Deterministic Networking Architecture

Abstract

This document provides 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 within a network
domain. Techniques used include 1) reserving data-plane resources
for individual (or aggregated) DetNet flows in some or all of the
intermediate nodes along the path of the flow, 2) providing explicit
routes for DetNet flows that do not immediately change with the
network topology, and 3) distributing data from DetNet flow packets
over time and/or space to ensure delivery of each packet's data in
spite of the loss of a path. DetNet operates at the IP layer and
delivers service over lower-layer technologies such as MPLS and Time-
Sensitive Networking (TSN) as defined by IEEE 802.1.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   This document provides the overall architecture for Deterministic
   Networking (DetNet), which provides a capability for the delivery of
   data flows with extremely low packet loss rates and bounded end-to-
   end delivery latency. DetNet is for networks that are under a single
   administrative control or within a closed group of administrative
   control; these include campus-wide networks and private WANs. DetNet

is not for large groups of domains such as the Internet.

DetNet operates at the IP layer and delivers service over lower-layer technologies such as MPLS and IEEE 802.1 Time-Sensitive Networking (TSN). DetNet provides a reliable and available service by dedicating network resources such as link bandwidth and buffer space to DetNet flows and/or classes of DetNet flows, and by replicating packets along multiple paths. Unused reserved resources are available to non-DetNet packets as long as all guarantees are fulfilled.

The "Deterministic Networking Problem Statement" [RFC8557] introduces DetNet, and "Deterministic Networking Use Cases" [RFC8578] summarizes the need for it. See [DETNET-FRAMEWORK] for specific techniques that can be used to identify DetNet flows and assign them to specific paths through a network.

A goal of DetNet is a converged network in all respects, including the convergence of sensitive non-IP networks onto a common network infrastructure. The presence of DetNet flows does not preclude non-DetNet flows, and the benefits offered DetNet flows should not, except in extreme cases, prevent existing Quality-of-Service (QoS) mechanisms from operating in a normal fashion, subject to the bandwidth required for the DetNet flows. A single source-destination pair can trade both DetNet and non-DetNet flows. End systems and applications need not instantiate special interfaces for DetNet flows. Networks are not restricted to certain topologies; connectivity is not restricted. Any application that generates a data flow that can be usefully characterized as having a maximum bandwidth should be able to take advantage of DetNet, as long as the necessary resources can be reserved. Reservations can be made by the application itself, via network management, centrally by an application's controller, or by other means, for instance, by placing on-demand reservation via a distributed Control Plane, e.g., leveraging the Resource Reservation Protocol (RSVP) [RFC2205]. QoS requirements of DetNet flows can be met if all network nodes in a DetNet domain implement DetNet capabilities. DetNet nodes can be interconnected with different sub-network technologies (Section 4.1.2) where the nodes of the subnet are not DetNet aware (Section 4.1.3).

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

2. Terminology

2.1. Terms Used in This Document

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

allocation
   The dedication of resources to support a DetNet flow. Depending
   on an implementation, the resource may be reused by non-DetNet
   flows when it is not used by the DetNet flow.

App-flow
   The payload (data) carried over a DetNet service.

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

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

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

DetNet edge node
   An instance of a DetNet relay node that acts as a source and/or
   destination at the DetNet service sub-layer. For example, it can
   include a DetNet service sub-layer proxy function for DetNet
   service protection (e.g., the addition or removal of packet
   sequencing information) for one or more end systems, it can start
   or terminate resource allocation at the DetNet forwarding sub-
   layer, or it can aggregate DetNet services into new DetNet flows.
   It is analogous to a Label Edge Router (LER) or a Provider Edge
   (PE) router.

DetNet flow
   A sequence of packets that conforms 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.

DetNet forwarding sub-layer
   DetNet functionality is divided into two sub-layers. One of them
   is the DetNet forwarding sub-layer, which optionally provides
   resource allocation for DetNet flows over paths provided by the
   underlying network.

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

DetNet node
   A DetNet edge node, a DetNet relay node, or a DetNet transit node.

DetNet relay node

A DetNet node that includes a service sub-layer function that
interconnects different DetNet forwarding sub-layer paths to
provide service protection. A DetNet relay node participates in
the DetNet service sub-layer. It typically incorporates DetNet
forwarding sub-layer functions as well, in which case it is
collocated with a transit node.

DetNet service sub-layer
   DetNet functionality is divided into two sub-layers. One of them
   is the DetNet service sub-layer, at which a DetNet service (e.g.,
   service protection) is provided.

DetNet service proxy
   A proxy that maps between App-flows and DetNet flows.

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

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

DetNet transit node
   A DetNet node, operating at the DetNet forwarding sub-layer, that
   utilizes link-layer and/or network-layer switching across multiple
   links and/or sub-networks to provide paths for DetNet service sub-
   layer functions. It typically provides resource allocation over
   those paths. An MPLS Label Switch Router (LSR) is an example of a
   DetNet transit node.

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

end system
   Commonly called a "host" in the RFC series and an "end station" in
   IEEE 802 standards. End systems of interest to this document are
   either sources or destinations of DetNet flows, and they may or
   may not be aware of DetNet forwarding sub-layers or DetNet service
   sub-layers.

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

Packet Elimination Function (PEF)
   A function that eliminates duplicate copies of packets to prevent
   excess packets flooding the network or duplicate packets being
   sent out of the DetNet domain. A PEF can be implemented by a
   DetNet edge node, a DetNet relay node, or an end system.

Packet Replication Function (PRF)
   A function that replicates DetNet flow packets and forwards them

to one or more next hops in the DetNet domain. The number of
packet copies sent to the next hops is a parameter specific to the
DetNet flow at the point of replication. A PRF can be implemented
by a DetNet edge node, a DetNet relay node, or an end system.

PREOF
A collective name for Packet Replication, Elimination, and
Ordering Functions.

Packet Ordering Function (POF)
A function that reorders packets within a DetNet flow that are
received out of order. This function can be implemented by a
DetNet edge node, a DetNet relay node, or an end system.

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

2.2. Dictionary of Terms Used by TSN and DetNet

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

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

Relay system
The term used by IEEE 802.1 for a DetNet intermediate node.

Stream
The term used by IEEE 802.1 for a DetNet flow.

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

3. Providing the DetNet Quality of Service

3.1. Primary Goals Defining the DetNet QoS

The DetNet QoS can be expressed in terms of:

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

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

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

It is a distinction of DetNet that it is concerned solely with worst-

case values for the end-to-end latency, jitter, and misordering.
Average, mean, or typical values are of little interest, because they
do not affect the ability of a real-time system to perform its tasks.
In general, a trivial priority-based queuing scheme will give better
average latency to a data flow than DetNet; however, it may not be a
suitable option for DetNet because of its worst-case latency.

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

*   Resource allocation (Section 3.2.1)

*   Service protection (Section 3.2.2)

*   Explicit routes (Section 3.2.3)

Resource allocation operates by assigning resources, e.g., buffer
space or link bandwidth, to a DetNet flow (or flow aggregate) along
its path. Resource allocation greatly reduces, or even eliminates
entirely, packet loss due to output packet contention within the
network, but it can only be supplied to a DetNet flow that is limited
at the source to a maximum packet size and transmission rate. As
DetNet flows are assumed to be rate limited and DetNet is designed to
provide sufficient allocated resources (including provisioned
capacity), the use of transport-layer congestion control [RFC2914]
for App-flows is not required; however, if resources are allocated
appropriately, use of congestion control should not impact
transmission negatively.

Resource allocation addresses two of the DetNet QoS requirements:
latency and packet loss. Given that DetNet nodes have a finite
amount of buffer space, resource allocation necessarily results in a
maximum end-to-end latency. Resource allocation also addresses
contention-related packet loss.

Other important contributions to packet loss are random media errors
and equipment failures. Service protection is the name for the
mechanisms used by DetNet to address these losses. The mechanisms
employed are constrained by the need to meet the users' latency
requirements. Packet replication and elimination (Section 3.2.2.2)
and packet encoding (Section 3.2.2.3) are described in this document
to provide service protection, but other mechanisms may also be
found. For instance, packet encoding can be used to provide service
protection against random media errors, while packet replication and
elimination can be used to provide service protection against
equipment failures. This mechanism distributes the contents of
DetNet flows over multiple paths in time and/or space, so that the
loss of some of the paths does need not cause the loss of any
packets.

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

These three techniques can be applied individually or applied
together; it results that eight combinations, including none (no

DetNet), are possible. Some combinations, however, are of wider
utility than others. This separation keeps the protocol stack
coherent and maximizes interoperability with existing and developing
standards in the IETF and other Standards Development Organizations.
The following are examples of typical expected combinations:

* The combination of explicit routes and service protection is the
   technique employed by seamless redundancy mechanisms applied on a
   ring topology, e.g., as described in [IEC-62439-3]. In this
   example, explicit routes are achieved by limiting the physical
   topology of the network to a ring. Sequentialization,
   replication, and duplicate elimination are facilitated by packet
   tags added at the front or the end of Ethernet frames. [RFC8227]
   provides another example in the context of MPLS.

