DetNet
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

N. Finn
Huawei Technologies Co. Ltd
B. Varga
J. Farkas
Ericsson

October 30, 2017

DetNet Bounded Latency
draft-finn-bounded-latency-00

Abstract

This document a model for DetNet to achieve bounded latency and zero congestion loss using existing and in-progress standards from IEEE 802 and RFCs from IETF.

Status of This Memo

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

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

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

This Internet-Draft will expire on May 3, 2018.

Copyright Notice

Table of Contents

1. Introduction

The ability for IETF Deterministic Networking (DetNet) or IEEE 802.1 Time-Sensitive Networking (TSN) to provide bounded latency and zero congestion loss depends upon A) configuring and allocating network resources for the exclusive use of DetNet/TSN flows; B) identifying, in the data plane, the resources to be utilized by any given packet, and C) the detailed behavior of those resources, especially transmission queue selection, so that latency bounds can be reliably assured. Thus, DetNet is an example of an INTSERV Guaranteed Quality of Service [RFC2212]

As explained in [I-D.ietf-detnet-architecture], DetNet flows are characterized by 1) a maximum bandwidth, guaranteed either by the transmitter or by strict input metering; and 2) a requirement for a guaranteed worst-case end-to-end latency. That latency guarantee, in turn, provides the opportunity to supply enough buffer space to guarantee zero congestion loss. To be of use to the applications identified in [I-D.ietf-detnet-use-cases], it must be possible to calculate, before the transmission of a DetNet flow commences, the worst-case network latency and the amount of buffer space required at each hop to ensure against congestion loss. The detailed behavior of the mechanism(s) used to select the next packet for transmission at each output port is critical in making this determination. A detailed timing model, breaking down the time taken for each packet to traverse each element in the model, along with possible variations, is required, because seemingly minor implementation variations can generate large uncertainties in the number of required

buffers. Such inconsistencies must be identified, and where possible, minimized. This timing model must also include non-TSN/ DetNet queuing techniques insofar their use can affect the DetNet flows.

The IEEE 802.1 Working Group has standardized a number of specific techniqueues that can be used by routers or hosts. These documents include [IEEE8021Q] (Clause 34), [IEEE802.1Qch], [IEEE802.1Qci], [IEEE8021Qbv], [IEEE8021Qbu], [IEEE8023br].

[[NOTE (to be removed from a future revision): The queuing and transmission selection methods defined in IEEE 802.1Q and its amendments are all in the context of implementing those methods in an 802.1Q bridge; they are not all specified for use in an end station, much less in a router. It is the intention of the authors of this draft to create a document in some Standards Development Organization (SDO) that provides normative reference points for a document from any SDO describing any device, e.g. a host or a router. That would make the 802.1 queuing techniques readily available to a router or host. As that document develops, so too will this draft evolve.]]

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

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

4. Timing Model

4.1. Delay Model

In Figure 1 we see a breakdown of the per-hop latency experienced by a packet in terms that are suitable for computing both hop-by-hop latency, and per-hop buffer requirements.

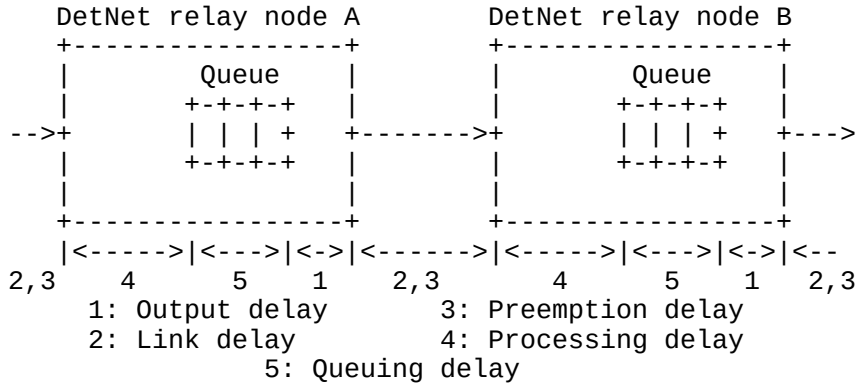

Figure 1: Timing model for DetNet or TSN

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

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

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

3. Preemption delay
   If the packet is interrupted (e.g. [IEEE8023br] preemption) in order to transmit another packet or packets, an arbitrary delay can result.

