DetNet
Internet-Draft
Intended status: Standards Track
Expires: November 3, 2017

B. Varga, Ed.
J. Farkas
Ericsson

May 2, 2017

DetNet Service Model
draft-varga-detnet-service-model-02

Abstract

This document describes service model for scenarios requiring deterministic networking.

This new version 02 of the DetNet Service Model draft is primarily intended to prevent it from expiring. Major parts of this document were moved to the architecture draft, but some remaining text is under discussion in the workgroup (e.g., QoS, etc.).

Status of This Memo

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

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

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

This Internet-Draft will expire on November 3, 2017.

Table of Contents

1. Introduction

A Deterministic Networking (DetNet) service provides a capability to carry a unicast or a multicast data flow for an application with constrained requirements on network performance, e.g., low packet loss rate and/or latency. During the discussion of DetNet use cases, DetNet architecture, and various related networking scenarios, several confusions have been raised due to different service model interpretations. This document defines service reference points, service components and proposes naming for service scenarios to achieve common understanding of the DetNet service model.

2. Conventions Used in This Document

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

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

3. Terminology and Definitions

Additional terms to [I-D.ietf-detnet-architecture] used in this draft.

DetLink: Direct link between two entities (node/end system) used for deterministic transport.

DetNet flow: A DetNet flow is a sequence of packets to which the DetNet service is to be provided, see [I-D.ietf-detnet-architecture]. This document distinguishes the following three formats of DetNet flows:

App-flow: An App-flow is a data flow between the applications requiring deterministic service. An App-flow does not contain any DetNet related attributes.

DetNet-s-flow: A DetNet-s-flow is an App-flow extended with some DetNet service layer attributes.

DetNet-st-flow: A DetNet-st-flow is an App-flow extended with both DetNet service layer and DetNet transport layer attributes, i.e., encapsulated according to the forwarding paradigm of the DetNet domain.

DetNet-NNI: NNI between DetNet domains.

DetNet-UNI: UNI of a DetNet edge node to provide DetNet service for a connected node or end system.

DetNetwork: Transport network between DetNet-st-flow endpoints.

4. End systems connected to DetNet

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

A DetNet flow [I-D.ietf-detnet-architecture] can have different formats during while it is transported between the peer end systems. Therefore, the following possible formats of a DetNet flow are distinguished in this document:

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

- DetNet-s-flow: specific format of a DetNet flow. It is an App-flow extended with some DetNet service related attributes (i.e., Flow-ID and/or Seq-num).

- DetNet-st-flow: specific format of a DetNet flow. It is an App-flow extended with both DetNet service layer and DetNet transport layer attributes, i.e., encapsulated according to the forwarding paradigm of the DetNet domain.

App-flow and DetNet-s-flow are generated by end systems. DetNet-st-flow can be generated by a DetNet edge node or an end system that is an integral part of a DetNet domain. Further details are described below. This document uses the exact DetNet flow type where it is important to distinguish the flow type; otherwise, the generic term, i.e., DetNet flow is used.

The native data flow between the source/destination end systems is referred to as application-flow (App-flow) as shown in Figure 1. The traffic characteristics of an App-flow can be CBR (constant bit rate) or VBR (variable bit rate) and can have L1 or L2 or L3 encapsulation (e.g., TDM (time-division multiplexing), Ethernet, IP).

[Note: Interworking function for L1 application-flows is out-of-scope in this document, therefore, not depicted in figures.]

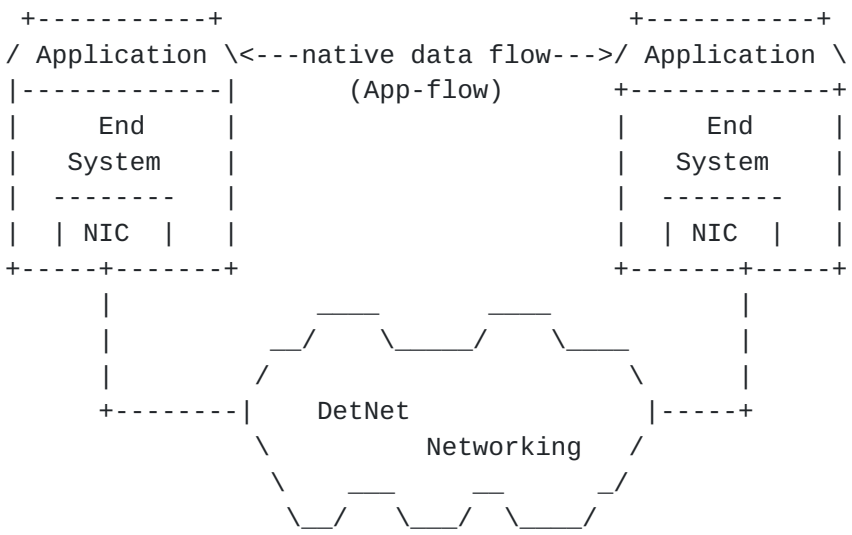

Figure 1: End systems connected to DetNet

An end system may or may not be DetNet transport layer aware or DetNet service layer aware, see [I-D.ietf-detnet-architecture]. That is, an end system may or may not contain DetNet specific functionality. End systems with DetNet functionalities may have the same or different transport layer as the connected DetNet domain. Grouping of end systems are shown in Figure 2. (Note: A "TSN end system" of [I-D.ietf-detnet-dp-alt] is an example for a "DetNet unaware end system".)

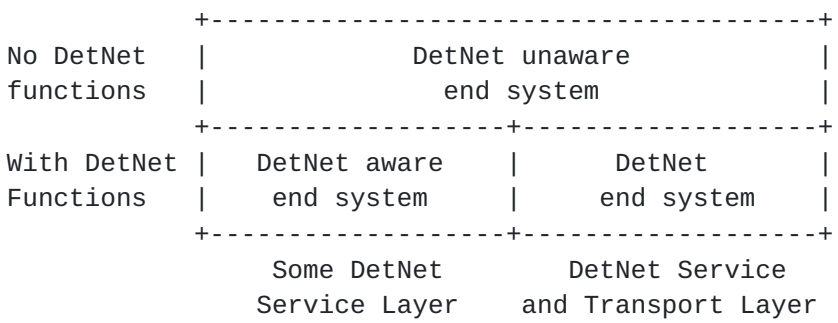

Figure 2: Grouping of end systems

End system(s) may or may not be directly connected to the DetNet transport network. This document assumes direct connection in the remaining part. The end system types are:

- A "DetNet unaware end system" originates a native data flow (App-flow). Such end systems usually assume dedicated (and direct) connectivity to their peers, which is replaced by the DetNet

network. Its connection to a DetNet network requires a DetNet edge node, that creates a DetNet-st-flow (with proper Flow-ID and Seq-num attributes) by encapsulating the native data flow according to the forwarding paradigm of the connected DetNet domain.

- A "DetNet aware end system" may contain some DetNet specific service functionalities and it extends the App-flow with related DetNet specific flow attributes (i.e., Flow-ID and/or Seq-num). The resulting flow is referred to as DetNet-s-flow as it contains service layer specific fields, but the format of the DetNet-s-flow encapsulation is not identical with the forwarding paradigm (i.e., the transport layer) of the DetNet domain. Therefore, it has to be connected to a DetNet edge node. DetNet aware end systems can be, e.g., an IP end system with some DetNet service functions connected to an MPLS-based DetNet domain.

- A "DetNet end node" has DetNet functionalities and the same forwarding paradigm as the connected DetNet domain. It can be treated as an integral part of the DetNet domain, therefore, it is connected to a DetNet relay node (or to a DetNet transit node). It originates a DetNet-st-flow (i.e., the App-flow is extended within the end system with all the DetNet specific flow attributes used inside the DetNet domain).

These end systems are shown in Figure 3. A DetNet-UNI ("U" on Figure 3) is assumed in this document to be a packet-based reference point and provides connectivity over the DetNet domain. A DetNet-UNI may add forwarding technology specific encapsulation to the App-flow / DetNet-s-flow and transport it as a DetNet-st-flow over the network.