* Resource allocation alone was originally offered by Audio Video
   Bridging as defined by IEEE 802.1 [IEEE802.1BA]. As long as the
   network suffers no failures, packet loss due to output packet
   contention can be eliminated through the use of a reservation
   protocol (e.g., the Multiple Stream Registration Protocol
   [IEEE802.1Q]), shapers in every bridge, and proper dimensioning.

* Using all three together gives maximum protection.

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

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

3.2. Mechanisms to Achieve DetNet QoS

3.2.1. Resource Allocation

3.2.1.1. Eliminate Contention Loss

The primary means by which DetNet achieves its QoS assurances is to
reduce, or even completely eliminate, packet loss due to output
packet contention within a DetNet node as a cause of packet loss.
This can be achieved only by the provision of sufficient buffer
storage at each node through the network to ensure that no packets
are dropped due to a lack of buffer storage. Note that App-flows are
generally not expected to be responsive to implicit [RFC2914] or
explicit congestion notification [RFC3168].

Ensuring adequate buffering requires, in turn, that the source and every DetNet node along the path to the destination (or nearly every node; see Section 4.3.3) be careful to regulate its output to not exceed the data rate for any DetNet flow, except for brief periods when making up for interfering traffic. Any packet sent ahead of its time potentially adds to the number of buffers required by the next-hop DetNet node and may thus exceed the resources allocated for a particular DetNet flow. Furthermore, rate limiting (e.g., using traffic policing) and shaping functions (e.g., shaping as defined in [RFC2475]) at the ingress of the DetNet domain must be applied. This is needed for meeting the requirements of DetNet flows as well as for protecting non-DetNet traffic from potentially misbehaving DetNet traffic sources. Note that large buffers have some issues (see, e.g., [BUFFERBLOAT]).

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

3.2.1.2. Jitter Reduction

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

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

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

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

Jitter reduction is provided by the mechanisms described in

Section 4.5 that also provide resource allocation.

3.2.2. Service Protection

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

3.2.2.1. In-Order Delivery

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

3.2.2.2. Packet Replication and Elimination

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

The DetNet service sub-layer includes the PRF, PEF, and POF for use
in DetNet edge, relay node, and end-system packet processing. These
functions can be enabled in a DetNet edge node, relay node, or end
system. The collective name for all three functions is Packet
Replication, Elimination, and Ordering Functions (PREOF). The packet
replication and elimination service protection method altogether
involves four capabilities:

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

*   The PRF replicates these packets into multiple DetNet member flows
    and typically sends them along multiple different paths to the
    destination(s), e.g., over the explicit routes described in
    Section 3.2.3. The location within a DetNet node and the
    mechanism used for the PRF are left open for implementations.

*   The PEF eliminates duplicate packets of a DetNet flow based on the

sequencing information and a history of received packets. The
output of the PEF is always a single packet. This may be done at
any DetNet node along the path to save network resources further
downstream, in particular if multiple replication points exist.
But the most common case is to perform this operation at the very
edge of the DetNet network, preferably in or near the receiver.
The location within a DetNet node and the mechanism used for the
PEF is left open for implementations.

* The POF uses the sequencing information to reorder a DetNet flow's
   packets that are received out of order.

The order in which a DetNet node applies PEF, POF, and PRF to a
DetNet flow is left open for implementations.

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

In the simplest case, this amounts to 1) replicating each packet in a
source that has two interfaces and 2) conveying them through the
network along separate (Shared Risk Link Group (SRLG) disjoint) paths
to the similarly dual-homed destinations that 3) reorder the packets
and 4) discard the duplicates. This ensures that one path remains,
even if some DetNet intermediate node fails. The sequencing
information can also be used for loss detection and for reordering.

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

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

```
            > > > > > > > > > relay > > > > > > > > >
             > /------------+ R node E +------------\ >
            > /             v + ^               \ >
 end     R +               v + ^                   + E end
 system    +               v | ^                   +   system
            > \             v + ^               / >
             > \------------+ R relay E +-----------/ >
              > > > > > > > > >  node > > > > > > > >
```
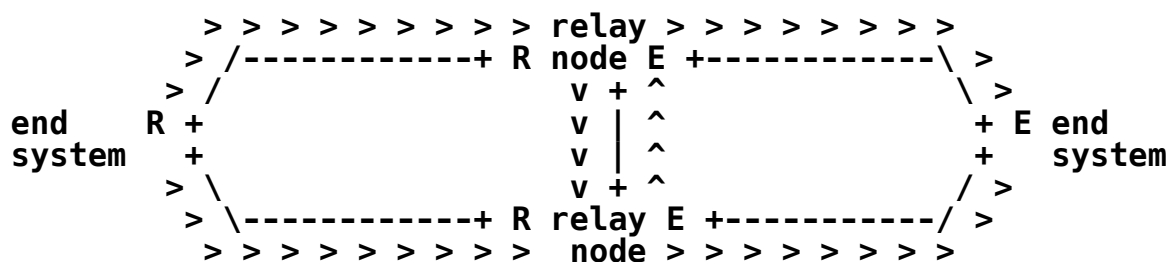
                 Figure 1: Packet Replication and Elimination

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

routing and bridging protocols.

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

3.2.2.3. Packet Encoding for Service Protection

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

3.2.3. Explicit Routes

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

Many real-time networks rely on physical rings of two-port devices, with a relatively simple ring control protocol. This supports redundant paths for service protection with a minimum of wiring. As an additional benefit, ring topologies can often utilize different topology management protocols from those used for a mesh network, with a consequent reduction in the response time to topology changes. Of course, this comes at some cost in terms of increased hop count, and thus latency, for the typical path.

In order to get the advantages of low hop count and still ensure against even very brief losses of connectivity, DetNet employs explicit routes where the path taken by a given DetNet flow does not change, at least not immediately and likely not at all, in response to network topology events. Service protection (see Sections 3.2.2 and 3.2.2.3) over explicit routes provides a high likelihood of continuous connectivity. Explicit routes can be established in various ways, e.g., with RSVP-TE [RFC3209], with Segment Routing (SR) [RFC8402], via a SDN approach [RFC8453], with IS-IS [RFC7813], etc. Explicit routes are typically used in MPLS TE (Traffic Engineering) Label Switched Paths (LSPs).

Out-of-order packet delivery can be a side effect of distributing a single flow over multiple paths, especially when there is a change from one path to another when combining the flow. This is irrespective of the distribution method used and also applies to service protection over explicit routes. As described in Section 3.2.2.1, out-of-order packets influence the jitter of a flow and impact the amount of buffering needed to process the data;

therefore, the guarantees of a DetNet service include a maximum
amount of misordering as a constraint. The use of explicit routes
helps to provide in-order delivery because there is no immediate
route change with the network topology, but the changes are plannable
as they are between the different explicit routes.

3.3. Secondary Goals for DetNet

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

3.3.1. Coexistence with Normal Traffic

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

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

*   DetNet flows can be shaped or scheduled, in order to ensure that
    the highest-priority non-DetNet packet is also ensured a worst-
    case latency.

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

Starvation of non-DetNet traffic must be avoided, for example, by
traffic policing and shaping functions (e.g., [RFC2475]). Thus, the
net effect of the presence of DetNet flows in a network on the non-
DetNet flows is primarily a reduction in the available bandwidth.

3.3.2. Fault Mitigation

Robust real-time systems require reducing the number of possible
failures. Filters and policers should be used in a DetNet network to
detect if DetNet packets are received on the wrong interface, at the
wrong time, or in too great a volume. Furthermore, filters and
policers can take actions to discard the offending packets or flows,
or trigger shutting down the offending flow or the offending
interface.

It is also essential that filters and service remarking be employed
at the network edge to prevent non-DetNet packets from being mistaken
for DetNet packets and thus impinging on the resources allocated to
DetNet packets. In particular, sending DetNet traffic into networks
that have not been provisioned in advance to handle that DetNet

traffic has to be treated as a fault. The use of egress traffic
filters, or equivalent mechanisms, to prevent this from happening are
strongly recommended at the edges of DetNet networks and DetNet
supporting networks. In this context, the term 'provisioned' has a
broad meaning, e.g., provisioning could be performed via an
administrative decision that the downstream network has the available
capacity to carry the DetNet traffic that is being sent into it.

Note that the sending of App-flows that do not use transport-layer
congestion control per [RFC2914] into a network that is not
provisioned to handle such traffic has to be treated as a fault and
prevented. PRF-generated DetNet member flows also need to be treated
as not using transport-layer congestion control even if the original
App-flow supports transport-layer congestion control because PREOF
can remove congestion indications at the PEF and thereby hide such
indications (e.g., drops, ECN markings, increased latency) from end
systems.

