Indication (RDI) notification to the DetNet node performing continuity checking.
10. DetNet OAM MUST support performance measurement methods.

11. DetNet OAM MAY support hybrid performance measurement methods.
 12. DetNet OAM MUST support unidirectional performance measurement methods. Calculated performance metrics MUST include but are not limited to throughput, packet loss, delay and delay variation metrics. [RFC6374] provides excellent details on performance measurement and performance metrics.
 13. DetNet OAM MUST support defect notification mechanism, like Alarm Indication Signal. Any DetNet node in the given DetNet domain MAY originate a defect notification addressed to any subset of nodes within the domain.
 14. DetNet OAM MUST support methods to enable survivability of the DetNet domain. These recovery methods MAY use protection switching and restoration.
 15. DetNet OAM MUST support the discovery of Packet Replication, Elimination, and Order preservation sub-functions locations in the domain.
 16. DetNet OAM MUST support testing of Packet Replication, Elimination, and Order preservation sub-functions in the domain.
 17. DetNet OAM MUST support monitoring any sub-set of paths traversed through the DetNet domain by the DetNet flow.
4. Active OAM for DetNet Networks with MPLS Data Plane

OAM protocols and mechanisms act within the data plane of the particular networking layer. And thus it is critical that the data plane encapsulation supports OAM mechanisms in such a way to comply with the above-listed requirements. One of such examples that require special consideration is requirement #5:

DetNet OAM packets MUST be in-band, i.e., follow precisely the same path as DetNet data plane traffic both for unidirectional and bi-directional DetNet paths.

The Det Net data plane encapsulation in transport network with MPLS encapsulation specified in [I-D.ietf-detnet-mpls]. For the MPLS underlay network, DetNet flows to be encapsulated analogous to pseudowires (PW) over MPLS packet switched network, as described in [RFC3985], [RFC4385]. Generic PW MPLS Control Word (CW), defined in [RFC4385], for DetNet displayed in Figure 1.

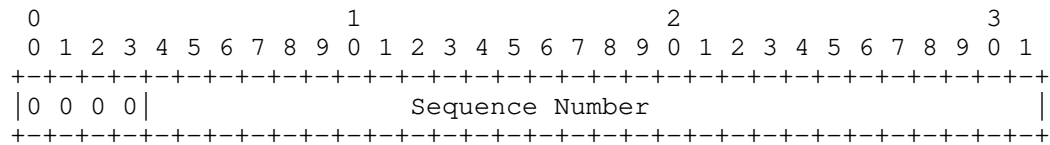


Figure 1: DetNet Control Word Format

PREF in the DetNet domain composed by a combination of nodes that perform replication and elimination sub-functions. The elimination sub-function always uses the S-Label and packet sequencing information, e.g., the value in the Sequence Number field of DetNet CW (d-CW). The replication sub-function uses the S-Label information only. For data packets Figure 2 presents an example of PREF in DetNet domain.

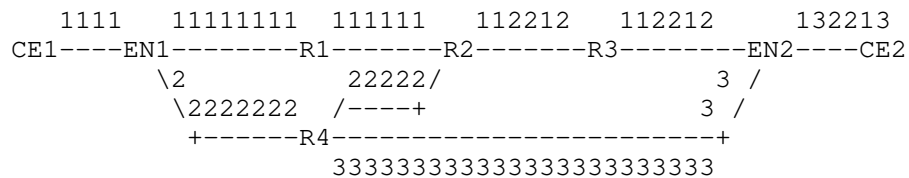


Figure 2: DetNet Data Plane Based on PW

4.1. DetNet Active OAM Encapsulation

DetNet OAM, like PW OAM, uses PW Associated Channel Header defined in [RFC4385]. Figure 3 displays the encapsulation of a DetNet MPLS [I-D.ietf-detnet-mpls] active OAM packet.