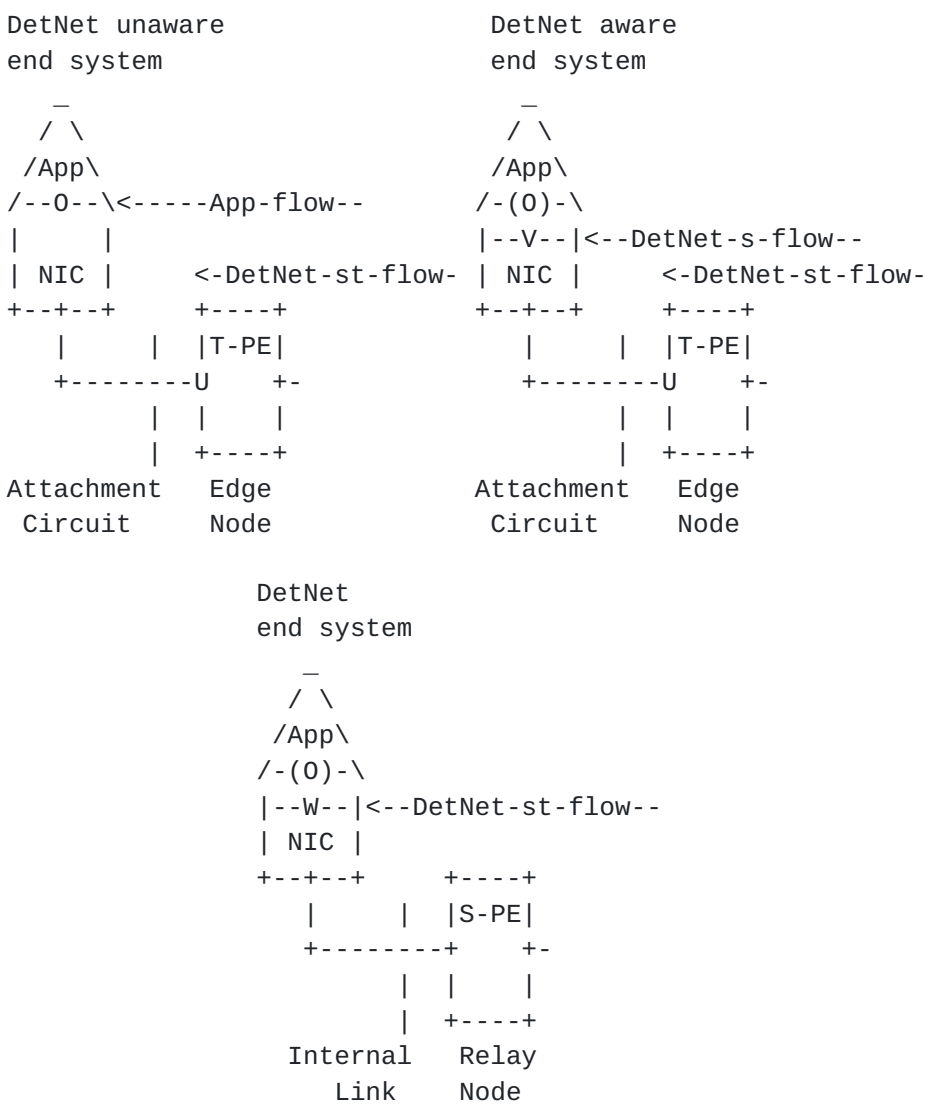

Figure 3: Types of end systems

[Note: DetNet aware end systems can be also treated as a special mix of a DetNet unaware system and a DetNet end system. It is similar to a DetNet end system as its data flow contains DetNet attributes, however, those attributes cannot be used directly inside the DetNet domain, e.g., due to the different transport layer. Therefore, it is also similar to DetNet unaware end systems as it must be connected to a DetNet edge node to adapt, e.g., the encapsulation of the DetNet flow to the forwarding paradigm of the DetNet domain. A typical example showing a DetNet aware end system can be the following scenario: an end system encapsulates its App-flow in IP-RTP packets. It assumes a single connection to its peer, therefore, the Seq-num field is not used by the end-system. It is connected to an MPLS-

based DetNet domain that has redundant paths and applies service protection via the duplication and elimination functionality. As per [I-D.ietf-detnet-architecture], the addition or removal of packet sequencing information is the job of a DetNet edge node. As forwarding is MPLS-based, the Seq-num required for service protection is created and added to the DetNet-s-flow by the DetNet edge node (in the PW control-word field).]

5. DetNet service model

5.1. Service parameters

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

Delay and loss parameters are somewhat correlated because the effect of late delivery can be equivalent to loss. However, not all applications require hard limits on both parameters (delay and loss). For example, some real-time applications allow graceful degradation if loss happens (e.g., sample-based processing, media distribution). Some others may require high-bandwidth connections that make the usage of techniques like flow duplication economically challenging or even impossible. Some applications may not tolerate loss, but are not delay sensitive (e.g., bufferless sensors).

Primary transport service attributes for DetNet transport are:

- Bandwidth parameter(s),
- Delay parameter(s),
- Loss parameter(s),
- Connectivity type.

Time/loss sensitive applications may have somewhat special requirements especially for loss (e.g., no loss in two consecutive communication cycles; very low outage time, etc.).

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

5.2. Service overview

The figures below show the DetNet service related reference points and components for various end system scenarios (Figure 4 and Figure 5).

