Network
Internet-Draft
Intended status: Standards Track
Expires: 27 December 2025

Shaofu. Peng
ZTE Corporation
Zongpeng. Du
China Mobile
Kashinath. Basu
Oxford Brookes University
Zuopin. Cheng
Zhejiang P&T College
Dong. Yang
Beijing Jiaotong University
Chang. Liu
China Unicom

25 June 2025

Deadline Based Deterministic Forwarding
draft-peng-detnet-deadline-based-forwarding-17

Abstract

This document describes a deadline based deterministic forwarding mechanism for IP/MPLS network with the corresponding resource reservation, admission control, scheduling and policing processes to provide guaranteed latency bound. It employs a latency compensation technique with a stateless core, to replace reshaping, making it suitable for the Differentiated Services (Diffserv) architecture [RFC2475].

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 27 December 2025.

Table of Contents

1. Introduction

[RFC8655] describes the architecture of deterministic network and defines the QoS goals of deterministic forwarding: Minimum and maximum end-to-end latency from source to destination, timely delivery, and bounded jitter (packet delay variation); packet loss ratio under various assumptions of the operational states of the nodes and links; an upper bound on out-of-order packet delivery. In order to achieve these goals, deterministic networks use resource reservation, explicit routing, service protection and other means. Resource reservation provides dedicated resources (such as bandwidth, buffer space, time slots, etc.) to DetNet flows. Explicit routing ensures the stability of the route and does not change with the real-time change of network topology. Service protection reduces the packet loss by sending multiple DetNet flows along multiple disjoint paths at the same time.

[P802.1DC] described some Quality of Service (QoS) features specified in IEEE Std 802.1Q, such as per-stream filtering and policing, queuing, transmission selection, stream control and preemption, in a network system which is not a bridge. The internal structure of IP/MPLS routers may also be based on these components to describe the scheduling process of packets. In the presence of admission check, policing and reshaping, a large number of packet scheduling techniques can provide bounded latency. However, many solutions may result in an inefficient use of network resources, or provide an overestimated latency. Currently the underlying scheduling mechanisms in IP/MPLS networks generally use SP (Strict Priority) and WFQ (Weighted Fair Queuing), and manage a small number of priority based queues. They are rate based schedulers.

For SP, the highest priority queue can consume the total port bandwidth, while for WFQ scheduler, each queue may be configured with a pre-set rate limit. Both of them can provide the worst-case latency, but evaluation is generally overestimated. In the case where the network core supports reshaping per flow (or optimized reshaping as provided by [IR-Theory]), the worst-case latency of a flow is approximately equal to the aggregated burst of the traffic class divided by the rate limit of that traffic class. A rate-based scheduler may refer to [Net-Calculus] to obtain its rate-latency service curve and get a more tighter evaluation. When the network core does not implement reshaping, multiple flows sharing the same priority may form burst cascade, making it more difficult or even impossible to evaluate the worst-case latency of a single flow. [EF-FIFO] discusses the SP scheduling behavior in this core-stateless situation, which requires the overall network utilization level to be limited to a small portion of its link capacity in order to provide an appropriate bounded latency.

To address the overestimation issue in rate-based scheduling where achieving low latency may require allocating a high service rate, [EDF-algorithm] prioritizes packets based on their deadlines, selecting the packet with the earliest deadline for transmission. It is considered optimal for bounded-delay services in the sense that it can support the delay bounds for any set of connections that can be supported by some other scheduling method. EDF is a delay-based scheduler, which distinguishes flows in terms of time urgency, rather than rough traffic classes.

The academic community has conducted extensive research on EDF. [RPQ-EDF] proposed a method for implementing a rotating queue for EDF and its schedulability conditions. [Jitter-EDF] proposed a combination of damper and EDF to achieve low jitter. [RC-EDF] and [RC-EDF-para] proposed combining re-shaping per hop with EDF. [CQ-EDF] proposed programmable calendar queues that enables the

efficient realization of EDF algorithm. [SCED] defined a deadline allocation algorithm that guarantees that a flow does have a minimum service curve.

