DetNet Working Group
Internet-Draft
Intended status: Standards Track
Expires: 20 April 2026

Y. Ryoo
ETRI

17 October 2025

On-time Forwarding with Push-In First-Out (PIFO) queue
draft-ryoo-detnet-ontime-forwarding-04

Abstract

This document describes operations of data plane and controller plane for Deterministic Networking (DetNet) to forward packets to meet minimum and maximum end-to-end latency requirements, while utilizing Push-In First-Out (PIFO) queue.

According to the solution described in this document, forwarding nodes do not need to maintain flow states or to be time-synchronized with each other.

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 20 April 2026.

Copyright Notice

Table of Contents

1. Introduction

Deterministic Networking (DetNet) whose architecture is defined in [RFC8655] provides the capability to carry specified unicast or multicast flows with extremely low packet loss rates and bounded end-to-end latency.

On-time forwarding is a critical feature of deterministic networks, especially of networks dealing with industrial process control signaling. The on-time forwarding is characterized as packets belonging to a flow are delivered within minimum end-to-end latency (MinLatency) and maximum end-to-end latency (MaxLatency) requirements for the flow. The difference between MaxLatency and MinLatency is the end-to-end latency variation, which becomes smaller as the requirement for on-time delivery precision becomes stricter. When MinLatency does not require to be guaranteed, it can be viewed as in-time forwarding.

This document describes operations of data plane and controller plane for DetNet to forward packets to meet minimum and maximum end-to-end latency requirements, while utilizing Push-In First-Out (PIFO) queue. Given MinLatency and MaxLatency requirements for a flow and non-queuing delays and available buffer resources on the path selected for the flow, the controller calculates lower and upper node delay bounds for each node on the path. When a packet arrives at a node, the node computes minimum departure time, nominal departure time, and maximum departure time for the packet based on the lower and upper node delay bounds calculated by the controller for the node. Using the PIFO queue, the packets are arranged in the ascending order of their nominal departure times in the PIFO queue and forwarded between their minimum and maximum departure times.

2. Terminology

2.1. Symbols Used in This Document

E2E_F	end-to-end fixed delay
E2E_VL	end-to-end variable delay lower bound
E2E_VU	end-to-end variable delay upper bound
MaxLatency	maximum end-to-end latency that must be guaranteed
MinLatency	minimum end-to-end latency that must be guaranteed
N_L	node delay lower bound
N_U	node delay upper bound
R_L	remaining end-to-end latency lower bound
R_U	remaining end-to-end latency upper bound

2.2. Abbreviations

3. Requirements Language

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

4. Temporal Model

This document separates end-to-end latency into two components: end-to-end variable delay and end-to-end fixed delay. The end-to-end variable delay is the sum of variable delays occurring in nodes and links on the path, and has its upper and lower bounds, which are denoted by E2E_VU and E2E_VL, respectively. Using the terms defined in [RFC9320], one obvious example of a variable delay is a queuing delay. Other delays, such as output delay, link delay, and

processing delay, can be classified as variable delays depending on implementation. On the other hand, the end-to-end fixed delay, denoted by E2E_F, is the sum of fixed delays occurring in links and nodes on the path. Some or all of the delays except the queuing delay can be included in the E2E_F depending on how the nodes and links are implemented. When an implementation can provide a fixed value for any non-queuing delay, that delay is considered a fixed delay in this document. An example of a fixed delay is the first-bit-out to first-bit-in delay of the link delay [RFC9320] unless the link is formed virtually. When a flow consists of packets of a constant size, the first-bit-in to last-bit-in delay of the link delay [RFC9320] also becomes a fixed delay. In this document, we assume that the first-bit-out to first-bit-in delay, which is commonly called link propagation delay, is classified as a fixed delay that depends on length of a link.

When a flow is requested, the non-queuing delays are known to a controller by considering network topology, port speeds, link lengths, maximum and minimum processing and output delays of nodes, and maximum and minimum packet sizes of the flow.

In order to guarantee MaxLatency and MinLatency, the sum of E2E_F and E2E_VU MUST be less than or equal to MaxLatency, and the sum of E2E_F and E2E_VL MUST be greater than or equal to MinLatency. Figure 1 shows the relationship among the aforementioned end-to-end parameters.