The mechanisms to support these requirements are both Data Plane and
implementation specific. Solutions that are data-plane specific will
be specified in the relevant data-plane solution document. There
also exist techniques, at present and/or in various stages of
standardization, that can support these fault-mitigation tasks that
deliver a high probability that misbehaving systems will have zero
impact on well-behaved DetNet flows with the exception, of course, of
the receiving interface(s) immediately downstream from the
misbehaving device. Examples of such techniques include traffic
policing and shaping functions (e.g., those described in [RFC2475]),
separating flows into per-flow rate-limited queues, and potentially
applying active queue management [RFC7567].

4. DetNet Architecture

4.1. DetNet Stack Model

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

4.1.1. Representative Protocol Stack Model

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

```
        |  packets going  |     ^  packets coming   ^
        v down the stack  v     |   up the stack     |
       +--------------------+   +--------------------+
       |       Source       |   |    Destination     |
       +--------------------+   +--------------------+
       |  Service sub-layer: |   |  Service sub-layer: |
       |   Packet sequencing |   | Duplicate elimination |
       |   Flow replication  |   |     Flow merging    |
```

```
|   Packet encoding   |   |   Packet decoding   |
+---------------------+   +---------------------+
| Forwarding sub-layer:|  | Forwarding sub-layer:|
|  Resource allocation |  |  Resource allocation |
|   Explicit routes    |  |   Explicit routes    |
+---------------------+   +---------------------+
|    Lower layers     |   |    Lower layers     |
+---------------------+   +---------------------+
          v                         ^
           _____/
```

                 Figure 2: DetNet Data-Plane Protocol Stack

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

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

Packet sequencing
    As part of the DetNet service sub-layer, the packet sequencing
    function supplies the sequence number for packet replication and
    elimination for DetNet service protection (Section 3.2.2.2); thus,
    its peer is duplicate elimination. This sub-layer is not needed
    if a higher-layer protocol is expected to perform any packet
    sequencing and duplicate elimination required by the DetNet flow
    replication.

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

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

Flow merging
    As part of the DetNet service sub-layer, the flow merging function
    combines DetNet member flows together for packets coming up the
    stack belonging to a specific DetNet compound flow. DetNet flow
    merging, together with packet sequencing, duplicate elimination,
    and DetNet flow replication perform packet replication and
    elimination (Section 3.2.2). Its peer is DetNet flow replication.

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

Packet decoding
   As part of DetNet service protection, as an alternative to flow
   merging and duplicate elimination, packet decoding takes packets
   from different DetNet member flows and computes from those packets
   the original DetNet packets from the compound flows input to
   packet encoding. Its peer is packet encoding.

Resource allocation
   The DetNet forwarding sub-layer provides resource allocation. See
   Section 4.5. The actual queuing and shaping mechanisms are
   typically provided by the underlying subnet. These can be closely
   associated with the means of providing paths for DetNet flows.
   The path and the resource allocation are conflated in this figure.

Explicit routes
   Explicit routes are arrangements of fixed paths operated at the
   DetNet forwarding sub-layer that are determined in advance to
   avoid the impact of network convergence on DetNet flows.

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

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

4.1.2. DetNet Data-Plane Overview

A "Deterministic Network" will be composed of DetNet-enabled end
systems, DetNet edge nodes, and DetNet relay nodes, which
collectively deliver DetNet services. DetNet relay and edge nodes
are interconnected via DetNet transit nodes (e.g., LSRs), which
support DetNet but are not DetNet service aware. All DetNet nodes
are connected to sub-networks, where a point-to-point link is also
considered a simple sub-network. These sub-networks provide DetNet-
compatible service for support of DetNet traffic. Examples of sub-
network technologies include MPLS TE, TSN as defined by IEEE 802.1,
and OTN (Optical Transport Network). Of course, multilayer DetNet

systems may also be possible, where one DetNet appears as a sub-
network and provides service to a higher-layer DetNet system. A
simple DetNet concept network is shown in Figure 3. Note that in
this and following figures, "Forwarding" and "Fwd" refer to the
DetNet forwarding sub-layer, and "Service" and "Svc" refer to the
DetNet service sub-layer; both of these sub-layers are described in
detail in Section 4.1.1.

```
 TSN                   Edge            Transit          Relay            DetNet
 End System            Node            Node             Node             End System

 +----------+    +.........+                                      +----------+
 |  Appl.   |<--:Svc Proxy:-- End-to-End Service -------->|  Appl.   |
 +----------+    +----------+                   +---------+    +----------+
 |   TSN    |    |TSN| |Svc|<- DetNet flow --: Service :-->| Service  |
 +----------+    +---+ +---+    +---------+    +---------+    +----------+
 |Forwarding|    |Fwd| |Fwd|    |   Fwd   |    |Fwd| |Fwd|    |Forwarding|
 +-------.--+    +-.-+ +-.-+    +--.----.+     +-.-+ +-.-+    +---.------+
         : Link :     / ,-----. \  : Link :     / ,-----. \
         +........+   +-[  Sub- ]-+ +.......+   +-[  Sub- ]-+
                        [network]                 [network]
                        `-----'                   `-----'
```
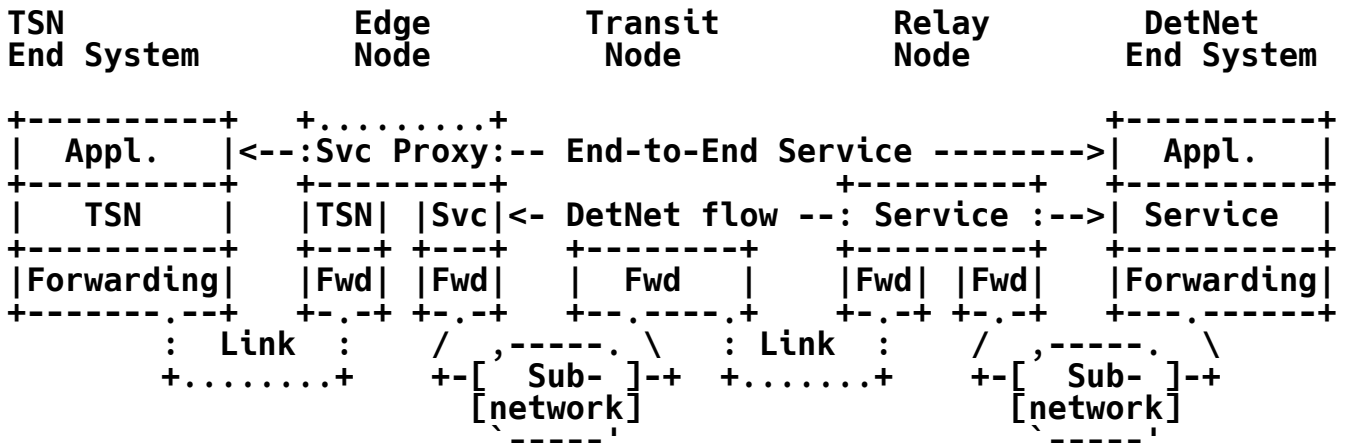
                Figure 3: A Simple DetNet-Enabled Network

DetNet Data Plane is divided into two sub-layers: the DetNet service
sub-layer and the DetNet forwarding sub-layer. This helps to explore
and evaluate various combinations of the data-plane solutions
available. Some of them are illustrated in Figure 4. This
separation of DetNet sub-layers, while helpful, should not be
considered a formal requirement. For example, some technologies may
violate these strict sub-layers and still be able to deliver a DetNet
service.

```
              .
              .
              .
  +------------------------------+
  |   DetNet Service sub-layer   | PW, UDP, GRE
  +------------------------------+
  | DetNet Forwarding sub-layer  | IPv6, IPv4, MPLS TE LSPs, MPLS SR
  +------------------------------+
              .
              .
```
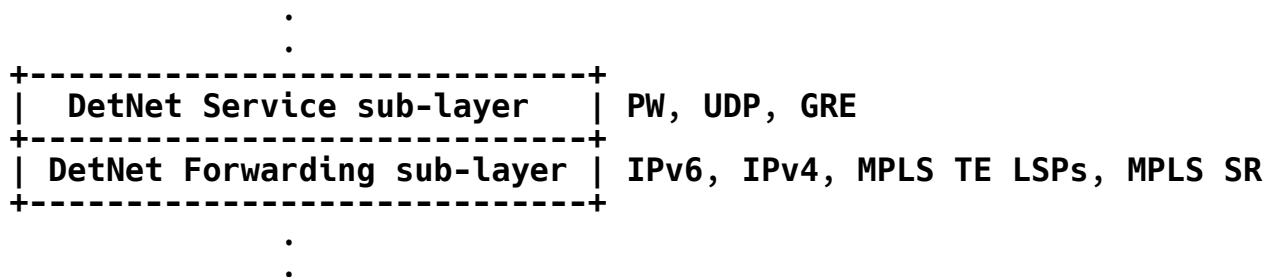
                Figure 4: DetNet Adaptation to Data Plane

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

There are many valid options to create a data-plane solution for
DetNet traffic by selecting a technology approach for the DetNet
service sub-layer and also selecting a technology approach for the

DetNet forwarding sub-layer. There are a large number of valid
combinations.