4. Processing delay
   This delay covers the time from the reception of the last bit of the packet to that packet being eligible, if there were no other packets in the queue, for selection for output. This delay can be variable, and depends on the details of the operation of the forwarding node.

5. Queuing delay
   This is the time spent from the insertion of the packet into a queue until the packet is selected for output on the next link. We assume that this time is calculable based on the details of the queuing mechanism and the sum of the variability in delay times 1-4.

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

The initial and final measurement point in this analysis (that is, the definition of a "hop") is the point at which a packet is selected for output. In general, any queue selection method that is suitable for use in a DetNet network includes a detailed specification as to exactly when packets are selected for transmission. Any variations in any of the delay times 1-4 result in a need for additional buffers in the queue. If all delays 1-4 are constant, then any variation in the time at which packets are inserted into a queue depends entirely on the timing of packet selection in the previous node. If the delays 1-4 are not constant, then additional buffers are required in the queue to absorb these variations. Thus:

- Variations in output delay (1) require buffers to absorb that variation in the next hop, so the output delay variations of the previous hop (on each input port) must be known in order to calculate the buffer space required on this hop.

- Variations in processing delay (4) require additional output buffers in the queues of that same Detnet relay node. Depending on the details of the queueing delay (5) calculations, these variations need not be visible outside the DetNet relay node.

4.2. Achieving zero congestion loss

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

Imagine a completely saturated DetNet network, in which all is part of some number of DetNet flows, and 100% of each link's bandwidth is allocated to some number of DetNet Flows using that link. Every source is transmitting at exactly its allotted rate. The DetNet flows traverse the network in all directions; no two DetNet flows take exactly the same path through the network. Imagine that there are no variations in the output delay (1), link delay (2), and processing delay (4), and there is no preemption delay (3).

Imagine now that one DetNet flow, DetNet flow A, stops. On some output port through which DetNet flow A was passing, when the transmission opportunity for one of DetNet flow A's packets comes up, the DetNet relay node must either output nothing, or output a packet belonging to some other DetNet flow B. If it outputs a packet from DetNet flow B, then in the long term, it is exceeding the normal rate for DetNet flow B, and runs the risk of overflowing the queues for DetNet flow B in the next hop. With sufficient analysis, it may be possible to determine the limits for how much excess data in DetNet flow B, or DetNet flow C, from this and from other ports feeding the next hop, can be accommodated before causing an overflow.

However, this analysis is very difficult. DetNet avoids the analysis by transmitting nothing (or transmitting a non-DetNet packet) when it has nothing to transmit for a given DetNet flow. This leads to DetNet making the following requirement for DetNet relay nodes:

For every DetNet flow traversing a DetNet relay node, sufficient data is buffered in that a DetNet relay node to ensure that a transmission opportunity for that DetNet flow is never missed, unless the source of the DetNet flow slows or stops. That is, for every DetNet flow, over some finite time scale, the input rate equals the output rate.

5. Queuing model

5.1. Queuing data model

Sophisticated QoS mechanisms are available in Layer 3 (L3), see, e.g., [RFC7806] for an overview. In general, we assume that "Layer 3" queues, shapers, meters, etc., are instantiated hierarchically above the "Layer 2" queuing mechanisms, among which packets compete for opportunities to be transmitted on a physical (or sometimes, logical) medium. These "Layer 2 queuing mechanisms" are not the province solely of bridges; they are an essential part of any DetNet relay node. As illustrated by numerous implementation examples, the "Layer 3" some of mechanisms described in documents such as [RFC7806] are often integrated, in an implementation, with the "Layer 2" mechanisms also implemented in the same system. An integrated model is needed in order to successfully predict the interactions among the

different queuing mechanisms needed in a network carrying both DetNet flows and non-DetNet flows. See Section 6 for a more complete discussion of the expanded model.

Figure 2 shows the (very simple) model for the flow of packets through the queues of an IEEE 802.1Q bridge. Packets are assigned to a class of service. The classes of service are mapped to some number of physical FIFO queues. IEEE 802.1Q allows a maximum of 8 classes of service, but it is more common to implement 2 or 4 queues on most ports.