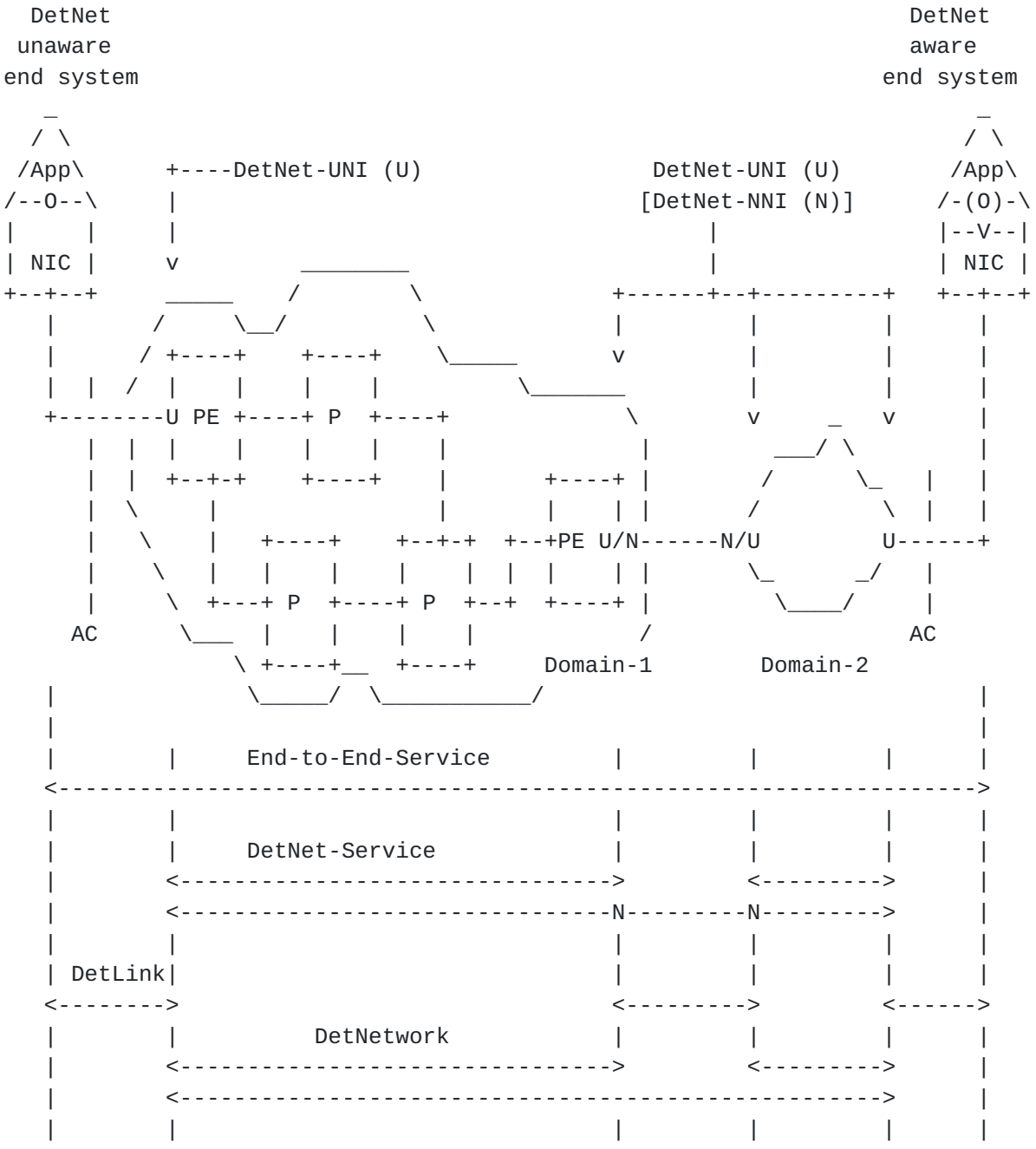

Figure 4: DetNet Service Reference Model (multi-domain)

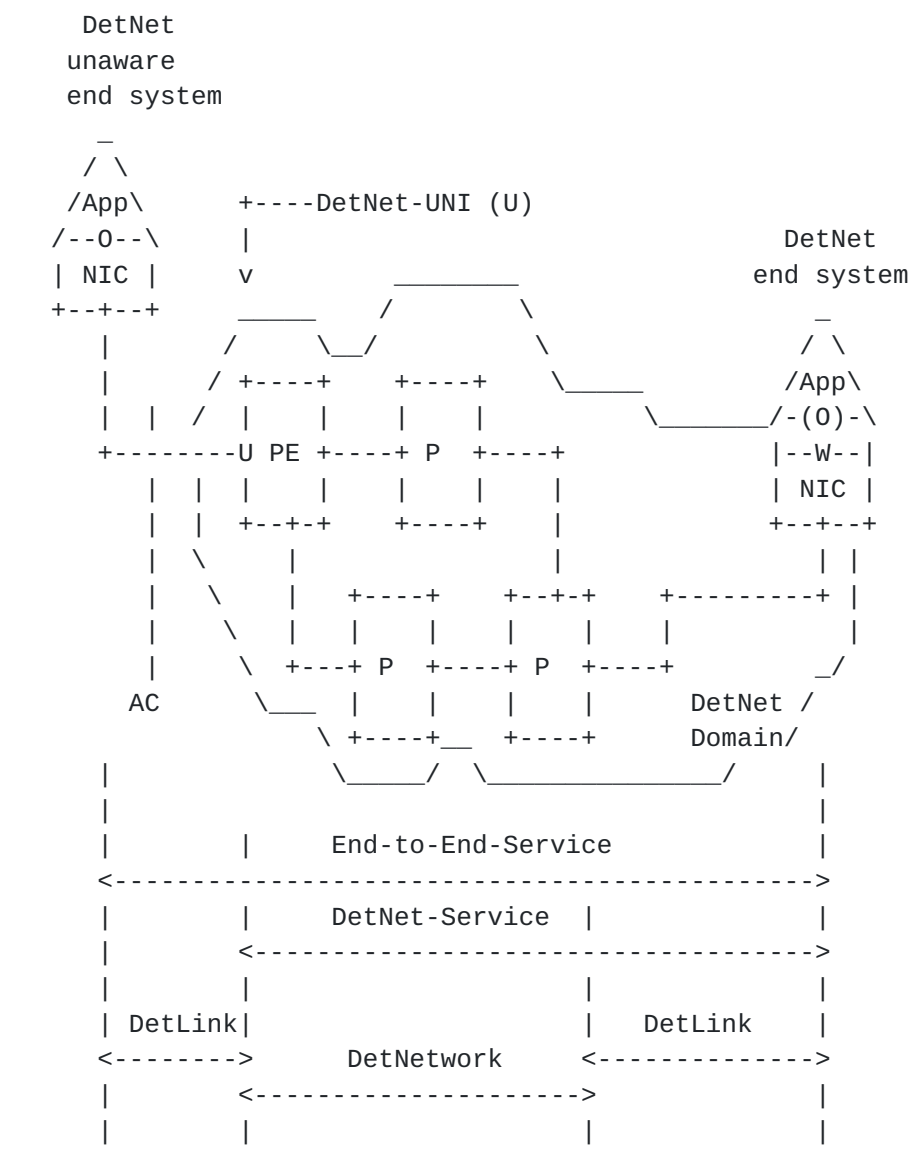

Figure 5: DetNet Service Reference Model (single domain)

5.3. Reference Points

From service model design perspective a fundamental question is the location of the service endpoints, i.e., where the service starts and ends. The following reference points can be distinguished for the DetNet use cases:

- App-flow endpoint: End system's internal reference point ("O") for the native data flow.

- DetNet-s-flow endpoint: DetNet aware end system's internal reference point ("V").

- DetNet-st-flow endpoint: DetNet edge node UNI ("U") or DetNet end system's internal reference point ("W").

- DetNet-UNI: UNI interface ("U") on a DetNet edge node.

- DetNet-NNI: NNI interface ("N") between DetNet domains.

Data flow endpoints ("O", "V" and "W" in Figure 4 and Figure 5) are more challenging from control perspective as they are internal reference points of end systems. They are providing access to deterministic transport for the native data flow (App-flow).

DetNet-UNI and DetNet-NNI ("U" and "N" in Figure 4) are assumed in this document to be packet-based reference points and provide connectivity over the packet network and between domains. A DetNet-UNI adds networking technology specific encapsulation to the App-flow / DetNet-s-flow in order to transport it as a DetNet-st-flow over the network. There are many similarities regarding the functions of a DetNet-st-flow endpoint ("W") and a DetNet-UNI ("U") but there may be some differences. For example, in-order delivery is expected in end system internal reference points, whereas it is considered optional over the DetNet-UNI.

5.4. Service scenarios

Using the above defined reference points, two major service scenarios can be identified:

- End-to-End-Service: the service reaches out to final source or destination nodes, so it is an e2e service between application hosting devices (end systems).

- DetNet-Service: the service connects networking islands, so it is a service between the borders of network domain(s).

End-to-End-Service is defined between App-flow endpoints, whereas DetNet-Service is between DetNet-st-flow endpoints. This allows the peering of same layers/functions.

5.5. Data flows

Three possible DetNet flow formats are distinguished for unambiguous references:

- App-flow: data flow requiring deterministic transport between two App-flow endpoints; data format is application specific (e.g., bit stream directly mapped to Ethernet frames, etc.). It does not contain any DetNet attributes.

- DetNet-s-flow: similar to the App-flow, but extended with some DetNet attributes as DetNet aware end systems have some DetNet service layer functionalities. However, the encapsulation format differs from the forwarding paradigm of the connected DetNet domain, so those attributes cannot be used directly.

- DetNet-st-flow: data flow between DetNet-UNIs ("U") and/or DetNet end systems ("W"). This flow is extended with both DetNet service layer and DetNet transport layer attributes. This format allows simple flow recognition/transport/etc. during forwarding in the DetNet domain.

5.6. Service components/segments

The follwoing building blocks are used as reference to service components/segments:

- DetLink: direct link between two entities (node/end system) used for deterministic transport.

- DetNetwork: network between DetNet nodes.

