

DetNet
Internet-Draft
Intended status: Informational
Expires: January 13, 2022

J. Dang, Ed.
Huawei
Z. Du
China Mobile
July 12, 2021

Services Deployment Guideline in DetNet Network
draft-dang-detnet-deployment-00

Abstract

Deterministic Networking (DetNet) defined in [RFC8655] provides a capability for the delivery of data flows with extremely low packet loss rates and bounded end-to-end delivery latency. DetNet network administrators worldwide can deploy DetNet services into their networks. This document aims to provide a guideline for DetNet network administrators.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 13, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
2. Requirements Language	2
3. Terminology & Abbreviations	3
4. Preparation of DetNet networks	3
5. How to Introduce Deterministic Flow into DetNet network . . .	4
5.1. Parameter Specification	4
5.1.1. Definition of Deterministic Flow	5
5.1.2. Establishing Explicit Path	6
5.2. DetNet Network Element Configuration and Functions . . .	9
6. How to Maintain Deterministic Flow in DetNet network	9
7. How to Withdraw Deterministic Flow in DetNet network	10
8. Deployment Trial Experience	10
9. Security Considerations	10
10. Acknowledgements	10
11. Normative References	10
Authors' Addresses	11

1. Introduction

Deterministic Networking (DetNet) defined in [RFC8655] provides a capability for the delivery of data flows with extremely low packet loss rates and bounded end-to-end delivery latency. The diverse industries in [RFC8578] have in common a need for "deterministic flows". How to introduce deterministic flows to the DetNet network is required.

While the DetNet technologies are becoming mature, the DetNet deployment is about to enter the live network experiment and even to large-scale commercial deployment. The DetNet network is actively managed by a network operations entity (the "administrator", whether a single person or a department of administrators). A network administrator is responsible for the deployment of DetNet services, who can master the skills of how to introduce deterministic flows into DetNet networks and the related maintenance.

This document is intended as guidance for DetNet network administrators.

2. Requirements Language

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

3. Terminology & Abbreviations

DetNet UPE

A DetNet edge node, which connects DetNet flows into DetNet network.

DetNet P

A DetNet relay node or DetNet transit node.

DetNet PE

A DetNet edge node, where DetNet flows leave DetNet network.

DetNet source

An end system is capable of originating a DetNet flow.

DetNet Destination

An end system is capable of terminating a DetNet flow.

4. Preparation of DetNet networks

The premise of this section is that the network has not yet enabled DetNet capability. First of all, a network administrator must enable the DetNet capability of the network on demand.

The DetNet network administrator must plan the scope of DetNet network, DetNet network topology and DetNet network element roles. As usual, the network controller has collected the topology of the entire network. So the DetNet network administrators only need to specify the scope of DetNet networks, DetNet network topology and DetNet network element roles on the controller interface. When the scope of the DetNet network is determined, the DetNet network administrators can naturally get the DetNet network topology. At that time, the DetNet network administrators must figure out whether the DetNet network is in a single domain or in multiple domains.

- o If in a single domain, it contains DetNet Ingress UPE nodes, DetNet P nodes, DetNet PE nodes. In fact, a P node may be connected to multiple different UPE devices or PE nodes or P node.
- o If in multiple domains, it also contains ASBR nodes in addition to Ingress UPE nodes, DetNet P nodes and DetNet PE nodes, for the purpose of cross-domain interconnection.

The example is shown in Figure 1, which contain DetNet Ingress UPE node, DetNet P nodes, DetNet PE node. In fact, a P node may be connected to multiple different UPE devices or PE nodes or P node.

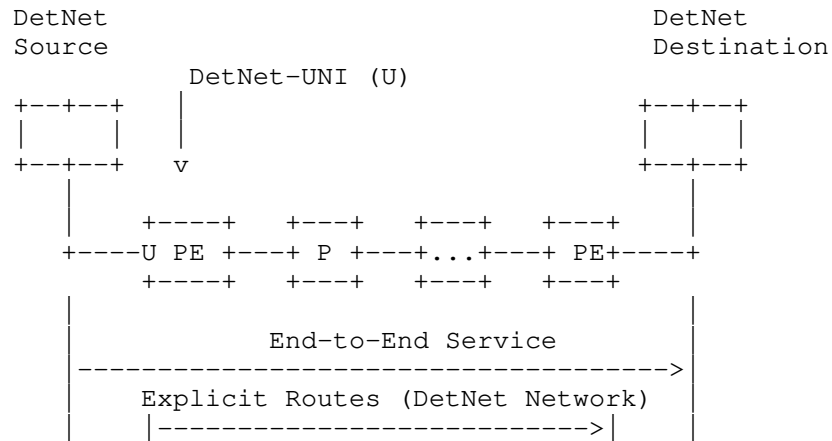


Figure-1: DetNet Network

5. How to Introduce Deterministic Flow into DetNet network

For the next work, the DetNet network administrator must specify the following information on the controller.

1. Definition of Deterministic Flow
2. Establishing Explicit Path for Deterministic Flow
 - * Definition of Deterministic Flow
 - * Specifying Encapsulation Type of Networking Technology
 - * Specifying Type of Queuing Mechanism
 - * Definition of Service Protection
 - * Network Resource Evaluation and Reservation

The section 5.1 focus on how to use these parameters.

5.1. Parameter Specification

5.1.1. Definition of Deterministic Flow

A DetNet network administrator must figure out

- o how to identify a deterministic flow,
- o the related DetNet SLA requirements,
- o which node the DetNet flow is accessed from and which node the DetNet flow leaves from.

This above information must be given the DetNet network administrator by DetNet service providers who initiate or terminate DetNet flows.

The flow identification in [RFC9016] let the DetNet UPE node identify the flow. Flow identification for MPLS and IP Data Planes are described in [RFC8939] , [RFC8964], and Ethernet information (such as MAC address, VLAN) respectively.

- o IP Data plane: five tuple
- o Ethernet data plane: MAC address or VLAN
- o MPLS or SR data plane: label

The SLA information of DetNet flow in section 5.9 of [RFC9016] are listed as follows.

- o MinBandwidth
- o MaxLatency
- o MaxLoss
- o MaxConsecutiveLossTolerance
- o MaxMisordering

If the deterministic flow has requirement for Jitter, a new parameter named jitter needs to be added.

In the follow-up work, the DetNet network administrator creates explicit route defined in section 3.2.3 of [RFC8655] according to the information which node the DetNet flow is accessed from and which node the DetNet flow leaves from.

5.1.2. Establishing Explicit Path

The DetNet network administrator must pay attention to the encapsulation type of the explicit route, which is added to the DetNet flows when DetNet flow enters the UPE node. The DetNet network administrator may freely choose encapsulation type of the networking technology according to his/her preferences. The way of IP over SR or [IP-Over-MPLS] or IP over SR) is recommended.

5.1.2.1. Encapsulation Type of Networking Technology

The DetNet network administrator must pay attention to the encapsulation type of the explicit route, which is added to the DetNet flows when DetNet flow enters the UPE node. The DetNet network administrator may freely choose encapsulation type of the networking technology according to his/her preferences. The way of IP over SR or [IP-Over-MPLS] or IP over SR) is recommended.

5.1.2.2. Type of Queuing Mechanism

The DetNet network administrator obtains or sets the queuing type used by the network. If the cyclic queuing mechanism is used in the network, the parameters of the queuing need to be set as follows. This mechanism must allow multiple deterministic flows to share a periodic buffer.

- o `CyclicBufferSize`: the length of the cyclic buffer
- o `CyclicInterval`: duration of periodic scheduling
- o `BufferNumber`: the number of the cyclic buffer
- o `MinBurstSize`: the minimum burst size that can be tolerated by cyclic queue mechanism, which is specified in octets per second and excludes additional DetNet header (if any). Bandwidth used above the Committed Information rate is called Burst traffic. It is used when the bandwidth available is more than CIR. `MinBurstSize` is the minimum burst size that has to be guaranteed for the DetNet traffic. The queuing mechanism needs to pay attention to how to shape burst size traffic into buffers.

5.1.2.3. Definition of Service Protection

The DetNet network administrator can consider how to do with service protection to meet `MaxLoss`, `MaxConsecutiveLossTolerance` and `MaxMisordering` of a deterministic flow. The premise of service protection is that there are multiple available explicit paths for a DetNet flow. These types of packet loss can be greatly reduced by

spreading the data over multiple disjointed forwarding paths. The PREOF embeded in the PE node ensures that packets are not out of order.

5.1.2.4. Network Resource Evaluation and Reservation

The DetNet network administrator can enable network resource evaluation and reservation of the controller. In fact, the network may support a distributed protocol similar to RSVP defined in [draft-trossen-detnet-rsvp-tsn], so this function can rely on the distributed protocol.

The DetNet SLA requirements to the DetNet flow generally have deterministic bandwidth, bounded latency and bounded jitter. But in fact these three parameters are interrelated. For example, the insufficient bandwidth reservation might introduce the additional delay or the additional jitter. Therefore, the bandwidth reservation should consider the latency and jitter requirements.

There are three methods here to do with, one is to get it through centralized calculation provided by controller or other centralized systems, the other is to get it through negotiation between DetNet Nodes along the explicit routes, and the third is to rely on the human brain. When the scale of the network becomes larger or the types of services become more, the third method is difficult to handle. Therefore, the first and the second methods are recommended. These centralized and distributed solutions can cooperate with each other, for example, if the centralized system is offline, the distributed system functions will be enabled. Or in order to support rapid network decision-making, the priority is given to using distributed systems for deployment, and the centralized systems are responsible for global optimization.

The algorithm on the network resource reservation is not discussed now in this document.

5.1.2.4.1. DetNet Bandwidth Evaluation and Reservation

The DetNet network administrator must know the bandwidth resource evaluation and reservation can be divided into service access interface part on the DetNet UPE node and explicit route part.

- o Service access interface part on the DetNet UPE node: The bandwidth of service access interface part on the DetNet UPE is reserved according to the MinBandwidth of the DetNet flow.

- o Explicit route part: This mechanism ensures that the available bandwidth along the explicit path can meet MinBandwidth of DetNet flow.

The P node should take into account that there are multiple explicit routes passing in the same direction. For example, if one interface of P node accesses 3 explicit paths, the reserved bandwidth of the interface is the total required bandwidth of the 3 explicit paths.

It is emphasized that the remaining bandwidth of the interface on the DetNet nodes can also be used for non-DetNet flows.

5.1.2.4.2. DetNet Latency Evaluation

The DetNet network administrator can let the controller collect the network-wide delay information for calculation and evaluation, and obtain the queuing type.

Given that DetNet nodes have a finite amount of buffer space, the resource allocation necessarily results in a maximum end-to-end latency. The overall latency of the explicit route can be calculated based on the queue scheduling mechanism on the data plane of the DetNet nodes. The queue scheduling mechanisms have various types, such as DiffServ, Qch[IEEE802.1QCH] and so on.

[DetNet-Bounded-Latency] provides end-to-end delay bound and backlog bound computations for such mechanisms that can be used by the control plane to provide DetNet QoS. If the CQF is used, CyclicBufferSize, CyclicInterval and BufferNumber of queuing mechanism can be included in the calculation factors that affect the E2E delay.

The controller evaluates the path delay based on the resources of the entire network, and judges whether it meets the MaxLatency of the deterministic flow.

5.1.2.4.3. DetNet Jitter Evaluation

The DetNet network administrator can figure out that there are two aspects to reduce network jitter. The first is through resource reservation in section 4.4.1 to 4.4.2, and the second is through effective queuing control methods. The former is not easy to evaluate jitter, but the latter is very convenient. The DetNet network administrator also can know the queuing type, because not all queuing mechanisms have a jitter control mechanism. The CQF is recommend to effectively solve the uncertainty of jitter. Under this mechanism, the end to end jitter can be controlled within $2 * \text{CyclicInterval}$.

5.2. DetNet Network Element Configuration and Functions

After the information is input by the DetNet network administrator, the controller will convert the information into the network configuration and send it to the DetNet network element node, using a protocol such as NETCONF [RFC6241]/YANG[RFC6020]. Deterministic Networking (DetNet) YANG Model defined in [DetNet-YANG] contains the specification for the Deterministic Networking YANG Model for configuration and operational data for DetNet Flows.

After DetNet network equipment receives the configuration, it starts to execute. As Figure 2 is shown, the functions of each DetNet network element is clearly visible.

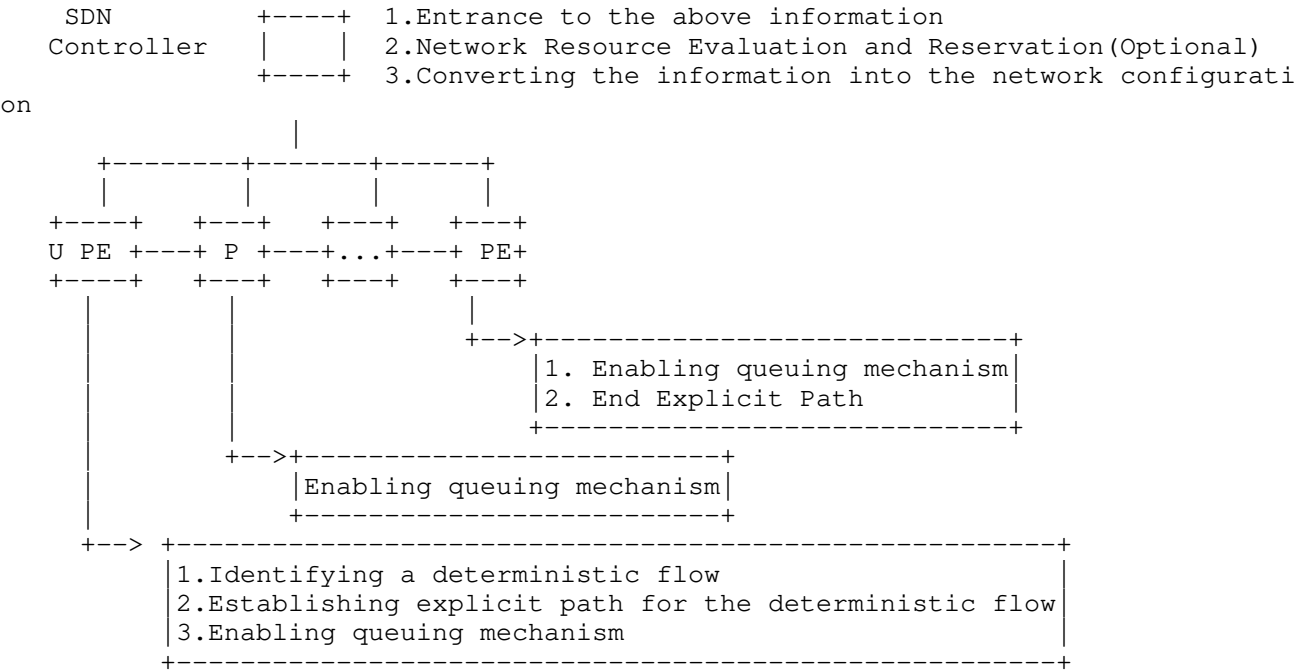


Figure-2: DetNet Network Functions

6. How to Maintain Deterministic Flow in DetNet network

TBD

If a new DetNet flow needs to be added into the existing DetNet network, the network administrators will operate according to section 4.1~4.5.

7. How to Withdraw Deterministic Flow in DetNet network

TBD

If a DetNet flow deployed needs to be canceled, the network administrator will execute the corresponding undo operation through the controller, and the network will release the corresponding resources.

8. Deployment Trial Experience

TBD

9. Security Considerations

TBD

10. Acknowledgements

TBD

11. Normative References

[DetNet-Bounded-Latency]

"DetNet Bounded Latency", <<https://www.rfc-editor.org/info/draft-ietf-detnet-bounded-latency>>.

[DetNet-YANG]

"Deterministic Networking (DetNet) YANG Model",
<<https://www.rfc-editor.org/info/draft-ietf-detnet-yang-12>>.

[draft-trossen-detnet-rsvp-tsn]

"RSVP for TSN Networks", <<https://www.rfc-editor.org/info/draft-trossen-detnet-rsvp-tsn>>.

[IEEE802.1QCH]

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

[IP-Over-MPLS]

"DetNet Data Plane: IP over MPLS", <<https://www.rfc-editor.org/info/draft-ietf-detnet-ip-over-mpls>>.

- [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>>.
- [RFC3209] "RSVP-TE: Extensions to RSVP for LSP Tunnels", <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC6020] "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", <<https://www.rfc-editor.org/info/RFC6020>>.
- [RFC6241] "Network Configuration Protocol (NETCONF)", <<https://www.rfc-editor.org/info/RFC6241>>.
- [RFC8402] "Segment Routing Architecture", <<https://www.rfc-editor.org/info/RFC8402>>.
- [RFC8578] "Deterministic Networking Use Cases", <<https://www.rfc-editor.org/info/rfc8578>>.
- [RFC8655] "Deterministic Networking Architecture", <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC8934] "Deterministic Networking (DetNet) Data Plane: MPLS", <<https://www.rfc-editor.org/info/rfc8934>>.
- [RFC8939] "Deterministic Networking (DetNet) Data Plane: IP", <<https://www.rfc-editor.org/info/rfc8939>>.
- [RFC8964] "Deterministic Networking (DetNet) Data Plane: MPLS", <<https://www.rfc-editor.org/info/rfc8964>>.
- [RFC9016] "Flow and Service Information Model for Deterministic Networking (DetNet)", <<https://www.rfc-editor.org/info/RFC9016>>.
- [RFC9023] "Deterministic Networking (DetNet) Data Plane: IP over IEEE 802.1 Time-Sensitive Networking (TSN)", <<https://www.rfc-editor.org/info/rfc9023>>.

Authors' Addresses

Joanna Dang (editor)
Huawei
No.156 Beiqing Road
Beijing, P.R. China 100095
China

Email: dangjuanna@huawei.com

Zongpeng Du
China Mobile
32 Xuanwumen West St
Beijing, P.R. China 100053
China

Email: duzongpeng@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 12, 2022

Z. Du
P. Liu
China Mobile
July 11, 2021

Micro-burst Decreasing in Layer3 Network for Low-Latency Traffic
draft-du-detnet-layer3-low-latency-03

Abstract

It is complex to support deterministic forwarding in a large scale network because there is too much dynamic traffic in the network and the data model becomes hard to predict after aggregation in the intermediate nodes. This document introduces the problem of micro-bursts in layer3 network, and proposed a method to decrease the micro-bursts in layer3 network for low-latency traffic.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 12, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

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

Table of Contents

1. Introduction	2
2. Gaps for Large-scale Layer 3 Deterministic Network	3
3. Rethinking the Problem in IP Forwarding	3
4. Method to Decrease Micro-bursts	5
4.1. Working Flow of the Method	5
4.2. Process of Edge Node	6
4.3. Process of Forwarding Node	6
5. Analysis of the Proposed Method	7
6. IANA Considerations	7
7. Security Considerations	7
8. Acknowledgements	8
9. References	8
9.1. Normative References	8
9.2. Informative References	8
Authors' Addresses	9

1. Introduction

The DetNet architecture in RFC 8655 [RFC8655] is supposed to work in campus-wide networks and private WANs, including the large-scale ISP network scenario, such as the 5G bearing network, as mentioned in RFC 8578 [RFC8578]. It is essential for the large-scale ISP network to be able to provide the low-latency service. The low-latency requirement exists in both L2 and L3 networks, and in both small and large networks.

However, as talked in [I-D.qiang-detnet-large-scale-detnet], deploying deterministic services in a large-scale network brings a lot of new challenges. A novel method called LDN (Large-scale Deterministic Network) is introduced in [I-D.qiang-detnet-large-scale-detnet], which explores the deterministic forwarding over a large-scale network. Meanwhile, the problem is introduced in [I-D.dang-queuing-with-multiple-cyclic-buffers].

This document also explores the deterministic service in the large-scale layer 3 network, and proposed a method based on micro-burst

decreasing, which can benefit the forwarding of low-latency traffic in a large-scale network.

2. Gaps for Large-scale Layer 3 Deterministic Network

In this document, the large-scale network means there are many dynamic flows in the network, but it is hard to do per-flow shaping for the intermediate nodes because they have high pressure on forwarding on the data plane.

According to RFC 8655 [RFC8655], DetNet operates at the IP layer and delivers service over lower-layer technologies such as MPLS and IEEE 802.1 Time-Sensitive Networking (TSN). However, the TSN mechanisms are designed for L2 network originally, and cannot be directly used in the large-scale layer 3 network because of various reasons. Some of them are described as below.

Some TSN mechanisms need synchronization of the network equipments, which is easier in a small network, but hard in a large network. It brings in some complex maintenance jobs across a long distance that are not needed before.

Some TSN mechanisms need a per-flow state in the forwarding plane, which is un-scalable. Aggregation methods need to be considered.

Some TSN mechanisms need a constant and forecastable traffic characteristics, which is more complicated in a large network which includes much more flows joining in or leaving randomly and the traffic characteristics are more dynamic.

The main aspects of the problems are the simplicity and the scalability. The former can ensure that the mechanism is easy to deploy, and the second can ensure that the mechanism is able to bear a large number of deterministic services.

3. Rethinking the Problem in IP Forwarding

The current IP forwarding mechanism is considered to be a good example fulfilling the requirements of simplicity and scalability. However, traditional IP network is based on statistical multiplexing, and can only provide Best Effort service, short of SLA guaranteed mechanisms.

When we rethink the problem in the current IP forwarding mechanism, we can find that in the current IP network, a long delay in queuing, or some packet losses due to burst are acceptable; however, it may be unacceptable in the deterministic forwarding. Therefore, they have different design principles in a low layer.

The current forwarding mechanism in an IP router, which is based on statistical multiplexing, cannot provide the deterministic service because of various reasons. Even be given a high priority, a deterministic packet can experience a long congestion delay or be lost in a relatively light-loaded network, which is caused by micro-bursts in the network.

Micro-burst is a special case of network congestion, which typically lasts a short period, at the granularity of millisecond. In a micro-burst, a lot of data are received on the interface suddenly, and the temporary bandwidth requirement would be tens of or hundreds of the average bandwidth requirement, or even exceed the interface bandwidth.

In most cases, the buffer on the equipment can handle the micro-bursts. However, in some corner cases, micro-bursts bring in a long delay (for example, at the granularity of millisecond) or even packet loss.

The following paragraphs introduce the causes of the micro-burst.

Firstly, IP traffic has an instinct of burstiness no matter in the macro or micro aspect, i.e., it does not have a constant traffic model even after aggregations.

Secondly, IP network has a flexible topology, where the incoming traffic may exceed the bandwidth of the outgoing interface. For example, an interface with a large bandwidth may need to send traffic to an interface with a smaller bandwidth, or multiple flows from several incoming interfaces may need to occupy the same outgoing interface.

Thirdly, the IP node has been designed to send traffic as quickly as possible, and it is not aware whether the downstream node's buffer can handle the traffic. For example, Figure 1 below shows the problem of the current IP scheduling mechanism. Before the scheduling in an IP network, the packets are well paced, but after the scheduling, the packets will be gathered even the total traffic rate is unchanged. When an IP outgoing interface receives multiple critical flows from several incoming interfaces, the situation becomes worse. However, an IP router will try to send them as soon as possible, so occasionally, in some later hops, micro-bursts will emerge.

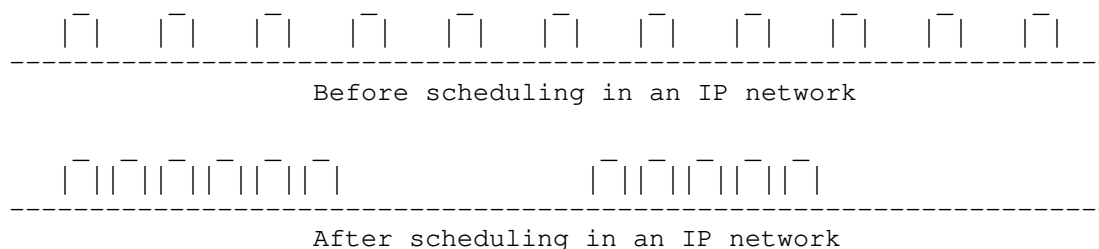


Figure 1: Change of the traffic characteristics in an IP network

This document proposes a method to support the low latency traffic bearing in an IP network, such as the 5G bearing network, by avoiding micro-bursts in the network as much as possible. The principle in this method is to forward critical and BE traffic separately, and do not distinguish different critical flows in the intermediate nodes on the forwarding plane.

4. Method to Decrease Micro-bursts

The method needs the cooperation of the edge nodes and the forwarding/core nodes in an IP network.

4.1. Working Flow of the Method

Generally, the method contains two steps:

Step1: per flow schedule in the edge node. The purpose is to make sure that each critical traffic has a constant traffic model.

Step2: per interface schedule in the core node. Traffic are aggregated to ensure the scalability, and the pacing also makes sure that they do not gather. The purpose is to make the critical traffic be forwarded as the shape when outgoing the edge, not as quickly as possible. We assume that the sending rate of the buffer for the critical traffic is the same as the receiving rate (maybe an algorithm is needed here). If all work well, the buffer will be maintained with a proper depth.

Other requirements include an RSVP liked mechanism with a good scalability, which should be used to make sure the bandwidth is not exceeded on the interface.

4.2. Process of Edge Node

The edge node of the IP network can recognize each critical flows just as in the TSN network, and then give them individually a good shaping. In fact, in TSN mechanisms, no micro-burst will emerge for critical traffic, and each TSN mechanism is proved to be effective under certain conditions.

This document suggests the edge node to shape the critical traffic by using the CBS method in IEEE 802.1Qav, or the shaping methods in IEEE 802.1Qcr. Generally, the shaping methods can generate a paced traffic for each critical flow.

The parameters of the shaper, such as the sending rate, can be configured for each flow by some means.

4.3. Process of Forwarding Node

For the forwarding node, it is uneasy to recognize each critical flow because of the high pressure of forwarding a large amount of packets. It is suggested that no per-flow state is maintained in the forwarding node. It is to say that, in the forwarding node, the critical flows should be aggregated and handled together.

This document suggests that the forwarding node can deploy a specific queue at each outgoing interface. The queue will buffer all critical traffic that need to go out through that interface, and will pace them by using methods mentioned in the last section.