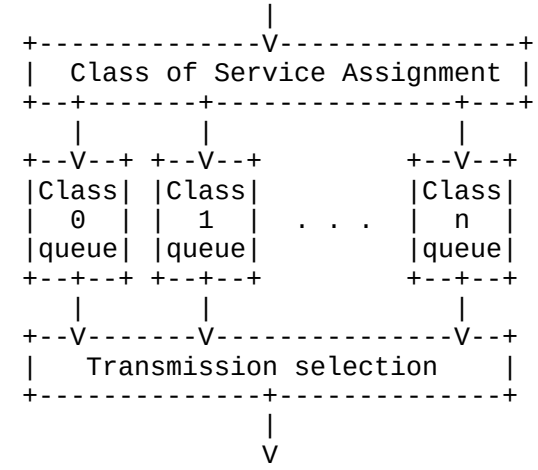

Figure 2: IEEE 802.1Q Queuing Model: Data flow

Some relevant mechanisms are hidden in this figure, and are performed in the "Class n queue" box:

- Discarding packets because a queue is full.
- Discarding packets marked "yellow" by a metering function, in preference to discarding "green" packets.

The Class of Service Assignment function can be quite complex, since the introduction of [IEEE802.1Qci]. In addition to the Layer 2 priority expressed in the 802.1Q VLAN tag, a bridge can utilize any of the following information to assign a packet to a particular class of service (queue):

- Input port.
- Selector based on a rotating schedule that starts at regular, time-synchronized intervals and has nanosecond precision.

- MAC addresses, VLAN ID, IP addresses, Layer 4 port numbers, DSCP. (Work items expected to add MPC and other indicators.)

- The Class of Service Assignment function can contain metering and policing functions.

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

5.2. Queuing Data Model with Preemption

Figure 2 must be modified if the output port supports preemption ([IEEE8021Qbu] and [IEEE8023br]). This modification is shown in Figure 3.