This document introduces EDF scheduling mechanism to IP/MPLS network, as well as corresponding resource reservation, admission control, policing, etc, to provide guaranteed latency, as a supplement to IEEE 802.1 TSN mechanisms. It is a layer-3 solution and can operate over different types of QoS sensitive layer 2 including TSN but is not an alternative to TSN. A latency compensation-based option is recommended as a replacement for reshaping to ensure compatibility with the DiffServ architecture [RFC2475]. This document also discusses two scheduling behaviors: in-time scheduling and on-time scheduling. The former only provide bounded delay, while the latter further provide bounded jitter.

1.1. 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. EDF Scheduling Overview

The EDF scheduler assigns a deadline for each incoming packet, which is equal to the time the packet arrives at the node plus the latency limit, i.e., planned residence time (D), see Section 2.1. The EDF scheduling algorithm always selects the packet with the earliest deadline for transmission.

The precondition for EDF to work properly is that any DetNet flow must always satisfy the given traffic constraint function when it reaches a certain EDF scheduler. Therefore, it should generally implement traffic regulation at the network entrance to ensure that the admitted traffic complies with the constraints; And, implement reshaping on each intermediate node to temporarily cache packets to ensure that packets entering the EDF scheduler queue comply with the constraints. However, reshaping per flow is a challenge in large-scaling networks. Some core stateless optimization method need to be considered.

Another challenge of EDF scheduling is that queued packets must be sorted and stored according to their deadline, and whenever a new packet arrives at the scheduler, it needs to perform search and insert operations on the corresponding data structure, e.g., a Push-in First-out (PIFO) queue, at line rate. Some approximate methods for sorted queues can be considered.

According to the above two challenges and the potential optimization methods, we will obtain four combination solutions. Operators should choose appropriate solutions based on the actual network situation. This document suggests using option-3 or option-4, which are referred to as latency Compensation EDF (CEDF). CEDF adjusts and sorts the arriving flows by the latency compensation factor carried in the packets, ensuring that the flows arrived at the EDF scheduler always conform to their constraints. This approach avoids the need for the network core to maintain per-flow state, thereby supporting large scalability requirements.

* option-1: Reshaping plus sorted queue.

* option-2: Reshaping plus Rotating Priority Queue (RPQ).

* option-3: Latency Compensation plus sorted queue.

* option-4: Latency Compensation plus RPQ.

2.1. Planned Residence Time of the DetNet Flow

The planned residence time (termed as D) of the packet is an offset time, which is based on the arrival time of the packet and represents the maximum time allowed for the packet to stay inside the node.

For a deterministic path, the end-to-end delay includes two parts, the accumulated residence time and the accumulated link propagation delay. The accumulated residence time may be shared equally by each node along the path to obtain the average planned residence time of each node, or each node may have different planned residence time. The link propagation delay is generally constant, but not always so, for example, it may vary with temperature changes. It is assumed that the tool for detecting the link propagation delay can sense the changes beyond the preset threshold and trigger the recalculation of the deterministic path.

The ingress PE node, when encapsulating DetNet flows, can explicitly insert the planned residence time into the packet according to SLA. The transit node, after receiving the packet, can directly obtain the planned residence time from the packet. Generally, either only a single average planned residence time needs to be carried in the

packet, which is shared with all nodes along the path, or a stack of planned residence time is inserted, one for each node. [I-D.peng-6man-deadline-option] defined a method to carry the shared planned residence time in the IPv6 packets. [I-D.pb-6man-deterministic-crh] and [I-D.p-6man-deterministic-eh] defined methods to carry the stack of planned residence time in the IPv6 packets.

An implementation should support the policy to forcibly override the planned residence time obtained from the packet.