The shaping method in TSN is used here instead of the original forwarding method in an IP router, which can make the critical traffic be forwarded orderly instead of as soon as possible. Therefore, micro-bursts can be decreased in the network.

If all the forwarding nodes can do their jobs properly, i.e., they can well pace the critical traffic, no or rare micro-bursts for the critical traffic would take place. In this way, the critical traffic will have a relatively low latency in the IP network with less uncertainties of micro-bursts.

As no per-flow state is maintained in the forwarding node, the sending rate of the shaper is hard to decide. In this document, the sending rate is suggested to be generated referring to the incoming rate of the queue. The purpose is to maintain a proper buffer depth for the queue.

Although it is claimed that the proposed method is simpler than the TSN mechanisms, forwarding/intermediate nodes also need to be

updated. The detailed realization of the method on the intermediate nodes is out of scope of this document.

5. Analysis of the Proposed Method

The method proposed does not need synchronization, just as the asynchronous mechanisms studied in IEEE 802.1 Qcr. Furthermore, the method has a larger aggregation granularity, which can fulfill the requirements of simplicity and scalability. However, it has a larger uncertainty in the forwarding than the TSN mechanisms.

We compare three mechanisms in the following paragraphs. The first is the priority based light-load mechanism. The second is the TSN mechanism, such as CQF. The third is the proposed mechanism.

If we only give a high priority to the critical traffic, the scalability of the deterministic system is good. However, the uncertainty in the forwarding plane perhaps can not fulfill the requirements in the industry network where SLA requirements are very essential. It should work well when only a small amount of critical traffic exist in the network.

If we use the scheduling method in the TSN, such as CQF. Its uncertainty is very low, but its scalability is not very good as said in Section 2. It should be noted that in a large deterministic system, the ISP can not guarantee 100 percent reliability, instead of which it perhaps is a value very close to.

The proposed method has a better scalability than traditional TSN mechanisms, and a better reliability than the priority based method. If we assume that different services need different deterministic levels, this method may be helpful for the service that does not need a very high deterministic level. For example, the method can be used in the consumption Internet, in which the deterministic service needs a relatively lower deterministic level than the industry Internet.

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

Detailed security considerations can refer to [I-D.ietf-detnet-bounded-latency] and [I-D.ietf-detnet-security].

8. Acknowledgements

Thanks for the valuable comments from Janos Farkas, Lou Berger, and David Black.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <<https://www.rfc-editor.org/info/rfc8578>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.

9.2. Informative References

- [I-D.dang-queuing-with-multiple-cyclic-buffers] Liu, B. and J. Dang, "A Queuing Mechanism with Multiple Cyclic Buffers", draft-dang-queuing-with-multiple-cyclic-buffers-00 (work in progress), February 2021.
- [I-D.ietf-detnet-bounded-latency] Finn, N., Boudec, J. L., Mohammadpour, E., Zhang, J., Varga, B., and J. Farkas, "DetNet Bounded Latency", draft-ietf-detnet-bounded-latency-05 (work in progress), April 2021.
- [I-D.ietf-detnet-security] Grossman, E., Mizrahi, T., and A. J. Hacker, "Deterministic Networking (DetNet) Security Considerations", draft-ietf-detnet-security-16 (work in progress), March 2021.
- [I-D.qiang-detnet-large-scale-detnet] Qiang, L., Geng, X., Liu, B., Eckert, T., Geng, L., and G. Li, "Large-Scale Deterministic IP Network", draft-qiang-detnet-large-scale-detnet-05 (work in progress), September 2019.

Authors' Addresses

Zongpeng Du
China Mobile
No.32 XuanWuMen West Street
Beijing 100053
China

Email: duzongpeng@foxmail.com

Peng Liu
China Mobile
No.32 XuanWuMen West Street
Beijing 100053
China

Email: liupengyjy@chinamobile.com

DETNET
Internet-Draft
Intended status: Informational
Expires: January 13, 2022

T. Eckert
Futurewei Technologies USA
S. Bryant
Stewart Bryant Ltd
July 12, 2021

Problems with existing DetNet bounded latency queuing mechanisms
draft-eckert-detnet-bounded-latency-problems-00

Abstract

The purpose of this memo is to explain the challenges and limitations of existing (standardized) bounded latency queuing mechanisms for desirable (large scale) MPLS and/or IP based networks to allow them to support DetNet services. These challenges relate to low-cost, high-speed hardware implementations, desirable network design approaches, system complexity, reliability, scalability, cost of signaling, performance and jitter experience for the DetNet applications. Many of these problems are rooted in the use of per-hop, per-flow (DetNet) forwarding and queuing state, but highly accurate network wide time synchronization can be another challenge for some networks.

This memo does not intend to propose a specific queuing solution, but in the same way in which it describes the challenges of mechanisms, it reviews how those problem are addressed by currently proposed new queuing mechanisms.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 13, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Summary	3
1.1. Problem: High speed forwarding, high scale fan-in/fan-out	3
1.2. Solution goal: Lightweight, per-hop, per-flow stateless transit hop forwarding	4
1.3. Requirement: Support for existing stateless / steering solutions	4
1.4. Requirement: PCE to ingress/egress LSR only flow signaling	4
1.5. Requirement: Support for DiffServ QoS model on transit hops.	4
1.6. Requirement: Low jitter bounded latency solutions. . . .	4
1.7. Requirement: Dynamic, application signalled DetNet flows	5
2. Evolution of IP/MPLS network technologies and designs	5
2.1. Guaranteed Service with RSVP	5
2.2. Hardware forwarding and DiffServ	6
2.3. MPLS and RSVP-TE	6
2.4. Path Computation Engines (PCE)	7
2.5. Segment Routing (SR)	8
2.6. BIER	8
2.7. Summary	8
3. Additional current considerations	9
3.1. Impact of application based state in networks	9
3.2. Experience from IP multicast	9
3.3. Service Provider and Private MPLS Networks	10
3.4. Mission-specific vs. shared infrastructures	11
3.5. PTP and challenges with clock synchronization	12
3.6. Jitter - in-time versus on-time	13
4. Challenges for high-speed packet forwarding hardware	15
5. A reference network design	16
6. Standardized Bounded Latency algorithms	19
6.1. Guaranteed Service (GS)	19

6.2.	TSN Asynchronous Traffic Shaping (TSN-ATS)	19
6.3.	Cyclic Queuing and Forwarding (CQF)	20
7.	Candidate solution directions	22
7.1.	Packet tagging based CQF	22
7.2.	Packet tagging based CQF with SR	23
7.3.	Per-hop latency indications for Segment Routing	23
7.4.	Latency Based Forwarding	24
8.	Conclusions	25
9.	Security Considerations	25
10.	IANA Considerations	25
11.	Acknowledgements	25
12.	Informative References	26
	Authors' Addresses	28

1. Summary

The architectural evolution of IP/MPLS networks (Section 2) in service provider and other "larger-than-building" (Section 3.3), shared-infrastructure service networks (Section 3.4) has led to a range of requirements against per-hop forwarding mechanisms which are currently not supported by the current DetNet MPLS forwarding plane [RFC8964] and per-hop, per-flow queueing model [RFC8655], Section 3.2, especially with respect to the QoS support of per-hop bounded latency. The authors of this memo think that solutions for these requirements are relatively easily added to the existing DetNet architecture by adding support for already existing and/or proposed, but not standardized per-hop forwarding and queueing options.

The following sub-sections summarize the problem, solution goals and requirements as perceived by the authors. The reasoning for these is explained in the following sections.

Note that requirements are somewhat overlapping in so far as solving one of them also solves others, but each addresses the problems from a different perspective, and are therefore easier understood for different stakeholders. For example: Operators that do want to see support of DetNet for example for Segment Routing (SR) would not think that this is "naturally" the same as DetNet supporting the DiffServ architecture, even though solutions would have a hard time to support only one of the two.

1.1. Problem: High speed forwarding, high scale fan-in/fan-out

Forwarders with bounded latency need to support interface speeds of 100 Gbps up to Tbps, likely over a period of 10 years from initial deployment of possible DetNet solutions. Hundreds of interfaces may need to be supported in a single forwarder (fan-in/fan-out).

Supporting bounded latency at these speeds and fan-in/fan-out raises cost and feasibility challenges beyond those that had led to past IETF IntServ (GS) standards ([RFC2210], [RFC2212]) or more recent TSN bounded latency solutions.

Note that these high speed and scale requirements even cause challenges when DetNet bounded latency traffic is intended to be used for only a small percentage of the interfaces traffic.

1.2. Solution goal: Lightweight, per-hop, per-flow stateless transit hop forwarding

Both high-speed hardware and network architecture design (for reasons of simplicity and minimization of shared risk functions) do favor architectures that support a lightweight transit hop forwarding plane design that requires no forwarding plane or control plane operations whose scale support depends on the number of services/service-instances (e.g.: DetNet flows) offered, but at best only on the size of the network (e.g.: no per-flow, per-hop state).

1.3. Requirement: Support for existing stateless / steering solutions

There should be DetNet bounded latency options that work in conjunction with per-transit-hop stateless traffic forwarding such as through Shortest Path First (SPF) routing with IP/MPLS), engineered steering (e.g.: SR) and stateless replication, such as Bit Indexed Explicit Replication with/without Tree Engineering (BIER, BIER-TE).

1.4. Requirement: PCE to ingress/egress LSR only flow signaling

There should be DetNet bounded latency options that for the purpose of traffic engineering (including assurance of bounded latency across the network) only require per-flow Path Computation Engine (PCE) signaling to network ingress/egress router, but not to transit hop routers.

1.5. Requirement: Support for DiffServ QoS model on transit hops.

There should be DetNet bounded latency options that support the DiffServ QoS model instead of only the IntServ model.

1.6. Requirement: Low jitter bounded latency solutions.

There should be DetNet bounded latency options that together with the other requirements also provide a better than worst-case jitter for DetNet traffic.

1.7. Requirement: Dynamic, application signalled DetNet flows

The DetNet architecture should support signaling and forwarding that would make support for automatically application instantiated DetNet flows scalable and lightweight to operate.

2. Evolution of IP/MPLS network technologies and designs

To help readers understand especially the per-hop stateless requirement from above, the following sections summarizes the historical evolution of technologies and operational principles that the authors think are relevant to understand the requirements outlined above and asks to see supported in DetNet.

2.1. Guaranteed Service with RSVP

The original (first and only) IETF standardized packet forwarding layer standardized queuing option for bounded latency in the IETF is "Guaranteed Service", [RFC2212] (GS), see the DetNet bounded latency document, [DNBL] section 6.5. At the time the RFC was published (1997), the standardized signaling was proposed to be RSVP [RFC2205], and the use of RSVP with GS was standardized in [RFC2210].

The function to support GS bounded latency in the forwarding plane is the per-flow reshaping on every forwarder hop along the path where GS packets of one flow may get delayed in the egress interface queue due to packets from other GS flows. In typical networks, this is every hop along the path.

Early (1990/2000) forwarders for which RSVP was implemented where using so-called "software" forwarding. This meant that the forwarding plane was implemented through a general purpose CPU without additional hardware support for QoS functions such as shaping or queuing. While these forwarders did support traffic flow shaping, GS was never implemented on them and their RSVP implementations did also not support (but ignored) the RSVP TSPEC/RSPEC signaling parameters used for bounded latency. Instead, RSVP implementations only supported the parameters for bandwidth reservation, which was henceforth called Call Admission Control (CAC).

In one instance, a software forwarder implementation with RSVP supported the Controlled Load (CL) service [RFC2211], which does not provide for bounded but instead for controlled latency. This service is achieved by creating a per-flow queue and applying weighted fair queuing (WFQ) with weights according to the reserved bandwidth of the flows (see [RFC2211], section 11). This functionality did not proliferate into later generations of routers because the execution cost of WFQ was too high for a multitude of flows and the scheduling

accuracy was too inaccurate in interrupt driven CPU software forwarding with higher speed interfaces (100Mbps...1Gbps).

2.2. Hardware forwarding and DiffServ

With the rise of forwarding planes with "acceleration" through ASIC based Forwarding Plane Elements (FPE) instead of general purpose CPUs and/or dedicated QoS hardware, the ability of forwarders to support shaping evolved to only be supported, if at all, on DiffServ (DS) boundary nodes, but not on DS interior nodes. This included both shaping as well as complex queuing such as WFQ.

The DS architecture, [RFC2475], was specifically targeted to enable the evolving, now common Service Provider network services architecture, in which "high-touch" service functions are only performed on so-called Provider Edge (PE) routers, which as required are DS boundary nodes, whereas the hop-by-hop forwarding through so-called Provider (P) (core) routers is meant to utilize only a reduced set of forwarding functions, specifically excluding per-hop, per-flow QoS forwarding plane functions such as shaping or policing. DiffServ therefore allowed to build higher speed, lower cost forwarding plane P routers. It also enabled to build equally higher speed, lower costs PE routers by supporting boundary node functions only on (lower speed) customer facing interfaces/line cards, but not on core facing interfaces.

2.3. MPLS and RSVP-TE

With the advent of MPLS [RFC3031], RSVP was extended to support MPLS through the RSVP-TE [RFC3209] extensions. RSVP-TE manages p2p (later on also p2mp) MPLS Label Switched Paths (LSP), which when signaled through RSVP-TE are also called RSVP-TE tunnels. These can be seen as the equivalent of IP flows that RSVP manages for IP. RSVP-TE tunnels can support a variety of traffic engineering functions, but none of the implementations known to the authors ever implemented GS or CL services, specifically because hardware forwarding for service provider networks was not designed to support these QoS functions for P Label Switched Routers (LSR).

Because CL/GS were not targeted with RSVP-TE, the signaling extensions for Interior Gateway Protocols (IGP) required in the classical RSVP-TE reservation model (such as [RFC8570] for IS-IS) have no parameters to signal per-hop GS queuing latency or buffer capacity utilization. In result, the existing IGP signaling for RSVP-TE only supports RSVP-TE to perform bandwidth but not non-queuing path latency resource calculations and therefore no latency based traffic engineering.

2.4. Path Computation Engines (PCE)

Even though RSVP-TE implementations support only DiffServ (but not GS/CL) with respect to per-hop QoS functions, its traffic-steering (path selection) and signaling model introduced per-flow (per-tunnel) control plane and forwarding plane overhead onto every P-hop. Through the 200x's, this RSVP-TE overhead was seen as undesirable complexity and overhead by many service providers using it. There was also a much larger number of service providers that desired some of the benefits provided by RSVP-TE, but who were not willing to commit to the complexity, costs and operational risk introduced into the network by complex per-flow signaling of RSVP-TE. The on-path, per-hop signaling of RSVP-TE for example introduced so much overhead, that reconvergence of RSVP-TE paths after a failure or recovery took as much as 20 minutes in networks with 10,000 or more RSVP-TE tunnels.

The design of RSVP-TE's (decentralized) on path signaling model specifically showed problematic under high resource utilization. In the original, decentralized RSVP-TE deployment model, ingress PE LSR would perform so-called Constrained Shortest Path Forwarding (CSPF) calculations to determine the shortest path with enough free resources for a new flow. Afterwards the ingress PE would signal the path via RSVP-TE. The IGP would signal to all ingress PE how many (bandwidth) resources were left on every link. Under high load, when multiple ingress PE were performing this process in parallel this would cause high load, churn and reservation collisions.

These problems of de-centralized RSVP-TE plus IGP signaling lead to the introduction of a so-called Path Computation Element (PCE) based architecture, in which the (competing and uncoordinated) traffic engineering computations on every de-centralized RSVP-TE ingress LSR were replaced by a centralized PCE function (or at least a coordinated PE function), which would send the calculated results back as a path object to the headend LSR, in result limiting the functions of RSVP-TE to the signaling of a steered traffic path through the network to establish the hop-by-hop LSP. The use of a PCE can likewise eliminate all the reservation state dependent signaling from the RSVP-TE IGP extensions, because all the reservation calculations solely need to happen only on the PCE. Nevertheless, the PCE does not eliminate the per-hop signaling overhead of RSVP-TE to establish LSPs and hence it did not eliminate for example the majority of the platform and convergence cost of RSVP-TE in the network, especially for the control plane of P nodes and could hence not resolve the concerns of service providers who had chosen not to adopt RSVP-TE.

2.5. Segment Routing (SR)

The introduction of centralized PCE had obsoleted most of the reasons for RSVP: headends did not need to do path calculation, and P router did not need to manage the available and allocated bandwidth for TE tunnels. In most service-provider use-cases this left RSVP-TE only serving as a very complex solution to do traffic steering, and the PCE was doing the rest. This ultimately lead to the design of the Segment Routing [RFC8402] architecture, and its mapping to the MPLS forwarding plane, SR-MPLS [RFC8660]. Later, a mapping to IPv6 was defined with SRv6 [RFC8986]. SR relies on strict or loose hop-by-hop hop source routing information, contained in each packet header, therefore eliminating the need to set up per-path flow state via RSVP-TE, and allowed in conjunction with DiffServ for hop-by-hop QoS a complete per-hop, per-flow stateless forwarding solution, arguably therefore lightweight, easy to implement at high performance and scalable to large number of flows.

2.6. BIER

In the same way as SR eliminated the need for hop-by-hop traffic steering forwarding state from RSVP-TE in P-routers for unicast traffic, Bit Indexed Explicit Replication [RFC8279] (BIER) solves this problem for shortest path multicast replication state across P-routers, by replacing it with a BIER packet header [RFC8296] and therefore eliminating any per-application/flow, per-hop forwarding state for multicast in P-routers. BIER also removed the associated overhead of prior ingress replication solutions Service Providers where looking into to avoid the per-hop state.

Finally, BIER-TE [I-D.ietf-bier-te-arch] adds traffic steering with replication to the BIER architecture and calls this Tree Engineering. Likewise, this is without the need for per-hop/per-flow steering or replication state.

2.7. Summary

Service Provider networks have evolved especially in the past 25 years into an architecture, where high-speed, low-cost and high-reliability are based on designs that eliminate or reduce as much as possible any form of unnecessary control-plane and even more so per-flow, per-application plane complexity from P-routers/transit-nodes.

This has led to the development of the DiffServ QoS architecture that eliminated IntServ/per-flow QoS from P-routers, and later on to the evolution from MPLS/RSVP-TE to SR and BIER that eliminated per-flow/tunnel forwarding/steering and replication state from the same P-nodes.

Finally, early experience with Traffic Engineering churn under high load and today's requirements for often NP-complete optimization lead to an architectural preference for off-path/centralized model for TE calculations via PCE to also free P-routers from signaling complexity and perform dynamic/service-dependent signaling only to PE-routers.

3. Additional current considerations

The following subsections look at further into the background for why per-hop, per-flow state can be problematic and discuss problems beyond this core issue.

3.1. Impact of application based state in networks

RSVP-TE was (and is) solely used for services where the operator of a domain explicitly provisions RSVP-TE tunnels across its domain (for example using a PCE) and can therefore fairly easily know the worst-case scaling impact. For example the number of tunnels does not arise as a chance value arising through dynamic subscriber action, and the number of tunnels in the network is primarily impacted by topological changes and the (over time relatively rare) occurrences of additional services and/or service instances being provisioned. For RSVP-TE there was never (to the knowledge of the authors) an end-to-end application layer interface such as there was for RSVP over IP, for example as supported by earlier versions of Microsoft Windows QoS enabled IP sockets.

When per-flow operations including per-hop signaling or even worse per-hop forwarding plane or QoS state is not a result of well-controlled provisioning or well-plannable/predictable failure behavior but instead driven by applications not under the control of network operators, the per-hop state requirements can become much more an operational and cost problem, because of its unpredictability.

3.2. Experience from IP multicast

The widest experience with dynamic, application based signaling in Service Provider networks likely exist for IP multicast, where creation of per-hop forwarding/replication state is triggered by applications not under the control of network operations but by customer managed applications/application-instances. Managing the amount of state and the control plane load on P-routers was and is one of the major concerns when operationalizing IP Multicast services in SPs.

Service Provider L2-VPN and L3-VPN services can offer IP Multicast via architectures such as [RFC6513] that attempt to solve/reduce the

problem of customer application driven, per-multicast application in a variety of ways, but they all come with their own problems:

- o In ingress-replication, the ingress-PE sends a separate unicast copy to every egress-PE. This creates significant excess traffic on links close to the ingress-PE and potentially higher-cost ingress-PE attachment speeds.
- o In L3VPN aggregates-trees, the traffic for multiple trees is sent across a common tree reaching the superset of all egress-PE of all included trees. This reduces the number of trees from one per-customer application to a lower number of aggregates this, but it creates potentially significant excess traffic towards egress-PE that do not need all the aggregated traffic and may even result in a requirement for access core access link speeds for those egress routers.

Finally, the per P-router stateless BIER solution solved these issues. It does not require any per P-router, per tree state creation, and achieves a 256x better traffic efficiency than ingress replication (with 256 long BIER bit strings).

3.3. Service Provider and Private MPLS Networks

With DetNet services being targeted primarily for so-called private networks such as (but not limited to) those for industrial, theme parks, power supply systems, road, river, airport and train transportation networks, it is important to understand how concerns for SP networks will apply to such private networks:

While the aforementioned evolution of MPLS networks focused on large-scale service provider networks, the very same architectural evolution is or will also happen in any private MPLS networks in the same way as the DiffServ architecture equally became the only widely adopted QoS architecture in any larger scale (campus or beyond) private networks.

While some of the scaling, cost, performance and reliability issues mentioned above for service providers may not equally apply to smaller scale private networks, past experience has shown that that it is unlikely for a critical mass for different solutions to develop across a large variety of vertical private type of networks. For this reason, in the past any larger scale enterprise networks have preferred to adopt solutions that had proven themselves through SP deployments and that were based on cross-vendor IETF based architecture principles and widely, interoperable vendor implementations.

Another reason for private network operators looking for service provider calls designs is that it also simplifies potential service provider based management of the network and/or outsourcing of the network to a service provider. This was seen often when large enterprises that had to support multi-tenants evolved from ad-hoc network virtualization solutions (such as VRF-lite) over to BGP/MPLS-VPN designs and later outsourced those very networks.

In that same line of future proofing, networking technologies first developed for enterprises would also be picked up and reused in Service Provider networks as long as they would fit. IP Multicast for example had (since about 1996) ca. 10 years of deployment for business critical enterprise use cases (such as financial market data distribution), before it was adopted widely for IPTV in service providers.

3.4. Mission-specific vs. shared infrastructures

Whereas the previous section points to the practice and benefits to share technologies between private and SP network, this section highlights one core additional requirement of SP networks not found in most private networks from which pre-DetNet deterministic service requirements will likely originate.

In architectural terms, the desire and need to minimize or avoid per-application/flow forwarding/control-plane state and per-hop control plane interactions (be it through on-path signaling or direct PCE to P-router signaling) is not primarily a matter of SP/private networks or not even of size, but foremost a matter of whether or not the network itself is seen as the (a) communications fabric of a large distributed application or (b) as an independently running shared infrastructure across a potentially wide variety of application/services with diverging requirements.

(a) is the dominant view of the network specifically from many (single) mission specific networks such as many industrial networks and even non-public High Performance Compute (HPC) center architectures. In either of these case, it is a single architectural entity that can control both network infrastructure and application to build a mission optimized compound.

For example, switches in HPC Data Centers had traditionally very shallow interface packet buffering for cost reasons, resulting in inferior performance under peak load with predominant older TCP congestion control stacks. Instead of using better, more expensive switches, it was easier to improve application device TCP stacks, leading for example to BBR TCP. While this is very much in line with the desired Internet architecture that is putting a significant

responsibility onto transport layer protocols in hosts (not limited to TCP) to behave "fair" or "ideal", the reality even in many private missions centric networks such as manufacturing plant is different. Dealing with misbehaving user devices or applications is one of the main challenge. In the example, that is the case when a DC is offering public cloud services, where TCP stacks can not be controlled, and hence deeper buffers and/or better AQM are a core requirement.

In general: In networks following the (b) shared infrastructure design principle, any resource that needs to be shared across different services or even service instances becomes a potential three party reliability and costing issue between the provider running the network and the two (or more) parties whose services utilize the common resource. Minimizing the total amount of complex, failure-prone and hard to quantify in a cost-effective manner shared resources is thus at the base of any shared infrastructure network design.

This again points to the model, where all network control can happen on the edge, and due to the absence of per-hop, per-flow state there simply is no shared flow state table that needs to be managed across multiple different services/subscribers.

3.5. PTP and challenges with clock synchronization

Some bounded latency solution require accurate clock synchronization across network nodes performing the bounded latency algorithm. The most commonly used (family of) protocol(s) for this is the Precision Time Protocol (PTP), standardized in IEEE1588 and various market specific profiles thereof.

PTP can achieve long-term Maximum Time Interval Errors (MTIE) of as little as 10th of nsec. MTIE is the maximum time difference between the clocks of two PTP nodes measured over long period of time.

Implementing PTP in devices comes at a range of design requirements. At high degree of accuracy, PTP requires accordingly accurate local oscillators that includes hardware such as regulated heating to operate under constant temperature. It includes accurate distribution of clock across all components of the system, which can be especially challenging in modular, large-scale devices, and accurate insertion and retrieval of timestamp field into packet headers.

While PTP is becoming more and more widely available, consistent support of high accuracy across all target type of switches and routers in wide area networks cannot be taken for granted to be a

feasible new requirement raised for DetNet when it did not exist in before. Today, PTP is often found in mobile network fronthauls, but not their backhauls or any other broadband aggregation, distribution or core networks. This is because there is, as of today, no strong business case requirement for PTP at high precision in those networks, whereas technologies such as eCPRI raise such requirements against mobile fronthauls. Instead, those other networks most often resort to at best msec accuracy NTP protocol deployments which is typically sufficient for control-plane and operational event tracing as its main, accuracy defining use-case.

The larger the network and more multi-vendor varied the deployed equipment is, the higher will also be the operational cost of maintaining and controlling the accuracy of a PTP service. This primarily has been cited in the past as a reason to not deploy PTP even if hardware was supporting it. This operational challenge will especially apply when PTP support may be required for only a small percentage of traffic in a high speed wide area network. The revenue from the service needs to cover the operational cost incurred by its exclusive components (hardware, software and operations).

3.6. Jitter - in-time versus on-time

This section discusses how low-jitter bounded latency applications can be highly beneficial for DetNet applications.