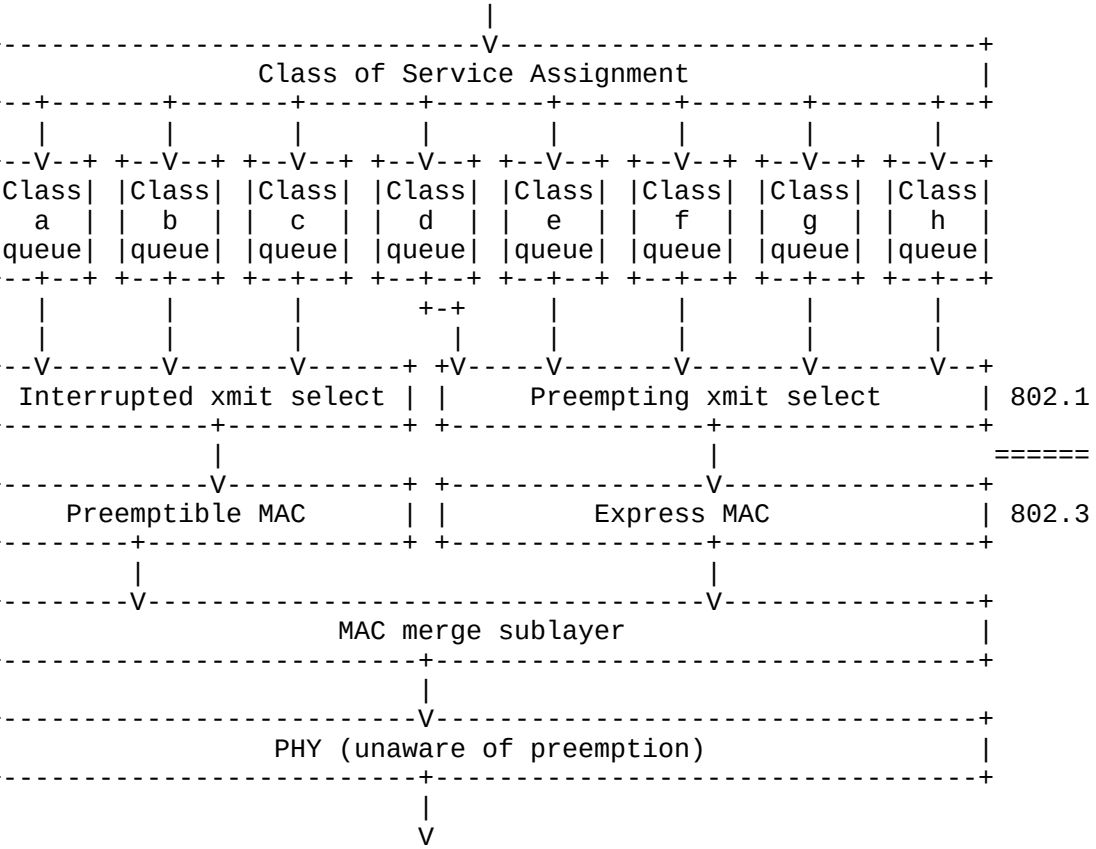

Figure 3: IEEE 802.1Q Queuing Model: Data flow with preemption

From Figure 3, we can see that, in the IEEE 802 model, the preemption feature is modeled as consisting of two MAC/PHY stacks, one for packets that can be interrupted, and one for packets that can interrupt the interruptible packets. The Class of Service (queue) determines which packets are which. In Figure 3, the classes of service are marked "a, b, ..." instead of with numbers, in order to avoid any implication about which numeric Layer 2 priority values correspond to preemptible or preempting queues. Although it shows three queues going to the preemptible MAC/PHY, any assignment is possible.

5.3. Transmission Selection Model

In Figure 4, we expand the "Transmission selection" function of Figure 3.

Figure 4 does NOT show the data path. It shows an example of a configuration of the IEEE 802.1Q transmission selection box shown in Figure 2 and Figure 3. Each queue m presents a "Class m Ready" signal. These signals go through various logic, filters, and state machines, until a single queue’s "not empty" signal is chosen for presentation to the underlying MAC/PHY. When the MAC/PHY is ready to take another output packet, then a packet is selected from the one queue (if any) whose signal manages to pass all the way through the transmission selection function.

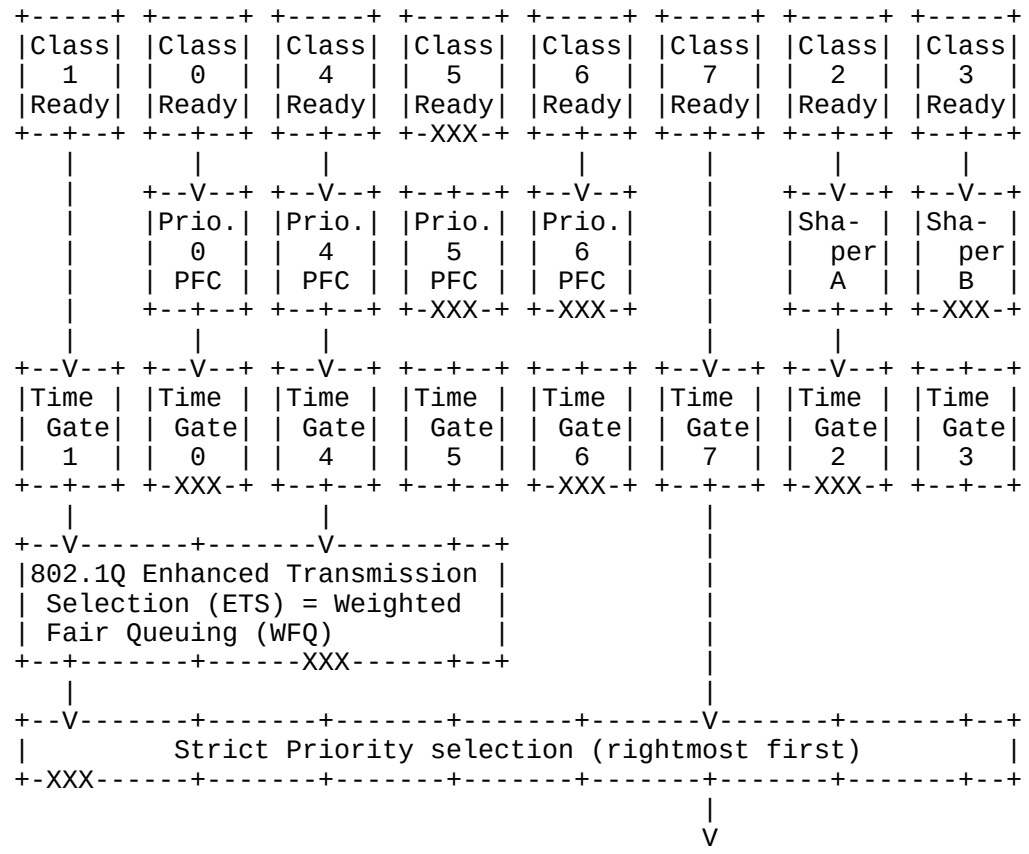

Figure 4: 802.1Q Transmission Selection

The following explanatory notes apply to Figure 4

- The numbers in the "Class n Ready" boxes are the values of the Layer 2 priority that are assigned to that Class of Service in this example. The rightmost CoS is the most important, the leftmost the least. Classes 2 and 3 are made the most important, because they carry DetNet flows. It is all right to make them more important than the priority 7 queue, which typically carries critical network control protocols such as spanning tree or IS-IS, because the shaper ensures that the highest priority best-effort queue (7) will get reasonable access to the MAC/PHY. Note that Class 5 has no Ready signal, indicating that that queue is empty.