2.2. Delay Levels Provided by the Network

The network may provide multiple delay levels on the outgoing port, each with its own delay resource pool. For example, some typical delay levels may be 10us, 20us, 30us, etc.

In theory, any additional delay level can be added dynamically, as long as the buffer and remaining bandwidth on the data plane allow.

The quantification of delay resource pool for each delay level is actually based on the schedulability conditions of EDF. This document introduces two types of resources per delay level:

* Burst: It represents the amount of bits bound that a delay level provides.

* Bandwidth: It represents the amount of bandwidth bound that a delay level provides. The bandwidth possessed by a certain delay level is also known as the service rate of that delay level.

For more information on the construction of resource pools, please refer to Section 3.2 and Section 4.3.

2.3. Relationship Between Planned Residence Time and Delay Level

The planned residence time (D) is the per-hop latency requirement of the flow, while the delay level (d) is the capability provided by the link.

Generally, only a limited number of delay levels are required to support a larger number of per-hop latency requirement. For example, there are delay levels such as d_1, d_2, ..., and d_n. In the resource management of the control plane, d_i resources is assigned to all D that meet d_i <= D < d_(i+1).

2.4. Relationship Between Service Burst Interval and Delay Level

Although it is generally preferred to have the service burst interval (SBI) greater than the maximum delay level, there is actually no necessary association between SBI and delay level.

A flow with a small SBI (such as 10 us) can request a larger delay level (such as 100 us). During the extended residence time caused by the larger delay level, there will be multiple rounds of burst interval packets leading to bursts accumulation. However, these packets can be distinguished and sent in sequence. In fact, the original SBI can be multiplied several times to obtain the expanded SBI (which includes multiple original bursts), with a length greater than the requested delay level, to get the preferred paradigm.

Similarly, a flow with a large SBI (such as 1 ms) can also request a smaller delay level (such as 10 us).

3. Sorted Queue

[PIFO] is a priority queue that maintains the scheduling order or time. A PIFO allows elements to be pushed into an arbitrary position based on an element’s rank (the scheduling order or scheduling time), but always dequeues elements from the head.

3.1. Scheduling Mode for Push-in First-out (PIFO)

A PIFO queue may be configured as either in-time or on-time scheduling mode, but cannot support both modes simultaneously.

In the in-time scheduling mode, as long as the queue is not empty, packets always depart from the head of queue (HoQ) for transmission. The actual bandwidth consumed by the scheduler may exceed its preset service rate C.

In the on-time scheduling mode, if the queue is not empty and the rank of the HoQ packet is equal to or earlier than the current system time, then the HoQ packet will be sent, otherwise, not.

3.2. Schedulability Condition for PIFO

[RPQ-EDF] has given the schedulability condition for classic EDF that is based on any type of sorted queue with in-time scheduling mode.

Suppose for any delay level d_i, the corresponding accumulated constraint function is A_i(t). Let d_i < d_(i+1), then the schedulability condition is:

sum{A_i(t-d_i) for all i} <= C*t (Equation-1)

where, C is service rate of the EDF scheduler.

It should be noted that for a delay level d_i, its residence time is actually contributed by its own flows and all other more urgent delay levels. Based on the schedulability conditions, the traffic arrival constraint function could be selected according to the preset delay level, or alternately the delay level could be selected according to the preset traffic arrival constraint function.

When setting up new flows in the network, admission check based on schedulability condition must be executed on each link that the flow passes through.

Here, A_i(t) defines the upper limit of eligible arrivals of delay level d_i, and should not be treated as the actual arrivals (the actual arrivals is denoted as a_i(t) for distinction). As described in this document, a_i(t) may contain ineligible arrivals that need first to be converted (or sorted) into eligible arrivals, e.g., by method of regulation (Section 5) or latency compensation (Section 6), and then processed by the EDF scheduler.