Any DetNet service scenario can be described using DetLink and DetNetwork components/segments. For example, the service between the App-flow endpoints in Figure 4 can be composed as a DetLink-1 (between the end system on the left and the edge node of Domain-1) + DetNetwork-1 (of Domain-1) + DetLink-2 (between Domain-1 and Domain-2) + DetNetwork-2 (of Domain-2) + DetLink-3 (between edge node of Domain-2 and the end system on the right).

6. DetNet service instances

6.1. Attributes used by DetNet functions

The three DetNet functions (congestion protection, explicit routes, service protection) require two data flow related attributes to work properly:

- Flow-ID and

- Sequence number (Seq-Num).

These attributes are extracted from the ingress packets of the node [I-D.ietf-detnet-architecture]. Flow-ID is used by all the three DetNet functions, but sequence number is used only by the duplicate elimination functionality.

Flow-ID must be unique per network domain. Its encoding format is specific to the forwarding paradigm of the domain and to the capabilities of intermediate nodes to identify data flows. For example, in case of "PW over MPLS", one option is to construct the Flow-ID by the PW label and the LSP label (denoted as [PW-label;LSP-label]). In such a case, intermediate P nodes have to check all labels to identify a DetNet flow, what may not be a valid option in some deployment scenarios. Another possible option is to use a dedicated LSP per data flow, so the LSP label itself can be used as a Flow-ID (denoted as [LSP-label]). In such a case, the intermediate P nodes do not have to check the whole label stack to recognize a data flow (DetNet flow), however, it results in larger L-FIB tables on the MPLS nodes.

[Note: Seq-num requires a control-word in the label stack in MPLS domains, which should be recognized by intermediate S-PE (relay) nodes.]

6.2. Service instance for DetNet flows

The DetNet network reference model is shown in Figure 6 for a DetNet-Service scenario (i.e. between two DetNet-UNIs). In this figure, the end systems ("A" and "B") are connected directly to the edge nodes of the PSN ("PE1" and "PE2"). End-systems participating DetNet communication may require connectivity before setting up an App-flow that requires the DetNet service. Such a service instance and the one dedicated for DetNet service share the same attachment circuit. Packets belonging to a DetNet flow are selected by a filter configured on the attachment circuit ("F1" and "F2"). As a result, data flow specific attachment circuits ("AC-A + F1" and "AC-B + F2") are terminated in the flow specific service instance ("SI-1" and "SI-2"). A PSN tunnel is used to provide connectivity between the service instances. The encapsulation used over the PSN tunnel are described in [I-D.ietf-detnet-dp-alt].

The PSN tunnel is used to transport exclusivelly the packets of the DetNet flow between "SI-1" and "SI-2". The service instances are configured to implement a flow specific routing or bridging function depending on what connectivity the participating end systems require (L3 or L2). The service instance and the PSN tunnel may or may not be shared by multiple DetNet flows. Sharing the service istance by multiple DetNet flows requires properly populated forwarding tables of the service instance.

Serving regular traffic and DetNet flows by the same service instance is out-of-scope in this draft, but some related thoughts are described in Annex 1. Such a combination can provide the required connectivity before setting up a DetNet service.

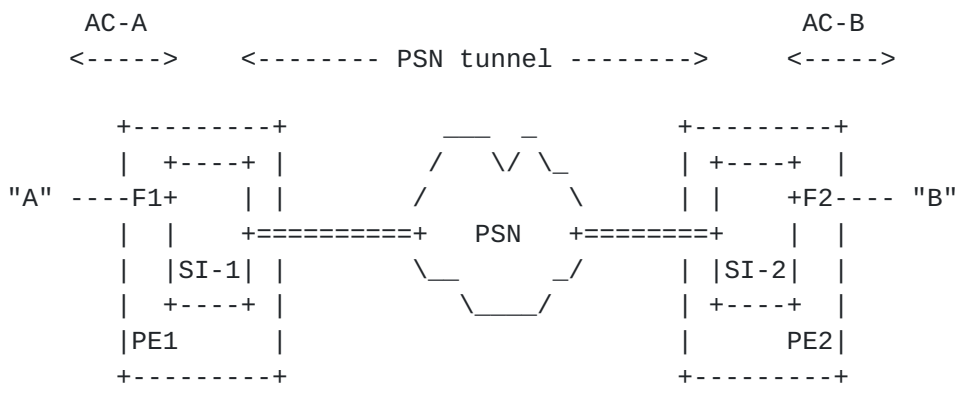

Figure 6: DetNet network reference model

[Note: There are differences in the usage of a "packet PW" for DetNet traffic compared to the network model described in [RFC6658]. In the DetNet scenario, the packet PW is used exclusivelly 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 ... This packet pseudowire is used to transport all of the required layer 2 and layer 3 protocols between LSR1 and LSR2".]

7. DetNet flows over multiple technology domains

7.1. Flow attribute mapping between layers

Transport of DetNet flows over multiple technology domains may require that lower layers are aware of specific flows of higher layers. Such an "exporting of flow identification" (see section 4.7 in [I-D.ietf-detnet-architecture]) is needed each time when the forwarding paradigm is changed on the transport path (e.g., two LSRs are interconnected by a L2 bridged domain, etc.). The three main forwarding methods considered for deterministic networking are:

- IP routing
- MPLS label switching
- Ethernet bridging

The simplest solution for generalized flow identification could be to define a unique Flow-ID triplet per DetNet flow (e.g., [IP: "IPv6-flow-label"+"IPv6-address"; MPLS: "PW-label"+"LSP-label"; Ethernet: "VLAN-ID"+"MAC-address"). This triplet can be used by the DetNet

encoding function of technology border nodes (where forwarding paradigm changes) to adapt to capabilities of the next hop node. They push a further (forwarding paradigm specific) Flow-ID to packet header ensuring that flows can be easily recognized by domain internal nodes. This additional Flow-ID might be removed when the packet leaves a given technology domain.

[Note: Seq-num attribute may require a similar functionality at technology border nodes.]

The additional (domain specific) Flow-ID can be

o created by a domain specific function or

o derived from the Flow-ID added to the App-flow,

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

7.2. 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 7 shows a scenario where an IP end system ("IP-A") is connected via two Ethernet switches ("ETH-n") to an IP router ("IP-1").