- Below the Class Ready signals are shown the Priority Flow Control gates (IEEE Std 802.1Qbb-2011 Priority-based Flow Control, now [IEEE8021Q] clause 36) on Classes of Service 1, 0, 4, and 5, and

two 802.1Q shapers, A and B. Perhaps shaper A conforms to the IEEE Std 802.1Qav-2009 (now [IEEE8021Q] clause 34) credit-based shaper, and shaper B conforms to [IEEE8021Qcr] Asynchronous Traffic Shaper. Any given Class of Service can have either a PFC function or a shaper, but not both.

- Next are the IEEE Std 802.1Qbv time gates ([IEEE8021Qbv]). Each one of the 8 Classes of Service has a time gate. The gates are controlled by a repeating schedule that restarts periodically, and can be programmed to turn any combination of gates on or off with nanosecond precision. (Although the implementation is not necessarily that accurate.)

- Following the time gates, any number of Classes of Service can be linked to one ore more instances of the Enhanced Transmission Selection function. This does weighted fair queuing among the members of its group.

- A final selection of the one queue to be selected for output is made by strict priority. Note that the priority is determined not by the Layer 2 priority, but by the Class of Service.

- An "XXX" in the lower margin of a box (e.g. "Prio. 5 PFC" indicates that the box has blocked the "Class n Ready" signal.

- IEEE 802.1Qch Cyclic Queuing and Forwarding [IEEE802.1Qch] is accomplished using two or three queues (e.g. 2 and 3 in the figure), using sophisticated time-based schedules in the Class of Service Assignment function, and using the IEEE 802.1Qbv time gates [IEEE8021Qbv] to swap between the output buffers.

6. Extending the queuing model

6.1. Complex delay models

Using the model of Section 4, we can model any system, even one that is very complex, including separate line cards, MAC/PHY modules, mid-planes, backplanes, control/forwarding boards, etc. However, in a complex case, the variations in the processing delay (4) may become so large as to make any latency or buffer requirement analysis relatively useless.

If a DetNet node is sufficiently complex that simply assigning a minimum and maximum to the some delay (typically, the processing delay, 4) results in insufficiently accurate computations for latency or buffer requirements, the DetNet node can be modeled as a federation of DetNet relay nodes, each conforming to the model.

In the simplest example, system with input queues on each port could be modeled having a two-port DetNet relay node inserted into each input port, each with some number of output queues (which model the input queues).

6.2. Extending the 802.1Q model to routers

Extending the models described in Section 5 to routers requires a number of steps:

1. The Class of Service Assignment function of Figure 2 needs extension to the DetNet flow identification techniques use in [I-D.ietf-detnet-dp-alt].

2. Some applications will require more than 8 Classes of Service (queues).

3. The Layer 3 queues, such as are defined in [RFC7806], must be integrated with the 802.1Q queues. In some cases, this means identifying an [RFC7806] queue with an 802.1Q CoS queue, and having it compete with the other queues as shown in Figure 4. In other cases, the [RFC7806] queues may form a unit, as in Figure 2 that is separate from any specific port, and feeds a forwarding engine. Alternatively, some number of [RFC7806] queues can feed one of the Figure 2 queues.

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 5 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 may, in some implementations, 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 may be needed for DetNet packets than other packets when they get to Layer 2 queuing from Layer 3 queuing via Brdr-c.

Certainly, in the 802.1/802.3 model, different SAPs are needed for the express and for the preemptible 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 3. It depends on the implementation whether or not Brdr-a exists.