3.2.1. Schedulability Conditions Analysis for In-time Mode using Leaky Bucket Constraint Function

Assuming that n delay levels (d_1, d_2,..., d_n) in the network needs to be supported, and the traffic arrival constraint function of each delay level d_i is the leaky bucket arrival curve A_i(t) = b_i + r_i * t where b_i and r_i are the burst and rate of delay level d_i. Then, Equation-1 can be expressed as:

b_1 <= C*d_1 - M

b_1 + b_2 + r_1*(d_2-d_1) <= C*d_2 - M

b_1 + b_2 + b_3 + r_1*(d_3-d_1) + r_2*(d_3-d_2) <= C*d_3 - M

... ...

sum(b_1+...+b_n) + r_1*(d_n-d_1) + r_2*(d_n-d_2) + ... + r_n_1*(d_n-d_n_1) <= C*d_n - M

where, C is the service rate of the EDF scheduler, M is the maximum size of the interference packet.

Note that the preset value of b_i does not depend on r_i, but r_i generally refers to b_i (and burst interval) for during setup. For example, the preset value of r_i may be small, while the value of b_i may be large. Such parameter design is more suitable for transmitting traffic with large service burst interval, and large service burst size, but small bandwidth requirements.

An extreme example is that the preset r_i of each level d_i is close to 0 (this is because the burst interval of the served flow is too large), but the preset b_i is close to the maximum value (e.g., b_1 = C*d_1 - M), then when the concurrent flow of all delay levels is scheduled, the time 0~d_1 is all used to send the burst b_1, the time d_1~d_2 is all used to send the burst b_2, the time d_2~d_3 is all used to send the burst b_3, and so on.

However, the typical allocation scheme is that the preset r_i of each level d_i will divide C roughly equally. For example, b_1 may first be pre-allocated as b_1 = C*d_1 - M, r_1 = C/n where n is the number of delay levels; Then recursively b_2 is pre-allocated as b_2 = C*(d_2-d_1)*(n-1)/n, r_2 = C/n; And so on. The pre-allocated parameters b_i and r_i of each level d_i constitute the resources of that delay level of the link. A path can reserve required burst and bandwidth resources of the specific delay level d_i, and the reservation is successful only if the two resources are successfully reserved at the same time. If either b_i or r_i is full, the delay resource of level d_i is exhausted.

Alternatively, a more tight allocation scheme is to not preset the parameters of A_i(t), but to dynamically accumulate the parameters of A_i(t) based on the actual flows setup demand, and always check whether the schedulability condition is met based on the updated A_i(t) during the flow setup procedure. In this case, it is still necessary to set a resource limit for each delay level to prevent the flows of a certain delay level from consuming all resources. For example, the resource limit of each delay level d_i may be set to b_i_limit = C*(d_i - d_(i-1)) - M, r_i_limit = C/n. In this case, the dynamically updated b_i and r_i should be treated as utilized resources, and participate in schedulability condition checks.

Note that for some delay level d_i, its resource may be explicitly set to empty, i.e., b_i = 0, r_i = 0. This brings flexibility, and resources can be freed up for later delay levels with lower priority to use.

In a specific scenario, if the ideal arrival packet interval (by the method of re-shaping or latency compensation) of all service flows is large, not less than the maximum delay level d_n, the schedulability condition can be further simplified as follows:

```
b_1 <= C*d_1 - M, r_1 = b_1 / d_n;

b_1 + b_2 <= C*d_2 - M, , r_2 = b_2 / d_n;

b_1 + b_2 + b_3 <= C*d_3 - M, r_3 = b_3 / d_n;

... ...

sum(b_1+...+b_n) <= C*d_n - M, r_n = b_n / d_n;
```

The above simplified condition implies that the total number of bursts contained within any time interval d_n does not exceed sum(b_1+...+b_n). This is true because for any flow i it never contains two packets in a single time interval d_n. In this case, it can support a larger service scale than the original condition.