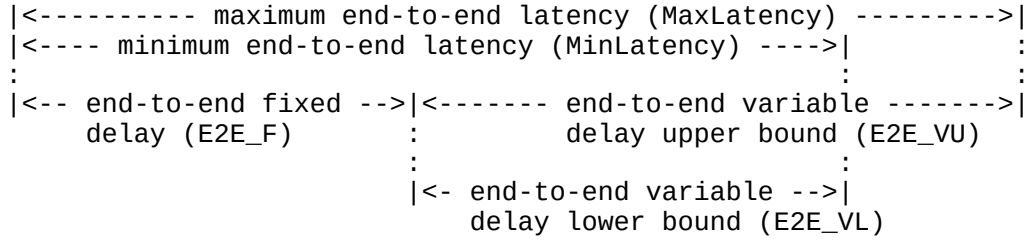

Figure 1: Relationship among end-to-end latency parameters

For the data plane operation described in the document, E2E_VU and E2E_VL are divided into all the nodes on the path, and the controller assigns a node delay upper bound (N_U) and a node delay lower bound (N_L) to each node. N_U and N_L are upper and lower bounds of the time a packet can reside in a node, and their values may be different for each node. For the sake of brevity, we omit the index for a node in this document. How the controller determines the values of N_U and N_L is described in Section 6.

Once N_U and N_L are given, a node performs the data plane operation as described in Section 5.

5. Data Plane Operation

As mentioned in the previous section, N_L and N_U are assumed to be set in each node on the path. The values of (MinLatency-E2E_F) and (MaxLatency-E2E_F) are also assumed to be known at the first node on the path. In addition to the normal DetNet encapsulation, such as DetNet control word, service label, and forwarding labels in case of MPLS, this document assumes that fields containing upper and lower bounds of remaining end-to-end latency, called R_L and R_U, are available. A node is assumed to measure a residence time, which is defined as the time a packet resides in the node. To be more precise, the residence time is the time duration between the time that the last bit of a packet comes in and the time that the last bit of the packet leaves the node.

When a packet arrives at the first node on the path, the node performs the queuing operation as described in Section 5.1 based on N_L and N_U values assigned to the first node. When the packet departs from the first node, R_L and R_U fields are set by subtracting its residence time from (MinLatency-E2E_F) and (MaxLatency-E2E_F), respectively.

Each node except the first and last nodes on the path performs the queuing operation as described in Section 5.1 based on N_L and N_U values assigned to the node, and updates R_L and R_U fields by subtracting its residence time from received R_L and R_U values. If the resulting value of R_L is negative, then the R_L field is updated with zero.

The last node also performs the queuing operation as described in Section 5.1, but uses R_L and R_U received from its previous node instead of N_L and N_U values assigned to the last node. If R_U is greater than N_U of the last node, N_U is used.

5.1. Queuing Operation

When a packet is processed within a node, it can experience variable delays, which are generally categorized into three types of delay: processing delay, queuing delay, and output delay. N_U and N_L represent the total variable delay that the node must guarantee. Processing delay and output delay vary depending on the packet size, etc, but these delays are hard to adjust, so the queuing delay must be adjusted to guarantee N_U and N_L. Therefore, when a packet is received by a node, the expected output delay that may be required based on the packet size is subtracted from N_U and N_L, and the remaining values are determined as the packet's minimum, and maximum departure times. Since the output delay can also have a variable value, it is calculated as follows. When a packet arrives at time t, a minimum departure time, which is defined as t plus N_L minus del_min, and a maximum departure time, which is defined as t plus N_U minus del_max, are calculated. del_min and del_max are defined as the minimum and maximum times it takes from the time a packet leaves the queue until it completely leaves the node. del_min or del_max can be calculated with the size of the packet, port speed and minimum or maximum output delay, respectively. In addition, a nominal departure time, which is defined as the midpoint between the minimum departure time and maximum departure time, is calculated. The difference between (N_U - del_max) and (N_L - del_min) is called forwarding budget. Figure 2 shows the relationship among the minimum, nominal, and maximum departure times.