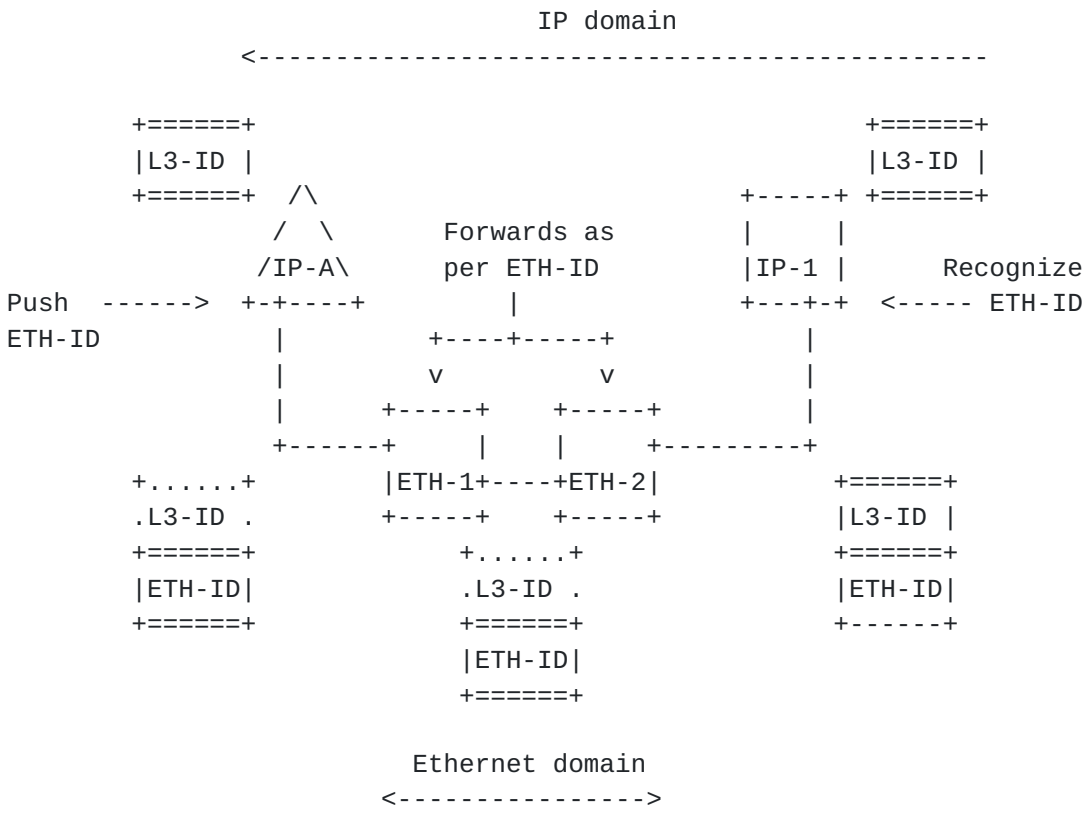

Figure 7: IP nodes interconnected by an Ethernet domain

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

Figure 8 shows a scenario where MPLS domain nodes ("PE-n" and "P-m") are connected via two Ethernet switches ("ETH-n").

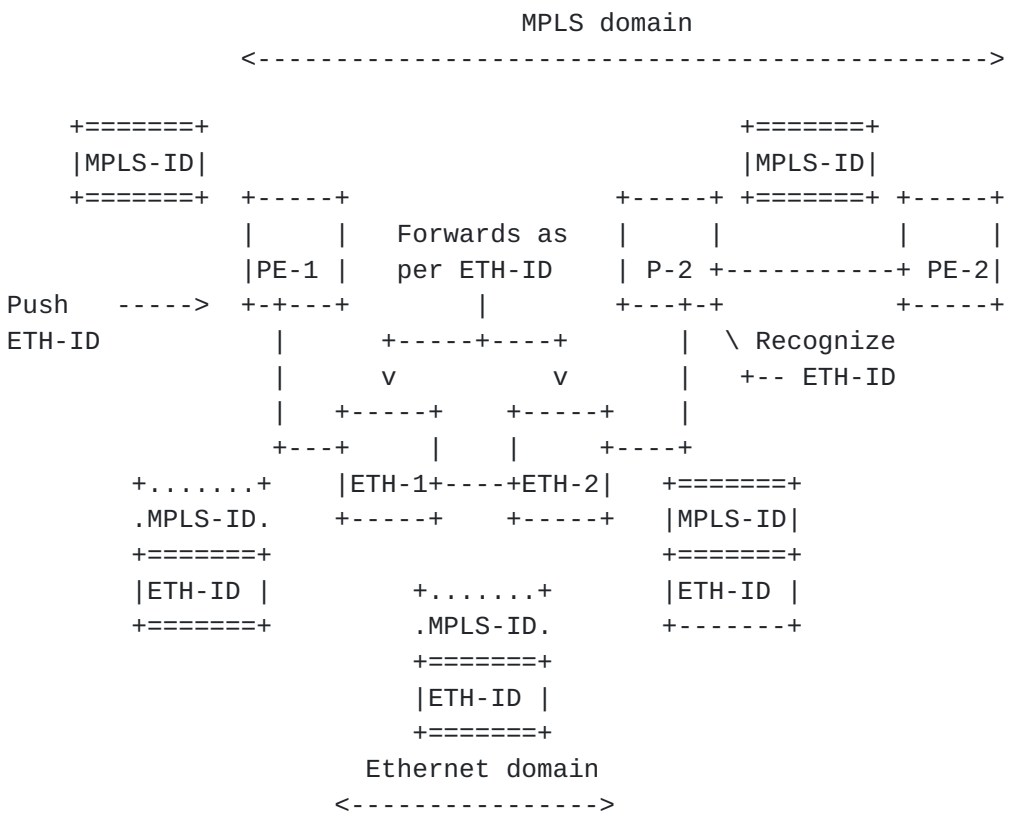

Figure 8: MPLS nodes interconnected by an Ethernet domain

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

8. Summary

This document describes DetNet service model.

9. IANA Considerations

N/A.

10. Security Considerations

N/A.

11. Acknowledgements

The authors wish to thank Lou Berger, Norman Finn, Jouni Korhonen and the members of the data plane design team for their various contributions, comments and suggestions regarding this work.

12. Annex 1 - Service Instance shared by DetNet and regular traffic

This Annex contains some thoughts about scenarios where the service instance is shared by DetNet and regular traffic.

12.1. L2 service instance shared by regular and DetNet traffic

In case of a L2 VPN transport, the service instance implements bridging. In MPLS-based PSN, there is a full mesh of PWs between service instances of PE nodes. Adding DetNet flows to the network results in a somewhat modified PW structure, as a DetNet flow requires its unique Flow-ID to be encoded in the labeled packet.

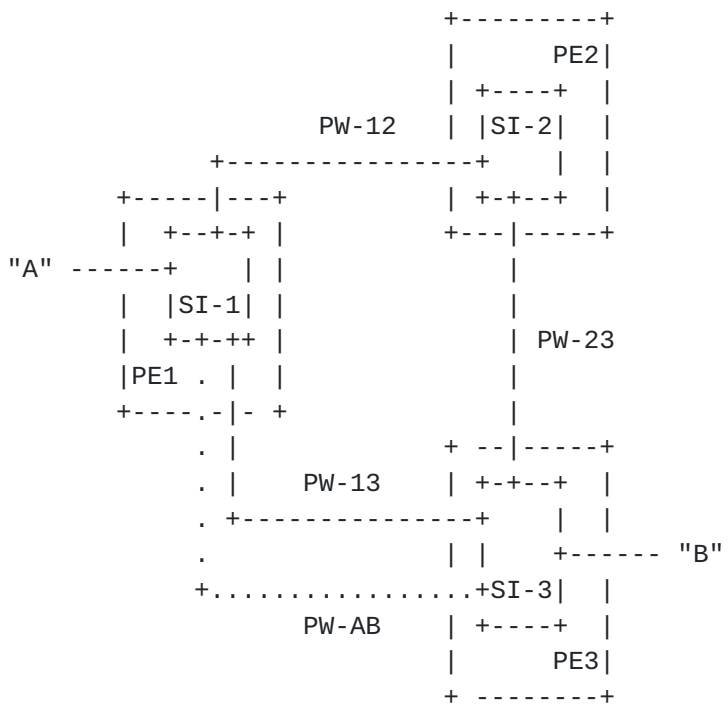

Figure 9: DetNet L2 VPN Service

Figure 9 shows a scenario where there is a DetNet flow between the end systems ("A" and "B"). "SI-n" denotes the L2 VPN service instance of "PEn". Regular traffic of the L2 VPN instance use "PW-12", "PW-13" and "PW-23". However, for transport of DetNet traffic between "A" and "B" a separate PW ("PW-AB") has to be used. "PW-AB" is a somewhat special PW (called here "virtual PW") and it is treated differently than PWs used by regular traffic (i.e., PW-13, PW-12, and PW-23). Namely, "PW-AB" is used exclusivelly by the DetNet flow between "A" and "B". "PW-AB" does not participate in flooding and no MAC addresses are associated with it (not considered for the MAC learning process). "PW-AB" may use the same LSP as "PW-13" or a dedicated one.

Regular traffic between "A" and "B" has an encapsulation [PW-13_label ; LSP_label], whereas DetNet flow has [PW-AB_label ; LSP_label].

12.2. L3 service instance shared by regular and DetNet traffic

In case of a L3 DetNet service, the service instance implements routing. In MPLS-based PSN, such a "routing service" can be provided by IP VPNs ([RFC4364]). However, the IP VPN service adds only a single label (VPN label) during forwarding, therefore, the label stack does not contain a "control word" (i.e., there is no field to encode a sequence number). Therefore, transport of DetNet flows requires the combination of IP VPN and PW technologies.