Depending on the bounded latency algorithm, the jitter experienced by packets varies based on the amount of competing traffic. In algorithms and their resulting end-to-end service which this memo calls "in-time", such as GS and [TSN-ATS], the experienced latency in the absence of any competing traffic is zero, and in the presence of the maximum amount of permissible competing traffic, latency is the maximum, guaranteed bounded latency. In result, the jitter provided by these algorithms is the highest possible.

In algorithms and their resulting end-to-end service which this memo calls "on-time", the experienced latency is completely or most significantly independent of the amount of competing traffic and the jitter therefore null or minimal. In these algorithms, the network buffers packets when they are earlier than guaranteed, whereas in-time algorithms deliver packets (almost) as fast as possible.

This memo argues that on-time queuing algorithms provide an additional value-add over in-time algorithms, especially for use in metropolitan or wide-area networks. Whatever algorithm is used, the receiving application only has a guarantee for the maximum bounded latency, and the real (shorter) latency of any received packet is no indication for the latency of the next packet. Instead, the receiver

application has to be prepared for each and any future packet to arrive with the worst possible, e.g.: the bounded latency.

The majority of applications require some higher layer function synchronously to the sender application: Rendering of audio/video and other media information needs to happen at the same frequency or event intervals at which the media was encoded. When these applications receive packets earlier than the time at which they can be processed (which is equal or close to the bounded latency), these applications buffer media in a so-called playout buffer and release them only at that target time. Likewise, remote control loops including industrial Programmable Logic Controller (PLC) loops or remote controlling of robots or cars is typically based on synchronous operations. In these applications, early packets are also delayed to then be processed "synchronously" later.

In all cases, where applications need to buffer (or otherwise remember) received data when it is too early, in-time queueing latency raises the challenge to application developers to be able to predict the networks worst possible jitter, and this can be particularly challenging for embedded, if not constrained receiver devices with minimum memory to buffer/remember. When these devices are designed against one particular type of network with well-known low jitter, then they will not necessarily operate correctly in networks with larger jitter. And in metropolitan and WAN networks, jitter with in-time services can be highly variable based on its design and the relative location of the communicating nodes in the topology (see Section 5 for an example network design).

One example of such issues was encountered when digital TV receivers (Set Top Boxes, STB) designed for (mostly synchronous) digital cable transmission where evolved to become IPTV STB, but the playout buffer of < 50 msec was not sufficient to compensate for a > 50 msec jitter experienced in IP metropolitan networks.

Note that this section does not claim that all applications will benefit from on-time service, nor that no application would benefit more from in-time service than from on-time service. Nevertheless, the authors are not aware of instances of [RFC8578] application for whom in-time service would be more beneficial than on-time service. Of course, this comparison is only about the benefit to the application and other factors such as the cost/scale of the service for the network itself have also to be taken into account.

4. Challenges for high-speed packet forwarding hardware

The problems of cost and operational feasibility in shared-infrastructure networks specifically applies to scaling of hardware resources such as per-application-flow forwarding or QoS state in high-speed network routers: Even if the business case makes it clear that only e.g. 1 Gbps worth of traffic may require this advanced state (such as multicast replication or per-flow shaping for bounded latency), it will be more expensive to build this functionality into a 100 Gbps transit switch/router than into a 1 Gbps switch/router. This too is based on experience from migrating services of low-speed mission specific networks, such as IP multicast onto high speed, shared-infrastructure service provider networks.

The reason for this higher cost at higher speed is that the 1 Gbps worth of "advanced" traffic still has to be built into 100 times faster hardware and each of the "advanced" packets forwarded would need to be replicated/shaped 100 times faster.

This packet processing issue may look like it applies equally to both per-hop, per-flow stateful based forwarding as well as solely in-packet based mechanisms, in practice, per-flow state may require a lot more high-speed memory access because of the need to access an entry from a state table. In most cases, this table space can only be made to work at line rate packet processing when it is on-chip, hence it is not only most expensive, it is also crucial to scale right. And as the 1 vs. 100 Gbps example above showed, it is very hard to come by an appropriate scale smaller than "would work for 100% of traffic" - because network operator providing shared infrastructure networks really do not want to be responsible for predicting how individual services may grow in adoption by making a specific hardware selection that constrains any such growth.

Last, but not least, on-chip high-speed state tables become even more expensive when they do not only have to be read only, but also when they have to be written at line rate and even worse, when they have to operate for line-rate speed read/write/read control loops:

The main issue with scaling state in hardware routers is that designs will be hesitant to work against unclear growth predictions. Even if at some point in time only 1 Gbps of DetNet traffic was expected to be required on a 100 Gbps platform, hardware designers will be much more likely to scale against the worst (best) case service growth expectation so that customers will not feel that they would buy into a product that becomes obsolete under success.

Whereas steering state, such as MPLS label entries can easily scale to hundreds of thousands, the same is not clear about shapers or

interleaved regulators. They are more challenging because they require fast (on-chip) read-write memory for the state variables, especially when forwarding is parallelized across multiple execution unit. This does incur additional complexity to split up the state and its packets across multiple execution units and/or to provide consistent cross-execution units shared read/writeable memory.

Even only writeable (but not cross-execution units then also readable) memory has traditionally been a sparse resource the faster the forwarding engines are. This can be seen from (often very limited) scale of packet monitoring state such as for IPfix.

But the main issue of per-hop, per-flow forwarding state that could be quite dynamic because it might be triggered by applications is the control plane to forwarding-plane-state interactions. Updating hardware forwarding engine state tables is often one of the key performance limits of routers. Adding significant additional state with likely ongoing changes is easily seen as a big contributor to churn in the control plane and likely reason for stability and reduced peak performance under key events such as reconvergence of all or large parts of IGP or BGP routing tables.

5. A reference network design

The following picture shows an example, worst-case network topology of interest (in the opinion of the authors) for bounded latency considerations. This section does not claim that greenfield rollouts may or want to use all aspects of this topology. What this memo does claim is that many existing brownfield networks, especially large metropolitan areas show all or many of these aspects, and that it would be prudent for bounded latency network technologies to support networks like these so as to not create new constraints against network designers by only supporting physical network topologies optimized for a particular type of service (bounded latency).

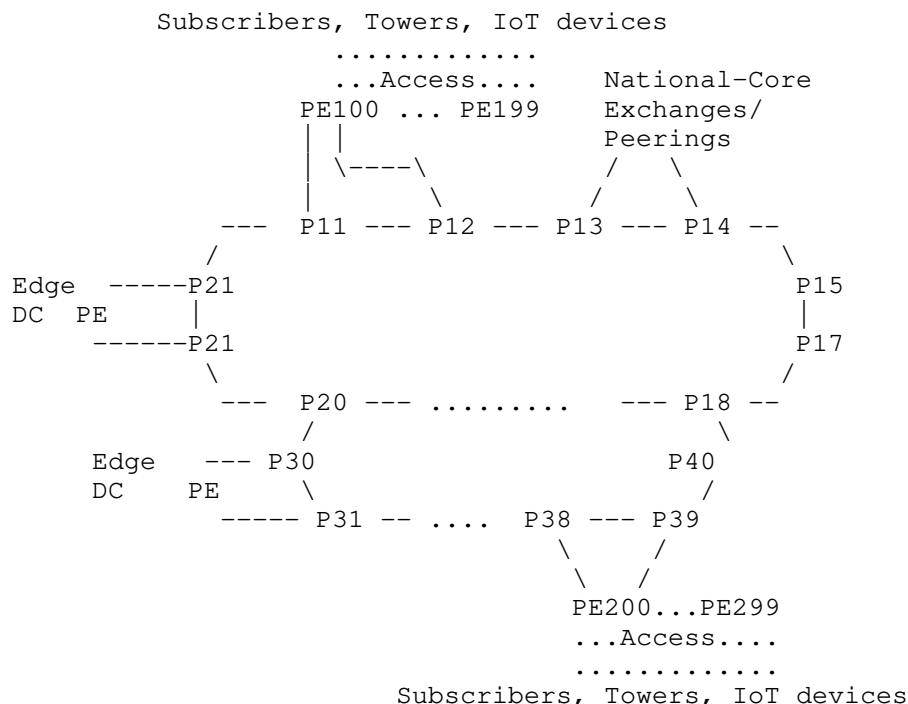


Figure 1: Reference Network Topology

An example metropolitan scale network as shown in Figure 1 may consist of one or more rings of forwarders. A ring provides the minimum cost $n+1$ redundancy between the ring nodes, especially when, as is common in metropolitan networks, new fibre cannot cost-effectively be put into new optimum trenches, but existing fibre and/or trenches have to be used. This is specifically true when the area includes not dense populated suburban areas (higher cost per subscriber and mile for rollouts).

Multiple, so-called subtended rings typically occur when existing networks are expanded into new areas: A new ring is simply connected at two most economic points into the existing infrastructure. Likewise, such a topology may become more complicated over time by addition of capacity, which resulting from TE planning calculations may not follow any of the pre-existing ring paths.

Edge Data-Center (DC), connections to Exchanges/Peerings or national cores of the provider itself, as well as all subscribers including Mobile Network Towers, and IoT devices connect to these ring directly via PE edge-forwarders and (more often) via additional CE type devices. P nodes may also double as PE nodes.

In densely populated regions, P, or PE nodes may have a high number of attached devices, shown in the picture with the example of 100 PE forwarder connecting to a single P forwarder (or rather two P for redundancy and therefore support of PREOF).

In summary, the following aspects of these networks are relevant for bounded latency:

- o Link speeds today are at least 100 Gbps and will be Tbps in the near future. Even if only a small percentage of that traffic has to support bounded latency, the queuing mechanism need to support these high-speed interfaces.
- o Fan-in/out at PE or P nodes may be (worst case) in the order of hundred(s) of incoming interfaces. Bounded latency mechanisms whose number of queues depend on the number (#I) of interfaces in a more than linear fashion, such as $(\#I^2)$ in the case of [TSN-ATS], may introduce significant challenges for cost-effective hardware.
- o Through the advent of decentralized edge Data Center and peerings between different operators and content providers, traffic flows of interest will not solely be between one central site from/to subscribers hub&spoke. Instead arbitrary, traffic engineered paths across the topology between any two edges need to be supportable in scale with the bounded latency queuing mechanism.
- o The total number of edge (#E) nodes (PE or CE) for a bounded latency service can easily be in the thousands. Aggregation of bounded latency flows on the order of $(\#E^2)$, which is the best option in per-hop, per-flow solutions such as [TSN-ATS], is likely insufficient to significantly reduce the number of flows that need to be managed across P nodes in such bounded latency queuing mechanisms.
- o The total number of P nodes may be in the hundreds and bounded latency flows in the tenths of thousands. It should also be expected that such flows are not necessarily long-term static but may need to be provisionable in the time-scale order of for example telephone calls (such as flows supporting remote control of devices or operations). Bounded latency solutions that require per-flow, per-node state maintenance on the P nodes themselves may therefore be undesirable from a network operational/complexity/reliability perspective, but also from a hardware engineering cost perspective, especially with respect to the control plane cost of dynamically setting up per-flow bounded latency for flow whenever there is a new flow or all of them

whenever there are topology or load changes that make rerouting desirable.

Beyond queuing concerns, path selection too specifically for deterministic services is a challenge in these networks:

- o Path lengths may be significantly longer than e.g. 3 hops. In large metropolitan networks, they can reach 20 or more hops. Speed of light end-to-end in these networks will be in the order of low number of msec. End-to-end queuing latency can be in the same range, if not higher.
- o To avoid undesirable re-routing under failure when PREOF and engineered disjoint paths are used, traffic steering needs to support efficiently supportable hop-by-hop traffic steering. In networks designed for source-routing (e.g.: SR routing), efficiently encoded strict-hop-by-hop steering for as much as those (e.g.: 20) hops may be desirable to support.

6. Standardized Bounded Latency algorithms

[DNBL] gives an overview of the math for the most well-known existing deterministic bounded latency algorithms/solutions. This section reviews the relevant currently standardized algorithms from the perspective of the above listed problems for high-speed, high-scale, shared services infrastructures and to provide additional background about them.

6.1. Guaranteed Service (GS)

GS is described in section 6.5 of [DNBL]. Section 2.1 describes its historical evolution and challenges. We skip further detailing of its issues here to concentrate on IEEE Time Synchronous Networking - Asynchronous Traffic Shaping [TSN-ATS], which in general is seen as superior to GS for high speed hardware implementation. All the concerns described in the TSN-ATS section apply equally or even more to GS.

6.2. TSN Asynchronous Traffic Shaping (TSN-ATS)

Section 6.4 of [DNBL] describes the bounded latency used for TSN Asynchronous Traffic Shaping [TSN-ATS]. Like GS, this bounded latency solution also relies on per-flow shaper state, except that it uses optimized shapers called "Interleaved Regulator" as explained in section 4.2.1 of [DNBL].

The concept and simplification in interleaved regulators over traditional shapers and the concept of interleaved regulators is a

resulting from mathematical work done in the last 10 years starting with [UBS].

In a system with e.g. $N=10,000$ flows each with a shaper, the forwarder needs to have 10,000 shapers each of which would need to calculate the earliest feasible send-time of the first queued packet of the flow and all these send-times would need to be compared by a scheduler picking the absolute first packet to send. Of course it is unlikely that the router would have to queue at least one packet for all queues at any point in time, but the complexity to implement the scheduler scales with N .

With interleaved regulators, there is still the per-flow state required to hold each flows traffic parameters and its next-packet earliest departure time, but instead of requiring a scheduler to compare N entries, packets are queued into one out of $(\#IIF, \#PRIO)$ FIFO queues, one queue for all the packets arriving from the same Incoming InterFace (IIF) and targeted the same worst-case queuing latency/PRIOrity (PRIO) on this hop. The shaper now only needs to calculate the earliest departure time of the head of each of these $M = \#IIF * \#PRIO$ queues and the complexity of a scheduler to select the first packet across those interleave regulators is therefore reduced by a factor of $O(N/M)$.

Unfortunately, while industrial ethernet switches today often have no more than 24 IIF, aggregation routers in metropolitan networks may have thousands of IIF, so the benefit of interleaved regulators over per-flow shaper will likely be much higher in classical TSN environments than it would be for example likely DetNet target routers in metropolitan networks.

In addition, the aforementioned core problems for shapers (Section 4), namely control plane, read/write/read cycle access and scale equally apply to interleaved regulators, so the main optimization benefits of interleaved regulators is for the original targets of [UBS] / [TSN-ATS]: low-speed (1..10Gbps switches) with limited number of interfaces - but to a much lower degree for likely important type of DetNet deployments.

6.3. Cyclic Queuing and Forwarding (CQF)

TSN Cyclic Queuing and Forwarding as described in [DNBL], section 6.6, is a per-flow, per-transit-hop stateless forwarding mechanism, which solves the concerns with per-hop, per-flow state issues described earlier in this memo. It also provides an on-time service in which the per-hop and end-to-end jitter is very small, namely in the order of a cycle time.

[CQF] operates by forwarders sending packets in periodic cycles. These cycles are derived from clock synchronization: The start of each cycle (and by implication the end of the prior cycle) are simply periodically increasing clock timestamps that have to be synchronized across adjacent forwarders, usually via PTP. This method to operate cycles allows [CQF] to operate without additional [CQF] data packet headers, but it is also the reason for the two issues of [CQF], and both relate to the so-called dead time (DT).

For the receiving node to correctly associate a [CQF] packet to the same cycle as the sending node, the last bit of the last packet in the cycle on the sending node needs to be received by the receiving node before the cycle ends.

[DNBL] explains that DT is the sum of latencies 1,2,3,4 as of [DNBL] Figure 1, but that is missing the MTIE between the forwarders: If a cycle is for example 10 usec, and the PTP MTIE is 1 usec, then only 9 usec of the cycle could be used (without even yet considering the other factors contributing to MTIE). If MTIE is not taken into account, a packet might arrive in time from the perspective of the sending forwarder, but not in the perspective of the 1 usec earlier receiving node.

In practice, MTIE should be equal or lower than 1% of the cycle time. When forwarders and links increase in speed, cycle times could become proportionally smaller to reduce per-hop cycle time latency. When this is done, MTIE needs to equally become smaller, raising the costs of the solution. Therefore, [CQF] has a challenge with higher speed networks.

The second and even more important problem is that DT includes the link latency (2 in [DNBL], Figure 1). With a speed of light in fibre of 200,000 Km, link latency is 10 usec for 2 Km. This makes [CQF] very problematic and limited in metropolitan and wide-area networks. If the longest link of a network was 10 Km, this would cause a DT on that link of 50 usec and with a cycle time of 100 usec, only 50% bandwidth could be used for cycle-time (bounded latency) traffic (excluding all other DT factors).

When links are subject to thermal expansion also known as sag on hanging wires, such as broadband copper wires (Cable Networks), their length can also change by as much as 20% between noon and night temperatures, which without changes in the design has to be taken into account as part of DT.

In conclusion, [CQF] solves many of the problems discussed in this memo, but it's reliance on timestamp synchronized cycles may pose undesirable challenges with the required accuracy of PTP in high

speed network and especially limits [CQF] ability to support wider-scale networks due to DT.

7. Candidate solution directions

As this memo outlines, per-hop, per-flow stateless forwarding is the one core requirement for to support Gbps speed metropolitan or wide-area networks.

This section gives an overview and evaluation from the perspective of the authors of this memo of currently known non-standardized proposals for per-hop-stateless forwarding with the explicit goal and/or possibility of bounded latency forwarding and in relationship to the concerns and desires described in the previous sections.

7.1. Packet tagging based CQF

To overcome the challenges outlined in Section 6.3, [I-D.qiang-DetNet-large-scale-DetNet] and [I-D.dang-queuing-with-multiple-cyclic-buffers] (tagged-CQF) propose a modified [CQF] mechanism in which timestamp based cycle indication of [CQF] is replaced by indicating the senders cycle in an appropriate packet header field, so that the receiver can accordingly map the received packet to the right local cycle.

This approach completely eliminates the link-latency as a factor impacting the effectiveness of the mechanism, because in this approach, the link latency does not impact the DT. Instead the link latency is used to calculate which cycle from the sender needs to be mapped to which cycle on the receiver, and this is programmed during setup of links into the receiving routers cycle mapping table.

Depending on the number of cycles configured, it is also possible to compensate for variability in the link-latency and higher MTIE (picture TBD). If one more cycle is used for example, this would allow for MTIE to be the order of one cycle time as opposed to a likely target of 1% of cycle time as in [CQF], reducing the required PTP clock accuracy by a factor of 100. This possible reduction in required accuracy of operations by appropriate configuration does not only cover PTP but also extends into any forwarding operation within the nodes, e.g.: it could also reduce the cost of implementation of forwarding hardware at higher speeds accordingly.

In MPLS networks, packet tagged CQF with a small number of cycle tags (such as 3 or 4) could easily be realized and standardized by relying on E-LSP where 3 or 4 EXP code points would be used to indicate the cycle value. Given how such deterministic bounded latency traffic is not subject to congestion control, it also does not require

additional ECN EXP code points, so those would be available for e.g.: best-effort traffic that should use the same E-LSP.

7.2. Packet tagging based CQF with SR

[I-D.chen-DetNet-sr-based-bounded-latency] applies the tagged-CQF mechanisms to Segment Routing (SR) by proposing SR style header elements to indicate the per-segment/hop cycle. This eliminates the need to set up on every hop a cycle mapping table.

It is unclear to the authors of this memo how big a saving this is given how the PCE would need to update all the ingress router per-flow configurations where header imposition happens when links change, whereas the mapping table approach would require only localized changes on the affected routers.

7.3. Per-hop latency indications for Segment Routing

[I-D.stein-srtsn] describes a mechanism in which a source-routed header in the spirit of a Segment Routing (SR) header can be used to enable a per-transit-hop per-flow stateless latency control. For every hop, a maximum latency is specified. The draft outlines a control plane which similarly to packet tagging based CQF or [TSN-ATS] would put the work of admitting flows, determining their paths and admitting their resources along those paths to some form of PCE/SDN-Controller.

The basic principle of forwarding in this proposal is to put received packets into a priority heap and schedule them in order of their urgency (shortest latency) for this hop.

The draft explicitly does not prescribe specific algorithms on the forwarders to take the indicated latency for the hop into account in a way that the controller can calculate the resource availability, such as specific queuing or scheduling algorithms.

It is not entirely clear to the authors of this memo, if the sole indication of such deadline latencies is sufficient to completely eliminate per-transit-hop, per-flow state and still achieve deterministic latency because of the [UBS] work. Consider that the packets latency for a hop could be used to derive a priority queue on the hop relative to other packets with higher or lower latency for this hop,

As was shown in the research work leading up to [TSN-ATS], the priority queuing on each hop alone is not sufficient to achieve a simple, solely per-hop calculated latency bound under high load because of the problem of multi-hop burst aggregation and the

resulting hard to calculate incurred upper latency bound. To overcome that calculation issue, shapers or as in [TSN-ATS] their optimization, interleaved regulators, are used in [TSN-ATS] and GS. Shapers/interleaved requires to maintain across packets from the same flow per-flow state.

Nevertheless, appropriate mathematical models for SDN controllers may be possible to develop deterministic per-hop forwarding models relying not only on the per-hop indicated latency but also on additional constraints such as limited number of hops or sufficiently low degrees of maximum admitted amount of traffic. Or else this may be used for to be developed latency models that are not 100% deterministic, but close enough in probability such that the amount of late packets would be in the same order as otherwise unavoidable problems such as BER based packet loss.

To that end, the author of [I-D.stein-srtsn] has conducted simulations of the proposed mechanism, contrasting it with other mechanisms. These results, which will be published elsewhere, show that this mechanism excels in cases with high load and a small number of flows with tight budgets. However, some small percentage of packets will miss their end-to-end latency bounds, and must be treated as lost packets.

Depending on the algorithms chosen, solutions may or may not rely on strong, weak, or no clock synchronization across nodes.

7.4. Latency Based Forwarding

"High-Precision Latency Forwarding over Packet-Programmable Networks", NOMS 2020 conference [LBF] describes a framework for per-transit-hop, per-flow stateless forwarding based on three packet parameters: The minimum and maximum desired end-to-end latency, set by the sender and not changed by the network, and the experienced latency updated by every hop. Routers supporting this LBF mechanism do also extend their routing (e.g.: IGP) to be able to calculate the non-queueing latency towards the destination. Based on the in-packet parameters and the future latency prediction are used to prioritize packets in queuing including giving them higher priority when they are late due to prior hop incurred latency, or delaying them when they are too early.

LBF was started as a more fundamental research into how application experience could be improved when they are allowed to indicate such differential min/max latency Service Level Objectives (SLO). Benefits include the ability to compensate for prior hop incurred queuing latency, but also to automatically prioritize packets on a

single hop based on their future path length, all without the need for any explicit admission control.

The LBF algorithm is completely without need for clock synchronization across nodes. Instead, it assumes mechanisms to know or learn link latency and the remaining latencies (as defined in the DetNet architecture) can be calculated locally (e.g.: latency through a forwarder).

The authors have not yet tried to define a mathematical model that would allow to derive completely deterministic behavior for this original LBF algorithm in conjunction with a PCE/SDN controller. Due to the absence of per-flow (shaper/interleaved-regulator), the authors believe that deterministic solutions would as outlined above for SRTSN (Section 7.3) likely only be possible under additional assumed constraints.

8. Conclusions

Bounded Latency for DetNet have been designed by trying to adopt solutions developed either several decades ago (GS) or recently for limited scope and scale L2 networks [TSN-ATS].

To allow DetNet solutions to explore opportunities in larger speed & scale shared network infrastructures, both private and service provider networks, it is highly desirable for DetNet WG (and/or other IETF WGs claiming responsibility in conjunction with DetNet as the driver) to explore the opportunities to standardize additional, and in the opinion of the authors better per-hop forwarding models in support of (near) deterministic bounded latency by mean of standardizing per-flow stateless/"DiffServ" style per-hop forwarding behavior (PHB) with appropriate network packet header parameters.

9. Security Considerations

This document has no security considerations (yet?).

10. IANA Considerations

This document has no IANA considerations.

11. Acknowledgements

Thanks for Yaakov Stein for reviewing and proposing text for Section 7.3.

12. Informative References

- [CQF] IEEE Time-Sensitive Networking (TSN) Task Group., "IEEE Std 802.1Qch-2017: IEEE Standard for Local and Metropolitan Area Networks -- Bridges and Bridged Networks -- Amendment 29: Cyclic Queuing and Forwarding", 2017.
- [DNBL] Finn, N., Boudec, J. L., Mohammadpour, E., Zhang, J., Varga, B., and J. Farkas, "DetNet Bounded Latency", draft-ietf-detnet-bounded-latency-06 (work in progress), May 2021.
- [I-D.chen-DetNet-sr-based-bounded-latency] Chen, M., Geng, X., and Z. Li, "Segment Routing (SR) Based Bounded Latency", draft-chen-DetNet-sr-based-bounded-latency-01 (work in progress), May 2019.
- [I-D.dang-queuing-with-multiple-cyclic-buffers] Liu, B. and J. Dang, "A Queuing Mechanism with Multiple Cyclic Buffers", draft-dang-queuing-with-multiple-cyclic-buffers-00 (work in progress), February 2021.
- [I-D.ietf-bier-te-arch] Eckert, T., Cauchie, G., and M. Menth, "Tree Engineering for Bit Index Explicit Replication (BIER-TE)", draft-ietf-bier-te-arch-10 (work in progress), July 2021.
- [I-D.qiang-DetNet-large-scale-DetNet] Qiang, L., Geng, X., Liu, B., Eckert, T., Geng, L., and G. Li, "Large-Scale Deterministic IP Network", draft-qiang-DetNet-large-scale-DetNet-05 (work in progress), September 2019.
- [I-D.stein-srtsn] Stein, Y. (., "Segment Routed Time Sensitive Networking", draft-stein-srtsn-00 (work in progress), February 2021.
- [LBF] Clemm, A. and T. Eckert, "High-Precision Latency Forwarding over Packet-Programmable Networks", IEEE 2020 IEEE/IFIP Network Operations and Management Symposium (NOMS 2020), doi 10.1109/NOMS47738.2020.9110431, April 2020.
- [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>>.