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

4.1.3. Network Reference Model

Figure 5 shows another view of the DetNet service-related reference
points and main components.
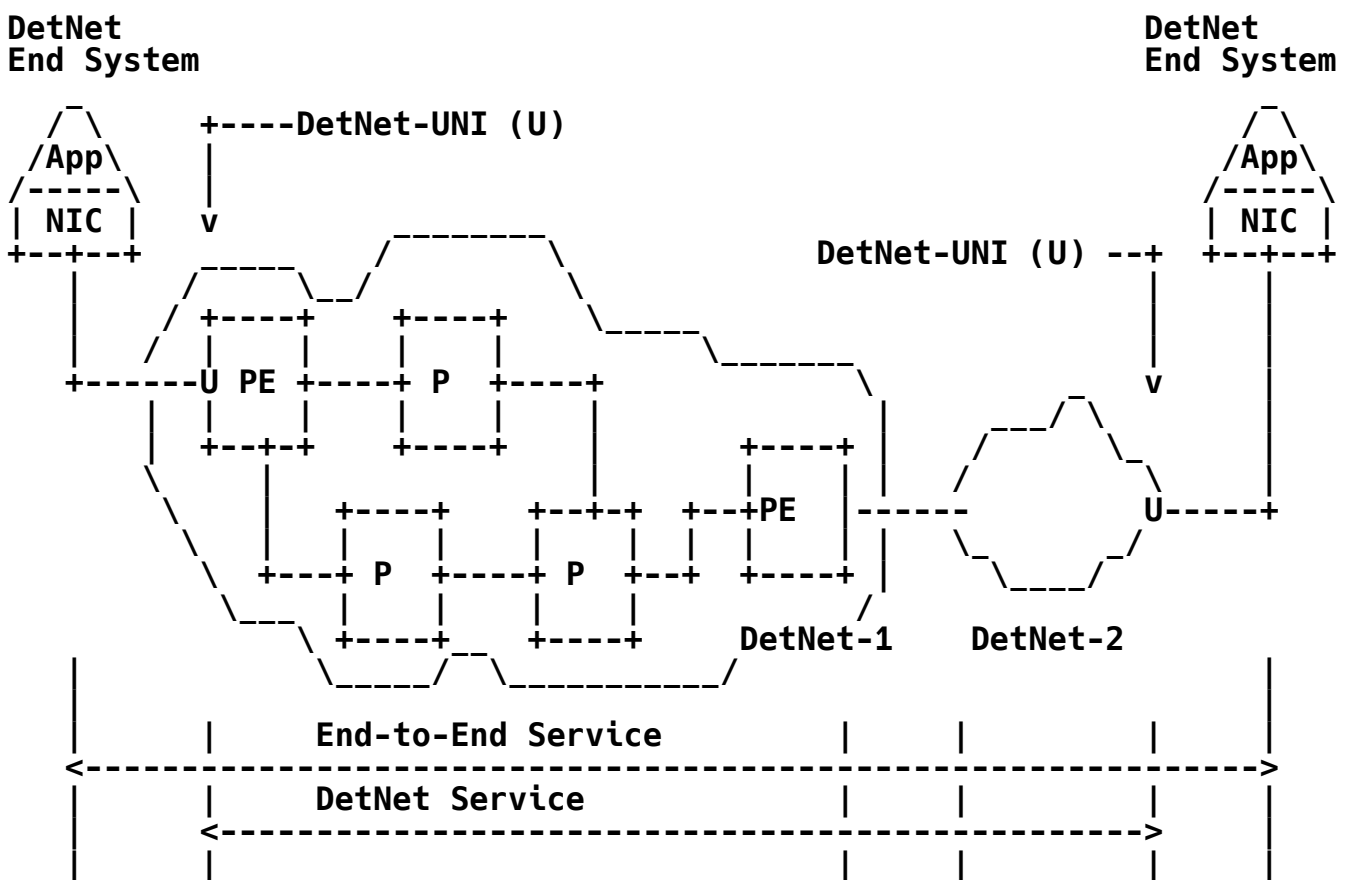
Figure 5: DetNet Service Reference Model (Multidomain)

DetNet User-to-Network Interfaces (DetNet-UNIs) ("U" in Figure 5) are
assumed in this document to be packet-based reference points and
provide connectivity over the packet network. A DetNet-UNI may
provide multiple functions. For example, it may:

* add encapsulation specific to networking technology to the DetNet
  flows if necessary,

*   provide status of the availability of the resources associated
    with a reservation,

*   provide a synchronization service for the end system, or

*   carry enough signaling to place the reservation in a network
    without a controller or in a network where the controller only
    deals with the network but not the end systems.

Internal reference points of end systems (between the application and
the Network Interface Card (NIC)) are more challenging from the
control perspective, and they may have extra requirements (e.g., in-
order delivery is expected in end system internal reference points,
whereas it is considered optional over the DetNet-UNI).

4.2. DetNet Systems

4.2.1. End System

The traffic characteristics of an App-flow can be CBR (constant bit
rate) or VBR (variable bit rate) and can have Layer 1, Layer 2, or
Layer 3 encapsulation (e.g., TDM (time-division multiplexing)
Ethernet, IP). These characteristics are considered as input for
resource reservation and might be simplified to ensure determinism
during packet forwarding (e.g., making reservations for the peak rate
of VBR traffic, etc.).

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

```
                   End system
                        |
                        |
                        |    DetNet aware ?
                        / \
               +------<   >------+
          NO   |        \ /      |   YES
               |         v       |
          DetNet-unaware         |
            End system           |
                                 |    Service/Forwarding
                                 |     sub-layer
                                 / \    aware ?
               +--------<   >-------------+
        f-aware |         \ /             |   s-aware
               |          v              |
               |          | both         |
               |          |              |
          DetNet f-aware  |         DetNet s-aware
            End system    |           End system
                          v
                   DetNet sf-aware
```
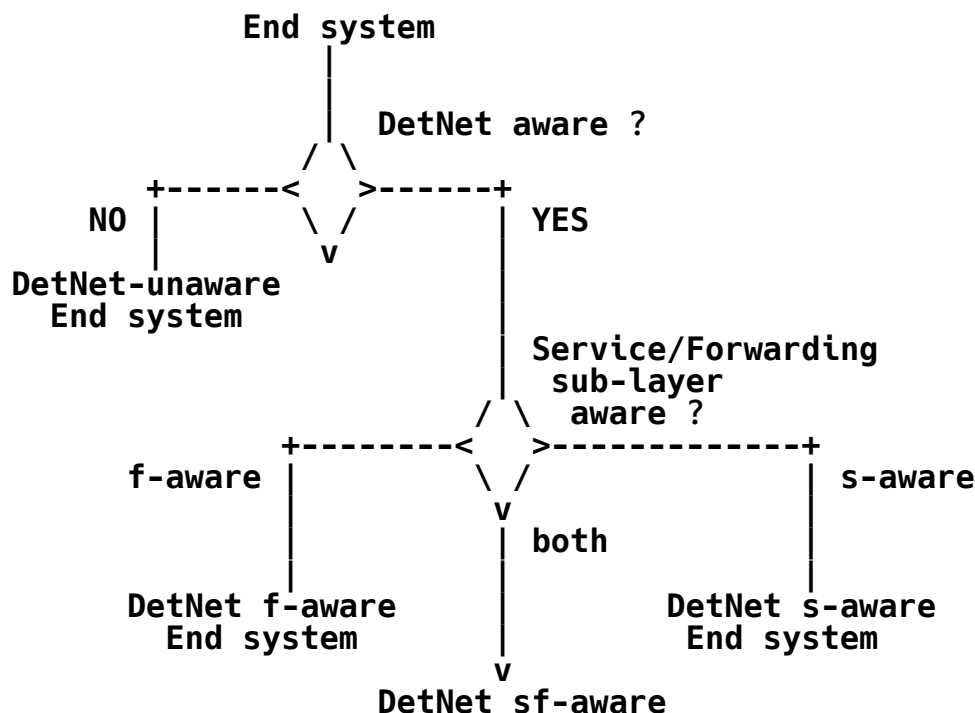
```
                         End system

              Figure 6: Categorization of End Systems

   The following are some known use case examples for end systems:

   DetNet unaware
      The classic case requiring service proxies.

   DetNet f-aware
      A system that is aware of the DetNet forwarding sub-layer.  It
      knows about some TSN functions (e.g., reservation) but not about
      service protection.

   DetNet s-aware
      A system that is aware of the DetNet service sub-layer.  It
      supplies sequence numbers but doesn't know about resource
      allocation.

   DetNet sf-aware
      A fully functioning DetNet end system.  It has DetNet
      functionalities and usually the same forwarding paradigm as the
      connected DetNet domain.  It can be treated as an integral part of
      the DetNet domain.
```

4.2.2. DetNet Edge, Relay, and Transit Nodes

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

In general, if a DetNet flow passes through one or more DetNet-unaware network nodes between two DetNet nodes providing the DetNet forwarding sub-layer for that flow, there is a potential for disruption or failure of the DetNet QoS. A network administrator needs to 1) ensure that the DetNet-unaware network nodes are configured to minimize the chances of packet loss and delay and 2) provision enough extra buffer space in the DetNet transit node following the DetNet-unaware network nodes to absorb the induced latency variations.