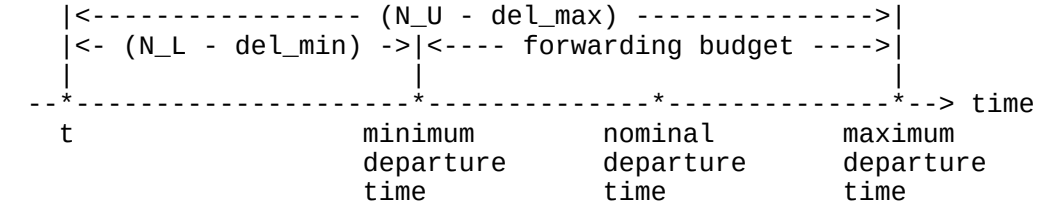

Figure 2: Relationship among the minimum, nominal, and maximum departure times

After calculating the minimum, nominal, and maximum departure times and performing necessary actions for packet forwarding, the packet is placed in the PIFO queue, where packets are arranged in the ascending order of their nominal departure times. When the minimum departure time of the packet in the head of queue (HoQ) has reached or passed current time, the packet is dequeued. Since the minimum and maximum departure times that a packet can stay in the queue are determined by

N_U and N_L, if the time taken from when the packet is received until it is processed and queued is long, the actual queuing time will automatically decrease. Conversely, if the time taken until it is queued is short, the queuing time will increase.

An example of the queuing operation is shown in Figure 3. Let us consider three incoming packets belonging to three different flows. The values of N_L and N_U are set to 1ms and 3 ms for the first flow, 0.34ms and 2ms for the second flow, and 0.3ms and 0.5ms for the third flow, respectively. To simplify the example, we assume del_min and del_max are zero.

* Assume that the first packet, P1, arrives at 0.2ms. Then, the minimum, nominal, and maximum departure times of P1 are calculated as 1.2ms, 2.2ms, and 3.2ms, respectively. P1 is placed at the HoQ and cannot leave the queue before 1.2ms.

* When the second packet, P2, arrives at 0.4ms, the minimum, nominal, and maximum times of P2 are determined as 0.74ms, 1.57ms, and 2.4ms. Since the nominal departure time of P2 is smaller than that of P1, P2 is placed at the HoQ and is scheduled to leave the queue at 0.74ms.

* The third packet, P3, is assumed to arrives at 0.6ms and its minimum, nominal, and maximum departure times are calculated as 0.9ms, 1.0ms, and 1.1ms, respectively. Since the nominal departure time of P3 is smaller than that of P2, P3 is placed at the HoQ and is followed by P2 and P1.

* At 0.9ms, P3 leaves the queue as its minimum departure time is 0.9ms. Following P3, P2 immediately leaves the queue as its minimum departure time (0.74ms) has passed.

* At 1.2ms, P1 is dequeued as its minimum departure time is 1.2ms.