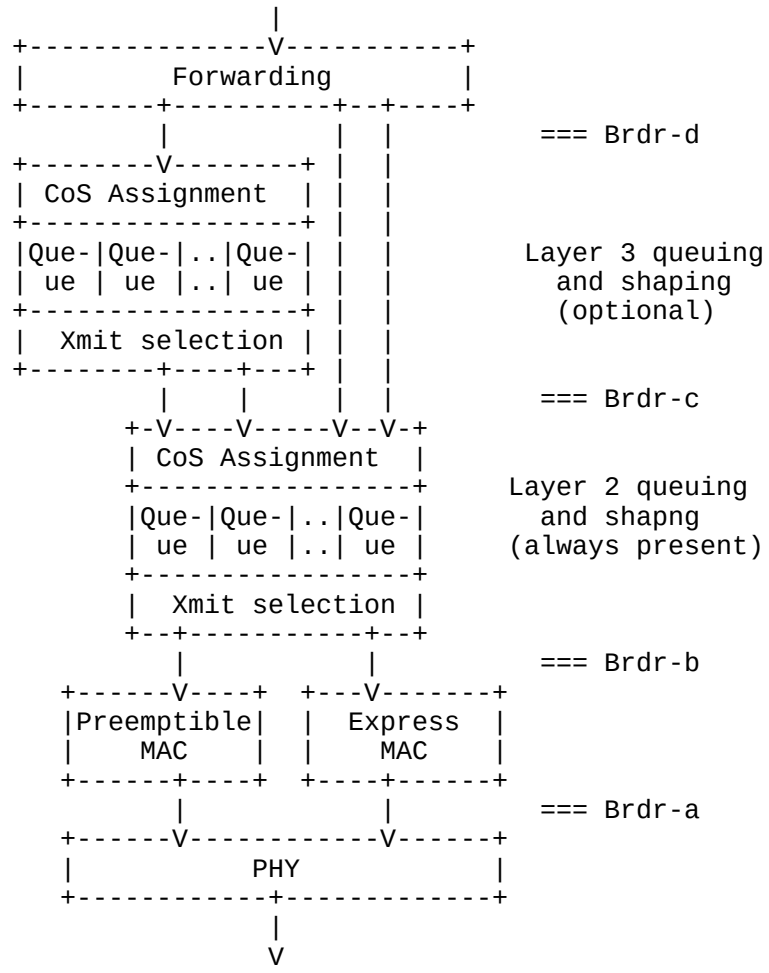

Figure 5: Combined L2/L3 Queueing Data Model

7. References

7.1. 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., Vizarreta, P., Huang, D., Geng, X., Dujovne, D., and M. Seewald, "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-13 (work in progress), September 2017.

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

[RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, DOI 10.17487/RFC2212, September 1997, <https://www.rfc-editor.org/info/rfc2212>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://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, <https://www.rfc-editor.org/info/rfc6658>.

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

7.2. Informative References

[IEEE802.1Qch]
IEEE, "IEEE Std 802.1Qch-2017 IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks Amendment 29: Cyclic Queuing and Forwarding (amendment to 802.1Q-2014)", 2017, <http://www.ieee802.org/1/files/private/ch-drafts/>.

[IEEE802.1Qci]
IEEE, "IEEE Std 802.1Qci-2017 IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment 30: Per-Stream Filtering and Policing", 2017, <http://www.ieee802.org/1/files/private/ci-drafts/>.

[IEEE8021Q]
IEEE 802.1, "IEEE Std 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, "IEEE Std 802.1Qbu-2016 IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment 26: Frame Preemption", 2016, <http://standards.ieee.org/getieee802/download/802.1Qbu-2016.zip>.

[IEEE8021Qbv]
IEEE 802.1, "IEEE Std 802.1Qbv-2015: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", 2015, <http://standards.ieee.org/getieee802/download/802.1Qbv-2015.zip>.

[IEEE8021Qcr]
IEEE 802.1, "IEEE P802.1Qcr: IEEE Draft Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment: Asynchronous Traffic Shaping", 2017, <http://www.ieee802.org/1/files/private/cr-drafts/>.

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

[IEEE8023]
IEEE 802.3, "IEEE Std 802.3-2015: 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 Std 802.3br-2016: IEEE Standard for Local and metropolitan area networks - Ethernet - Amendment 5: Specification and Management Parameters for Interspersing Express Traffic", 2016, <http://standards.ieee.org/getieee802/download/802.3br-2016.pdf>.

Authors' Addresses

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

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

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

Email: balazs.a.varga@ericsson.com

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

Email: janos.farkas@ericsson.com