- [RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, DOI 10.17487/RFC2210, September 1997, <<https://www.rfc-editor.org/info/rfc2210>>.
- [RFC2211] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", RFC 2211, DOI 10.17487/RFC2211, September 1997, <<https://www.rfc-editor.org/info/rfc2211>>.
- [RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, DOI 10.17487/RFC2212, September 1997, <<https://www.rfc-editor.org/info/rfc2212>>.
- [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>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [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>>.
- [RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <<https://www.rfc-editor.org/info/rfc6513>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.
- [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>>.

- [RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <<https://www.rfc-editor.org/info/rfc8570>>.
- [RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <<https://www.rfc-editor.org/info/rfc8578>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <<https://www.rfc-editor.org/info/rfc8660>>.
- [RFC8964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "Deterministic Networking (DetNet) Data Plane: MPLS", RFC 8964, DOI 10.17487/RFC8964, January 2021, <<https://www.rfc-editor.org/info/rfc8964>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.
- [TSN-ATS] Specht, J., "P802.1Qcr - Bridges and Bridged Networks Amendment: Asynchronous Traffic Shaping", IEEE , July 2020, <<https://1.ieee802.org/tsn/802-1qcr/>>.
- [UBS] Specht, J. and S. Samii, "Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks", IEEE 28th Euromicro Conference on Real-Time Systems (ECRTS), 2016.

Authors' Addresses

Toerless Eckert
Futurewei Technologies USA
2220 Central Expressway
Santa Clara CA 95050
USA

Email: tte@cs.fau.de

Stewart Bryant
Stewart Bryant Ltd

Email: sb@stewartbryant.com

DetNet
Internet-Draft
Intended status: Informational
Expires: 7 January 2022

G. Mirsky
ZTE Corp.
F. Theoleyre
CNRS
G.Z. Papadopoulos
IMT Atlantique
CJ. Bernardos
UC3M
6 July 2021

Framework of Operations, Administration and Maintenance (OAM) for
Deterministic Networking (DetNet)
draft-ietf-detnet-oam-framework-03

Abstract

Deterministic Networking (DetNet), as defined in RFC 8655, is aimed to provide a bounded end-to-end latency on top of the network infrastructure, comprising both Layer 2 bridged and Layer 3 routed segments. This document's primary purpose is to detail the specific requirements of the Operation, Administration, and Maintenance (OAM) recommended to maintain a deterministic network. With the implementation of the OAM framework in DetNet, an operator will have a real-time view of the network infrastructure regarding the network's ability to respect the Service Level Objective, such as packet delay, delay variation, and packet loss ratio, assigned to each DetNet flow.

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 7 January 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
1.1. Terminology	3
1.2. Acronyms	4
1.3. Requirements Language	5
2. Role of OAM in DetNet	5
3. Operation	6
3.1. Information Collection	6
3.2. Continuity Check	7
3.3. Connectivity Verification	7
3.4. Route Tracing	7
3.5. Fault Verification/detection	8
3.6. Fault Localization and Characterization	8
3.7. Use of Hybrid OAM in DetNet	8
4. Administration	9
4.1. Collection of metrics	9
4.2. Worst-case metrics	10
5. Maintenance	10
5.1. Replication / Elimination	10
5.2. Resource Reservation	11
5.3. Soft transition after reconfiguration	11
6. Requirements	11
7. IANA Considerations	13
8. Security Considerations	13
9. Acknowledgments	13
10. References	13
10.1. Normative References	13
10.2. Informative References	13
Authors' Addresses	15

1. Introduction

Deterministic Networking (DetNet) [RFC8655] has proposed to provide a bounded end-to-end latency on top of the network infrastructure, comprising both Layer 2 bridged and Layer 3 routed segments. That work encompasses the data plane, OAM, time synchronization, management, control, and security aspects.

Operations, Administration, and Maintenance (OAM) Tools are of primary importance for IP networks [RFC7276]. DetNet OAM should provide a toolset for fault detection, localization, and performance measurement.

This document's primary purpose is to detail the specific requirements of the OAM features recommended to maintain a deterministic/reliable network. Specifically, it investigates the requirements for a deterministic network, supporting critical flows.

In this document, the term OAM will be used according to its definition specified in [RFC6291]. DetNet expects to implement an OAM framework to maintain a real-time view of the network infrastructure, and its ability to respect the Service Level Objectives (SLO), such as in-order packet delivery, packet delay, delay variation, and packet loss ratio, assigned to each DetNet flow.

This document lists the functional requirements toward OAM for DetNet domain. The list can further be used for gap analysis of available OAM tools to identify possible enhancements of existing or whether new OAM tools are required to support proactive and on-demand path monitoring and service validation.

1.1. Terminology

This document uses definitions, particularly of a DetNet flow, provided in Section 2.1 [RFC8655]. The following terms are used throughout this document as defined below:

- * DetNet OAM domain: a DetNet network used by the monitored DetNet flow. A DetNet OAM domain (also referred to in this document as "OAM domain") may have MEPs on its edge and MIPs within.
- * DetNet OAM instance: a function that monitors a DetNet flow for defects and/or measures its performance metrics. Within this document, a shorter version, OAM instance, is used interchangeably.

- * Maintenance End Point (MEP): an OAM instance that is capable of generating OAM test packets in the particular sub-layer of the DetNet OAM domain.
- * Maintenance Intermediate endPoint (MIP): an OAM instance along the DetNet flow in the particular sub-layer of the DetNet OAM domain. A MIP MAY respond to an OAM message generated by the MEP at its sub-layer of the same DetNet OAM domain.
- * Control and management plane: the control and management planes are used to configure and control the network (long-term). Relative to a DetNet flow, the control and/or management plane can be out-of-band.
- * Active measurement methods (as defined in [RFC7799]) modify a DetNet flow by inserting novel fields, injecting specially constructed test packets [RFC2544]).
- * Passive measurement methods [RFC7799] infer information by observing unmodified existing flows.
- * Hybrid measurement methods [RFC7799] is the combination of elements of both active and passive measurement methods.
- * In-band OAM is an active OAM is considered in-band in the monitored DetNet OAM domain when it traverses the same set of links and interfaces receiving the same QoS and Packet Replication, Elimination, and Ordering Functions (PREOF) treatment as the monitored DetNet flow.
- * Out-of-band OAM is an active OAM whose path through the DetNet domain is not topologically identical to the path of the monitored DetNet flow, or its test packets receive different QoS and/or PREOF treatment, or both.
- * On-path telemetry can be realized as a hybrid OAM method. The origination of the telemetry information is inherently in-band as packets in a DetNet flow are used as triggers. Collection of the on-path telemetry information can be performed using in-band or out-of-band OAM methods.

1.2. Acronyms

OAM: Operations, Administration, and Maintenance

DetNet: Deterministic Networking

SLO: Service Level Objective

1.3. Requirements Language

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

2. Role of OAM in DetNet

DetNet networks expect to provide communications with predictable low packet delay and packet loss. Most critical applications will define an SLO to be required for the DetNet flows it generates.

To respect strict guarantees, DetNet can use an orchestrator able to monitor and maintain the network. Typically, a Software-Defined Network (SDN) controller places DetNet flows in the deployed network based on their SLO. Thus, resources have to be provisioned a priori for the regular operation of the network. OAM represents the essential elements of the network operation and necessary for OAM resources that need to be accounted for to maintain the network operational.

Many legacy OAM tools can be used in DetNet networks, but they are not able to cover all the aspects of deterministic networking. Fulfilling strict guarantees is essential for DetNet flows, resulting in new DetNet specific functionalities that must be covered with OAM. Filling these gaps is inevitable and needs accurate consideration of DetNet specifics. Similar to DetNet flows itself, their OAM needs careful end-to-end engineering as well.

For example, appropriate placing of MEPs along the path of a DetNet flow is not always a trivial task and may require proper design together with the design of the service component of a given DetNet flow.

There are several DetNet specific challenges for OAM. Bounded network characteristics (e.g., delay, loss) are inseparable service parameters; therefore, PM is a key topic for DetNet. OAM tools are needed to prove the SLO without impacting the DetNet flow characteristics. A further challenge is the strict resource allocation. Resources used by OAM must be considered and allocated to avoid disturbing DetNet flow(s).

The DetNet Working Group has defined two sub-layers: (1) DetNet service sub-layer, at which a DetNet service (e.g., service protection) is provided and (2) DetNet forwarding sub-layer, which optionally provides resource allocation for DetNet flows over paths

provided by the underlying network. OAM mechanisms exist for the DetNet forwarding sub-layer, nonetheless, OAM for the service sub-layer requires new OAM procedures. These new OAM functions must allow, for example, to recognize/discover DetNet relay nodes, to get information about their configuration, and to check their operation or status.

DetNet service sub-layer functions using a sequence number. That creates a challenge for inserting OAM packets in the DetNet flow.

Fault tolerance also assumes that multiple paths could be provisioned to maintain an end-to-end circuit by adapting to the existing conditions. The central controller/orchestrator typically controls the PREOF on a node. OAM is expected to support monitoring and troubleshooting PREOF on a particular node and within the domain.

Note that distributed controllers can also control PREOF in those scenarios where DetNet solutions involve more than one single central controller.

DetNet forwarding sub-layer is based on legacy technologies and has a much better coverage regarding OAM. However, the forwarding sub-layer is terminated at DetNet relay nodes, so the end-to-end OAM state of forwarding may be created only based on the status of multiple forwarding sub-layer segments serving a given DetNet flow (e.g., in case of DetNet MPLS, there may be no end-to-end LSP below the DetNet PW).

3. Operation

OAM features will enable DetNet with robust operation both for forwarding and routing purposes.

It is worth noting that the test and data packets MUST follow the same path, i.e., the connectivity verification has to be conducted in-band without impacting the data traffic. Test packets MUST share fate with the monitored data traffic without introducing congestion in normal network conditions.

3.1. Information Collection

Information about the state of the network can be collected using several mechanisms. Some protocols, e.g., Simple Network Management Protocol, send queries. Others, e.g., YANG-based data models, generate notifications based on the publish-subscribe method. In either way, information is collected and sent to the controller.

Also, we can characterize methods of transporting OAM information relative to the path of data. For instance, OAM information may be transported in-band or out-of-band relative to the DetNet flow. In case of the former, the telemetry information uses resources allocated for the monitored DetNet flow. If an in-band method of transporting telemetry is used, the amount of generated information needs to be carefully analyzed, and additional resources must be reserved. [I-D.ietf-ippm-ioam-data] defines the in-band transport mechanism where telemetry information is collected in the data packet on which information is generated. Two tracing methods are described - end-to-end, i.e., from the ingress and egress nodes, and hop-by-hop, i.e., like end-to-end with additional information from transit nodes. [I-D.ietf-ippm-ioam-direct-export] and [I-D.mirsky-ippm-hybrid-two-step] are examples of out-of-band telemetry transport. In the former case, information is transported by each node traversed by the data packet of the monitored DetNet flow in a specially constructed packet. In the latter, information is collected in a sequence of follow-up packets that traverse the same path as the data packet of the monitored DetNet flow. In both methods, transport of the telemetry can avoid using resources allocated for the DetNet domain.

3.2. Continuity Check

Continuity check is used to monitor the continuity of a path, i.e., that there exists a way to deliver the packets between two MEP A and MEP B. The continuity check detects a network failure in one direction, from the MEP transmitting test packets to the remote egress MEP.

3.3. Connectivity Verification

In addition to the Continuity Check, DetNet solutions have to verify the connectivity. This verification considers additional constraints, i.e., the absence of misconnection. The misconnection error state is entered after several consecutive test packets from other DetNet flows are received. The definition of the conditions of entry and exit for misconnection error state is outside the scope of this document.

3.4. Route Tracing

Ping and traceroute are two ubiquitous tools that help localize and characterize a failure in the network. They help to identify a subset of the list of routers in the route. However, to be predictable, resources are reserved per flow in DetNet. Thus, DetNet needs to define route tracing tools able to track the route for a specific flow. Also, tracing can be used for the discovery of the

Path Maximum Transmission Unit or location of elements of PREOF for the particular route in the DetNet domain.

DetNet is NOT RECOMMENDED to use multiple paths or links, i.e., Equal-Cost Multipath (ECMP) [RFC8939]. As the result, OAM in ECMP environment is outside the scope of this document.

3.5. Fault Verification/detection

DetNet expects to operate fault-tolerant networks. Thus, mechanisms able to detect faults before they impact the network performance are needed.

The network has to detect when a fault occurred, i.e., the network has deviated from its expected behavior. While the network must report an alarm, the cause may not be identified precisely. For instance, the end-to-end reliability has decreased significantly, or a buffer overflow occurs.

DetNet OAM mechanisms SHOULD allow a fault detection in real time. They MAY, when possible, predict faults based on current network conditions. They MAY also identify and report the cause of the actual/predicted network failure.

3.6. Fault Localization and Characterization

An ability to localize the network defect and provide its characterization are necessary elements of network operation.

Fault localization, a process of deducing the location of a network failure from a set of observed failure indications, might be achieved, for example, by tracing the route of the DetNet flow in which the network failure was detected. Another method of fault localization can correlate reports of failures from a set of interleaving sessions monitoring path continuity.

Fault characterization is a process of identifying the root cause of the problem. For instance, misconfiguration or malfunction of PREOF elements can be the cause of erroneous packet replication or extra packets being flooded in the DetNet domain.

3.7. Use of Hybrid OAM in DetNet

Hybrid OAM methods are used in performance monitoring and defined in [RFC7799] as:

Hybrid Methods are Methods of Measurement that use a combination of Active Methods and Passive Methods.

A hybrid measurement method may produce metrics as close to passive, but it still alters something in a data packet even if that is the value of a designated field in the packet encapsulation. One example of such a hybrid measurement method is the Alternate Marking method (AMM) described in [RFC8321]. As with all on-path telemetry methods, AMM in a DetNet domain with the IP data plane is natively in-band in respect to the monitored DetNet flow. Because the marking is applied to a data flow, measured metrics are directly applicable to the DetNet flow. AMM minimizes the additional load on the DetNet domain by using nodal collection and computation of performance metrics in combination with optionally using out-of-band telemetry collection for further network analysis.

4. Administration

The network SHOULD expose a collection of metrics to support an operator making proper decisions, including:

- * Queuing Delay: the time elapsed between a packet enqueued and its transmission to the next hop.
- * Buffer occupancy: the number of packets present in the buffer, for each of the existing flows.

The following metrics SHOULD be collected:

- * per a DetNet flow to measure the end-to-end performance for a given flow. Each of the paths has to be isolated in multipath routing strategies.
- * per path to detect misbehaving path when multiple paths are applied.
- * per device to detect misbehaving device, when it relays the packets of several flows.

4.1. Collection of metrics

DetNet OAM SHOULD optimize the number of statistics / measurements to collected, frequency of collecting. Distributed and centralized mechanisms MAY be used in combination. Periodic and event-triggered collection information characterizing the state of a network MAY be used.

4.2. Worst-case metrics

DetNet aims to enable real-time communications on top of a heterogeneous multi-hop architecture. To make correct decisions, the controller needs to know the distribution of packet losses/delays for each flow, and each hop of the paths. In other words, the average end-to-end statistics are not enough. The collected information must be sufficient to allow the controller to predict the worst-case.

5. Maintenance

In the face of events that impact the network operation (e.g., link up/down, device crash/reboot, flows starting and ending), the DetNet Controller need to perform repair and re-optimization actions in order to permanently ensure the SLO of all active flows with minimal waste of resources. The controller **MUST** be able to continuously retrieve the state of the network, to evaluate conditions and trends about the relevance of a reconfiguration, quantifying:

the cost of the sub-optimality: resources may not be used optimally (e.g., a better path exists).

the reconfiguration cost: the controller needs to trigger some reconfigurations. For this transient period, resources may be twice reserved, and control packets have to be transmitted.

Thus, reconfiguration may only be triggered if the gain is significant.

5.1. Replication / Elimination

When multiple paths are reserved between two MEPs, packet replication may be used to introduce redundancy and alleviate transmission errors and collisions. For instance, in Figure 1, the source device S is transmitting the packet to both parents, devices A and B. Each MEP will decide to trigger the packet replication, elimination or the ordering process when a set of metrics passes a threshold value.

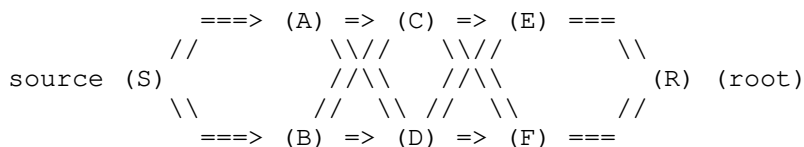


Figure 1: Packet Replication: S transmits twice the same data packet, to DP(A) and AP (B).

5.2. Resource Reservation

Because the quality of service criteria associated with a path may degrade, the network has to provision additional resources along the path. We need to provide mechanisms to patch the network configuration.

5.3. Soft transition after reconfiguration

Since DetNet expects to support real-time flows, DetNet OAM MUST support soft-reconfiguration, where the the additional resources are reserved before the those previously reserved but not in use are released. Some mechanisms have to be proposed so that packets are forwarded through the novel track only when the resources are ready to be used, while maintaining the global state consistent (no packet reordering, duplication, etc.)

6. Requirements

This section lists requirements for OAM in a DetNet domain:

1. It MUST be possible to initiate a DetNet OAM session from a MEP located at a DetNet node towards downstream MEP(s) within the given domain at a particular DetNet sub-layer. [Ed.note: FT: A MEP may be inside the detnet domain: for instance, for PREOF, an OAM session may be maintained between any pair of replicator / eliminator / egress / ingress.]
2. It MUST be possible to initialize a DetNet OAM session from a centralized controller.
3. DetNet OAM MUST support proactive and on-demand OAM monitoring and measurement methods.
4. DetNet OAM MUST support unidirectional OAM methods, continuity check, connectivity verification, and performance measurement.
5. OAM methods MAY combine in-band monitoring or measurement in the forward direction and out-of-bound notification in the reverse direction, i.e., towards the ingress MEP.
6. DetNet OAM MUST support bi-directional DetNet flows.
7. DetNet OAM MAY support bi-directional OAM methods for bidirectional DetNet flows. OAM test packets used for monitoring and measurements MUST be in-band in both directions.

8. DetNet OAM MUST support proactive monitoring of a DetNet device reachability for a given DetNet flow.
9. DetNet OAM MUST support Path Maximum Transmission Unit discovery.
10. DetNet OAM MUST support the discovery of PREOF along a route in the given DetNet domain.
11. DetNet OAM MUST support Remote Defect Indication (RDI) notification to the DetNet OAM instance performing continuity checking.
12. DetNet OAM MAY support hybrid performance measurement methods.
13. DetNet OAM MUST support unidirectional performance measurement methods. Calculated performance metrics MUST include but are not limited to throughput, packet loss, out of order, delay and delay variation metrics. [RFC6374] provides detailed information on performance measurement and performance metrics.
14. DetNet OAM MUST be able to measure metrics (e.g. delay) inside a collection of OAM sessions, specially for complex DetNet flows, with PREOF features.
15. DetNet OAM MUST support defect notification mechanism, like Alarm Indication Signal. Any DetNet device within the given DetNet flow MAY originate a defect notification addressed to any subset of DetNet devices within that flow.
16. DetNet OAM MUST support methods to enable availability of the DetNet domain. These recovery methods MAY use protection switching and restoration.
17. DetNet OAM MUST support the discovery of Packet Replication, Elimination, and Order preservation sub-functions locations in the domain.
18. DetNet OAM MUST support testing of Packet Replication, Elimination, and Order preservation sub-functions in the domain.
19. DetNet OAM MUST support monitoring levels of resources allocated for the particular DetNet flow. Such resources include but not limited to buffer utilization, scheduler transmission calendar.
20. DetNet OAM MUST support monitoring any sub-set of paths traversed through the DetNet domain by the DetNet flow.

7. IANA Considerations

This document has no actionable requirements for IANA. This section can be removed before the publication.

8. Security Considerations

This document lists the OAM requirements for a DetNet domain and does not raise any security concerns or issues in addition to ones common to networking and those specific to a DetNet discussed in [RFC9055].

9. Acknowledgments

The authors express their appreciation and gratitude to Pascal Thubert for the review, insightful questions, and helpful comments.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.

10.2. Informative References

- [I-D.ietf-ippm-ioam-data] Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-data-14, 24 June 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-data-14>>.

- [I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-direct-export-03, 17 February 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-direct-export-03>>.
- [I-D.mirsky-ippm-hybrid-two-step]
Mirsky, G., Lingqiang, W., Zhui, G., and H. Song, "Hybrid Two-Step Performance Measurement Method", Work in Progress, Internet-Draft, draft-mirsky-ippm-hybrid-two-step-10, 17 May 2021, <<https://datatracker.ietf.org/doc/html/draft-mirsky-ippm-hybrid-two-step-10>>.
- [RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <<https://www.rfc-editor.org/info/rfc2544>>.
- [RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, DOI 10.17487/RFC6291, June 2011, <<https://www.rfc-editor.org/info/rfc6291>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.
- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <<https://www.rfc-editor.org/info/rfc7276>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <<https://www.rfc-editor.org/info/rfc8321>>.

- [RFC8939] Varga, B., Ed., Farkas, J., Berger, L., Fedyk, D., and S. Bryant, "Deterministic Networking (DetNet) Data Plane: IP", RFC 8939, DOI 10.17487/RFC8939, November 2020, <<https://www.rfc-editor.org/info/rfc8939>>.
- [RFC9055] Grossman, E., Ed., Mizrahi, T., and A. Hacker, "Deterministic Networking (DetNet) Security Considerations", RFC 9055, DOI 10.17487/RFC9055, June 2021, <<https://www.rfc-editor.org/info/rfc9055>>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com, gregory.mirsky@ztetx.com

Fabrice Theoleyre
CNRS
300 boulevard Sebastien Brant - CS 10413
67400 Illkirch - Strasbourg
France

Phone: +33 368 85 45 33
Email: theoleyre@unistra.fr
URI: <http://www.theoleyre.eu>

Georgios Z. Papadopoulos
IMT Atlantique
Office B00 - 102A
2 Rue de la Châtaigneraie
35510 Cesson-Sévigné - Rennes
France

Phone: +33 299 12 70 04
Email: georgios.papadopoulos@imt-atlantique.fr

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
28911 Leganes, Madrid
Spain

Phone: +34 91624 6236
Email: cjbc@it.uc3m.es

URI: <http://www.it.uc3m.es/cjbc/>

DetNet
Internet-Draft
Intended status: Standards Track
Expires: 23 December 2021

P. Thubert, Ed.
Cisco Systems
21 June 2021

IPv6 Hop-by-Hop Options for DetNet
draft-pthubert-detnet-ipv6-hbh-04

Abstract

RFC 8938, the Deterministic Networking Data Plane Framework relies on the 6-tuple to identify an IPv6 flow. But the full DetNet operations require also the capabilities to signal meta-information such as a sequence within that flow, and to transport different types of packets along the same path with the same treatment, e.g., Operations, Administration, and Maintenance packets and/or multiple flows with fate and resource sharing. This document introduces new IPv6 Hop-by-Hop options that signal that path and redundancy information to the intermediate DetNet relays.

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 23 December 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
2. Terminology	4
3. Applicability	4
4. The DetNet Options	5
4.1. DetNet Redundancy Information Option	6
4.2. DetNet Path Options	9
4.2.1. DetNet Strict Path Option	9
4.2.2. DetNet Loose Path Option	11
4.3. RPL Packet Information	12
5. Security Considerations	12
6. IANA Considerations	12
6.1. New Subregistry for the Redundancy Type	12
6.2. New Hop-by-Hop Options	13
7. Acknowledgments	13
8. References	13
8.1. Normative References	14
8.2. Informative References	15
Author's Address	16

1. Introduction

Section 2 of the Deterministic Networking Problem Statement [DetNet-PBST] introduces the concept of Deterministic Networking (DetNet) to the IETF. DetNet extends the reach of lower layer technologies such as Time-Sensitive Networking (TSN) [IEEE 802.1 TSN] and Timeslotted Channel Hopping (TSCH) [IEEE Std. 802.15.4] over IPv6 and MPLS [RFC8938], to provide bounded latency and reliability guarantees over an end-to-end layer-3 nailed-down path.

The "Deterministic Networking Architecture" [DetNet-ARCH] details the contribution of layer-3 protocols, and defines three planes: the Application (User) Plane, the Controller Plane, and the Network Plane. [DetNet-ARCH] places an emphasis on the centralized model whereby a controller instantiates a DetNet state in the routers that is located based on matching information in the packet. For IPv6 flows, this document proposes a layer-3 signaling to index that state, using an IPv6 Extension Header (EH).

The "6TiSCH Architecture" [6TiSCH-ARCH] leverages RPL, the "Routing Protocol for Low Power and Lossy Networks" [RPL] and introduces concept of a Track as a highly redundant RPL Destination Oriented Directed Acyclic Graph (DODAG) rooted at the Track Ingress. A Track

may for instance be installed using RPL route projection [RPL-PDAO]. In that case, the TrackId is an index from a namespace associated to one IPv6 address of the Track Ingress node, and the Track that an IPv6 packet follows is signaled by the combination of the source address (of the Track Ingress node), and the TrackID placed in a RPL Option [RFC6553] located in an IPv6 Hop-by-Hop (HbH) Options Header [IPv6] in the IPv6 packet.

The "Reliable and Available Wireless (RAW) Architecture/Framework" [RAW-ARCH], extends the DetNet Network Plane to accomodate one or multiple hops of homogeneous or heterogeneous wireless technologies, e.g. a Wi-Fi6 Mesh or parallel radio access links combining Wi-Fi and 5G. The RAW Architecture reuses the concept of Track and introduces a new dataplane component, the Path Selection Engine (PSE), to dynamically select a subpath and maintain the required quality of service within a Track in the face of the rapid evolution of the medium properties.

With [IPv6], the behavior of a router upon an IPv6 packet with a HbH Options Header has evolved, making the examination of the header by routers along the path optional, as opposed to previously mandatory. Additionally, the Option Type for any option in a HbH Options Header encodes in the leftmost bits whether a router that inspects the header should drop the packet or ignore the option when encountering an unknown option. Combined, these capabilities enable a larger use of the header beyond the boundaries of a limited domain, as exemplified by the change of behavior of the RPL data plane, that was changed to allow a packet with a RPL option to escape the RPL domain in the larger Internet [RFC9008].