4.3. DetNet Flows

4.3.1. DetNet Flow Types

A DetNet flow can have different formats while its packets are forwarded between the peer end systems depending on the type of the end systems. Corresponding to the end system types, the following possible types/formats of a DetNet flow are distinguished in this document. The different flow types have different requirements to DetNet nodes.

```
   App-flow
      The payload (data) carried over a DetNet flow between DetNet-
      unaware end systems.  An App-flow does not contain any DetNet-
```

related attributes and does not imply any specific requirement on
DetNet nodes.

DetNet-f-flow
   The specific format of a DetNet flow. It only requires the
   resource allocation features provided by the DetNet forwarding
   sub-layer.

DetNet-s-flow
   The specific format of a DetNet flow. It only requires the
   service protection feature ensured by the DetNet service sub-
   layer.

DetNet-sf-flow
   The specific format of a DetNet flow. It requires both the DetNet
   service sub-layer and the DetNet forwarding sub-layer functions
   during forwarding.

4.3.2. Source Transmission Behavior

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

In contrast, asynchronous DetNet flows are not coordinated with a
fine-grained schedule, so relay and end systems must assume worst-
case interference among DetNet flows contending for buffer resources.
Asynchronous DetNet flows are characterized by:

* A maximum packet size;

* An observation interval; and

* A maximum number of transmissions during that observation
   interval.

These parameters, together with knowledge of the protocol stack used
(and thus the size of the various headers added to a packet), provide
the bandwidth that is needed for the DetNet flow.

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

a long period of time.

There is no expectation in DetNet for App-flows to be responsive to congestion control [RFC2914] or explicit congestion notification [RFC3168]. The assumption is that a DetNet flow, to be useful, must be delivered in its entirety. That is, while any useful application is written to expect a certain number of lost packets, the real-time applications of interest to DetNet demand that the loss of data due to the network is a rare event.

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

All nodes in a DetNet domain are expected to support the data behavior required to deliver a particular DetNet service. If a node itself is not DetNet service aware, the DetNet nodes that are adjacent to them must ensure that the node that is non-DetNet aware is provisioned to appropriately support the DetNet service. For example, a TSN node (as defined by IEEE 802.1) may be used to interconnect DetNet-aware nodes, and these DetNet nodes can map DetNet flows to 802.1 TSN flows. As another example, an MPLS-TE or MPLS-TP (Transport Profile) domain may be used to interconnect DetNet-aware nodes, and these DetNet nodes can map DetNet flows to TE LSPs, which can provide the QoS requirements of the DetNet service.

4.3.3. Incomplete Networks

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

4.4. Traffic Engineering for DetNet

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

Because of its very nature of establishing explicit optimized paths, DetNet can be seen as a new, specialized branch of TE, and it inherits its architecture with a separation into planes.

The DetNet architecture is thus composed of three planes: a (User) Application Plane, a Controller Plane, and a Network Plane. This echoes the composition of Figure 1 of "Software-Defined Networking (SDN): Layers and Architecture Terminology" [RFC7426] and the controllers identified in [RFC8453] and [RFC7149].

4.4.1. The Application Plane

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

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

4.4.2. The Controller Plane

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

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

One or more CPFs collaborate to implement the requests from the FME as per-flow, per-hop behaviors installed in the DetNet nodes for each individual flow. The CPFs place each flow along a deterministic arrangement of DetNet nodes so as to respect per-flow constraints such as security and latency, and to optimize the overall result for metrics such as an abstract aggregated cost. The deterministic arrangement can typically be more complex than a direct arrangement and include redundant paths with one or more packet replication and elimination points. Scaling to larger networks is discussed in Section 4.9.

4.4.3. The Network Plane

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

The Network Plane comprises the Network Interface Cards (NICs) in the end systems, which are typically IP hosts, and DetNet nodes, which are typically IP routers and MPLS switches.

A Southbound (Network) Interface enables the entities in the Controller Plane to communicate with devices in the Network Plane as illustrated in Figure 7. This interface leverages and extends TEAS to describe the physical topology and resources in the Network Plane.

```
   End                                                          End
   System                                                       System


   -+-+-+-+-+-+-+ Northbound -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

            CPF           CPF              CPF              CPF

   -+-+-+-+-+-+-+ Southbound -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

            DetNet        DetNet        DetNet        DetNet
            Node          Node          Node          Node
   NIC                                                          NIC
            DetNet        DetNet        DetNet        DetNet
            Node          Node          Node          Node
```
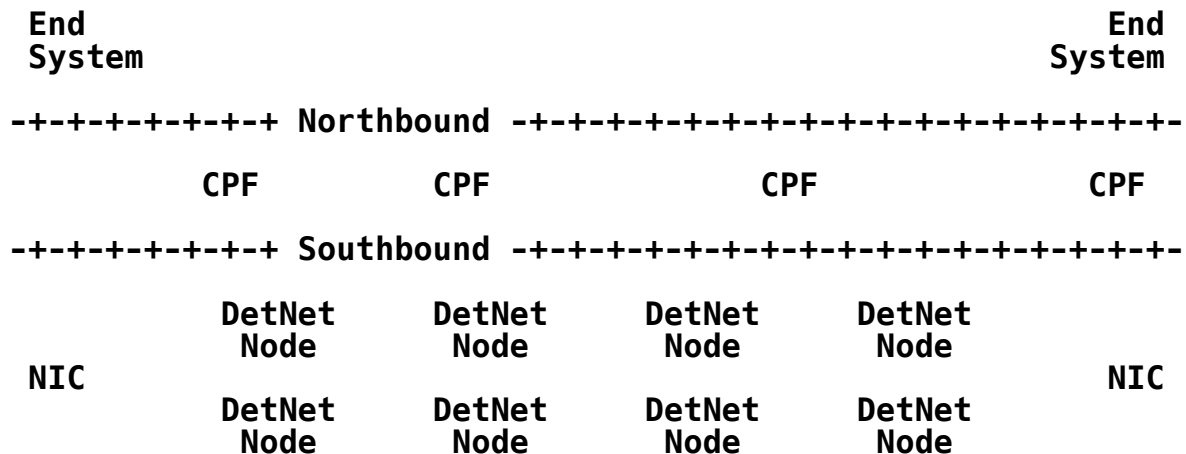
             Figure 7: Northbound and Southbound Interfaces

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

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

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

4.5. Queuing, Shaping, Scheduling, and Preemption

DetNet achieves bounded delivery latency by reserving bandwidth and buffer resources at each DetNet node along the path of the DetNet flow. The reservation itself is not sufficient, however. Implementors and users of a number of proprietary and standard real-time networks have found that standards for specific data-plane techniques are required to enable these assurances to be made in a multivendor network. The fundamental reason is that latency

variation in one DetNet system results in the need for extra buffer space in the next-hop DetNet system(s), which in turn increases the worst-case per-hop latency.

Standard queuing and transmission-selection algorithms allow TE (Section 4.4) to compute the latency contribution of each DetNet node to the end-to-end latency, to compute the amount of buffer space required in each DetNet node for each incremental DetNet flow, and most importantly, to translate from a flow specification to a set of values for the managed objects that control each relay or end system. For example, the IEEE 802.1 WG has specified (and is specifying) a set of queuing, shaping, and scheduling algorithms that enable each DetNet node, and/or a central controller, to compute these values. These algorithms include:

*   A credit-based shaper [IEEE802.1Qav] (incorporated to [IEEE802.1Q]).

*   Time-gated queues governed by a rotating time schedule based on synchronized time [IEEE802.1Qbv] (incorporated to [IEEE802.1Q]).

*   Synchronized double (or triple) buffers driven by synchronized time ticks. [IEEE802.1Qch] (incorporated to [IEEE802.1Q]).