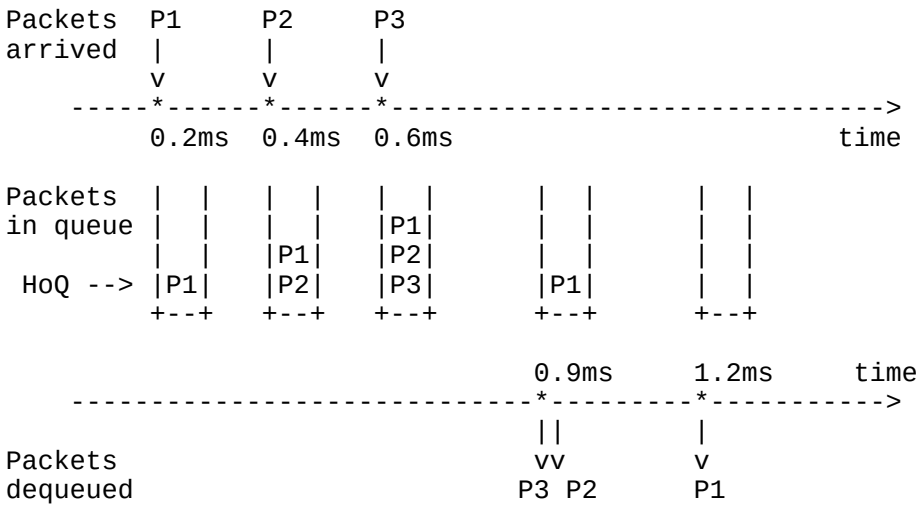

Figure 3: Example of queuing using PIFO queue

In this queuing operation, if packets with nominal departure times smaller than the nominal departure time of the HoQ packet continue to arrive, the packet with a small forwarding budget may exceed its maximum departure time. Therefore, the forwarding budget MUST be set to be larger than the time required to transmit any preceding packets of all the flows at the speed of the output port. This requirement of the forwarding budget needs to be confirmed through admission control in the controller plane when the flow is set up.

6. Controller Plane Operation

A controller collects network topology, PIFO queue resource, and various delay-related information such as port speed, link length, maximum and minimum processing and output delays of nodes, and maximum and minimum packet sizes of the flow, etc.

If a new DetNet flow is requested, the controller selects a path that satisfies the following conditions:

1. The E2E_F of the path MUST NOT exceed the MaxLatency required for the flow.

2. The sum of the available buffer resources of all nodes on the path MUST be large enough to provide a delay greater than E2E_VL minus minimum end-to-end variable non-queuing delay.

3. The available buffer resource of each node on the path MUST be large enough to provide a delay equal to N_U minus minimum node variable non-queuing delay.

4. The forwarding budget of each node MUST be larger than the time required to transmit any preceding packets of all the flows at the speed of the output port.

The first condition can be easily checked with the fixed delay values collected by the controller.

The second condition represents the minimum end-to-end queuing delay which is the minimum packet queuing time that nodes along the end-to-end path must guarantee. Since the maximum queuing delay can be obtained by dividing the buffer size by the service rate, whether condition 2 is satisfied can be checked as 奸措h=0,H] {buffer sizes} / the service rate >= E2E_VL - minimum e2e variable non-queuing delay.

In the second condition, the minimum end-to-end variable non-queuing delay is defined as the sum of lower bounds of variable delays except queuing delays occurring in nodes and links on the path, and the controller is assumed to be able to calculate from the information collected from the network. Likewise, in the third condition, the minimum node variable non-queuing delay is defined as the sum of lower bounds of variable delays except a queuing delay in a node, and the controller is assumed to be able to calculate from the information collected from the node.

In order to check the third and fourth conditions, N_L and N_U for each node need to be determined. There can be various ways to determine the values of N_L and N_U. In the following, we describe how both N_U and N_L can be obtained as one of the possible ways.

Considering the fact that the last node is the node that can take final actions to ensure the E2E_VL and E2E_VU for packets requiring on-time delivery, the value of N_U of the last node MUST be determined first. It is RECOMMENDED to set the value of N_U of the last node as large as possible as long as the buffer resource of the last node allows for the flow. N_U is computed as the maximum queuing delay plus the minimum node variable non-queuing delay. Since the maximum queuing delay is determined by the buffer size allocated to a flow, If a policy is defined, such as setting a maximum buffer size per flow to prevent a single flow from occupying all the buffer resources of the node, the maximum queuing delay is calculated based on the maximum buffer size specified by the policy. If no such policy exists, the operator considers the buffer size allocated per flow and determines N_U to be as large as possible without exceeding the total buffer size of the node. Then, the

remaining value after subtracting N_U of the last node from E2E_VL is divided into all other nodes. The value divided into each node is used as N_L for the node. The N_L of the last node is set to the time required to transmit any preceding packets of all the flows at the speed of the output port of the last node.

The value of N_U of each node except the last node is determined by dividing the remaining value after subtracting N_L of the last node from E2E_VU into all nodes except the last node on the path. Figure 4 shows the relationship between the variable delay of end-to-end and nodes

If the available buffer resource of each node on the path can support the value of N_U minus minimum node variable non-queuing delay, the third condition is satisfied. The fourth condition can be checked with N_U and N_L.