"IPv6 Hop-by-Hop Options Processing Procedures" [HbH-UPDT] further specifies the procedures for how IPv6 Hop-by-Hop options are processed to make their processing even more practical and increase their use in the Internet. In that context, it makes sense to consider Hop-by-Hop Options to transport the information that is relevant to DetNet.

The "Deterministic Networking Data Plane Framework" [RFC8938] relies on the 6-tuple to identify an IPv6 flow. But the full DetNet operations require also the capabilities to signal meta-information such as a sequence within that flow, and to transport different types of packets along the same path with the same treatment. For instance, it is required that Operations, Administration, and Maintenance (OAM) [RFC6291] packets and/or multiple flows share the same fate and resource sharing over the same Track or the same Traffic Engineered (TE) [RFC3272] DetNet path.

This document introduces new IPv6 Hop-by-Hop options that signal the needful DetNet path and redundancy information to the intermediate relays in an abstract form that is pure layer-3 and agnostic of the transport layer.

This pure layer-3 technique aligns DetNet with the IPv6 architecture and opens to the progress / extensions done elsewhere for IPv6; e.g., if the DetNet path leverages Segment routing (SRv6) [RFC8402] for some reason - there are plausible ones in RAW -, the Source Route Header (SRH) will be inserted after the HbH EH by the PE and both are readily accessible for the on-path routers without the need of a deeper inspection of the packet (up to and beyond the transport header).

2. Terminology

Timestamp semantics and timestamp formats used in this document are defined in "Guidelines for Defining Packet Timestamps" [RFC8877].

The Deterministic Networking terms used in this document are defined in the "Deterministic Networking Architecture" [DetNet-ARCH].

The terms Track and TrackID are defined in the "6TiSCH Architecture" [6TiSCH-ARCH].

3. Applicability

Transported in IPv6 HbH Options, the DetNet options are available early in the header chain of the packet. A DetNet-aware end system (see section 4.2 of [DetNet-ARCH]) may place the options in the header chain when constructing the packet, in which case there is no need of an encapsulation.

Alternatively, the source end system may signal the flow information some other way, or it may lack the full DetNet awareness; in that case the DetNet path endpoints are the provider Edge (PE) routers (see Figure 1 reproducing figure 5 of [DetNet-ARCH]) and the Ingress PE needs to encapsulate the packets to add the HbH options.

In Figure 1, the DetNet end systems may be f-aware and signal an IPv6 flow using the 6-tuple for the End-to-End service, but may not be s-aware, and may not sequence the packets for Packet Replication, Elimination, and Ordering Functions (PREOF), which operate at the detNet Service Layer. In that case, the Ingress PE will encapsulate the packets for this and possibly other flows to provide a common DetNet Service with OAM and PREOF, across the DetNet-l service provider network, terminating the tunnel at the Egress PE router.

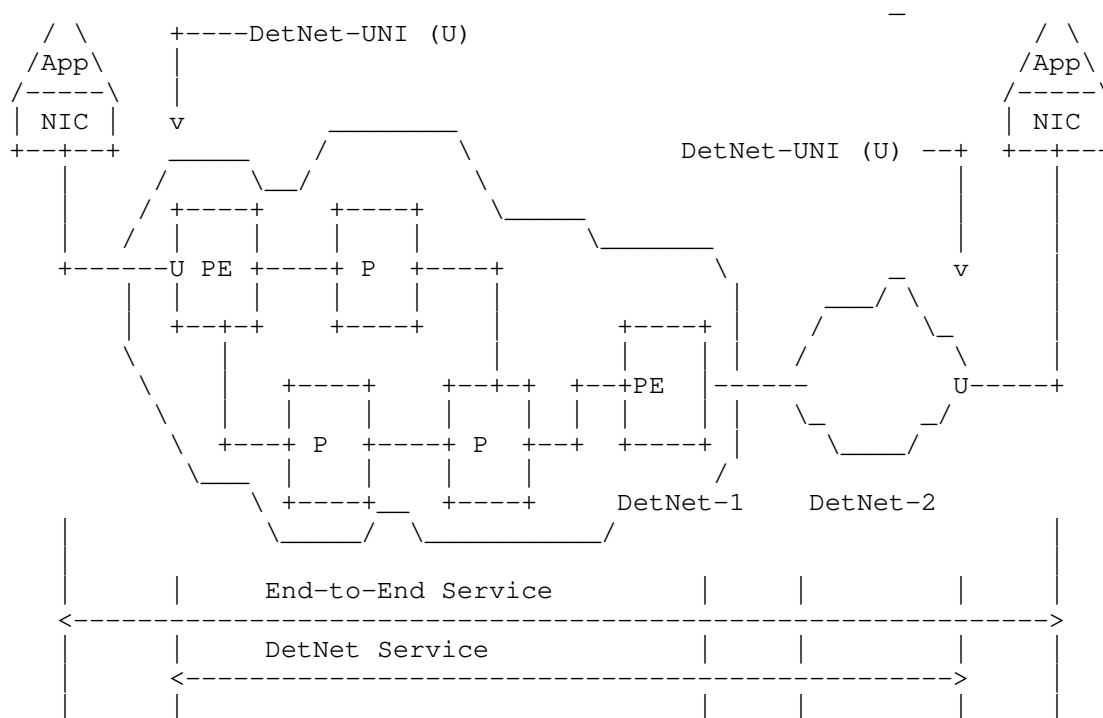


Figure 1: Figure 5 of RFC 8655, Reproduced

4. The DetNet Options

This document defines new IPv6 options for DetNet to signal path and a reliability information (e.g., sequencing) to the DetNet layers. Those options are to be placed in the IPv6 HbH Options Header, which is found right after the outer IPv6 header in the DetNet packet and immediately reachable for the forwarding engine. The format of the options follow the generic definition in section 4.2 of [IPv6]. For each type of option, the draft allows to express the information in different fashions, depending on the use case, and possibly carrying an information that plays the same role at another layer, in which case the format of the information is opaque.

The reliability information may be inherited from another layer as long as the value is guaranteed to be unique within a reasonable set of sequential packet so all packets with the same value are redundant. Timestamping can be used as an alternate sequencing technique, that avoids maintaining per-path state at the path ingress, which is feasible for nodes that maintain a very precise sense of time (e.g., from GPS or PTP) for their DetNet operations.

As long as the time granularity is in the order of a few bytes transmission, the system timestamp provides an absolute sense of ordering over a very long period across all paths for which this node is ingress, and thus within any of those. Alternatively, the draft allows to combine a rough time stamp (e.g., from a system clock synchronized by NTP) and a sequence counter that differentiates the packets that are stamped within the timer resolution.

If a DetNet Path option (see Section 4.2), including the RPL Option, is present in the same HbH Option Header as a DetNet Redundancy Information option (see Section 4.1), then the redundancy information applies to the signaled path across all flows that traverse that path; else the redundancy information applies to the flow indicated by the 6-tuple [RFC8938].

4.1. DetNet Redundancy Information Option

The DetNet Redundancy Information Option helps discriminate copies of a same packet vs. different packets, and is useful for service-sublayer Packet Replication Elimination and Ordering Functions (PREOF). The typical expression redundancy information is a sequence counter, but it is not the only way to identify a packet. It is also possible that a packet is divided in elements such as network-coded fragments. In that case, the pieces are discriminated with an opaque 8-bit fragment tag.

A packet sequence can be expressed uniquely as a wrapping counter, represented as an unsigned integer in the option. In that case, the size of the representation MUST be large enough to cover at least 3 times the upper bound on out-of-order packet delivery in terms of number of packets. The sequence counter may be copied from a field in another protocol, and it is possible that the value 0 is reserved when wrapping, to the option offers both possibilities, wrapping to either 0 or to 1.

This specification also allows to use a time stamp for the packet redundancy information, in conformance with the recommendations in [RFC8877]. This can be accomplished by utilizing the Precision Time Protocol (PTP) format defined in IEEE Std. 1588 [IEEE Std. 1588] or Network Time Protocol (NTP) [RFC5905] formats. In that case, the timestamp resolution at the origin node that builds the option MUST be fine enough to ensure that two consecutive packets are never stamped with the same value. There is no requirement for this particular stamping function that the sense of time at the origin node is synchronized with the rest of the DetNet network.

IEEE TSN [IEEE 802.1 TSN] defined a redundancy tag (R-Tag) for the IEEE Std. 802.1CB Frame Replication and Elimination for Reliability (FRER). The R-Tag is a structured field and its content is subject to evolve; but the expectation for this specification is that the overall size remains 48 bits and that the 48-bit value is different for a large number of contiguous frames. When transporting TSN frames in a DetNet packet, it is possible to leverage the R-Tag as Redundancy information, though it cannot be assumed that the R-Tag is sequentially incremented; so it can be used for packet duplicate elimination but it is not suitable not for packet re-ordering.

This specification also allows for an hybrid model with a coarse grained packet sequence within a coarse grained time stamp. In that case, both a time stamp option and a wrapping counter options are found, and the counter is used to compare packets with the same time stamp and ignored otherwise In that case, the size of the representation of the counter **MUST** be large enough to cover at least 3 times the number of packets that may be sent with the same value of time stamp.

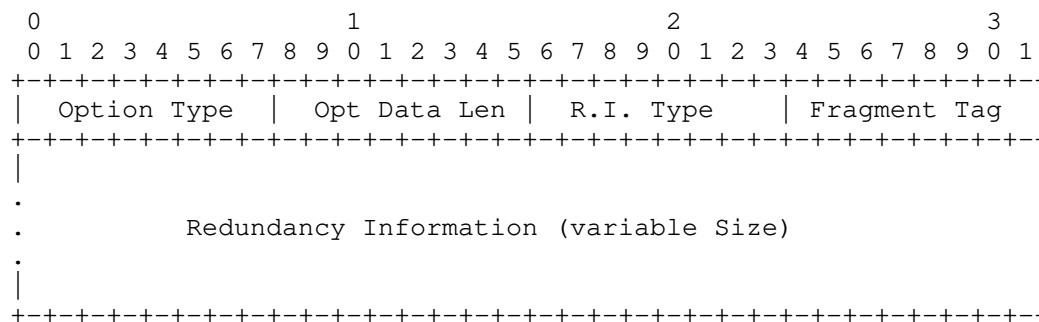


Figure 2: Redundancy Information Option Format

Redundancy Information Option fields:

Option Type: 8-bit identifier of the type of option. Value TBD by IANA; if the processing IPv6 node does not recognize the Option Type it MUST skip over this option and continue processing the header (act =00); the Option Data of that option cannot change en route to the packet's final destination (chg=0). The

Opt Data Len: 8-bit length of the option data.

Fragment Tag: 8-bit field, set to 0 when the packet is sent in

entirety; packets with the same Redundancy Information and different fragments tags MUST be considered as different by the elimination function and are not subject to ordering based on the Tag.

Redundancy Information Type: 8-bit identifier of the type of Redundancy information. Value to be confirmed by IANA.

Seq. Type Value	Category	Common Name	Redundancy Information Format
1	Wrapping Counter	Basic Sequence Counter	32-bit unsigned integer
2	Wrapping Counter	Zero-avoiding Sequence Counter	32-bit unsigned integer, wraps to 1
3	Wrapping Counter	RPL Sequence Counter	8-bit RPL sequence, see section 7. of [RPL]
11	Time Stamp	Fractional NTP	NTP 64-bit Timestamp Format, see section 4.2.1. of [RFC8877]
12	Time Stamp	Short NTP	NTP 32-bit Timestamp Format, see section 4.2.2. of [RFC8877]
13	Time Stamp	PTP	PTP 80-bit Timestamp Format, see [IEEE Std. 1588]
14	Time Stamp	Short PTP	PTP 64-bit Truncated Timestamp Format, see section 4.3. of [RFC8877]
24	Structured Unique Tag	TSN Redundancy Tag	48-bit opaque

Table 1: Redundancy Information Type values (suggested)

Redundancy Information: Variable size, as indicated in Table 1.

4.2. DetNet Path Options

The DetNet Architecture [DetNet-ARCH] assigns a DetNet flow "to specific paths through a network", but is not specific on how the path is then signaled in the packet. The DetNet Data Plane Framework [RFC8938] relies on the 6-tuple to identify an IPv6 flow and implicitly the path could be indexed by the flow identification. But this requires to maintain one path per flow and makes it difficult to assign other traffic such as OAM to the same path.

This draft provides additional means to signal the path in which the flow is placed separately from the flow identification, and independantly of the transport layer, so a path can be shared between one or more flows and OAM packets across IP address families. All the packets that are assigned to the same path are subject to the same DetNet forwarding treatment.

the DetNet expectation is that a PCE sets up a state at the DetNet forwarding sublayer to instruct each hop on how to process the DetNet flows. The DetNet Path Options when present contains information that MUST be used to select the DetNet state installed and if the DetNet state does not exist then the packet cannot be forwarded.

4.2.1. DetNet Strict Path Option

In complement to the RPL option, this specification defines a protocol-independent Strict Path Identifier, which is also taken from a namespace indicated by the IPv6 source address of the packet.

The DetNet Strict Path Option is to be used in a limited domain and all routers along the path are expected to support the option. The path indicated therein may also be used by the service sublayer, to signal the scope where the redundancy information is unique across a number of packets large enough to ensure that a forwarding node never has to handle different packets with the same redundancy information, though the same value may be found for packets with a different path information.

The typical DetNet path is typically contained under a single administrative control or within a closed group of administrative control; these include campus-wide networks and private WANs [DetNet-ARCH]. The typical expectation is that all nodes along a DetNet path are aware of the path and actively maintain a forwarding state for it. The DetNet Strict Path Option (see Section 4.2.1) is designed for that environment; if a packet escapes the local domain, a router that does not support the option will intercept it and return an error to the source.

In other environments such as RAW, it might be that the service-layer protection concentrates on just segments of the end-to-end path. In that case, the service-sublayer protection may require the signaling of both redundancy and path information, though the path information is potentially not used by some of the intermediate routers and may not be used for forwarding at all. The path information may also relate to segments that are installed along the path using a DetNet forwarding state as opposed to, say, source routing. In either case the DetNet Loose Path Option Section 4.2.2 can be used to signal the path without incurring an ICMP Error from an intermediate node.

An intermediate router that supports the DetNet Strict Path Option but is missing the necessary state to forward along the indicated path must drop the packet and return an ICMP error.code 0 pointing at the offset of the Strict Path ID in the DetNet Strict Path Option.

DetNet can also leverage the RPL Option that signals a Track in the RPL Packet Information (RPI) [RFC6553]. There are 2 versions of the RPL option, defined respectively in [RPL] with the act bits [IPv6] set to dropped the packet when the option is unknown, that defined in [RFC9008] which let the option be ignored.

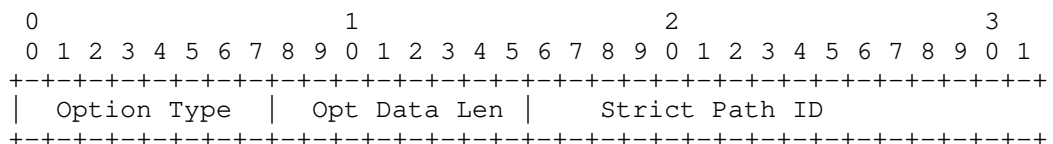


Figure 3: DetNet Strict Path Option Format

Redundancy Option fields:

Option Type: 8-bit identifier of the type of option. Value TBD by IANA; if the processing IPv6 node does not recognize the Option Type it must discard the packet and send an ICMP Parameter Problem, Code 2, message to the packet's Source Address (act =10); the Option Data of that option cannot change en route to the packet's final destination (chg=0).

Opt Data Len: 8-bit length of the option data, set to 2.

Strict Path ID: 16-bit identifier of the DetNet Path, taken from a local namespace associated with the IPv6 source address of the packet.

4.2.2. DetNet Loose Path Option

The DetNet Loose Path Option transports a Loose Path identifier which is taken from a namespace indicated by the Origin Autonomous System (AS). When the DetNet path is contained within a single AS, the Origin Autonomous System field can be left to 0 indicating local AS.

The DetNet Loose Path Option is to be used to signal a path that may be loose and may exceed the boundaries of a local domain; a portion of the hops may traverse routers in the wider internet that will not leverage the option and are expected to ignore it.

An intermediate router that supports the DetNet Loose Path Option but is missing the necessary state to forward along the indicated path must ignore the DetNet Loose Path Option.

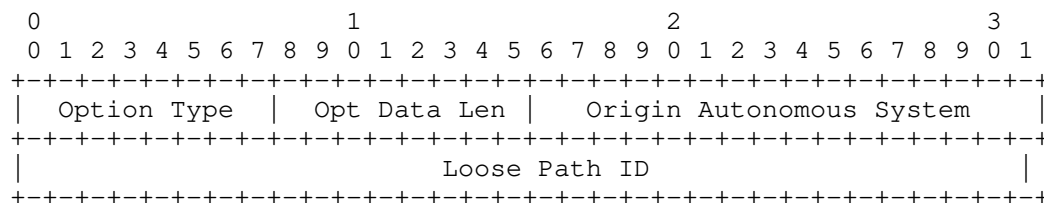


Figure 4: DetNet Loose Path Option Format

Redundancy Option fields:

Option Type: 8-bit identifier of the type of option. Value TBD by IANA; if the processing IPv6 node does not recognize the Option Type it MUST skip over this option and continue processing the header (act =00); the Option Data of that option cannot change en route to the packet's final destination (chg=0).

Opt Data Len: 8-bit length of the option data, set to 6.

Origin Autonomous System: 16-bit identifier of the Autonomous Systems (AS) that originates the path. The value of 0 signals a DetNet path that is constrained within the local AS or the local administrative DetNet domain.

Loose Path ID: 32-bit identifier of the DetNet Path, taken from a

local namespace associated with the origin AS of the DetNet path.

4.3. RPL Packet Information

6TiSCH [6TiSCH-ARCH] and RAW [RAW-ARCH] signal a Track using a RPL Option [RFC6553] with a RPLInstanceID used as TrackID. This specification reuses the RPL option as a method to signal a DetNet path. In that case, the Projected-Route 'P' flag [RPL-PDAO] MUST be set to 1, and the O, R, F flags, as well as the Sender Rank field, MUST be set to 0 by the originator, forwarded as-is, and ignored on reception.

5. Security Considerations

6. IANA Considerations

6.1. New Subregistry for the Redundancy Type

This specification creates a new Subregistry for the "Redundancy Type of the Redundancy Option" under the "Internet Protocol Version 6 (IPv6) Parameters" registry [IPV6-PARMS].

- * Possible values are 8-bit unsigned integers (0..255).
- * Registration procedure is "IETF Review" [RFC8126].
- * Initial allocation is as Suggested in Table 2:

Suggested Value	Meaning	Reference
1	Basic Sequence Counter	THIS RFC
2	Zero-avoiding Sequence Counter	THIS RFC
3	RPL Sequence Counter	THIS RFC
11	Fractional NTP time stamp	THIS RFC
12	Short NTP time stamp	THIS RFC
13	PTP time stamp	THIS RFC
14	Short PTP time stamp	THIS RFC
24	TSN Redundancy Tag	THIS RFC

Table 2: Redundancy Information Type values

6.2. New Hop-by-Hop Options

This specification updates the "Destination Options and Hop-by-Hop Options" under the "Internet Protocol Version 6 (IPv6) Parameters" registry [IPV6-PARMS] with the (suggested) values below:

Hexa	act	chg	rest	Description	Reference
0x12	00	0	10010	DetNet Redundancy Information Option	THIS RFC
0x93	10	0	10011	DetNet Strict Path Option	THIS RFC
0x14	00	0	10100	DetNet Loose Path Option	THIS RFC

Table 3: DetNet Hop-by-Hop Options

7. Acknowledgments

TBD

8. References

8.1. Normative References

- [RPL] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, DOI 10.17487/RFC6553, March 2012, <<https://www.rfc-editor.org/info/rfc6553>>.
- [IPv6] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8877] Mizrahi, T., Fabini, J., and A. Morton, "Guidelines for Defining Packet Timestamps", RFC 8877, DOI 10.17487/RFC8877, September 2020, <<https://www.rfc-editor.org/info/rfc8877>>.
- [HbH-UPDT] Hinden, R. M. and G. Fairhurst, "IPv6 Hop-by-Hop Options Processing Procedures", Work in Progress, Internet-Draft, draft-hinden-6man-hbh-processing-00, 3 December 2020, <<https://tools.ietf.org/html/draft-hinden-6man-hbh-processing-00>>.
- [DetNet-ARCH] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC9008] Robles, M.I., Richardson, M., and P. Thubert, "Using RPI Option Type, Routing Header for Source Routes, and IPv6-in-IPv6 Encapsulation in the RPL Data Plane", RFC 9008, DOI 10.17487/RFC9008, April 2021, <<https://www.rfc-editor.org/info/rfc9008>>.

[6TiSCH-ARCH]

Thubert, P., Ed., "An Architecture for IPv6 over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH)", RFC 9030, DOI 10.17487/RFC9030, May 2021, <<https://www.rfc-editor.org/info/rfc9030>>.

[RAW-ARCH] Thubert, P., Papadopoulos, G. Z., and R. Buddenberg, "Reliable and Available Wireless Architecture/Framework", Work in Progress, Internet-Draft, draft-pthubert-raw-architecture-05, 15 November 2020, <<https://tools.ietf.org/html/draft-pthubert-raw-architecture-05>>.

8.2. Informative References

[RPL-PDAO] Thubert, P., Jadhav, R. A., and M. Gillmore, "Root initiated routing state in RPL", Work in Progress, Internet-Draft, draft-ietf-roll-dao-projection-16, 15 January 2021, <<https://tools.ietf.org/html/draft-ietf-roll-dao-projection-16>>.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, DOI 10.17487/RFC6291, June 2011, <<https://www.rfc-editor.org/info/rfc6291>>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.

[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>>.

[DetNet-PBST]

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

[RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002, <<https://www.rfc-editor.org/info/rfc3272>>.

- [RFC8938] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC 8938, DOI 10.17487/RFC8938, November 2020, <<https://www.rfc-editor.org/info/rfc8938>>.
- [IEEE Std. 802.15.4]
IEEE standard for Information Technology, "IEEE Std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks".
- [IEEE 802.1 TSN]
IEEE 802.1, "Time-Sensitive Networking (TSN) Task Group", <<http://www.ieee802.org/1/pages/tsn.html>>.
- [IEEE Std. 1588]
IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Standard 1588, <<https://ieeexplore.ieee.org/document/4579760/>>.
- [IPV6-PARMS]
IANA, "Internet Protocol Version 6 (IPv6) Parameters", <<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml>>.

Author's Address

Pascal Thubert (editor)
Cisco Systems, Inc
France

Phone: +33 497 23 26 34
Email: pthubert@cisco.com

DetNet Working Group
INTERNET-DRAFT
Intended Status: Standards Track
Expires: January 8, 2022

D. Trossen
Huawei
F.-J. Goetz
J. Schmitt
Siemens
July 8, 2021

RSVP for TSN Networks
draft-trossen-detnet-rsvp-tsn-00.txt

Abstract

This document provides a solution for control plane signaling by virtue of proposing changes to RSVP signaling with deterministic services at the underlying TSN enabled layer. The solution covers distributed, centralized, and hybrid signaling scenarios in the TSN and SDN domain. The proposed changes to RSVP IntServ, called RSVP TSN in the remainder of this document, provide a better integration with Layer 2 technologies for resource reservation, for which we outline example API specifications for the realization of RSVP TSN.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
1.1. Terminology	3
2. Use Cases	4
3. Design Rationale	5
3.1. RSVP IntServ vs RSVP TSN Data Plane Model	6
3.2. RSVP IntServ vs RSVP TSN Resource Reservation Styles	6
3.3. RSVP IntServ vs RSVP TSN Object Definitions	7
3.4. RSVP IntServ vs RSVP TSN Flow Specification	7
4. RSVP TSN	8
4.1. Layer Interactions between RSVP TSN and Lower Layer Resource Allocation	9
4.2. API for Deterministic QoS (dQoS)	10
4.3. DnFlow Signaling Interface (DnFSI)	10
4.4. DnFlow Transport Interface (DnFTI)	12
4.5. RSVP TSN Message Formats	14
5. Security Considerations	14
6. IANA Considerations	14
7. Conclusion	14
8. References	14
8.1. Normative References	14
8.2. Informative References	15
Authors' Addresses	15

1. Introduction

The authors in [ID.malis-detnet-controller-plane-framework] provide an overview of the DetNet control plane architecture along three possible classes, namely (i) fully distributed control plane utilizing dynamic signaling protocols, (ii) a centralized, SDN-like, control plane, and (iii) a hybrid control plane.

The Time-Sensitive Networking (TSN) Task Group (TG) is a part of the IEEE 802.1 Working Group (WG) (<https://1.ieee802.org/tsn/>). The charter of the TSN TSG is to provide deterministic services for time sensitive applications through IEEE 802 networks, i.e., guaranteed packet transport with bounded latency, low packet delay variation, and low packet loss.

The TSN TG has developed basic data plane techniques for providing deterministic services within an IEEE 802.1Q network. Key aspects are to provide resource reservation for deterministic services by making use of a separate queue, access control, and determining the upper-, lower- and in-class interference on the egress side for bounded latency. This model for traffic from time sensitive applications, called TSN model, and the associated data plane techniques for time sensitive traffic can be applied to different lower layer network technologies and is not limited to IEEE 802.1Q bridges. DetNet uses for its DnFlows deterministic services provided by the lower layer network technologies.

When investigating the usage of RSVP [RFC2205] for the signaling of deterministic IP connectivity in combination with underlying Layer 2 mechanisms, specifically those developed for TSN, considerations arise for the development of a Layer2-specific RSVP protocol, called RSVP TSN in the following.

This document will outline use cases for RSVP TSN, followed by the design rationale and specification for the proposed RSVP TSN protocol.

Note that the document does NOT cover aspects of traffic engineering, specifically for a more detnet-focused revision of RSVP-TE. However, the work in this draft is meant to provide more insights into the possible working of RSVP for detnet (here focused over a specific L2 technology, namely TSN), which may in turn be used for a more general work on detnet-specific extensions needed for RSVP overall. As such, this document has been narrowed in scope from its previous version in [ID-trossen-detnet-control-signaling].