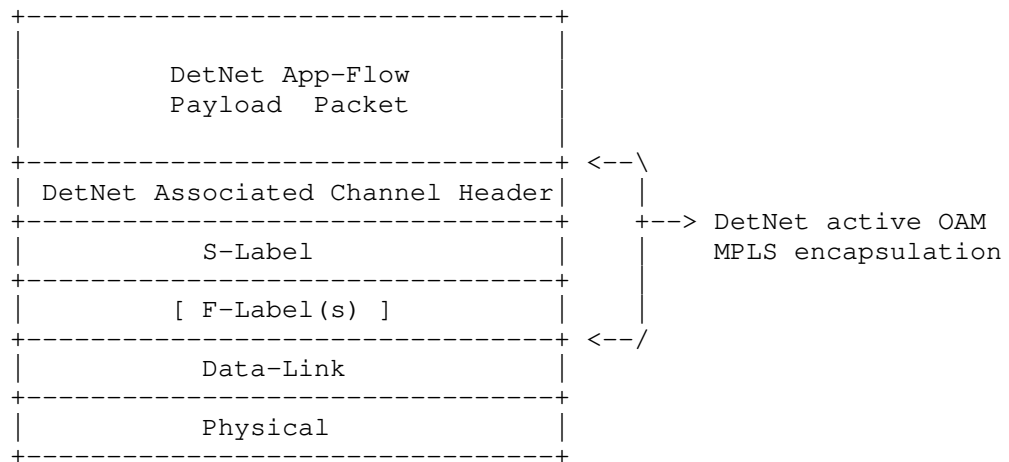


Figure 3: DetNet Active OAM Packet Encapsulation in MPLS Data Plane

Figure 4 displays encapsulation of a test packet of an active DetNet OAM protocol in case of MPLS-over-UDP/IP [I-D.ietf-detnet-mpls-over-udp-ip].

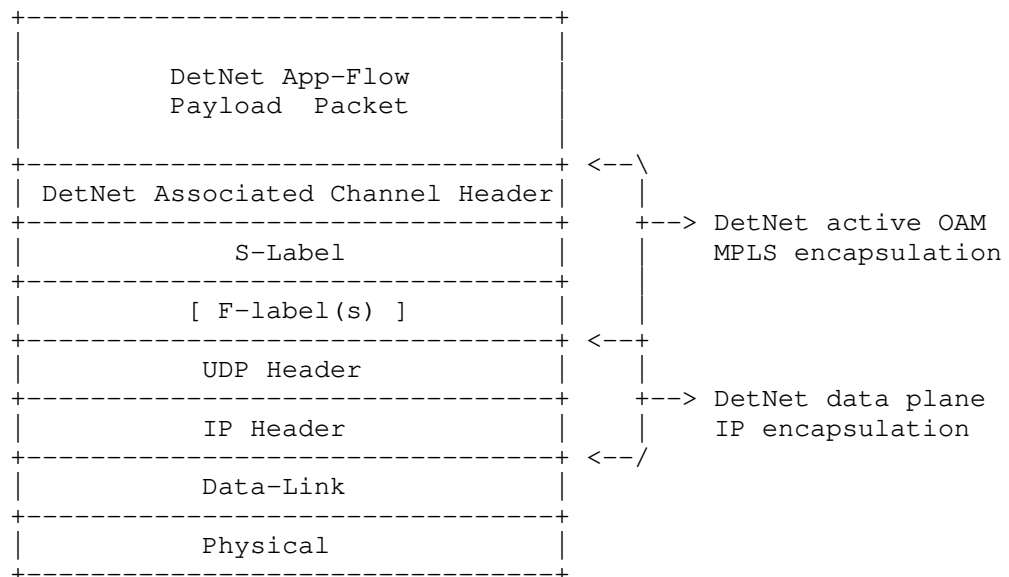


Figure 4: DetNet Active OAM Packet Encapsulation in MPLS-over-UDP/IP

Figure 5 displays the format of the DetNet Associated Channel Header (d-ACH).

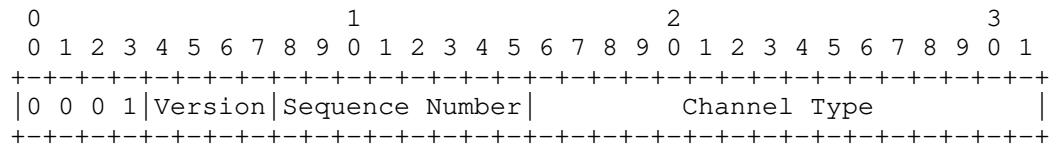


Figure 5: DetNet Associated Channel Header Format

The meanings of the fields in the d-ACH are:

Bits 0..3 MUST be 0b0001. This value of the first nibble allows the packet to be distinguished from an IP packet [RFC4928] and a DetNet data packet [I-D.ietf-detnet-mpls].