Once a path satisfying the aforementioned conditions is selected, the values of N_L and N_U are set to all nodes, and each node performs the operation described in Section 5 in the data plane. And, the buffer resources associated with N_U become unavailable for flows requested later.

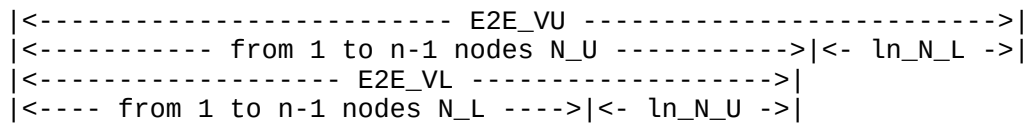

Figure 4: Relationship between the variable delay of end-to-end and nodes

7. Characteristics

7.1. Scaling requirements

The data and controller plane operations described in this document have the following characteristics for the requirements described in [I-D.ietf-detnet-scaling-requirements]. The item numbers below correspond to the numbers of the technical requirements in Section 3 of [I-D.ietf-detnet-scaling-requirements].

1. The solution described in this document does not require time synchronization. However, the solution measures the residence time and passes the remaining end-to-end latency values to the next node. As a specific delay value seen by all nodes must be the same amount, frequency synchronization is necessary.

2. The large single-hop propagation delay is supported. The solution describe in this document does not impose any limits on the amount of propagation delay.

3. Accommodation of the higher link speed is supported. It is considered possible to implement a PIFO queue supporting speeds of 100 Gbps or more.

4. The solution described in this document is scalable to the large number of flows as it does not require to maintain flow states in a node.

5. The solution described in this document is robust against node and link failures and topology changes, as the PREOF function can be applied.

6. Since the solution described in this document provides on-time forwarding while complying with the forwarding budget at all nodes, flow fluctuation inherently does not occur.

7. Since each node operates independently and the operation of the controller does not require any greater burden than existing typical network control, there are no scalability issues regarding the number of hops.

8. The solution described in this document uses a dedicated PIFO queue and clearly distinguishes the algorithm applied to it from that used for the existing FIFO queue. It supports multiple mechanisms by appropriately mapping each flow to a queue based on its SLA. Furthermore, it can support multiple algorithms across multiple domains by compartmentalizing the end-to-end delay requirements according to sections divided by differences in domain or link speed, and applying the upper and lower bounds of node delay for each section.

7.2. Taxonomy

Based on the draft of the taxonomy, latency-bound solutions are classified according to functional characteristics such as

* periodicity (periodic, non-periodic)

* network synchronization (phase and frequency synchronous, asynchronous)

* traffic granularity (flow level, flow aggregate level, class level)

* time bound (bounded, left-bounded, right-bounded, unbounded)

* service order (rate-based, time-based, arrival-based, priority-based)

The solution described in this document is a non-periodic, asynchronous, flow level, bounded, time-based solution.

The draft of the taxonomy also defines seven suitable categories for deterministic networking as follows.

* Right-bounded category

* Flow level periodic bounded category

* Class level periodic bounded category

* Flow level non-periodic bounded category

* Class level non-periodic bounded category

* Flow level rate based unbounded category

* Flow level rate based left-bounded category

The solution described in this document belongs to the "Flow level non-periodic bounded category", which is an on-time solution with a time-based service order characteristic that can adjust very low jitter and delay according to the user’s requirements.

8. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

9. Security Considerations

TBD

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>.

[RFC9320] Finn, N., Le Boudec, J.-Y., Mohammadpour, E., Zhang, J., and B. Varga, "Deterministic Networking (DetNet) Bounded Latency", RFC 9320, DOI 10.17487/RFC9320, November 2022, <https://www.rfc-editor.org/info/rfc9320>.

10.2. Informative References

[I-D.ietf-detnet-scaling-requirements] Liu, P., Li, Y., Eckert, T. T., Xiong, Q., Ryoo, J., zhushiyin, and X. Geng, "Requirements for Scaling Deterministic Networks", Work in Progress, Internet-Draft, draft-ietf-detnet-scaling-requirements-09, 7 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-detnet-scaling-requirements-09>.

Author’s Address

Yeoncheol Ryoo
ETRI
Email: dbduscjf@etri.re.kr