It should be noted that the burst and bandwidth resource of each delay level mentioned above always assumes that the flows it serves arrive concurrently from many incoming interfaces (i.e., with a large concurrency), which is a safe but conservative assumption. If operators are aware of the specific topology knowledge of the network, such as having very little (or even no) concurrency, they can design special resource pools.

For example, in the case of one incoming and one outgoing packet, there will be no queueing delay, and a single delay level can be used for all interleaved flows. In this case, the delay level value just equals the forwarding delay (F), plus the transmission delay of a single packet. There is no limit on burst resources, and the upper limit of bandwidth resources is still the service rate C. Alternatively, a simple FIFO queueing mechanism can also work in this case.

Alternately, in the case of multiple incoming one outgoing, if the eligible arrivals pattern of each incoming interface is known, the resolved size can be calculated (i.e., the maximum difference between the total arriving bursts and the sending capacity during the busy period.) and used to design a single delay level. In this case, the delay level value equals the forwarding delay (F), plus the transmission delay of the resolved size. Its burst and bandwidth resources are equal to the sum of the eligible arriving bursts and bandwidth of all incoming ports. It is also possible to design more delay levels, each for a different subset of flows. In this case, the burst resource of urgent delay level must be limited to avoid larger delay values for other delay levels.

3.2.2. Schedulability Condition Analysis for On-time Mode

Compared with in-time mode, on-time mode is non-work-conserving, which can be considered as the combination of damper and EDF scheduler. The on-time scheduling mode applied to a flow tries to maintain the packet interval between successive packets of that flow to be consistent with the regulated interval on the flow entrance node. The maintenance of the packet intervals helps to limit the bandwidth consumption by that flow and also restricts the arrival curve within the traffic constraint function so that the schedulability condition (i.e., Equation-1) can also be applied to the on-time scheduling mode. See Section 8 for more information about jitter control.

3.3. Buffer Size Design

The service rate of the EDF scheduler, termed as C, can reach the link rate, but generally only needs to be configured as part of the link bandwidth, such as 50%. It should allow provision for hierarchical scheduling, for example, the EDF scheduler may participate in higher-level WFQ scheduling along with other schedulers.

If flows are rate-controlled (i.e., reshaping is done inside the network, or on-time mode is applied), the maximum depth of the PIFO buffer should be C * d_n, where d_n is the maximum delay level. Otherwise, more buffer is necessary to absorb the burst accumulation. The PIFO buffer zone where the distance from HoQ exceeds the maximum delay level is just used to store accumulated bursts. Please refer to Section 12 for more considerations.

4. Rotation Priority Queues (RPQ)

[RPQ-EDF] described the rotating priority queues. Here, the priority granularity of the queue is set the same as the rotation interval. However, in our context, if the planned residence time of the flow is used as priority, it will require a lot of priority levels and corresponding queues. Therefore, this section provides an enhanced approach with a limited number of rotating priority queues with count-down time range whose rotation interval is more refined, with the following characteristics:

* Each queue has CT (Count-down Time) that is decreased by RTI (Rotation Time Interval). The CT difference between two adjacent queues is CTI (CT Interval). RTI must be less than or equal to CTI, with CTI = K * RTI, where K is a natural number greater than or equal to 1.

* The smaller the CT, the higher the priority. At the beginning, all queues have different initial CT values, i.e., staggered from each other, e.g., one queue has the minimum CT value (termed as MIN_CT), and one queue has the maximum CT value (termed as MAX_CT), and the CT values of all queues increase equally by CTI. Note that CT is just the countdown of the HoQ, and the countdown of the end of the queue (EoQ) is near CT+CTI. So the CT attribute of a queue is actually a range [CT, CT+CTI).

* For a queue whose CT is MIN_CT, after a new round of CTI, its CT will become MIN_CT - CTI and immediately return to MAX_CT.

The above CTI, RTI, MIN_CT and MAX_CT value should be chosen according to the hardware capacity. Each link can independently use different CTI. The general principle is that the larger is the bandwidth, the smaller is the CTI. The CTI must be designed large enough to include interference delay caused by a single packet with maximum size.