Version: this is the version number of the d-ACH. This specification defines version 0.

Sequence Number: this is unsigned eight bits-long field. The originating DetNet node MUST set the value of the Sequence Number field to a non-zero before packet being transmitted. The originating node MUST monotonically increase the value of the Sequence Number field for the every next active OAM packet.

Channel Type: the value of DetNet Associated Channel Type is one of values defined in the IANA PW Associated Channel Type registry.

The DetNet flow, according to [I-D.ietf-detnet-mpls], is identified by the S-label that MUST be at the bottom of the stack. Active OAM packet MUST have d-ACH immediately following the S-label.

Special consideration for DetNet active OAM with MPLS data plane interworking with OAM in IEEE 802.1 Time-Sensitive Networking (TSN) domain based on [I-D.ietf-detnet-mpls-over-tsn]:

- o Active OAM test packet MUST be mapped to the same TSN Stream ID as the monitored DetNet flow .
- o Active OAM test packets MUST be treated in the TSN domain based on its S-label and CoS marking (TC field value).

4.2. DetNet Replication, Elimination, and Ordering Sub-functions Interaction with Active OAM

At the DetNet service layer, special functions MAY be applied to the particular DetNet flow - PREF to potentially lower packet loss, improve the probability of on-time packet delivery and Packet Ordering Function (POF) to ensure in-order packet delivery. As data and the active OAM packets have the same Flow ID, S-label, sub-functions that rely on sequencing information in the DetNet service layer MUST process 28 MSBs of the d-ACH as the source of the sequencing information for the OAM packet.

5. Use of Hybrid OAM in DetNet

Hybrid OAM methods are used in performance monitoring and defined in [RFC7799] as:

Hybrid Methods are Methods of Measurement that use a combination of Active Methods and Passive Methods.

A hybrid measurement method may produce metrics as close to passive, but it still alters something in a data packet even if that is the value of a designated field in the packet encapsulation. One example of such a hybrid measurement method is the Alternate Marking method described in [RFC8321]. Reserving the field for the Alternate Marking method in the DetNet Header will enhance available to an operator set of DetNet OAM tools.

6. OAM of DetNet MPLS Interworking with OAM of DetNet IP

TBA

7. OAM of DetNet MPLS Interworking with OAM of TSN

TBA

8. IANA Considerations

TBA

9. Security Considerations

This document lists the OAM requirements for a DetNet domain and does not raise any security concerns or issues in addition to ones common to networking.

10. Acknowledgment

Authors extend their appreciation to Pascal Thubert for his insightful comments and productive discussion that helped to improve the document.

11. References

11.1. Normative References

- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", draft-ietf-
detnet-architecture-13 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS",
draft-ietf-detnet-mpls-01 (work in progress), July 2019.
- [I-D.ietf-detnet-mpls-over-tsn]
Varga, B., Farkas, J., Malis, A., Bryant, S., and J.
Korhonen, "DetNet Data Plane: MPLS over IEEE 802.1 Time
Sensitive Networking (TSN)", draft-ietf-detnet-mpls-over-
tsn-00 (work in progress), May 2019.
- [I-D.ietf-detnet-mpls-over-udp-ip]
Varga, B., Farkas, J., Berger, L., Malis, A., Bryant, S.,
and J. Korhonen, "DetNet Data Plane: MPLS over UDP/IP",
draft-ietf-detnet-mpls-over-udp-ip-01 (work in progress),
July 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

11.2. Informational References

- [RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation
Edge-to-Edge (PWE3) Architecture", RFC 3985,
DOI 10.17487/RFC3985, March 2005,
<<https://www.rfc-editor.org/info/rfc3985>>.

- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<https://www.rfc-editor.org/info/rfc4385>>.
- [RFC4928] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", BCP 128, RFC 4928, DOI 10.17487/RFC4928, June 2007, <<https://www.rfc-editor.org/info/rfc4928>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <<https://www.rfc-editor.org/info/rfc8321>>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com