Adding DetNet flows to the network results in a somewhat modified label stack structure, as a DetNet flow requires its packet PW encapsulation ([RFC6658]).

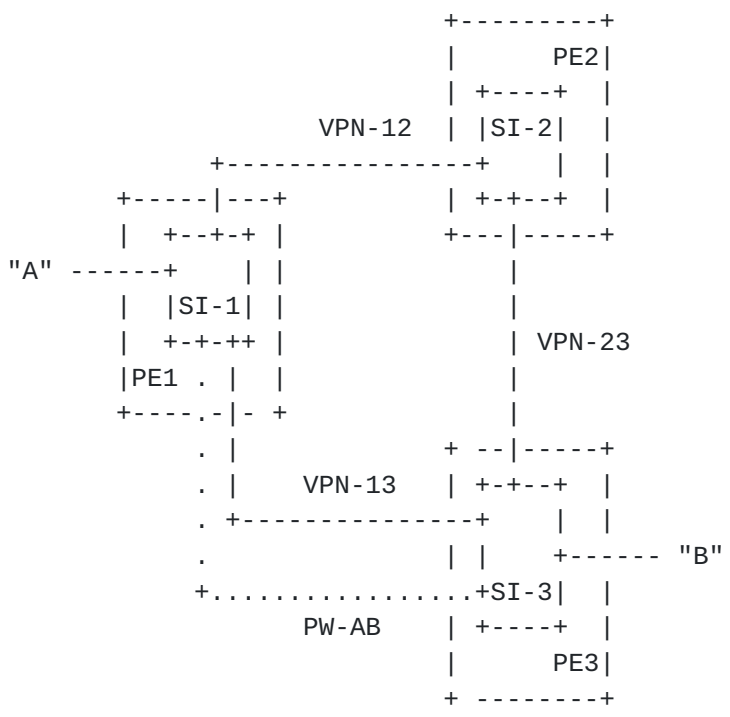

Figure 10: DetNet L3 VPN Service

Figure 10 shows a scenario where there is a DetNet flow between the end systems ("A" and "B"). "SI-n" denotes the L3 VPN service instance of "PEn". Regular traffic of the L3 VPN instance use as service label "VPN-12", "VPN-13" and "VPN-23". However, for transport of DetNet traffic between "A" and "B" a PW ("PW-AB") has to be used, what ensures that DetNet flow can be recognized by intermediate P nodes and a control world can be also present. "PW-AB" is used exclusivelly by the DetNet flow between "A" and "B". "PW-AB" may use the same LSP as regular traffic (labeled by "VPN-13") or a dedicated one.

Regular traffic between "A" and "B" has an encapsulation [VPN-13_label ; LSP_label], whereas DetNet flow has [PW-AB_label ; LSP_label].

13. Annex 2 - Integrating Layer 3 and Layer 2 QoS

Sophisticated QoS mechanisms are available in Layer 3 (L3), see, e.g., [RFC7806] for an overview. Although, Layer 2 (L2) QoS and queuing used to be simpler; it has been evolving, it is now equipped with Time-Sensitive Networking (TSN) features [IEEE8021TSN]. The TSN features may be beneficial or even essential for DetNet flows if Layer 2 links or sub-networks are included in their path. Therefore, it is worth investigating the problems arising when both Layer 3 and

Layer 2 QoS features are supported by a node; even without diving deep into solution/implementation details.

In IEEE Std 802.1Q-2005, eight traffic classes are supported, allowing separate queues for each priority as illustrated in Figure 11. Any traffic class-based transmission selection algorithm can be implemented in addition to the strict priority algorithm mandated by IEEE Std 802.1Q-2005. The priority information is encoded in the 3-bit field carried in a tag in the frame header. Note that the IEEE 802.1Q architecture specifies queuing at the output port; however, implementations may differ. Consequently, the following figures only show the queuing at the output port that is selected by the forwarding decision for the transmission of a frame.

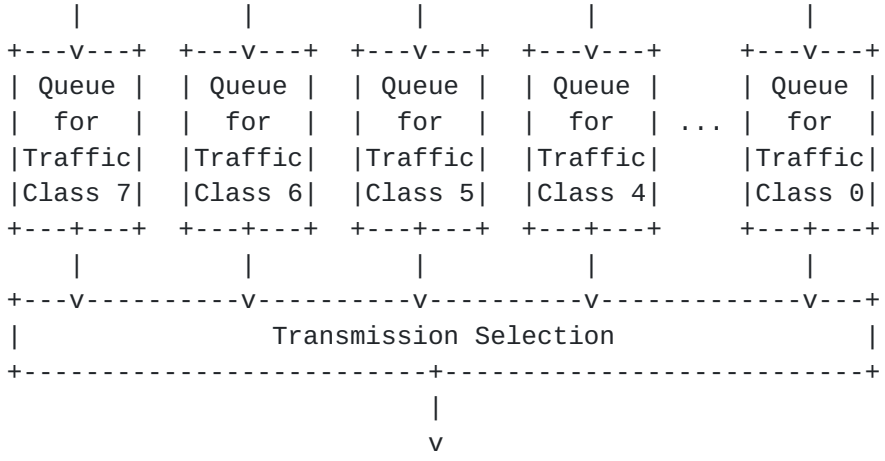

Figure 11: Queuing in IEEE 802.1Q-2005

The Layer 2 QoS architecture has been evolving, see, e.g., IEEE Std 802.1Q-2014 [IEEE8021Q], which specifies the Credit-Based Shaper (originally specified by IEEE Std 802.1Qav). There are recent IEEE 802.3 and 802.1 standards and ongoing projects to enhance the QoS supported by Ethernet and Layer 2 networks. For instance, frame preemption is specified by IEEE Std 802.3br ([IEEE8023br], to be amended to [IEEE8023]) and IEEE Std 802.1Qbu ([IEEE8021Qbu], to be amended to [IEEE8021Q]) where time-critical (express) frames can suspend the transmission of non-time-critical (preemptable) frames while one or more time-critical frames are transmitted. Another recently published specification is IEEE Std 802.1Qbv [IEEE8021Qbv], which specifies time-aware queue-draining controlled by transmission gates in order to schedule the transmission of frames relative to a known timescale, which can be provided by time synchronization. The architecture extended with time-aware queuing and frame preemption is illustrated in Figure 12. These time-sensitive networking extensions provide deterministic behavior in Layer 2 networks. The ongoing IEEE 802.1 projects provide further extensions to the QoS architecture,

e.g., ingress filtering and policing (P802.1Qci), cyclic queuing and forwarding (P802.1Qch), and asynchronous traffic shaping (P802.1Qcr), see [IEEE8021TSN].