1.1. Terminology

This document uses the terminology established in the DetNet Architecture [RFC8655], and the reader is assumed to be familiar with that document and its terminology.

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

2. Use Cases

A deterministic network [RFC8655] is composed of DetNet-enabled end systems and DetNet relay nodes which deliver deterministic services. As shown in Figure 1, TSN-enabled end systems can still make use of deterministic networking when they are connected to an DetNet edge node supporting service proxy function to establish a deterministic end-to-end service.

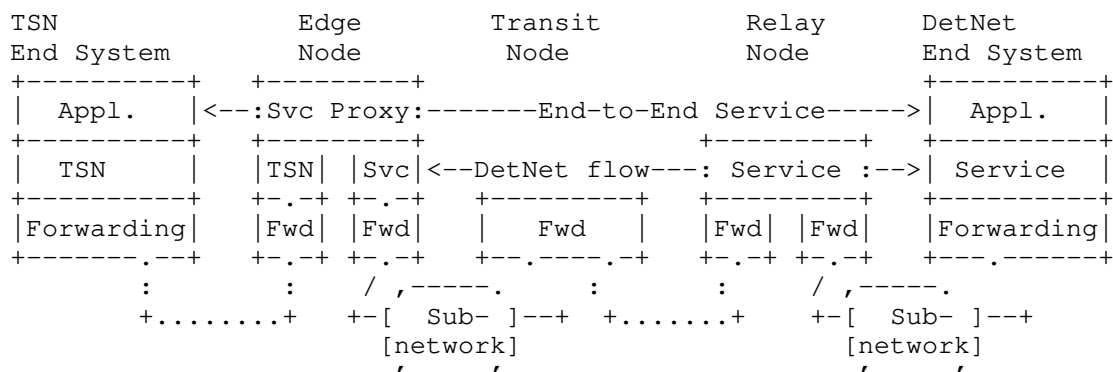


Figure 1 : Deterministic Network with TSN-enabled End Systems

In principle, three use cases are of interest for the establishment of deterministic end-to-end service over deterministic networks:

- DetNet-enabled edge nodes with service proxy on both side because the connected source and destination are TSN-enabled end systems
- Detnet enabled edge nodes with service proxy on one side because the connected source or destination are TSN-enabled end systems and on the other side the connected source or destination are Detnet-enabled end-systems
- DetNet-enabled end systems are connected to a network supporting end-to-end deterministic services

For the establishment of deterministic end-to-end connectivity based

on DnFlows, an end-2-end signaling protocol called RSVP TSN for DnFlows is proposed. To achieve deterministic QoS, access control for a DnFlow is required because each DnFlow must be known by the network supporting DetNet.

The establishment of deterministic end-to-end connectivity is in principle comparable with the establishment of TCP connectivity. The main difference is that all network elements must take active part in the establishment of a deterministic end-to-end connectivity.

RSVP TSN is an option which can be used to signal DnFlow information between

- a) DetNet-enabled edge nodes,
- b) DetNet-enabled edge nodes and DetNet-enabled end system,
- c) DetNet-enabled end systems,
- d) DetNet-enabled end system and first DetNet-enabled relay node,
- e) and DetNet-enabled relay node

Several years ago, the IETF has introduced RSVP Intserv to exchange flow information for integrated services. Because deterministic service based on TSN differs from integrated services, additional RSVP object definitions are required for RSVP TSN.

Goal of this contribution is to use RSVP TSN for signaling DnFlow information to establish deterministic end-to-end connectivity. DetNet-enabled end systems support RSVP TSN. There is no need for edge nodes with proxy services. DetNet-unaware or TSN-aware end-systems presume edge nodes supporting proxy services when they want have benefit from DetNet.

In the detnet stack model [RFC 8938], "Resource allocation" is located in the forwarding sub-layer. In this document, the term "Signaling" is used instead of the term "Resource allocation". One reason for using the term "Signaling" is because the lower layer network technologies like IEEE 802.1Q with TSN enhancements are responsible to allocate queuing, bandwidth and latency resources to provide deterministic services.

3. Design Rationale

IntServ and TSN have defined different models providing deterministic

QoS. This section will explore the design rationale behind the development of RSVP TSN. It also outlines aspects derived from the underlaying TSN capable lower layer network technology before highlighting key design considerations for the presentation of RSVP TSN in Section 4.

3.1. RSVP IntServ vs RSVP TSN Data Plane Model

The RSVP IntServ [RFC2212] model provides a flow bandwidth driven latency model with a separate transmission queue per flow. RSVP IntServ assumes a weighted fair queuing (WFQ) at the data plane, where a listener is able to influence therefore the latency through the reserved bandwidth per flow.

RSVP TSN assumes deterministic services are provided by lower layer network technologies supporting the TSN model. The TSN model itself is in contrasts with the RSVP IntServ [RFC2212] model. Lower layer network technologies providing deterministic services for traffic from time sensitive applications make use of separate queues, access control, resource reservation and determine the upper-, lower- and in-class interference on the egress side for bounded latency

3.2. RSVP IntServ vs RSVP TSN Resource Reservation Styles

RSVP IntServ has introduced the notion of 'sessions' to maintain different kinds of deterministic end-to-end connectivity and resource styles, namely fixed (i) filter style, (ii) shared explicit style, and (iii) wildcard filter style - see [RFC2205]. The receiver controls sender selection and resource styles selection. The receiver is also able to influence latency for a flow by requesting certain amount of bandwidth.

RSVP TSN splits the control over sender selection and resource styles, due to the given TSN data plane model. The resource style is controlled by the sender and the sender selection is controlled by the receiver. The Receiver cannot influence bandwidth for a DnFlow.

The resource style 'coordinated share' is introduced in RSVP TSN to support a large amount of small DnFlows with small data usage. Multiple separate resource reservations on lower layer for small DnFlows could become very inefficient.

Sender Selection	Resource Style		
	Distinct	Shared	NEW: Coordinated Shared
Explicit	supported	supported	supported

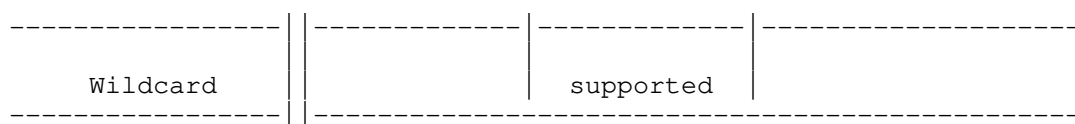


Figure 2: Resource Style and Sender Selection [RFC2205]

3.3. RSVP IntServ vs RSVP TSN Object Definitions

Due to the differences described above, not all object definitions from RSVP IntServ can be applied to RSVP TSN. Also, not all features are supported in the same way as is done by RSVP IntServ since RSVP TSN assumes a deterministic service to be provided by the lower network layer.

For instance, IEEE 802.1Q networks with TSN enhancements provides deterministic services by a layer 2 protocol for resource allocation for Sstreams [IEEE P802.1Qdd - Resource Allocation Protocol]. Such an allocated Sstream can transport one or multiple DnFlows. A StreamID is used for the identification at the layer 2 control plane.

To correlate DnFlow with their lower layer transport streams a stream identifier information must be distributed by RSVP TSN. This is only one of the reasons for introducing additional RSVP object definitions.

3.4 RSVP IntServ vs RSVP TSN Flow Specification

In RSVP IntServ, the flow specification describes both the characteristics of the traffic sent by the source and the service requirements of the application. The flow specification of RSVP IntServ is token bucket based. The sender TSpec is a description of the allowed traffic characteristics for which service is being requested. Each receiver describes by RSpec the service it desires to receive. The RSpec is carried from the receiver to the intermediate network elements and flows upstream towards the sender. It may be used or updated at the intermediate network elements before arriving the sender. The ADSpec object carries information which is generated at either data sources or intermediate network elements, is flowing downstream towards receivers.

In RSVP TSN, the sender TSpec information is also a description of the allowed traffic characteristics for which service is being requested. The receiver cannot describe the service it desires to receive. The traffic specification itself can be token bucket based but also variants based on intervals are supported. RSVP TSN does not support RSpec. It is not able to support heterogeneous receivers where each makes reservation requests with different QoS requirements on per DnFlow session.

These differences pose a number of questions:

1. Is RSVP IntServ (as defined in [RFC2212]) the right starting point to deliver DnFlow information and trigger resource allocation on lower layer network technologies supporting the TSN model?
2. How to efficiently map the different reservation styles of RSVP TSN (originally introduced by RSVP IntServ) onto the TSN data plane model?
3. What is the nature of the interface between RSVP TSN and lower layer resource reservation?
4. How does the binding between DnFlow signaling of RSVP TSN and lower layer resource reservation look like?
5. Which of the different RSVP TSN traffic specifications shall be supported?

Note: Different traffic specifications exist for an efficient mapping of traffic specification to scheduling model.

	Time based Scheduling	Token Bucket based Scheduling	Priority based (none shaping network nodes)
Stream/ Flow Based	Proposal: Dampers with Forward Traffic isolation	Asynchronous Traffic Shaping (ATS) (IEEE 802.1Qcr)	Highest (static) priority
Class Based	Cyclic Queuing & Forwarding (CQF) (IEEE 802.1Qch)	Credit-Based Shaper (CBSA) (IEEE 802.1Qav)	

Figure 3: Comparison of TSN and RSVP-IntServ Models

The proposal for dumper is discussed within the IEEE 802.1 TSN WG (see <https://www.ieee802.org/1/files/public/docs2020/new-specht-dampers-fti-0620-v02.pdf>).

For instance, the Resource-Allocation-Protocol (RAP) [RAP_IEEE] introduces templates to describe traffic class for streams with its scheduling model and the associated traffic specification for streams.

4. RSVP TSN

This section specifies the APIs for RSVP TSN, the message formats, and outlines the layer and node interactions in an RSVP TSN based system.

4.1. Layer Interactions between RSVP TSN and Lower Layer Resource Allocation

Figure 4 provides an overview of the interactions between lower layer resource allocation and DnFlow signaling in a network deployment as an elaboration of the elements in Figure 1, also illustrating the various interfaces described in the following sections.

The application utilizes a generalized API for deterministic QoS (dQoS), which controls and signals the establishment DnFlow via the upper API of RSVP TSN. The latter is called DnFlow-Signaling-Interface(uRSVPDnFSI) in this contribution.

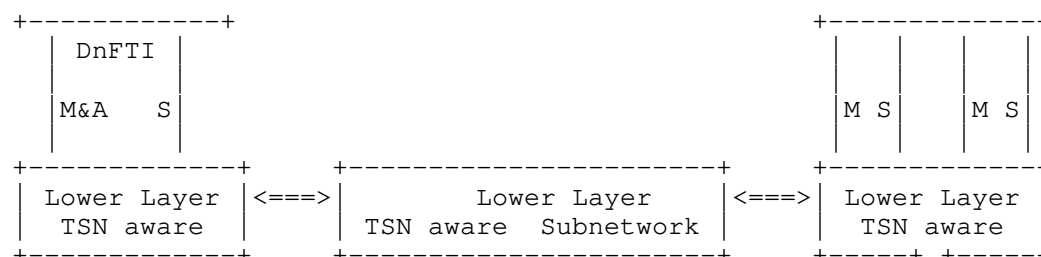
DetNet end nodes utilize RSVP TSN to distribute DnFlow information by end-to-end signaling over DetNet Route.

The lower API of RSVP TSN is called DnFlow-Transport-Interface (DnFTI) in this contribution. The DnFTI has connectivity with the lower network layer, which in turn provides deterministic services within a subnetwork based on the TSN model.

For instance, IEEE 802.1Q with extensions for TSN establish streams to transport DnFlows. For stream reservation the Resource-Allocation-Protocol (RAP) [RAP_IEEE] has defined the Reservation-Service-Interface (RSI).

The following figure illustrates the information flow within a DetNet end system and a DetNet relay node for the establishment of deterministic end-2-end services.





<----> DnFlow Signaling Service
 <====> Lower layer transport stream/flow reservation service
 <====> TSN Stream Reservation
 dQoS Deterministic QoS time sensitive application interface
 DnFSI DnFlow-Service-Interface (upper API of RSVP TSN)
 DnFTI DnFlow-Transport-Interface(lower API of RSVP TSN)
 C Control
 S Signaling
 M&A Maps and Aggregation

Figure 4: Layer Interactions between RSVP TSN and lower layer network supporting TSN

4.2. API for Deterministic QoS (dQoS)

The description of a generalized API to support deterministic QoS is not part of this document.

4.3. DnFlow Signaling Interface (DnFSI)

The definition of the DnFSI and the DnFTI is based on the DnFlow information model [ID-detnet-flow-information-model].

This interface is oriented on the interface specified by RSVP-IntServ (RFC 2205). Most of the changes are due to mapping resource reservation styles (see Section 3.2).

Sender

Call: Open Session (oriented to the RSVP-IntServ interface)

Request parameter (make use of pieces from the DnFlowSpecification)

- DestinationIpAddress, Protocol, DestinationPort

Response parameter:

- SessionID

Call: Add DnFlow

Request parameter (make use of pieces from the DnFlowSpecification)

- SessionID, SourceIpAddress, SourcePort, DSCP
- DnTrafficSpecification: Interval, MaxPacketsPerInterval, MaxPayloadSize, MinPayloadSize
- DnFlowRank
- Select one of the Resource Style: Distinct, Shared, CoordinatedShared
- Data TTL, PATH MTU size, LossRate

Notes for new parameter:

The DSCP is required to map DnFlows according their service class to offered service classes of the lower layer.

The resource style for an DnFlow is announced by the sender within the path message.

The LossRate is accumulated per DnFlow from Sender to Receiver.

Upcall: DnFlow

- Session ID
- One of the Info_type: RESV_EVENT; PATH_ERROR

Receiver

Call: Open Session

Request parameter (make use of pieces from the DnFlowSpecification)

- DestinationIpAddress, Protocol, DestinationPort

Response parameter

- SessionID

Call: Join DnFlow

Request parameter

- SessionID
- Select one of the DnFlow Source Selection: Wildcard, List of explicit sources with SourcePort
- MaximumPacketSize
- Extended Traffic Specification: MaximumExpectedLatency

Notes for new parameter:

The Source Selection is split from the RSVP-IntServ Reservation Style but still follows the rules defined by RSVP-IntServ.

The extended traffic specification MaximumExpectedLatency is propagated and merged to a minimum upstream from receiver to sender.

Upcall: DnFlow

- SessionID
- SourceIpAddress (Sender)
- SourcePort
- One of the Info_type: RESV_EVENT; PATH_ERROR

General

Call: Close Session

Request parameter

- SessionID

4.4. DnFlow Transport Interface (DnFTI)

Sender

Call: Add DnFlow

Request parameter

- SessionID, Interface, DnFlowID, DestinationIpAddress, DSCP
- DnTrafficSpecification: Interval, MaxPacketsPerInterval, MaxPayloadSize, MinPayloadSize, MinPacketsInterval
- One of the Resource Styles: Distinct, Shared, Coordinated Shared

Response parameter

- TransportFlowID (TSN StreamID)

Notes for new parameter:

The DnFlowID is a local parameter to correlate DnFlows to transport flows (e.g., TSN Stream).

The TransportFlowID correlates the DnFlow to the lower layer transport flow, e.g., TSN Stream ID.

Upcall: DnFlow

Response parameter

- SessionID
- TransportFlowID
- One of the Info_type: RESV_EVENT, RES_MODIFY_EVENT

Receiver

Call: Join DnFlow

Request parameter

- SessionID, Interface, DnFlowID, TransportFlowID
- MaximumPacketSize
- Extended Traffic Specification: MaximumExpectedLatency

Notes for new parameter:

(see notes above)

Upcall: DnFlow

Response parameter

- SessionID, TransportFlowID
- One of the Info_type: ANNOUNCE_EVENT, ANNOUNCE_MODIFY_EVENT

4.5. RSVP TSN Message Formats

TBD

5. Security Considerations

Editor's note: This section needs more details.

6. IANA Considerations

N/A

7. Conclusion

This draft outlines recommended changes to RSVP signaling in the form of RSVP TSN for a better alignment of the Layer 3 signaling with that of emerging Layer 2 solutions, together with suggested API specifications for the realization of the L3 to L2 interfaces in endpoints.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC2212] Shenker, S., Partridge, C., and Guerin, R., "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.
- [RFC2205] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jasmin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

- [RFC8655] N. Finn, B. Thubert, B. Vargas, J. Farkas, "Deterministic Networking Architecture", RFC8655, October 2019
- [RFC8938] B. Varga, Ed, J. Farkas, L. Berger, A. Malis, S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC8938, November 2020.

8.2. Informative References

- [ID.malis-detnet-controller-plane-framework] A. Malis, X. Geng, M. Chen, F. Qin, B. Varga, "Deterministic Networking (DetNet) Controller Plane Framework", draft-malis-detnet-controller-plane-framework-05 (work in progress), 2020
- [ID-detnet-flow-information-model] Balazs Varga, Janos Farkas, Rodney Cummings, Yuanlong Jiang, Don Fedyk, "DetNet Flow and Service Information Model", draft-ietf-detnet-flow-information-model-14 (work in progress), 2021
- [CHEN-IEEE] F. Chen, F.J. Goetz, M. Kiessling, J. Schmitt, " Support for uStream Aggregation in RAP (ver 0.3)" (work in progress), Jan 2019,
<<http://www.ieee802.org/1/files/public/docs2019/dd-chen-flow-aggregation-0119-v03.pdf>>
- [RAP_IEEE] IEEE, "P802.1Qdd - Resource Allocation Protocol", (work in progress), <<https://1.ieee802.org/tsn/802-1qdd/>>
- [ID-trossen-detnet-control-signaling] D. Trossen, F.-J. Goetz, J. Schmitt, "DetNet Control Plane Signaling", draft-trossen-detnet-control-signaling-01 (work in progress), 2021

Authors' Addresses

Dirk Trossen
Huawei Technologies Duesseldorf GmbH
Riesstr. 25C
80992 Munich
Germany

Email: Dirk.Trossen@Huawei.com

Franz-Josef Goetz
Siemens AG
Gleiwitzer-Str. 555
90475 Nuremberg

Germany

Email: franz-josef.goetz@siemens.com

Juergen Schmitt
Siemens AG
Gleiwitzer Str. 555
90475 Nuremberg
Germany

Email: juergen.jues.schmitt@siemens.com

DetNet
Internet-Draft
Intended status: Informational
Expires: December 11, 2021

B. Varga
J. Farkas
Ericsson
A. Malis
Malis Consulting
June 9, 2021

Deterministic Networking (DetNet): PREOF for DetNet IP
draft-varga-detnet-ip-preof-00

Abstract

This document describes how DetNet IP data plane can support the Packet Replication, Elimination, and Ordering Functions (PREOF) based on [RFC9025].

Status of This Memo

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

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

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

This Internet-Draft will expire on December 11, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

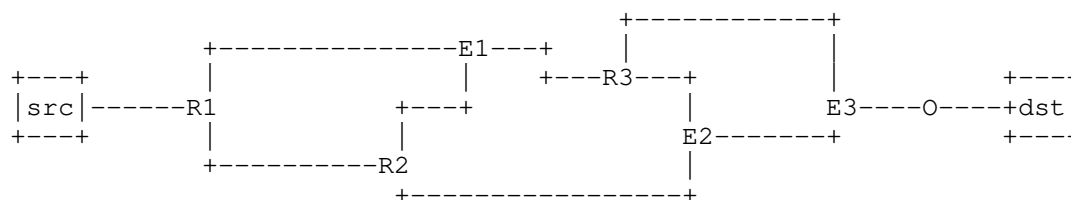
This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this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

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Terms Used in This Document	3
2.2. Abbreviations	3
2.3. Requirements Language	4
3. Requirements for adding PREOF to DetNet IP	4
4. Adding PREOF to DetNet IP	4
4.1. Solution Basics	4
4.2. Encapsulation	5
4.3. Packet Processing	6
4.4. Flow Aggregation	6
4.5. PREOF Procedures	7
4.6. PREOF capable DetNet IP domain	8
5. Control and Management Plane Parameters	8
6. Security Considerations	10
7. IANA Considerations	10
8. References	10
8.1. Normative References	10
8.2. Informative References	11
Authors' Addresses	11

1. Introduction

The DetNet Working Group has defined packet replication (PRF), packet elimination (PEF) and packet ordering (POF) functions to provide service protection by the DetNet service sub-layer [RFC8655]. The PREOF service protection method relies on copies of the same packet sent over multiple maximally disjoint paths and uses sequencing information to eliminate duplicates. A possible implementation of the PRF and PEF functions is described in [IEEE8021CB] and the related YANG data model is defined in [IEEEP8021CBcv]. A possible implementation of POF function is described in [I-D.varga-detnet-pof]. Figure 1 shows a DetNet flow on which PREOF functions are applied during forwarding from the source to the destination.



R: replication function (PRF)

E: elimination function (PEF)

O: ordering function (POF)

Figure 1: PREOF scenario in a DetNet network

In general, the use of PREOF functions require sequencing information to be included in the packets of a DetNet compound flow. This may be done by adding a sequence number or time stamp as part of DetNet encapsulation. Sequencing information is typically added once, at or close to the source.

The DetNet MPLS data plane [RFC8939] specifies how sequencing information is encoded in the MPLS header. However, the DetNet IP data plane described in [RFC8939] does not specify how sequencing information can be encoded in the IP header. This document describes a DetNet IP encapsulation that includes sequencing information based on the DetNet MPLS over UDP/IP data plane [RFC9025], i.e., leveraging the MPLS-over-UDP technology.

2. Terminology

2.1. Terms Used in This Document

This document uses the terminology established in the DetNet architecture [RFC8655], and the reader is assumed to be familiar with that document and its terminology.

2.2. Abbreviations

The following abbreviations are used in this document:

DetNet	Deterministic Networking.
PEF	Packet Elimination Function.
POF	Packet Ordering Function.
PREOF	Packet Replication, Elimination and Ordering Functions.

PRF Packet Replication Function.

2.3. Requirements Language

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

3. Requirements for adding PREOF to DetNet IP

The requirements for adding PREOF to DetNet IP are:

- o to reuse existing DetNet data plane solutions (e.g., [RFC8964], [RFC9025]).
- o to allow with minimal implementation effort the DetNet service sub-layer for IP packet switched networks.

The described solution practically gains from MPLS header fields without adding MPLS protocol stack complexity to the nodal requirements.

4. Adding PREOF to DetNet IP

4.1. Solution Basics

The DetNet IP encapsulation supporting DetNet Service sub-layer is based on the "UDP tunneling" concept. At the edge of a PREOF capable DetNet IP domain the DetNet flow is encapsulated in an UDP packet containing the sequence number used by PREOF functions within the domain. This solution maintains the 6-tuple-based DetNet flow identification in DetNet transit nodes, which operate at the DetNet forwarding sub-layer between the DetNet service sub-layer nodes; therefore, it is compatible with [RFC8939]. Figure 2 shows how the PREOF capable DetNet IP data plane fits into the DetNet sub-layers.

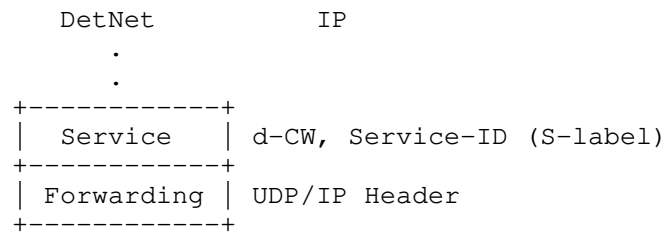


Figure 2: PREOF capable DetNet IP data plane

4.2. Encapsulation

The PREOF capable DetNet IP encapsulation builds on encapsulating DetNet PW directly over UDP. That is, it combines DetNet MPLS [RFC8964] with DetNet MPLS-in-UDP [RFC9025], without using any F-Labels as shown in Figure 3. DetNet flows are identified at the receiving DetNet service sub-layer processing node via the S-Label and/or the UDP/IP header information. Sequencing information for PREOF is provided by the DetNet Control Word (d-CW) as per [RFC8964]. The S-label is used to identify both the DetNet flow and the DetNet App-flow type. The UDP tunnel is used to direct the packet across the DetNet domain to the next DetNet service sub-layer processing node.

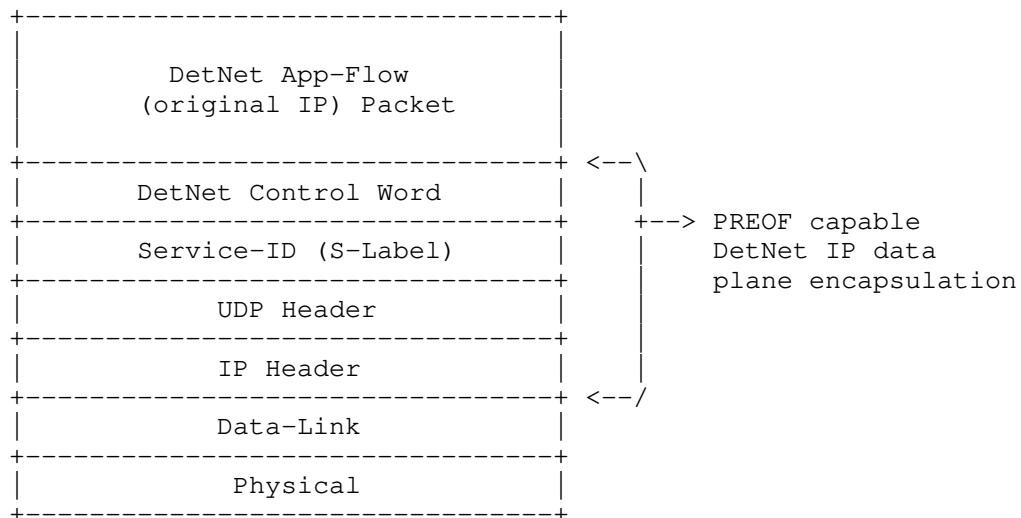


Figure 3: PREOF capable DetNet IP encapsulation

4.3. Packet Processing

IP ingress and egress nodes of the PREOF capable DetNet IP domain MUST add and remove a DetNet service-specific d-CW and Service-ID (i.e., S-Label). Relay nodes MAY change Service-ID values when processing a DetNet flow, i.e., incoming and outgoing Service-IDs of a DetNet flow can be different. Service-ID values MUST be provisioned per DetNet service via configuration, i.e., via the Controller Plane described in [RFC8938]. In some PREOF topologies, the node performing replication sends the packets to multiple nodes performing PEF or POF and the replication node may need to use different Service-ID values for the different member flows for the same DetNet service.