*   Preemption of an Ethernet packet in transmission by a packet with a more stringent latency requirement, followed by the resumption of the preempted packet [IEEE802.1Qbu] (incorporated to [IEEE802.1Q]) [IEEE802.3br] (incorporated to [IEEE802.3]).

While these techniques are currently embedded in Ethernet [IEEE802.3] and bridging standards, we can note that they are all, except perhaps for packet preemption, equally applicable to media other than Ethernet and to routers as well as bridges. Other media may have their own methods (see, e.g., [TSCH-ARCH] and [RFC7554]). Further techniques are defined by the IETF (e.g., [RFC8289] and [RFC8033]). DetNet may include such definitions in the future or may define how these techniques can be used by DetNet nodes.

4.6. Service Instance

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

The DetNet network general reference model is shown in Figure 8 for a DetNet service scenario (i.e., between two DetNet-UNIs). In this figure, end systems ("A" and "B") are connected directly to the edge nodes of an IP/MPLS network ("PE1" and "PE2"). End systems participating in DetNet communication may require connectivity before setting up an App-flow that requires the DetNet service. Such a connectivity-related service instance and the one dedicated for DetNet service share the same access. Packets belonging to a DetNet flow are selected by a filter configured on the access ("F1" and "F2"). As a result, data-flow-specific access ("access-A + F1" and "access-B + F2") is terminated in the flow-specific service instance ("SI-1" and "SI-2"). A tunnel is used to provide connectivity between the service instances.

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

```
               access-A                                      access-B
                <----->      <-------- tunnel ---------->     <----->

                           +---------+                +---------+
End system     | +----+ |                 _        | +----+ | End system
   "A" -------+F1+   | |         / ‾‾\ /‾\_      | |   +F2----- "B"
              |  |    | |       /        \       | |    |  |
              |  |SI-1| +=======+ IP/MPLS +======+ |SI-2|  |
              |  +----+ |       \__ Net._/        | +----+  |
              |PE1      |         \____/           |     PE2|
               +---------+                          +---------+
```
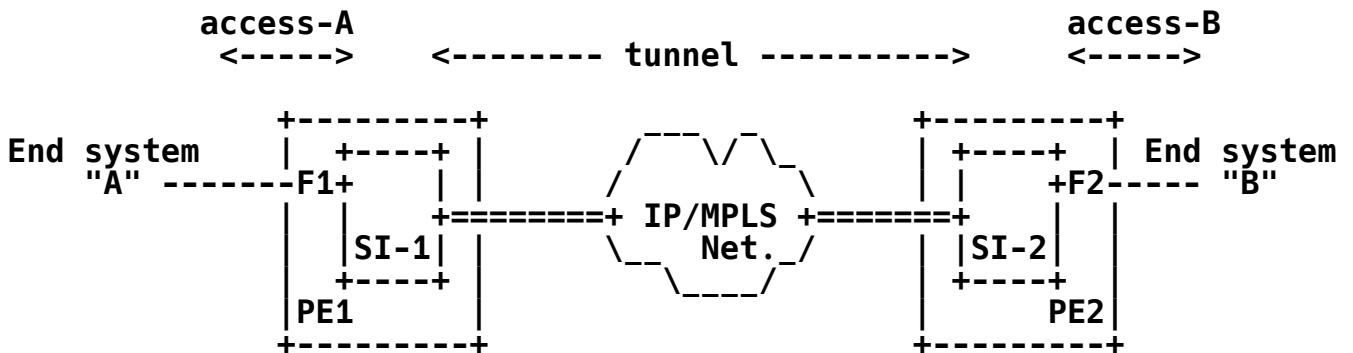
              Figure 8: DetNet Network General Reference Model

The tunnel between the service instances may have some special
characteristics. For example, in case of a DetNet L3 service, there
are differences in the usage of the PW for DetNet traffic compared to
the network model described in [RFC6658]. In the DetNet scenario,
the PW is likely to be used exclusively by the DetNet flow, whereas
[RFC6658] states:

| The packet PW appears as a single point-to-point link to the
| client layer. Network-layer adjacency formation and maintenance
| between the client equipments will follow the normal practice
| needed to support the required relationship in the client layer.

   and

| This packet pseudowire is used to transport all of the required
| layer 2 and layer 3 protocols between LSR1 and LSR2.

Further details are network technology specific and can be found in
[DETNET-FRAMEWORK].

4.7. Flow Identification at Technology Borders

This section discusses what needs to be done at technology borders
including Ethernet as one of the technologies. Flow identification
for MPLS and IP Data Planes are described in [DETNET-MPLS] and
[DETNET-IP], respectively.

4.7.1. Exporting Flow Identification

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

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

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

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

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

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

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

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

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

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

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

4.7.2. Flow Attribute Mapping between Layers

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

* IP routing

* MPLS label switching

* Ethernet bridging

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

```
    add/remove        add/remove
    Eth Flow-ID       IP Flow-ID
        |                 |
        v                 v
    +-------------------------------------------------------------+
    |      |      |      |                                        |
    | Eth  | MPLS |  IP  |        Application data                |
    |      |      |      |                                        |
    +-------------------------------------------------------------+
           ^
           |
        add/remove
      MPLS Flow-ID
```
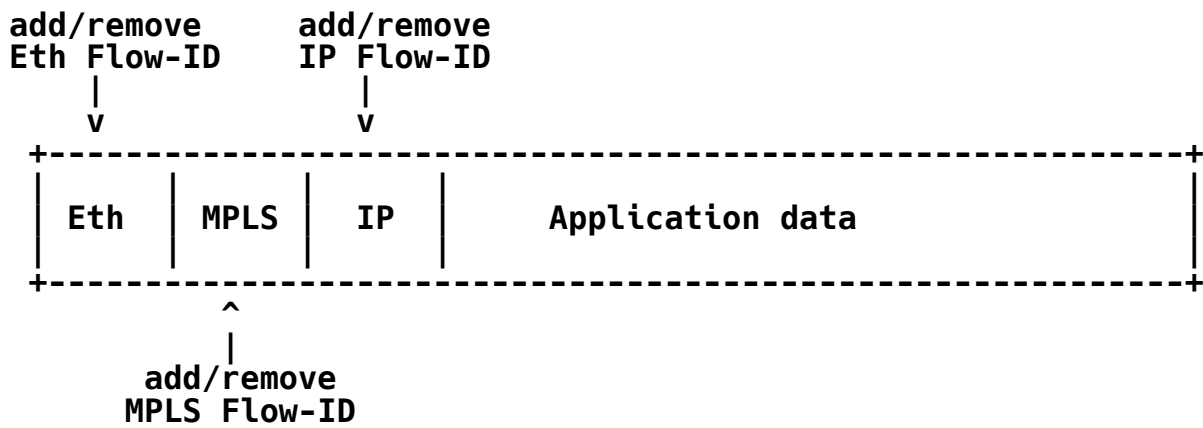
                Figure 9: Packet with Multiple Flow-IDs

The additional (domain-specific) Flow-ID can be:

* created by a domain-specific function or

* derived from the Flow-ID added to the App-flow.