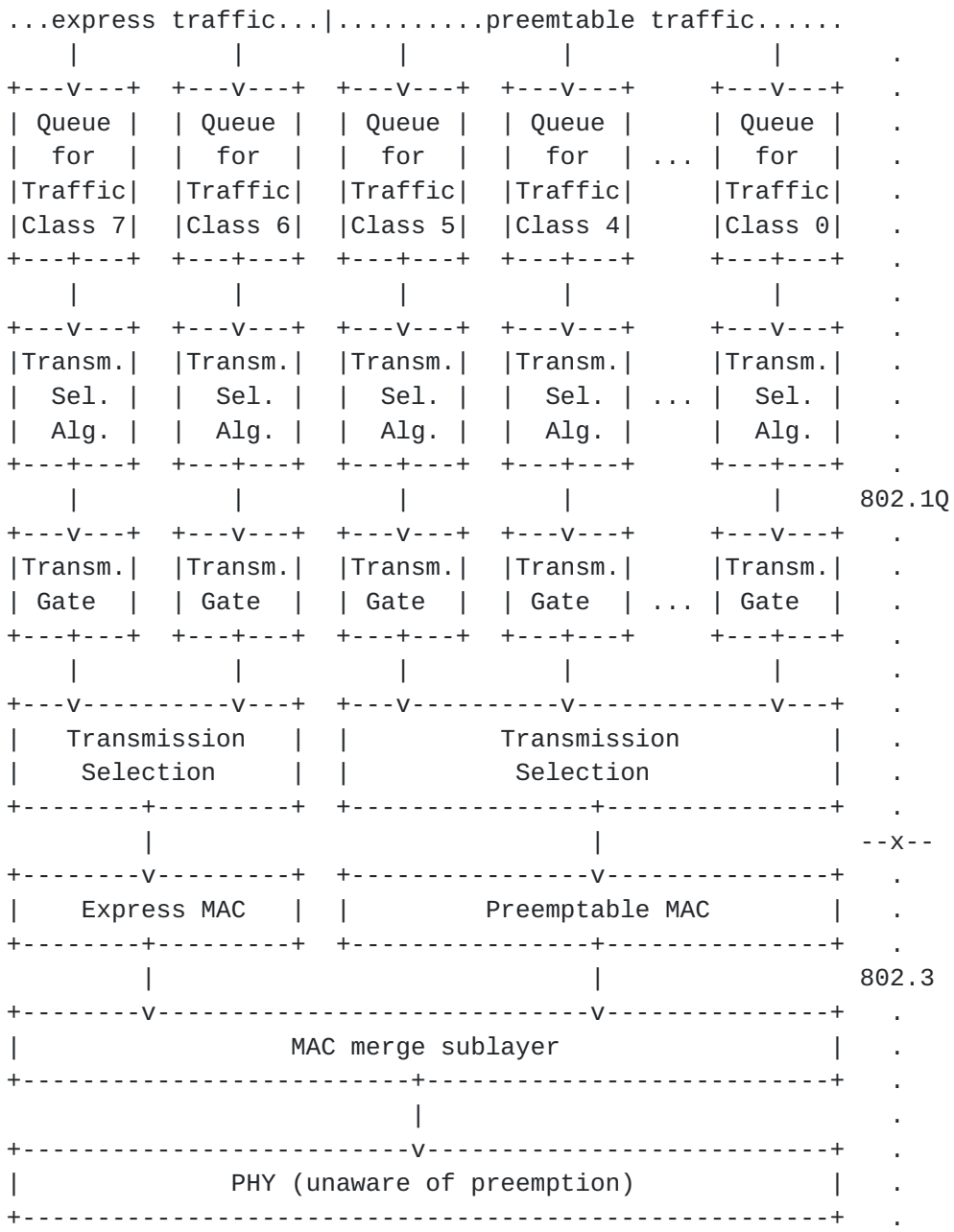

Figure 12: L2 queuing and frame preemption

A QoS architecture integrating both Layer 3 and Layer 2 features is necessary to exploit the benefits provided by the different layers if a DetNet network includes link(s) or sub-network(s) equipped with TSN features. For instance, it can be crucial for a time-critical DetNet

flow to leverage TSN features in a Layer 2 sub-network in order to meet the DetNet flow's requirements, which may be spoiled otherwise.

Figure 13 provides a theoretical illustration for the integration of the Layer 3 and Layer 2 QoS architecture. The figure only shows the queuing after the routing decision. The figure also illustrates potential implementation dependent borders (Brdr). The borders shown in the figure are critical in the sense that the high priority DetNet flows have to be transferred via a different Service Access Points (SAPs) through these borders than the low priority (background) flows. Having a single SAP for these very different traffic types may result in possible QoS degradation for the DetNet flows because packets of other flows could delay the transmission of DetNet packets. For instance, different SAPs are needed for the DetNet flows and other flows when they get to Layer 3 queuing after the routing decision via Brdr-d. Furthermore, a different SAP is needed for DetNet packets than other packets when they get to Layer 2 queuing from Layer 3 queuing via Brdr-c. Similarly, different SAPs are needed for the express and for the preemptable frames when they get to the MAC layer from Layer 2 queuing via Brdr-b, which is provided by the IEEE 802.1Q architecture as shown in Figure 12. It depends on the implementation whether or not Brdr-a exists.

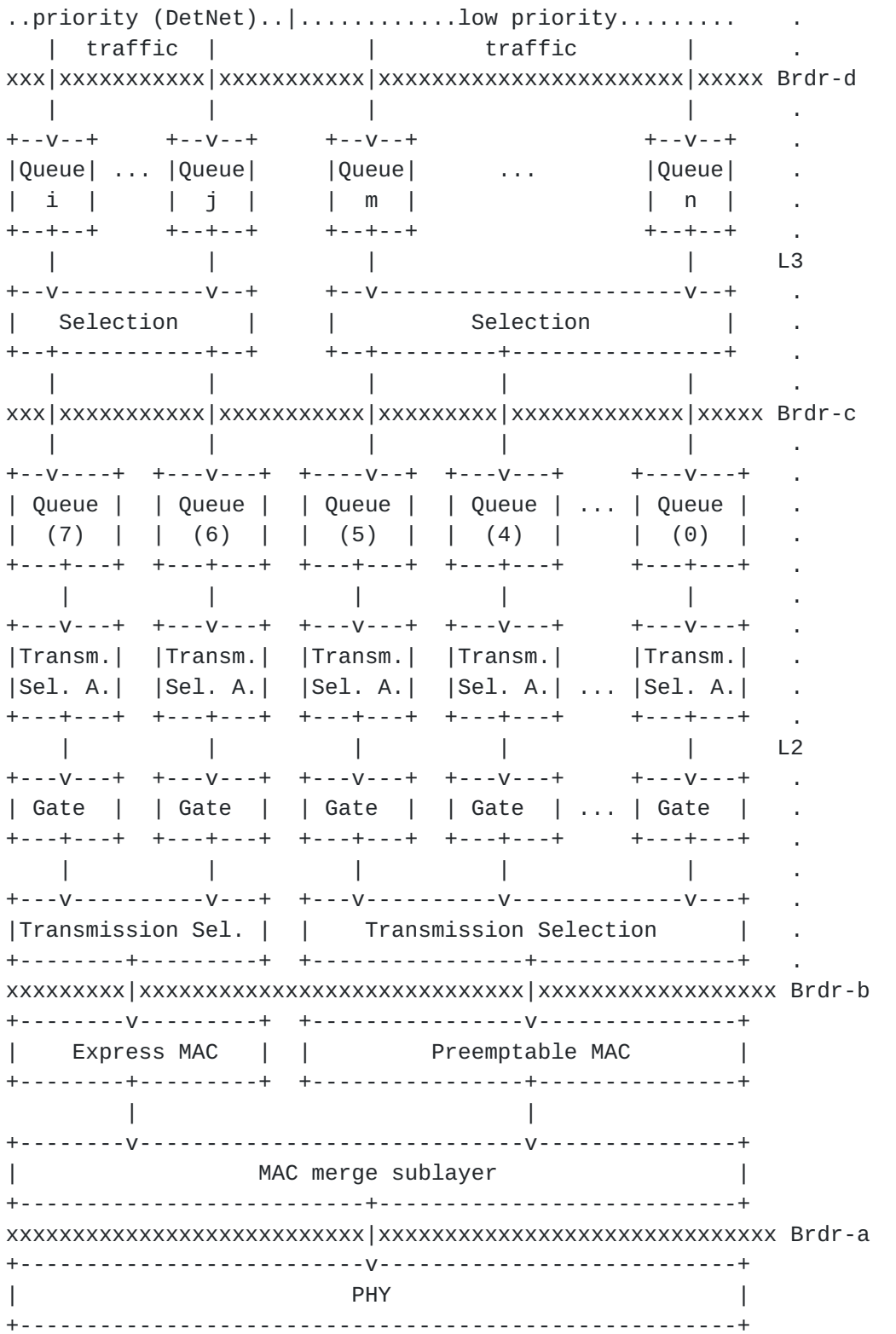

Figure 13: Integrated L3/L2 queuing architecture and implementation options

Not all the functions depicted in Figure 13 are necessarily present in an implementation. A function may be combined with another one or may be completely missing. For instance, it may be the case that there is no Layer 3 queuing for DetNet packets, but they get directly to the Layer 2 queues. Alternatively, an implementation may combine the Layer 3 queues and the Layer 2 queues such that there is a single level of queues. There are further alternatives in addition to the ones mentioned here.