Note, that Service-IDs provide identification at the downstream DetNet service sub-layer receiver, not the sender.

4.4. Flow Aggregation

Two methods can be used for flow aggregation:

- o aggregation using same UDP tunnel,
- o aggregating DetNet flows as a new DetNet flow.

In the first case, the different DetNet PWs use the same UDP tunnel, so they are treated as a single (aggregated) flow on all transit nodes.

For the second option, an additional Service-ID and d-CW tuple is added to the encapsulation. The Aggregate-ID is a special case of a Service-ID, whose properties are known only at the aggregation and de-aggregation end points. It is a property of the Aggregate-ID that it is followed by a d-CW followed by an Service-ID/d-CW tuple. Figure 4 shows the encapsulation in case of aggregation.

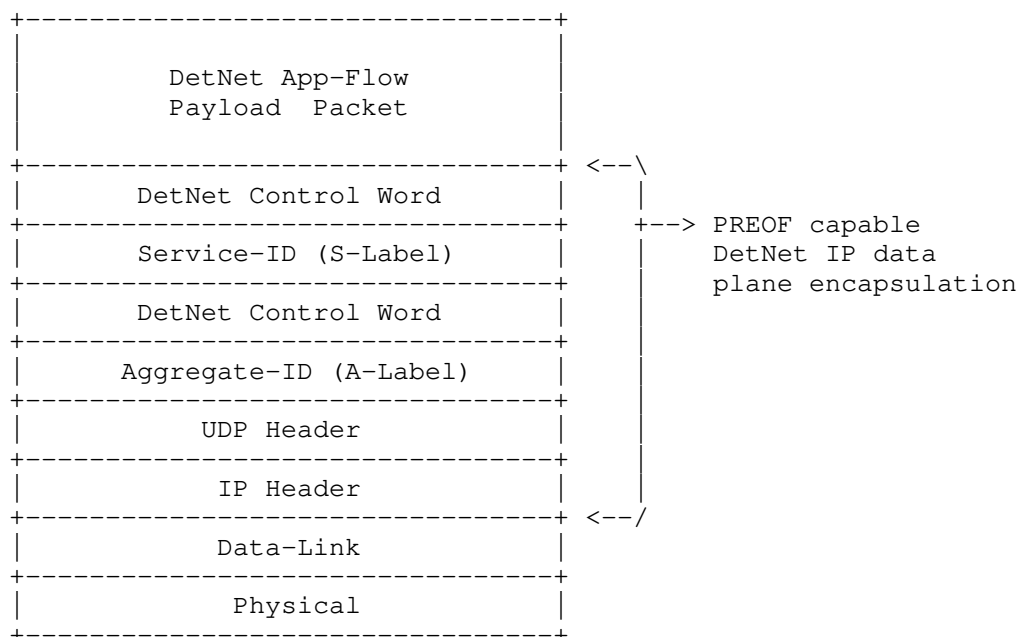


Figure 4: Aggregating DetNet flows as a new DetNet flow

4.5. PREOF Procedures

A node operating on a received DetNet flow at the DetNet service sub-layer uses the local context associated with a received Service-ID to determine which local DetNet operation(s) are applied to received packet. A Service-ID may be allocated to be unique and enabling DetNet flow identification regardless of which input interface or UDP tunnel the packet is received. It is important to note that Service-ID values are driven by the receiver, not the sender.

The DetNet forwarding sub-layer is supported by the UDP tunnel and is responsible for providing resource allocation and explicit routes.

To support outgoing PREOF capable DetNet IP encapsulation, an implementation MUST support the provisioning of UDP and IP header information. Note, when PRF is performed at the DetNet service sub-layer, there are multiple member flows, and each member flow requires the of their own Service-ID, UDP and IP header information. The headers for each outgoing packet MUST be formatted according to the configuration information, and the UDP Source Port value MUST be set to uniquely identify the DetNet flow. The packet MUST then be handled as a PREOF capable DetNet IP packet.

To support the receive processing, an implementation **MUST** also support the provisioning of received Service-ID, UDP and IP header information. The provisioned information **MUST** be used to identify incoming app-flows based on the combination of Service-ID and/or incoming encapsulation header information.

The challenge for POF initialization is that, for example, after a reset, it is not known whether the first received packet is in-order or out-of-order. The original initialization (see [I-D.varga-detnet-pof]) considers the first packet as in-order, so out-of-order packet(s) during "POFMaxTime"/"POFMaxTime_path_i" time - after the first packet was received - may not be corrected. The motivation behind such an initialization is POF implementation simplicity.

4.6. PREOF capable DetNet IP domain

Figure 5 shows using PREOF in a PREOF capable DetNet IP network.

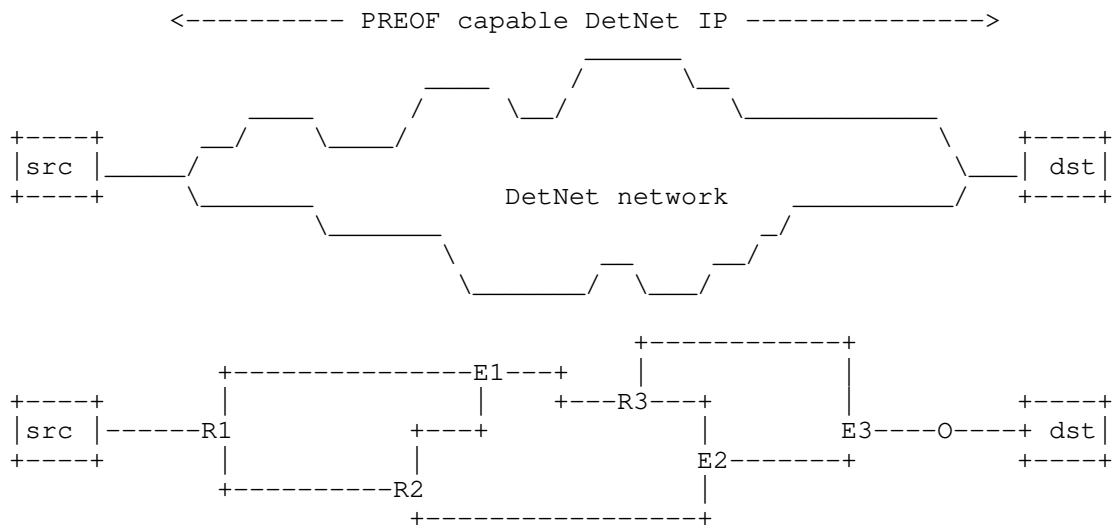


Figure 5: PREOF capable DetNet IP domain

5. Control and Management Plane Parameters

The information needed to identify individual and aggregated DetNet flows is summarized as follows:

- o Service-ID information to be mapped to UDP/IP flows. Note that, for example, a single Service-ID can map to multiple sets of UDP/IP information when PREOF is used.
- o IPv4 or IPv6 source address field.
- o IPv4 or IPv6 source address prefix length, where a zero (0) value effectively means that the address field is ignored.
- o IPv4 or IPv6 destination address field.
- o IPv4 or IPv6 destination address prefix length, where a zero (0) effectively means that the address field is ignored.
- o IPv4 protocol field set to "UDP".
- o IPv6 next header field set to "UDP".
- o For the IPv4 Type of Service and IPv6 Traffic Class Fields:
 - * Whether or not the DSCP field is used in flow identification as the use of the DSCP field for flow identification is optional.
 - * If the DSCP field is used to identify a flow, then the flow identification information (for that flow) includes a list of DSCPs used by the given DetNet flow.
- o UDP Source Port. Support for both exact and wildcard matching is required. Port ranges can optionally be used.
- o UDP Destination Port. Support for both exact and wildcard matching is required. Port ranges can optionally be used.
- o For end systems, an optional maximum IP packet size that should be used for that outgoing DetNet IP flow.

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

An implementation MUST support ordering of the set of information used to identify an individual DetNet flow. This can, for example, be used to provide a DetNet service for a specific UDP flow, with unique Source and Destination Port field values, while providing a different service for the aggregate of all other flows with that same UDP Destination Port value.

The minimum set of information for the configuration of the DetNet service sub-layer is summarized as follows:

- o App-flow identification information.
- o Sequence number length.
- o PREOF + related Service-ID(s).
- o Associated forwarding sub-layer information.
- o Service aggregation information.

The minimum set of information for the configuration of the DetNet forwarding sub-layer is summarized as follows:

- o UDP tunnel specific information.
- o Traffic parameters.

6. Security Considerations

There are no new DetNet related security considerations introduced by this solution.

7. IANA Considerations

This document makes no IANA requests.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC8938] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC 8938, DOI 10.17487/RFC8938, November 2020, <<https://www.rfc-editor.org/info/rfc8938>>.

- [RFC8939] Varga, B., Ed., Farkas, J., Berger, L., Fedyk, D., and S. Bryant, "Deterministic Networking (DetNet) Data Plane: IP", RFC 8939, DOI 10.17487/RFC8939, November 2020, <<https://www.rfc-editor.org/info/rfc8939>>.
- [RFC8964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "Deterministic Networking (DetNet) Data Plane: MPLS", RFC 8964, DOI 10.17487/RFC8964, January 2021, <<https://www.rfc-editor.org/info/rfc8964>>.
- [RFC9025] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane: MPLS over UDP/IP", RFC 9025, DOI 10.17487/RFC9025, April 2021, <<https://www.rfc-editor.org/info/rfc9025>>.

8.2. Informative References

- [I-D.varga-detnet-pof]
Varga, B., Farkas, J., Kehrler, S., and T. Heer,
"Deterministic Networking (DetNet): Packet Ordering
Function", draft-varga-detnet-pof-00 (work in progress),
April 2021.
- [IEEE8021CB]
IEEE, "IEEE Standard for Local and metropolitan area
networks -- Frame Replication and Elimination for
Reliability", DOI 10.1109/IEEESTD.2017.8091139, October
2017,
<https://standards.ieee.org/standard/802_1CB-2017.html>.
- [IEEEP8021CBcv]
Kehrler, S., "FRER YANG Data Model and Management
Information Base Module", IEEE P802.1CBcv
/D1.2 P802.1CBcv, March 2021,
<<https://www.ieee802.org/1/files/private/cv-drafts/d1/802-1CBcv-d1-2.pdf>>.

Authors' Addresses

Balazs Varga
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

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

Email: janos.farkas@ericsson.com

Andrew G. Malis
Malis Consulting

Email: agmalis@gmail.com

DetNet
Internet-Draft
Intended status: Informational
Expires: November 22, 2021

B. Varga, Ed.
J. Farkas
Ericsson
S. Kehrer
T. Heer
Hirschmann Automation and Control GmbH
May 21, 2021

Deterministic Networking (DetNet): Packet Ordering Function
draft-varga-detnet-pof-01

Abstract

Replication and Elimination functions of DetNet [RFC8655] may result in out-of-order packets, which may not be acceptable for some time-sensitive applications. The Packet Ordering Function (POF) algorithm described herein enables to restore the correct packet order when replication and elimination functions are used in DetNet networks.

Status of This Memo

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

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

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

This Internet-Draft will expire on November 22, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Terms Used in This Document	3
2.2. Abbreviations	3
2.3. Requirements Language	4
3. Requirements on POF Implementations	4
4. POF Algorithms	4
4.1. Prerequisites and Assumptions	4
4.2. POF building blocks	5
4.3. The Basic POF Algorithm	6
4.4. The Advanced POF Algorithm	7
4.5. Further enhancements of POF algorithms	8
4.6. Selecting and using the POF algorithm	9
5. Control and Management Plane Parameters for POF	9
6. Security Considerations	10
7. IANA Considerations	10
8. References	10
8.1. Normative References	10
8.2. Informative References	10
Authors' Addresses	11

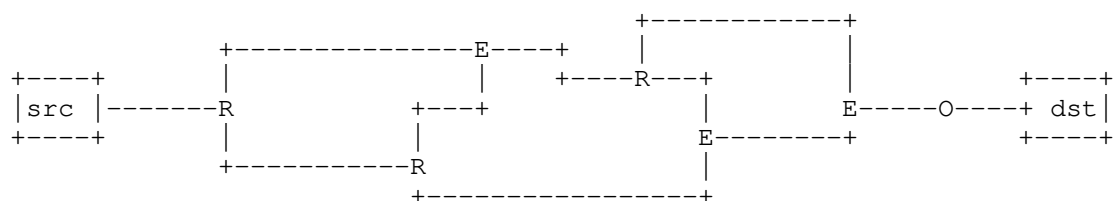
1. Introduction

The DetNet Working Group has defined packet replication (PRF) and packet elimination (PEF) functions for achieving extremely low packet loss. PRF and PEF are described in [RFC8655] and provide service protection for DetNet flows. This service protection method relies on copies of the same packet sent over multiple maximally disjoint paths and uses sequencing information to eliminate duplicates. A possible implementation of PRF and PEF functions is described in [IEEE8021CB] and the related YANG model is defined in [IEEEP8021CBcv].

In general, use of per packet replication and elimination functions may result in out-of-order delivery of packets, which may not be acceptable for some deterministic applications. Correcting packet order is not a trivial task, therefore details of a Packet Ordering Function (POF) are specified herein. The IETF DetNet WG has defined in [RFC8655] the external observable result of a POF function, i.e., that packets are reordered, but without any implementation details.

So far in packet networks, out-of-order delivery situations were handled at higher OSI layers at the end-points/hosts (e.g., in the TCP stack when packets are sent to application layer) and not within a network in nodes acting at the Layer-2 or Layer-3 OSI layers.

Figure 1 shows a DetNet flow on which PREOF functions are applied during forwarding from source to destination.



R: replication point (PRF)

E: elimination point (PEF)

O: ordering function (POF)

Figure 1: PREOF scenario in a DetNet network

Important to note, that application may react differently on out-of-order delivery. A single out-of-order packet (E.g., packet order: #1, #3, #2, #4, #5) may be interpreted by some applications as a single error, but some other applications may treat it as a 3 errors in-a-row situation. 3 errors in-a-row is a usual error threshold and may cause the application to stop (e.g., to transition to a fail safe state).

2. Terminology

2.1. Terms Used in This Document

This document uses the terminology established in the DetNet architecture [RFC8655], and the reader is assumed to be familiar with that document and its terminology.

2.2. Abbreviations

The following abbreviations are used in this document:

DetNet Deterministic Networking.

PEF Packet Elimination Function.

POF Packet Ordering Function.

PREOF Packet Replication, Elimination and Ordering Functions.

PRF Packet Replication Function.

2.3. Requirements Language

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

3. Requirements on POF Implementations

The requirements on a POF function are:

- o to solve the out-of-order delivery problem of the Replication and Elimination functions of DetNet networks.
- o to consider the delay bound requirement of a DetNet Flow.
- o to be simple and to require in network nodes only a minimum set of states/configuration parameters and resources per DetNet Flow.
- o to add only minimal or no delay to the forwarding process of packets.
- o not to require synchronization between PREOF nodes.

4. POF Algorithms

4.1. Prerequisites and Assumptions

The POF Algorithm discussed in this document makes some assumptions and tradeoffs regarding the characteristics of the sequence of received packets. In particular, the algorithm assumes that a Packet Elimination Function (PEF) is performed on the incoming packets before they are handed to the POF function. Hence, the sequence of incoming packets can be out of order or incomplete but cannot contain duplicate packets. However, the PREOF functions run independently without any state exchange required between the PEF and the POF or the PRF and the POF. Error cases in which the POF is presented duplicate packets may lead to out of order delivery of duplicate packets as well as to increased delays.

The algorithm further requires that the delay difference between two replicated packets that arrive at the PRF before the POF is bounded and known. Error cases that violate this condition (e.g., a packet that arrives later than this bound) will result in out-of order packets.

The algorithm also makes some tradeoffs. For simplicity, it is designed in a way that allows for some out of order packets directly after initialization. If this is not acceptable, Section 4.5 provides an alternative initialization scheme that prevents out-of-order packets in the initialization phase.

4.2. POF building blocks

The method described herein provides POF for DetNet networks. The configuration parameters of POF can be derived during engineering the DetNet flow through the network.

The POF method is provided via:

1. Conditional buffer: for buffering the out-of-order packets of a DetNet flow for a given time.
2. Delay calculator: buffering time considers (i) the delay difference of paths used for forwarding the replicated packets and (ii) the bounded delay requirement of the given DetNet flow.

Figure 2 shows the building blocks of a possible POF implementation.

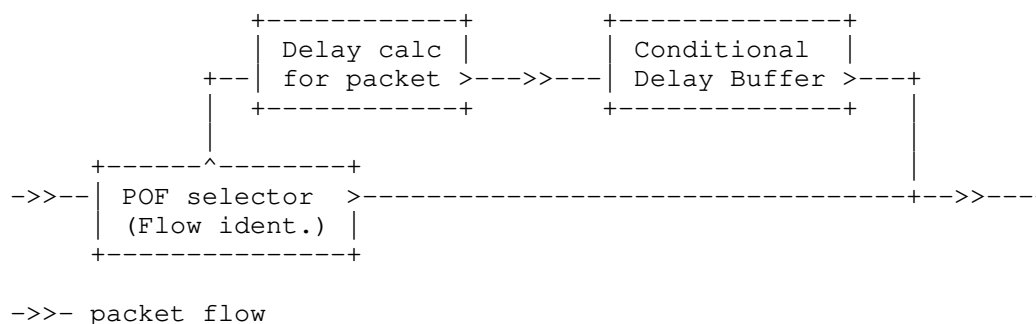


Figure 2: POF Building Blocks

4.3. The Basic POF Algorithm

The basic POF algorithm delays all out-of-order packets until all previous packet arrives or a given time (POFMaxDelay) elapses. The basic POF algorithm works as follows:

- o The sequence number of the last forwarded packet (POFLastSent) is stored for each DetNet Flow.
- o The sequence number (seq_num) of a received packet is compared to that of the last forwarded one (POFLastSent).
- o If (seq_num <= POFLastSent + 1)
 - * Then the packet is forwarded and "POFLastSent" is updated (POFLastSent = seq_num).
 - * Else the received packet is buffered.
- o A buffered packet is forwarded from the buffer when its seq_num becomes equal to "POFLastSent +1," OR a predefined time ("POFMaxDelay") elapses.
- o When a packet is forwarded from the buffer "POFLastSent" is updated with its seq_num (POFLastSent = seq_num).

Note: the difference of sequence number in consecutive packets is bounded due to the history window of the Elimination function before the POF. Therefore "<=" can be evaluated despite of the circular sequence number space.

The state used by the basic POF algorithm (i.e., "POFLastSent") needs initialization and maintenance. This works as follows:

- o The next received packet must be forwarded and the POFLastSent updated when the POF function was reset OR no packet was received for a predefined time ("POFTakeAnyTime").
- o The reset of POF erases all frames/packets from the time-based buffer used by POF.

The basic POF algorithm has two parameters to engineer:

- o "POFMaxDelay", which cannot be smaller than the delay difference of the paths used by the flow.
- o "POFTakeAnyTime", which is calculated based on several factors, for example the RECOVERY_TIMEOUT related settings of the

Elimination function(s) before the POF, the flow characteristics (e.g., inter frame/packet time), and the delay difference of the paths used by the flow.

Design of these parameters is out-of-scope in this document.

Note: multiple network failures may impact the POF function (e.g., complete outage of all redundant paths).

The basic POF algorithm increases the delay of packets with maximum "POFMaxDelay" time. Packets being in order are not delayed. This basic POF method can be applied in all network scenarios where the remaining delay budget of a flow at the POF point is larger than "POFMaxDelay" time.

Figure 3 shows the delay budget relations at the POF point.

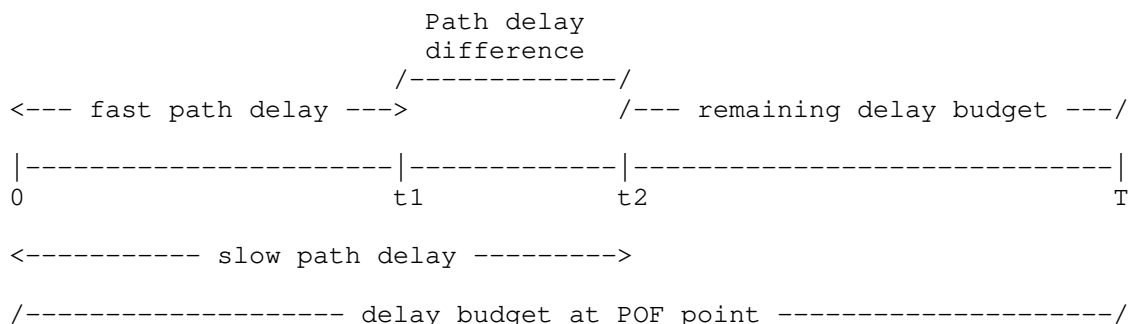


Figure 3: Delay Budget Relations at the POF Point

4.4. The Advanced POF Algorithm

In network scenario where the remaining delay budget of a flow at the POF point is smaller than "POFMaxDelay" time the basic method needs extensions.

The issue is that packets on the longest path cannot be buffered in order to keep delay budget of the flow. It must be noted that such a packet (i.e., forwarded over the longest path) needs no buffering as it is the "last chance" to deliver a packet with a given sequence number. This is because all replicas already must be arrived via shorter path(s).

The advanced POF algorithm needs two extensions of the basic POF algorithm:

- o to identify the received packet's path at the POF location and
- o to make the value of "POFMaxDelay" for buffered packets path dependent ("POFMaxDelay_i", where "i" notes the path the packet has used).

By identifying the path of a given frame, the POF algorithm can use this information to select what predefined time "POFMaxDelay_i" to apply for the buffered frame/packet. So, in the advanced POF algorithm "POFMaxDelay" is an array, that contains the predefined and path specific buffering time for each redundant path of a flow. Values in the "POFMaxDelay" array are engineered to fulfill the delay budget requirement.

The method for identification of the packet's path at the POF location depends on the network scenario. It can be implemented via various techniques, for example using ingress interface information, encoding the path in the packet itself (e.g., replication functions can set different FlowID per egress what can be used as a PathID), or in other means. Method for identification of the packet's path is out of scope in this document.

Note: in case of using the advanced POF algorithm it might be advantageous to combine PEF and POF locations in the DetNet network, as it can simplify the method used for identification of the packet's path at the POF location.

4.5. Further enhancements of POF algorithms

POF algorithms can be further enhanced by distinguishing the case of initialization from normal operation at the price of more states and more sophisticated implementation. Such enhancements could for example react better after some failure scenarios (e.g., complete outage of all paths of a DetNet flow) and may be dependent on the PEF implementation.

The challenge for POF initialization is that for example after a reset it is not known whether the first received packet is in-order or out-of-order. The original initialization (see before) considers the first packet as in-order, so out-of-order packet(s) during "POFMaxTime"/"POFMaxTime_path_i" time - after the first packet was received - may not be corrected. Motivation behind such an initialization is POF implementation simplicity.

A possible enhancement of POF initialization works as follows:

- o After a reset all received packets are buffered with their predefined timer ("POFMaxTime"/"POFMaxTime_path_i").
- o No basic/advanced POF rules are applied until the first timer expires.
- o When the first timer expires the packet with lowest seq_num in buffer is selected, forwarded, and "POFLastSent" is set with its seq_num.
- o The basic/advanced POF rules are applied for the packet(s) in the buffer and the subsequently received packets.

4.6. Selecting and using the POF algorithm

The selection of the POF algorithm depends on the network scenario and the remaining delay budget of a flow. Using POF and calculating its parameters require proper design. Knowing the path delay difference is essential for the POF algorithms described here. Failure scenarios breaking the design assumptions may impact the result of POF (e.g., packet received out of the expected worst-case delay window - calculated based on the path delay difference - may result in unwanted out-of-order delivery).

In DetNet scenarios there is always an Elimination function before the POF (therefore duplicates are not considered by the POF). Implementing them together in the same node allows POF to consider PEF events/states during the re-ordering. For example, under normal circumstances the difference of sequence number in consecutive packets is bounded due to the history window of PEF. However, in some scenarios (e.g., reset of sequence number) the difference can be much larger than the history window size.

5. Control and Management Plane Parameters for POF

POF algorithms needs setting of the following parameters:

- o Basic POF
 - * "POFMaxDelay"
 - * "POFTakeAnyTime"
- o Advanced POF
 - * "POFMaxDelay_i"
 - * "POFTakeAnyTime"

- * Network path identification related configuration(s)

Note, that in a proper design "POFTakeAnyTime" must be always larger than "POFMaxDelay".

6. Security Considerations

There are no POF related security considerations for DetNet.

7. IANA Considerations

This document makes no IANA requests.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.

8.2. Informative References

- [IEEE8021CB] IEEE, "IEEE Standard for Local and metropolitan area networks -- Frame Replication and Elimination for Reliability", DOI 10.1109/IEEESTD.2017.8091139, October 2017, <https://standards.ieee.org/standard/802_1CB-2017.html>.
- [IEEEP8021CBcv] Kehrer, S., "FRER YANG Data Model and Management Information Base Module", IEEE P802.1CBcv /D1.2 P802.1CBcv, March 2021, <<https://www.ieee802.org/1/files/private/cv-drafts/d1/802-1CBcv-d1-2.pdf>>.