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

4.7.3. Flow-ID Mapping Examples

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

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

                                 IP domain
            <---

```
        +======+                               +======+
        |L3-ID |                               |L3-ID |
        +======+   /\                  +-----+ +======+
                  /  \       Forward as |     |
                 /IP-A\      per ETH-ID |IP-1 |     Recognize
Push  ------>  +-+----+         |       +---+-+  <----- ETH-ID
ETH-ID         |              +----+-----+         |
               |              v         v          |
               |           +-----+   +-----+       |
               +------+    +------+   |     |   +---------+
        +......+     |ETH-1+----+ETH-2|   +-----+
        .L3-ID .     +-----+   +-----+      +======+
        +======+          +......+          |L3-ID |
        |ETH-ID|          .L3-ID .          +======+
        +======+          +======+          |ETH-ID|
                          |ETH-ID|          +------+
                          +======+


                        Ethernet domain
                      <---------------->


            Figure 10: IP Nodes Interconnected by an Ethernet Domain

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

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


                              MPLS domain
                  <------------------------------------------>

        +=======+                               +======+
        |MPLS-ID|                               |MPLS-ID|
        +=======+  +-----+                +-----+ +=======+ +-----+
                   |     |    Forward as   |     | |       | |     |
                   |PE-1 |    per ETH-ID   | P-2 +-----------+ PE-2|
Push  ----->  +-+----+         |          +---+-+          +-----+
ETH-ID         |              +-----+----+        \ Recognize
               |              v         v          +-- ETH-ID
               |           +-----+   +-----+       |
               +---+      +----+    |     |   +----+
        +.......+    |ETH-1+----+ETH-2|     +======+
        .MPLS-ID.    +-----+   +-----+      |MPLS-ID|
        +=======+          +.......+        +=======+
        |ETH-ID |          .MPLS-ID.        |ETH-ID |
        +=======+          +=======+        +-------+
```
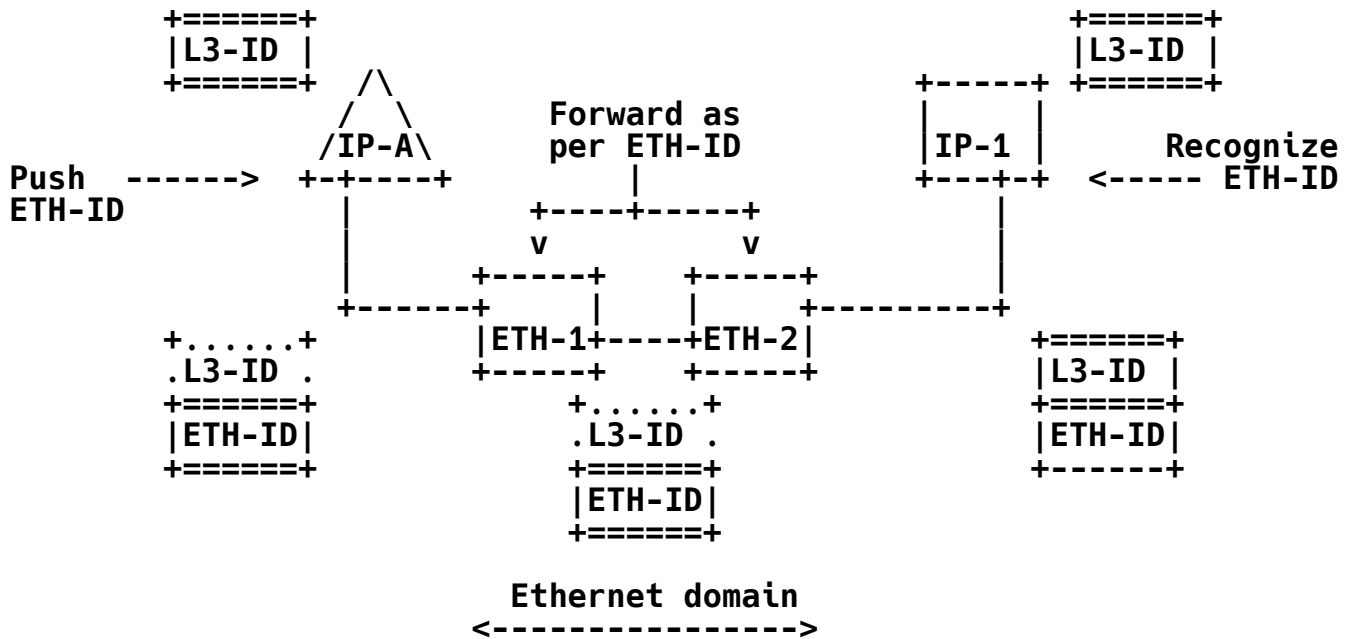
```
                          |ETH-ID |
                          +=======+
                         Ethernet domain
                      <---------------->
```
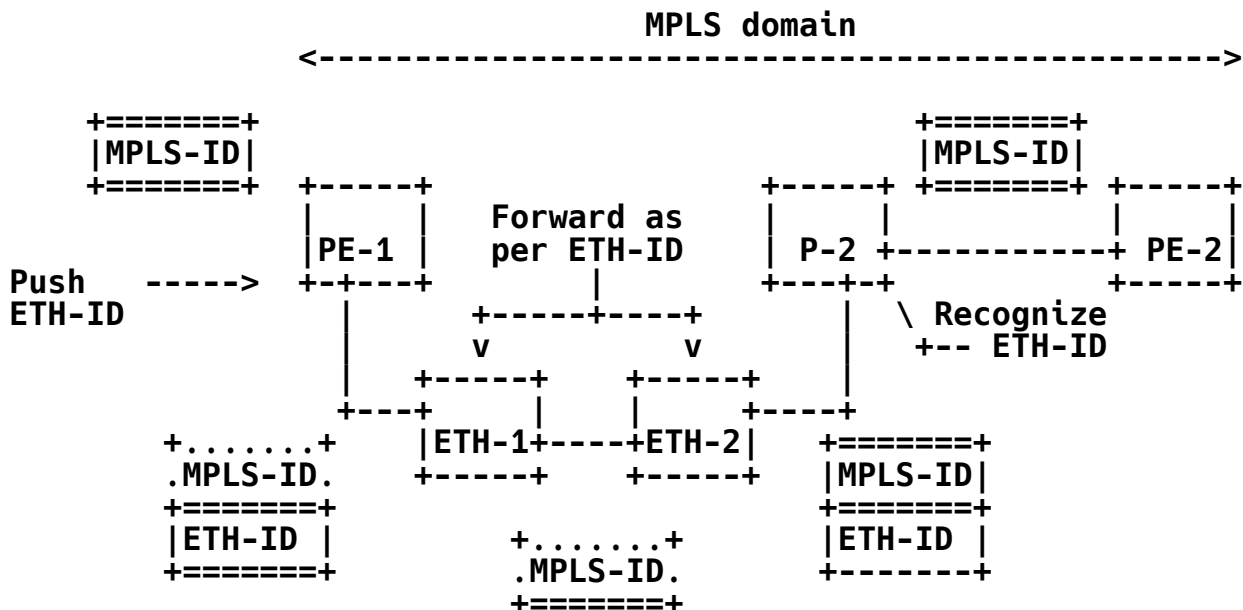
              Figure 11: MPLS Nodes Interconnected by an Ethernet Domain

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

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

4.8. Advertising Resources, Capabilities, and Adjacencies

   Provisioning of DetNet requires knowledge about:

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

   * The actual state of a DetNet node's DetNet resources.

   * The identity of the DetNet system's neighbors and the
      characteristics of the link(s) between the DetNet systems,
      including the latency of the links (in nanoseconds).

4.9. Scaling to Larger Networks

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

4.10. Compatibility with Layer 2

   Standards providing similar capabilities for bridged networks (only)
   have been and are being generated in the IEEE 802 LAN/MAN Standards

Committee. The present architecture describes an abstract model that
can be applicable both at Layer 2 and Layer 3, and over links not
defined by IEEE 802.

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

5. Security Considerations

Security considerations for DetNet are described in detail in
[DETNET-SECURITY]. This section considers exclusively security
considerations that are specific to the DetNet architecture.

Security aspects that are unique to DetNet are those whose aim is to
provide the specific QoS aspects of DetNet, which are primarily to
deliver data flows with extremely low packet loss rates and bounded
end-to-end delivery latency. A DetNet may be implemented using MPLS
and/or IP (including both v4 and v6) technologies and thus inherits
the security properties of those technologies at both the Data Plane
and the Controller Plane.

Security considerations for DetNet are constrained (compared to, for
example, the open Internet) because DetNet is defined to operate only
within a single administrative domain (see Section 1). The primary
considerations are to secure the request and control of DetNet
resources, maintain confidentiality of data traversing the DetNet,
and provide the availability of the DetNet QoS.

To secure the request and control of DetNet resources, authentication
and authorization can be used for each device connected to a DetNet
domain, most importantly to network controller devices. Control of a
DetNet network may be centralized or distributed (within a single
administrative domain). In the case of centralized control,
precedent for security considerations as defined for Abstraction and
Control of Traffic Engineered Networks (ACTN) can be found in
Section 9 of [RFC8453]. In the case of distributed control
protocols, DetNet security is expected to be provided by the security
properties of the protocols in use. In any case, the result is that
manipulation of administratively configurable parameters is limited
to authorized entities.

To maintain confidentiality of data traversing the DetNet,
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 by MACSec
[IEEE802.1AE] for Ethernet (Layer 2) flows.

DetNet flows are identified on a per-flow basis, which may provide
attackers with additional information about the data flows (when
compared to networks that do not include per-flow identification).
This is an inherent property of DetNet that 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 QoS, 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 traffic admission control applied at the input of a DetNet domain as described in Section 3.2.1 and through the fault-mitigation methods described in Section 3.3.2. 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.

Because DetNet mechanisms or applications that rely on DetNet can make heavy use of methods that require precise time synchronization, the accuracy, availability, and integrity of time synchronization is of critical importance. Extensive discussion of this topic can be found in [RFC7384].

DetNet use cases are known to have widely divergent security requirements. The intent of this section is to provide a baseline for security considerations that are common to all DetNet designs and implementations, without burdening individual designs with specifics of security infrastructure that may not be germane to the given use case. Designers and implementors of DetNet systems are expected to take use-case-specific considerations into account in their DetNet designs and implementations.

6. Privacy Considerations

DetNet provides a QoS, and the generic considerations for such mechanisms apply. In particular, such markings allow for an attacker to correlate flows or to select particular types of flow for more detailed inspection.