Different implementation approaches, i.e., different node designs are illustrated in Figure 14 and Figure 15. Figure 14 illustrates a monolithic node design where there is a single feature rich chip and relatively simple interfaces. The single chip implements all routing (and/or bridging) features as well as almost all QoS features. (Some aspects of frame preemption may be implemented on the interface.) Figure 15 illustrates a linecard-based design where each linecard has its own chip, which implements routing and QoS features.

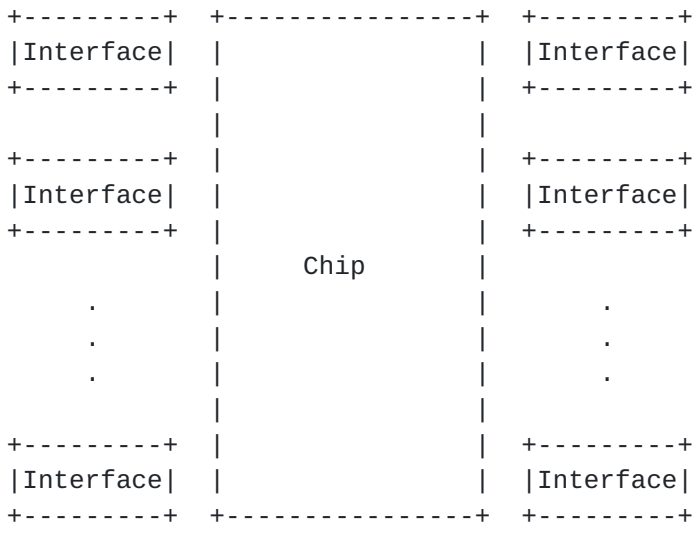

Figure 14: Monolithic node design

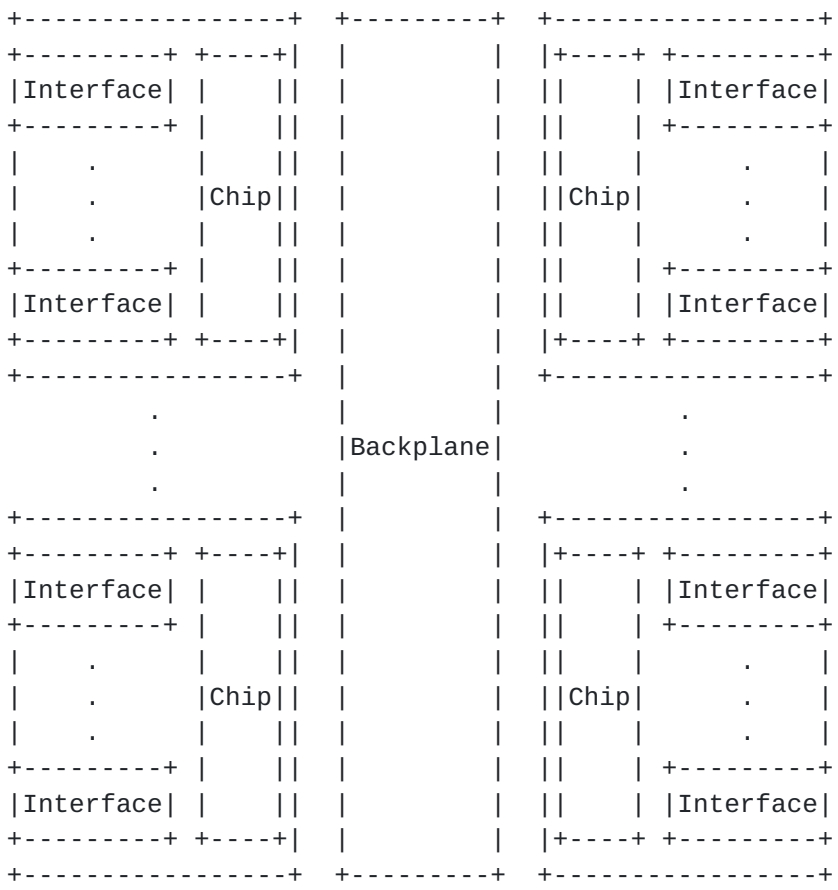

Figure 15: Linecard-based node design

Different implementations have different physical borders, which imply that different borders out of the ones illustrated in Figure 13 exist in a given implementation. For instance, there is no physical border corresponding to Brdr-d (Figure 13) in the monolithic implementation approach (Figure 14). However, Brdr-d is inevitably there in the linecard-based implementation approach (Figure 15) due to the backplane.

Altogether, it is essential to leverage the benefits of both Layer 3 and Layer 2 QoS features if Layer 2 is also involved in the support of a DetNet flow. Exploiting both layers requires attention to the aspects explained related to Figure 12. Nevertheless, the actually important aspects largely depend on the implementation approach chosen, see, e.g., Figure 14 vs. Figure 15.

14. References

14.1. Normative References

[I-D.ietf-detnet-architecture]
Finn, N. and P. Thubert, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-00 (work in progress), September 2016.

[I-D.ietf-detnet-dp-alt]
Korhonen, J., Farkas, J., Mirsky, G., Thubert, P., Zhuangyan, Z., and L. Berger, "DetNet Data Plane Protocol and Solution Alternatives", draft-ietf-detnet-dp-alt-00 (work in progress), October 2016.

[I-D.ietf-detnet-use-cases]
Grossman, E., Gunther, C., Thubert, P., Wetterwald, P., Raymond, J., Korhonen, J., Kaneko, Y., Das, S., Zha, Y., Varga, B., Farkas, J., Goetz, F., Schmitt, J., Vilajosana, X., Mahmoodi, T., Spirou, S., and P. Vizarreta, "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-12 (work in progress), April 2017.

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

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <http://www.rfc-editor.org/info/rfc4364>.

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

[RFC7806] Baker, F. and R. Pan, "On Queuing, Marking, and Dropping", RFC 7806, DOI 10.17487/RFC7806, April 2016, <http://www.rfc-editor.org/info/rfc7806>.

14.2. Informative References

[IEEE8021Q]
IEEE 802.1, "IEEE 802.1Q-2014: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks", 2014, <http://standards.ieee.org/getieee802/download/802-1Q-2014.pdf>.

[IEEE8021Qbu]
IEEE 802.1, "IEEE 802.1Qbu-2016: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks -- Amendment 26: Frame Preemption", 2016, <https://standards.ieee.org/findstds/standard/802.1Qbu-2016.html>.

[IEEE8021Qbv]
IEEE 802.1, "IEEE 802.1Qbv-2015: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks -- Amendment 25: Enhancements for Scheduled Traffic", 2015, <https://standards.ieee.org/findstds/standard/802.1Qbv-2015.html>.

[IEEE8021TSN]
IEEE 802.1, "IEEE 802.1 Time-Sensitive Networking (TSN) Task Group", <http://www.ieee802.org/1/>.

[IEEE8023]
IEEE 802.3, "IEEE 802.3-2015: IEEE Standard for Local and metropolitan area networks - Ethernet", 2015, <http://standards.ieee.org/getieee802/download/802.3-2015.zip>.

[IEEE8023br]
IEEE 802.3, "IEEE 802.3br-2016: IEEE Standard for Local and metropolitan area networks - Ethernet -- Amendment 5: Specification and Management Parameters for Interspersing Express Traffic", 2016, <https://standards.ieee.org/findstds/standard/802.3br-2016.html>.

Authors' Addresses

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

Email: balazs.a.varga@ericsson.com

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

Email: janos.farkas@ericsson.com