Authors' Addresses

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

Email: balazs.a.varga@ericsson.com

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

Email: janos.farkas@ericsson.com

Stephan Kehrer
Hirschmann Automation and Control GmbH
Stuttgarter Strasse 45-51.
Neckartenzlingen 72654
Germany

Email: Stephan.Kehrer@belden.com

Tobias Heer
Hirschmann Automation and Control GmbH
Stuttgarter Strasse 45-51.
Neckartenzlingen 72654
Germany

Email: Tobias.Heer@belden.com

DetNet
Internet-Draft
Intended status: Informational
Expires: December 8, 2021

B. Varga
J. Farkas
Ericsson
June 6, 2021

Deterministic Networking (DetNet): OAM Functions for The Service Sub-
Layer
draft-varga-detnet-service-sub-layer-oam-00

Abstract

Operation, Administration, and Maintenance (OAM) tools are essential for a deterministic network. The DetNet architecture [RFC8655] has defined two sub-layers: (1) DetNet service sub-layer and (2) DetNet forwarding sub-layer. OAM mechanisms exist for the DetNet forwarding sub-layer, nonetheless, OAM for the service sub-layer requires a new mechanism. This draft introduces OAM related procedures for the DetNet service sub-layer functions.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 8, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Terms Used in This Document	3
2.2. Abbreviations	3
2.3. Requirements Language	3
3. Requirements on OAM for DetNet Service Sub-layer	4
4. DetNet PING	4
4.1. Overview	4
4.2. OAM processing at the DetNet service sub-layer	5
4.2.1. Relay node with PRF	5
4.2.2. Relay node with PEF	6
4.2.3. Relay node with POF	6
4.2.4. Relay node without PREOF	7
5. Security Considerations	7
6. IANA Considerations	7
7. References	8
7.1. Normative References	8
7.2. Informative References	8
Authors' Addresses	9

1. Introduction

The DetNet Working Group has defined two sub-layers: (1) DetNet service sub-layer, at which a DetNet service (e.g., service protection) is provided and (2) DetNet forwarding sub-layer, which optionally provides resource allocation for DetNet flows over paths provided by the underlying network. In [RFC8655] new DetNet-specific functions have been defined for the DetNet service sub-layer, namely PREOF (a collective name for Packet Replication, Elimination, and Ordering Functions).

Framework of Operations, Administration and Maintenance (OAM) for Deterministic Networking (DetNet) is described in [I-D.ietf-detnet-oam-framework]. OAM for the DetNet MPLS data plane is described in [I-D.ietf-detnet-mpls-oam] and OAM for the DetNet IP data plane is described in [I-D.ietf-detnet-mpls-oam].

This draft has been submitted as an individual contribution to OAM discussions, in particular, to kick-off Working Group discussions on introducing OAM functions for the DetNet service sub-layer. It is also up to the Working Group discussions to which draft parts of this contribution may go, if any.

The OAM functions for the DetNet service sub-layer allow, for example, to recognize/discover DetNet relay nodes, to get information about their configuration, and to check their operation or status.

The approach described in this draft introduces a new OAM shim layer to achieve OAM for the DetNet service sub-layer. In the rest of the draft, this approach is referred to as "DetNet PING", which is an in-band OAM approach, i.e., the OAM packets follow precisely the same path as the data packets of the corresponding DetNet flow(s). The OAM packets provide DetNet service sub-layer specific information, like:

- o Identity of a DetNet service sub-layer node.
- o Discover Ingress/Egress flow specific configuration of a DetNet service sub-layer node.
- o Detect the status of the flow specific service sub-layer function.

DetNet PING is applicable both to IP and MPLS DetNet data planes.

2. Terminology

2.1. Terms Used in This Document

This document uses the terminology established in the DetNet architecture [RFC8655], and the reader is assumed to be familiar with that document and its terminology.

2.2. Abbreviations

The following abbreviations are used in this document:

DetNet	Deterministic Networking.
PEF	Packet Elimination Function.
POF	Packet Ordering Function.
PREOF	Packet Replication, Elimination and Ordering Functions.
PRF	Packet Replication Function.

2.3. Requirements Language

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

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

3. Requirements on OAM for DetNet Service Sub-layer

The requirements on OAM for a DetNet relay node are:

1. to provide OAM functions for the DetNet service sub-layer.
2. to discover DetNet relay nodes in a DetNet network.
3. to collect DetNet service sub-layer specific (e.g., configuration/operation/status) information from DetNet relay nodes.
4. to work for both DetNet data planes: (1) MPLS and (2) IP.

4. DetNet PING

4.1. Overview

The "DetNet PING" approach uses two types of OAM packets: (1) DetNet-Echo-Request and (2) DetNet-Echo-Reply. Their encapsulation is identical to that of the corresponding DetNet data flow, so they follow precisely the same path as the packets of the corresponding DetNet data flow. They target DetNet service sub-layer entities, so they may not be recognized as OAM packet by entities not implementing DetNet service sub-layer for a packet flow (e.g., transit nodes). Other entities treat them as packets belonging to the corresponding DetNet data flow.

The following relay node roles can be distinguished:

1. DetNet PING originator node,
2. Intermediate DetNet service sub-layer node,
3. DetNet PING targeted node.

An originator node sends (generates) DetNet-Echo-Request packet(s). DetNet-Echo-Request packet contains an OAM specific "PINGSeqNum", what can be used by the DetNet service sub-layer of relay nodes. Note that "PINGSeqNum" is originator specific.

An intermediate DetNet service sub-layer node executes DetNet flow specific service sub-layer functionality. Packet processing may be done in an OAM specific manner (see details in in Section 4.2).

A targeted node answers with DetNet-Echo-Reply packet for each DetNet-Echo-Request. DetNet-Echo-Reply packet provides DetNet service sub-layer specific information on (i) identities of DetNet service sub-layer node (e.g., Node-ID, local Flow-ID), (ii) ingress/egress flow related configuration (e.g., in/out member flow specific information (including forwarding sub-layer specifics)), and (iii) status of service sub-layer function (e.g., local PxF-ID, Action-Type=x, operational status, value of key state variable(s), function related counters).

4.2. OAM processing at the DetNet service sub-layer

Detailed OAM packet processing rules of various DetNet relay nodes are described in the next sections.

4.2.1. Relay node with PRF

A DetNet relay node with PRF processes DetNet OAM packets in a stateless manner.

If the relay node with PRF is the target of a DetNet-Echo-Request packet, then the DetNet-Echo-Request packet MUST NOT be further forwarded and an DetNet-Echo-Reply packet MUST be generated. If the relay node with PRF is not the target of a DetNet-Echo-Request packet, then the DetNet-Echo-Request packet MUST be sent over all DetNet flow specific member flow paths (i.e., it is replicated).

A DetNet-Echo-Reply packet MUST contain the following information:

- o Identities related to the DetNet service sub-layer node (e.g., Node-ID, local Flow-ID),
- o Ingress/Egress flow related configuration (e.g., in/out member flow specific information (including forwarding sub-layer specifics)),
- o Status of service sub-layer function (e.g., local PRF-ID, Action-Type=Replication, operational status, value of the flow related key state variable (e.g., "GenSeqNum" in [IEEE8021CB])).

A DetNet-Echo-Reply packet MAY contain the following information:

- o PRF function related local counters.

4.2.2. Relay node with PEF

A DetNet relay node with PEF processes DetNet OAM packets in a stateful manner.

If the relay node with PEF is the target of DetNet-Echo-Request packet, then the DetNet-Echo-Request packet MUST NOT be further forwarded and an DetNet-Echo-Reply packet MUST be generated. If the relay node with PEF is not the target of DetNet-Echo-Request packet, then elimination MUST be executed on the DetNet-Echo-Request packet(s) using the OAM specific "PINGSeqNum" in the packet. So only a single DetNet-Echo-Request packet is forwarded and all further replicas (having the same originator's sequence number) MUST be discarded.

Note, that PEF MAY use a simplified elimination algorithm for DetNet-Echo-Request packets (e.g., "MatchRecoveryAlgorithm" in [IEEE8021CB]) as OAM is a slow protocol.

A DetNet-Echo-Reply packet MUST contain the following information:

- o Identities related to the DetNet service sub-layer node (e.g., Node-ID, local Flow-ID),
- o Ingress/Egress flow related configuration (e.g., in/out member flow specific information (including forwarding sub-layer specifics)) ,
- o Status of service sub-layer function (e.g., local PEF-ID, Action-Type=Elimination, operational status, value of the flow related key state variable (e.g., "RecovSeqNum" in [IEEE8021CB])).

A DetNet-Echo-Reply packet MAY contain the following information:

- o PEF function related local counters.

4.2.3. Relay node with POF

A DetNet relay node with POF processes DetNet OAM packets in a stateless manner.

If the relay node with POF is the target of DetNet-Echo-Request packet, then the DetNet-Echo-Request packet MUST NOT be further forwarded and a DetNet-Echo-Reply packet MUST be generated. If the relay node with POF is not the target of DetNet-Echo-Request packet, then the DetNet-Echo-Request packet(s) MUST be forwarded without any POF specific action.

A DetNet-Echo-Reply packet MUST contain the following information:

- o Identities of the DetNet service sub-layer node (e.g., Node-ID, local Flow-ID),
- o Ingress/Egress flow related configuration (e.g., in/out member flow specific information (including forwarding sub-layer specifics)) ,
- o Status of service sub-layer function (e.g., local POF-ID, Action-Type=Ordering, operational status, value of the flow related key state variable (e.g., "POFLastSent" in [I-D.varga-detnet-pof])).

A DetNet-Echo-Reply packet MAY contain the following information:

- o POF function related local counters.

4.2.4. Relay node without PREOF

A DetNet relay node without PREOF processes DetNet OAM packets in a stateless manner.

If the relay node without PREOF is the target of DetNet-Echo-Request packet, then the DetNet-Echo-Request packet MUST NOT be further forwarded and an DetNet-Echo-Reply packet MUST be generated. If the relay node without PREOF is not the target of DetNet-Echo-Request packet, then the DetNet-Echo-Request packet(s) MUST be forwarded (as any data packets of the related DetNet flow).

DetNet-Echo-Reply packet MUST contain the following information:

- o Identities of the DetNet service sub-layer node (e.g., Node-ID, local Flow-ID),
- o Ingress/Egress flow related configuration (e.g., in/out member flow specific information (including forwarding sub-layer specifics)) .

5. Security Considerations

Tbd.

6. IANA Considerations

Tbd.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.

7.2. Informative References

- [I-D.ietf-detnet-ip-oam]
Mirsky, G., Chen, M., and D. Black, "Operations, Administration and Maintenance (OAM) for Deterministic Networks (DetNet) with IP Data Plane", draft-ietf-detnet-ip-oam-02 (work in progress), March 2021.
- [I-D.ietf-detnet-mpls-oam]
Mirsky, G. and M. Chen, "Operations, Administration and Maintenance (OAM) for Deterministic Networks (DetNet) with MPLS Data Plane", draft-ietf-detnet-mpls-oam-03 (work in progress), March 2021.
- [I-D.ietf-detnet-oam-framework]
Mirsky, G., Theoleyre, F., Papadopoulos, G. Z., and C. J. Bernardos, "Framework of Operations, Administration and Maintenance (OAM) for Deterministic Networking (DetNet)", draft-ietf-detnet-oam-framework-01 (work in progress), May 2021.
- [I-D.varga-detnet-pof]
Varga, B., Farkas, J., Kehrler, S., and T. Heer, "Deterministic Networking (DetNet): Packet Ordering Function", draft-varga-detnet-pof-00 (work in progress), April 2021.

[IEEE8021CB]

IEEE, "IEEE Standard for Local and metropolitan area networks -- Frame Replication and Elimination for Reliability", DOI 10.1109/IEEESTD.2017.8091139, October 2017,
<https://standards.ieee.org/standard/802_1CB-2017.html>.

Authors' Addresses

Balazs Varga
Ericsson
Magyar Tudosok krt. 11.
Budapest 1117
Hungary

Email: balazs.a.varga@ericsson.com

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

Email: janos.farkas@ericsson.com

IDR
Internet-Draft
Intended status: Standards Track
Expires: November 28, 2021

Q. Xiong
H. Wu
ZTE Corporation
May 27, 2021

BGP Flow Specification for DetNet Flow Mapping
draft-xiong-idr-detnet-flow-mapping-00

Abstract

This document proposes extensions to BGP Flow Specification for the flow mapping of Deterministic Networking (DetNet) when interconnected with IEEE 802.1 Time-Sensitive Networking (TSN). The BGP flowspec is used for the filtering of the packets that match the DetNet networks and the mapping between TSN streams and DetNet flows in the control plane.

Status of This Memo

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

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

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

This Internet-Draft will expire on November 28, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Requirements Language	3
3. The Flow Mapping of DetNet	3
4. BGP Extensions for Flow Specification Encoding	4
4.1. Filtering Rules for TSN Streams	4
4.2. Traffic Action for TSN Streams	5
4.3. Filtering Rules for DetNet Flows	6
4.4. Traffic Action for DetNet Flows	7
5. Security Considerations	8
6. Acknowledgements	8
7. IANA Considerations	8
8. Normative References	8
Authors' Addresses	9

1. Introduction

[RFC8655] specifies the architecture of Deterministic Networking (DetNet), which provide a capability for the delivery of data flows with extremely low packet loss rates and bounded end-to-end delivery latency. DetNet-enabled end systems and DetNet nodes can be interconnected by sub-networks, i.e., Layer 2 technologies such as IEEE 802.1 Time-Sensitive Networking (TSN).

As defined in [RFC8655], the DetNet IP and MPLS flows can be carried over TSN sub-networks. DetNet needs to be mapped to the sub-networks technology used to interconnect DetNet nodes. For example, a TSN node may be used to interconnect DetNet-aware nodes, and these DetNet nodes can map DetNet flows to TSN streams. When the Detnet provide the deterministic service for the TSN end system, a DetNet edge node may be used to interconnect the TSN end system, and the DetNet nodes can map the TSN streams to DetNet flows.

As described in [RFC8938], one of the primary requirements of the DetNet Controller Plane is restricting flows to IEEE 802.1 TSN and the requirement could use the centralized network management provisioning mechanisms such as BGP protocol. As defined in [RFC8955], the Flow Specifications for BGP is an n-tuple consisting of several matching criteria which is comprised of traffic filtering rules and is associated with actions that can be applied to the traffic flows. The DetNet edge nodes can provide the capability to process the traffic including classifying, shaping, rate limiting,

filtering, and redirecting packets based on the policies configured by the BGP Flow Specification.

This document proposes extensions to BGP Flow Specification for the interconnection of DetNet and TSN. The BGP flowspec is used for the filtering of the packets that match the DetNet networks and the mapping between TSN streams and DetNet flows in the control plane.

2. Conventions used in this document

2.1. Terminology

The terminology is defined as [RFC8655], [RFC8938], and [RFC8955].

2.2. Requirements Language

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

3. The Flow Mapping of DetNet

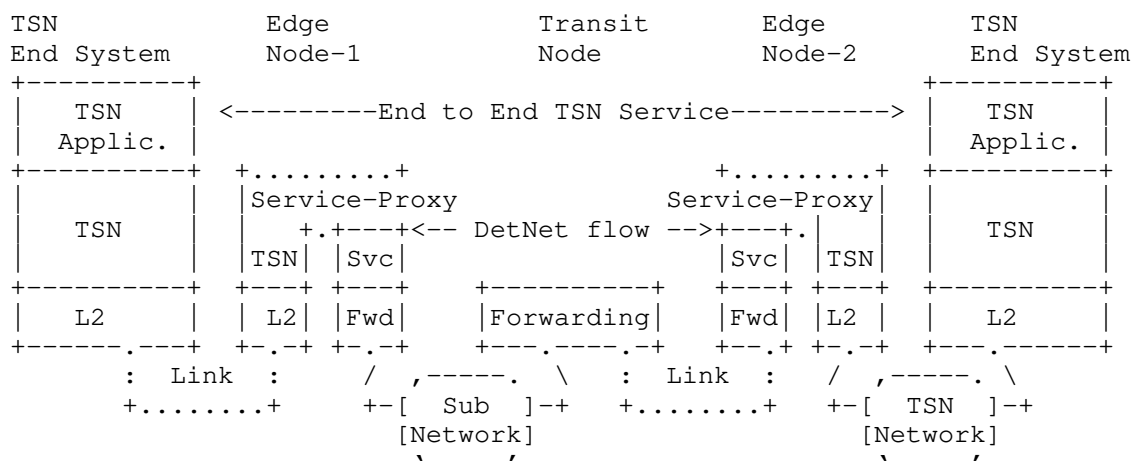
As described in [I-D.ietf-detnet-tsn-vpn-over-mpls], TSN networks can be interconnected over a DetNet MPLS Network. And as discussed in [I-D.ietf-detnet-ip-over-tsn] and [I-D.ietf-detnet-mpls-over-tsn], DetNet IP or MPLS networks can be operating over a TSN sub-network. The mapping between TSN Streams and DetNet flows is required for the service proxy function at DetNet Edge nodes. And the mapping table can be configured and maintained in the control plane. When a DetNet Edge Node receives a packet, it MUST identify and check whether such flow is present in its mapping table and decide to drop (when not match) or to forward the packet (when match) to the associated service. 1:1 and N:1 mapping (aggregating multiple TSN Streams in a single DetNet flow) MUST be supported.

As Figure 1 shows, it is required to configure the identification information when mapping received TSN Streams to the DetNet flows at Edge Node-1. Mechanisms and Parameters of TSN stream identification (e.g., Mask-and-Match Stream identification) defined in [IEEE8021CB] and [IEEE8021CBdb] can be used for service proxy function. After the identification of the TSN stream, it need to map the packet to the DetNet flow information such as S-Label, d-CW when in DetNet MPLS data plane and handle the packet as defined in [RFC8964].

When the DetNet Edge Node-2 receives a DetNet flow, it MUST identify the DetNet flow-ID information such as IP 6-tuple in DetNet IP data

plane or S-Label and d-CW information in DetNet MPLS data plane. Then the Service proxy function need to map the DetNet flow-ID and flow related parameters to the associated TSN Stream IDs and streams related parameters.

As defined in [RFC8955], the nodes that applied a Flow Specification can filter the received packets according to the matching criteria and can forward the flows based on the associated actions. This document proposes extensions to BGP Flow Specification for the mapping of DetNet flows and TSN streams by using the traffic filtering rules to identify the packet and using the associated action to map the packet to the related service.



Flow Mapping:

|TSN N:1 DetNet|<----- DetNet ----->|DetNet 1:N TSN|

Figure 1: Flow Mapping in TSN over DetNet Network

4. BGP Extensions for Flow Specification Encoding

4.1. Filtering Rules for TSN Streams

As IEEE Std 802.1Q defined, a Stream ID is a 64-bit field that uniquely identifies a stream and can be generated by the system offering the stream, or possibly a device controlling that system. But it is not carried in the header of the TSN Stream. As defined in [IEEE8021CB] and [IEEEP8021CBdb], five specific Stream identification functions are described: Null Stream identification, Source MAC and VLAN Stream identification, Active Destination MAC and VLAN Stream

identification, and IP Stream identification, and Mask-and-match Stream identification. It needs to examine the header of the streams such as destination_address, vlan_identifier, IP source address, IP destination address, DSCP, IP next protocol, source port, destination port and mac_service_data_unit.

As defined in [I-D.ietf-idr-flowspec-l2vpn], the Ethernet Layer 2 (L2) related fields have been covered by the L2 traffic filtering rules except the mac_service_data_unit in Mask-and-Match Stream identification. A mac_service_data_unit mask is defined to identify communication flows supported by various higher-layer protocols. This document proposes a new type in L2 components flowspec Type for TSN Streams.

Type TBD1 - Mac Service Data Unit

Encoding: <type (1 octet), length (1 octet), [op, value]+>

Defines a list of {operation, value} pairs used to match 6-octet Mac Service Data Unit field. Values are encoded as 6-octet quantities. op is encoded as specified in Section 4.2.1.1 of [RFC8955].

4.2. Traffic Action for TSN Streams

The action for an TSN traffic filtering flowspec is to accept the TSN streams that matches that particular rule and map the streams to the DetNet flows. The action for L3 traffic with extended communities types per [RFC8955] and [RFC8956] such as traffic-rate, traffic-marking, traffic-action, and redirect can be used for TSN to DetNet IP flow mapping.

The DetNet flow is identified by a S-Label and the DetNet Header consists of d-CW and F-Labels. The MPLS label related action for an TSN stream mapping to a DetNet MPLS network can use the Label-action defined in [I-D.ietf-idr-bgp-flowspec-label]. And the action for the sequence in d-CW field, this document specifies the following BGP extended community for TSN Streams as following shown.

type	extended community	encoding
TBD2	Sequence-action	bitmask

Table 1

The The Sequence-action extended community is shown as the Figure 2.

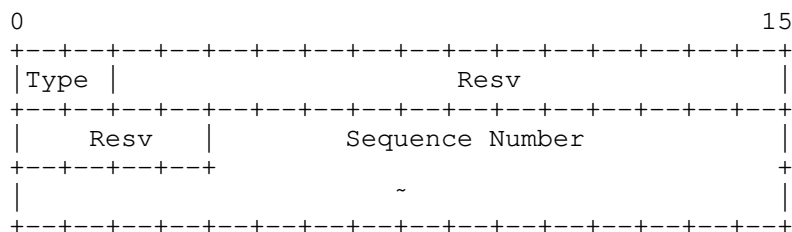


Figure 2: Sequence-action

Type: 2 bits, indicates the length of the sequence number:

0: 0 bits

1: 16 bits

2: 28 bits

Resv: 18 bits, reserved for future use. MUST be sent as zero and ignored on receipt.

Sequence Number: 28 bits, an unsigned value implementing the DetNet sequence number.

4.3. Filtering Rules for DetNet Flows

The L3 traffic filtering rules defined in [RFC8955] and [RFC8956] can be used for DetNet IP flow.

As defined in RFC8964, the MPLS-based DetNet data plane encapsulation consists of d-CW, S-Label and F-Labels. The MPLS label filtering rules have been defined in [I-D.ietf-idr-flowspec-mpls-match].

This document proposes a new community type in L3 components flowspec Type for DetNet MPLS flows.

Type TBD3 - d-CW

Encoding: <type (1 octet), length (1 octet), [op, value]+>

Defines a list of {operation, value} pairs used to match Sequence. Values are encoded as 4-octet quantities, where the four most significant bits are set to zero and ignored for matching and the 28 least significant bits contain the sequence value. op is encoded as specified in Section 4.2.1.1 of [RFC8955].

4.4. Traffic Action for DetNet Flows

The extended action for an DetNet traffic filtering flowspec is to accept the DetNet flows that matches that particular rule and map the flows to the TSN streams. This document specifies the following BGP extended communitiy as the following shown.

type	extended community	encoding
TBD4	TSN-action	bitmask

Table 2

The The TSN-action extended community is shown as the Figure 3.

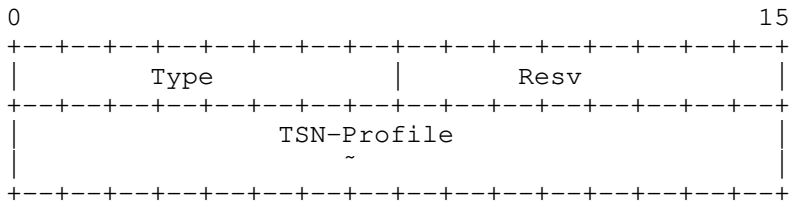


Figure 3: TSN-action

Type: 1-octet, indicates the type of TSN profiles. The value of the types is TBD:

Resv: 1-octet, reserved for future use. MUST be sent as zero and ignored on receipt.

TSN-profile: 4-octet, can be converted to the TSN Stream ID and stream related parameters and requirements as the following shown.

stream_handle: identifying the Stream to which the packet belongs in TSN networks.

sequence_number: identifying the order in which the packet was transmitted relative to other packets in the same Compound Stream in TSN networks.

traffic_scheduling: identifying the traffic scheduling mechanisms including traffic policy, queuing and forwarding methods in TSN networks.

5. Security Considerations

TBA

6. Acknowledgements

TBA

7. IANA Considerations

TBA

8. Normative References

[I-D.ietf-detnet-ip-over-tsn]

Varga, B., Farkas, J., Malis, A. G., and S. Bryant,
"DetNet Data Plane: IP over IEEE 802.1 Time Sensitive
Networking (TSN)", draft-ietf-detnet-ip-over-tsn-07 (work
in progress), February 2021.

[I-D.ietf-detnet-mpls-over-tsn]

Varga, B., Farkas, J., Malis, A. G., and S. Bryant,
"DetNet Data Plane: MPLS over IEEE 802.1 Time-Sensitive
Networking (TSN)", draft-ietf-detnet-mpls-over-tsn-07
(work in progress), February 2021.

[I-D.ietf-detnet-tsn-vpn-over-mpls]

Varga, B., Farkas, J., Malis, A. G., Bryant, S., and D.
Fedyk, "DetNet Data Plane: IEEE 802.1 Time Sensitive
Networking over MPLS", draft-ietf-detnet-tsn-vpn-over-
mpls-07 (work in progress), February 2021.

[I-D.ietf-idr-bgp-flowspec-label]

Liang, Q., Hares, S., You, J., Raszuk, R., and D. Ma,
"Carrying Label Information for BGP FlowSpec", draft-ietf-
idr-bgp-flowspec-label-01 (work in progress), December
2016.

[I-D.ietf-idr-flowspec-l2vpn]

Hao, W., Eastlake, D. E., Litkowski, S., and S. Zhuang,
"BGP Dissemination of L2 Flow Specification Rules", draft-
ietf-idr-flowspec-l2vpn-16 (work in progress), November
2020.

- [I-D.ietf-idr-flowspec-mpls-match]
Yong, L., Hares, S., Liang, Q., and J. You, "BGP Flow Specification Filter for MPLS Label", draft-ietf-idr-flowspec-mpls-match-01 (work in progress), December 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC8938] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC 8938, DOI 10.17487/RFC8938, November 2020, <<https://www.rfc-editor.org/info/rfc8938>>.
- [RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <<https://www.rfc-editor.org/info/rfc8955>>.
- [RFC8956] Loibl, C., Ed., Raszuk, R., Ed., and S. Hares, Ed., "Dissemination of Flow Specification Rules for IPv6", RFC 8956, DOI 10.17487/RFC8956, December 2020, <<https://www.rfc-editor.org/info/rfc8956>>.
- [RFC8964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "Deterministic Networking (DetNet) Data Plane: MPLS", RFC 8964, DOI 10.17487/RFC8964, January 2021, <<https://www.rfc-editor.org/info/rfc8964>>.

Authors' Addresses

Quan Xiong
ZTE Corporation
No.6 Huashi Park Rd
Wuhan, Hubei 430223
China

Email: xiong.quan@zte.com.cn

Haisheng Wu
ZTE Corporation
Nanjing, Jiangsu
China

Email: wu.haisheng@zte.com.cn