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

7. IANA Considerations

This document has no IANA actions.

8. Informative References

[BUFFERBLOAT]
          Gettys, J. and K. Nichols, "Bufferbloat: Dark Buffers in
          the Internet", DOI 10.1145/2063176.2063196, Communications
          of the ACM, Volume 55, Issue 1, January 2012,
          <https://doi.org/10.1145/2063176.2063196>.

[CCAMP]   IETF, "Common Control and Measurement Plane (ccamp)",
          October 2019,

          <https://datatracker.ietf.org/wg/ccamp/charter/>.

[DETNET-FRAMEWORK]
          Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
          Bryant, S., and J. Korhonen, "DetNet Data Plane
          Framework", Work in Progress, Internet-Draft, draft-ietf-
          detnet-data-plane-framework-02, 13 September 2019,
          <https://tools.ietf.org/html/draft-ietf-detnet-data-plane-
          framework-02>.

[DETNET-IP]
          Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
          Bryant, S., and J. Korhonen, "DetNet Data Plane: IP", Work
          in Progress, Internet-Draft, draft-ietf-detnet-ip-01, 1
          July 2019,
          <https://tools.ietf.org/html/draft-ietf-detnet-ip-01>.

[DETNET-MPLS]
          Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A.,
          Bryant, S., and J. Korhonen, "DetNet Data Plane: MPLS",
          Work in Progress, Internet-Draft, draft-ietf-detnet-mpls-
          01, 1 July 2019,
          <https://tools.ietf.org/html/draft-ietf-detnet-mpls-01>.

[DETNET-SECURITY]
          Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell,
          J., Austad, H., Stanton, K., and N. Finn, "Deterministic
          Networking (DetNet) Security Considerations", Work in
          Progress, Internet-Draft, draft-ietf-detnet-security-05,
          29 August 2019, <https://tools.ietf.org/html/draft-ietf-
          detnet-security-05>.

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

[IEEE802.1AE]
          IEEE, "IEEE Standard for Local and metropolitan area
          networks-Media Access Control (MAC) Security", IEEE
          802.1AE-2018,
          <https://ieeexplore.ieee.org/document/8585421>.

[IEEE802.1BA]
          IEEE, "IEEE Standard for Local and metropolitan area
          networks--Audio Video Bridging (AVB) Systems", IEEE
          802.1BA-2011,
          <https://ieeexplore.ieee.org/document/6032690>.

[IEEE802.1CB]
          IEEE, "IEEE Standard for Local and metropolitan area
          networks--Frame Replication and Elimination for
          Reliability", DOI 10.1109/IEEESTD.2017.8091139, IEEE
          802.1CB-2017, October 2019,

          <https://ieeexplore.ieee.org/document/8091139>.

[IEEE802.1Q]
          IEEE, "IEEE Standard for Local and Metropolitan Area
          Network--Bridges and Bridged Networks", IEEE 802.1Q-2018,
          <https://ieeexplore.ieee.org/document/8403927>.

[IEEE802.1Qav]
          IEEE, "IEEE Standard for Local and Metropolitan Area
          Networks - Virtual Bridged Local Area Networks Amendment
          12: Forwarding and Queuing Enhancements for Time-Sensitive
          Streams", IEEE 802.1Qav-2009,
          <https://ieeexplore.ieee.org/document/5375704>.

[IEEE802.1Qbu]
          IEEE, "IEEE Standard for Local and metropolitan area
          networks -- Bridges and Bridged Networks -- Amendment 26:
          Frame Preemption", IEEE 802.1Qbu-2016,
          <https://ieeexplore.ieee.org/document/7553415>.

[IEEE802.1Qbv]
          IEEE, "IEEE Standard for Local and metropolitan area
          networks -- Bridges and Bridged Networks - Amendment 25:
          Enhancements for Scheduled Traffic", IEEE 802.1Qbv-2015,
          <https://ieeexplore.ieee.org/document/7440741>.

[IEEE802.1Qch]
          IEEE, "IEEE Standard for Local and metropolitan area
          networks--Bridges and Bridged Networks--Amendment 29:
          Cyclic Queuing and Forwarding", IEEE 802.1Qch-2017,
          <https://ieeexplore.ieee.org/document/7961303>.

[IEEE802.1TSNTG]
          IEEE, "Time-Sensitive Networking (TSN) Task Group",
          <https://1.ieee802.org/tsn/>.

[IEEE802.3]
          IEEE, "IEEE Standard for Ethernet", IEEE 802.3-2018,
          <https://ieeexplore.ieee.org/document/8457469>.

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

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S.
          Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1
          Functional Specification", RFC 2205, DOI 10.17487/RFC2205,
          September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z.,
          and W. Weiss, "An Architecture for Differentiated
          Services", RFC 2475, DOI 10.17487/RFC2475, December 1998,
          <https://www.rfc-editor.org/info/rfc2475>.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41,
           RFC 2914, DOI 10.17487/RFC2914, September 2000,
           <https://www.rfc-editor.org/info/rfc2914>.

[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>.

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

[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>.

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path
           Computation Element (PCE)-Based Architecture", RFC 4655,
           DOI 10.17487/RFC4655, August 2006,
           <https://www.rfc-editor.org/info/rfc4655>.

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

[RFC6658] Bryant, S., Ed., Martini, L., Swallow, G., and A. Malis,
           "Packet Pseudowire Encapsulation over an MPLS PSN",
           RFC 6658, DOI 10.17487/RFC6658, July 2012,
           <https://www.rfc-editor.org/info/rfc6658>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined
           Networking: A Perspective from within a Service Provider
           Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014,
           <https://www.rfc-editor.org/info/rfc7149>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in
           Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384,
           October 2014, <https://www.rfc-editor.org/info/rfc7384>.

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

[RFC7554] Watteyne, T., Ed., Palattella, M., and L. Grieco, "Using
           IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the
           Internet of Things (IoT): Problem Statement", RFC 7554,

                   DOI 10.17487/RFC7554, May 2015,
                   <https://www.rfc-editor.org/info/rfc7554>.

[RFC7567]   Baker, F., Ed. and G. Fairhurst, Ed., "IETF
            Recommendations Regarding Active Queue Management",
            BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015,
            <https://www.rfc-editor.org/info/rfc7567>.

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

[RFC8033]   Pan, R., Natarajan, P., Baker, F., and G. White,
            "Proportional Integral Controller Enhanced (PIE): A
            Lightweight Control Scheme to Address the Bufferbloat
            Problem", RFC 8033, DOI 10.17487/RFC8033, February 2017,
            <https://www.rfc-editor.org/info/rfc8033>.

[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>.

[RFC8289]   Nichols, K., Jacobson, V., McGregor, A., Ed., and J.
            Iyengar, Ed., "Controlled Delay Active Queue Management",
            RFC 8289, DOI 10.17487/RFC8289, January 2018,
            <https://www.rfc-editor.org/info/rfc8289>.

[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>.

[RFC8453]   Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for
            Abstraction and Control of TE Networks (ACTN)", RFC 8453,
            DOI 10.17487/RFC8453, August 2018,
            <https://www.rfc-editor.org/info/rfc8453>.

[RFC8557]   Finn, N. and P. Thubert, "Deterministic Networking Problem
            Statement", RFC 8557, DOI 10.17487/RFC8557, May 2019,
            <https://www.rfc-editor.org/info/rfc8557>.

[RFC8578]   Grossman, E., Ed., "Deterministic Networking Use Cases",
            RFC 8578, DOI 10.17487/RFC8578, May 2019,
            <https://www.rfc-editor.org/info/rfc8578>.

[TEAS]   IETF, "Traffic Engineering Architecture and Signaling
            (teas)", October 2019,
            <https://datatracker.ietf.org/doc/charter-ietf-teas/>.

[TSCH-ARCH]
            Thubert, P., "An Architecture for IPv6 over the TSCH mode
            of IEEE 802.15.4", Work in Progress, Internet-Draft,
            draft-ietf-6tisch-architecture-26, 27 August 2019,
            <https://tools.ietf.org/html/draft-ietf-6tisch-

architecture-26>.

Authors' Addresses

Norman Finn
Huawei
3101 Rio Way
Spring Valley, California 91977
United States of America

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

Pascal Thubert
Cisco Systems
Batiment T3
Village d'Entreprises Green Side, 400, Avenue de Roumanille
06410 Biot - Sophia Antipolis
France

Phone: +33 4 97 23 26 34
Email: pthubert@cisco.com

Balázs Varga
Ericsson
Budapest
Magyar tudosok korutja 11
1117
Hungary

Email: balazs.a.varga@ericsson.com

János Farkas
Ericsson
Budapest
Magyar tudosok korutja 11
1117
Hungary

Email: janos.farkas@ericsson.com