The choice of RTI should consider the latency granularity of various DetNet flows, so that CT updated per RTI can match the delay requirements of different flows. An implementation may not choose to let CT be actually updated at the granularity of RTI, but at the granularity of CTI. For example, the elapsed time within CTI can be recorded, and (cur_CT - elapsed_time) can be used as the actual CT of the queue, where cur_CT is the current CT of the queue that has not been updated yet. Although the update of cur_CT is slow, the actual CT is sensitive enough.

According to different scheduling mode configured to the RPQ, MIN_CT may be designed to different values. For in-time mode, MIN_CT may be 0. For on-time mode with option E|D decoupling (see Section 8), MIN_CT may also be 0 where D is the planned residence time and E is the accumulated residence time deviation, also termed as latency deviation (E). For on-time mode with option E+D integration, MIN_CT may be -N*CTI, where N is the number of delay levels.

A specific example of RPQ configured with in-time scheduling mode is depicted in Figure 1.

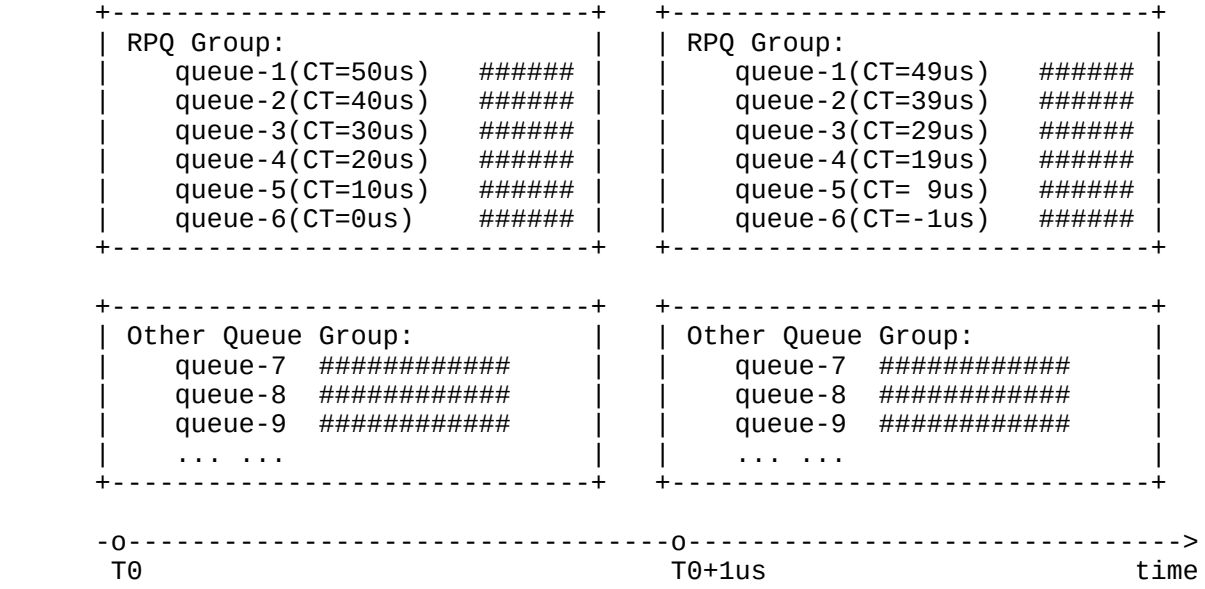

Figure 1: Example of RPQ Groups

In this example, the CTI for RPQ group is configured to 10us. Queue-1 ~ queue-6 are members of RPQ group. Each queue has its initial CT attribute, and the CT of all queues are staggered from each other. For example, the CT of queue-1 is 50us (MAX_CT), the CT of queue-2 is 40uS, ..., the CT of queue-6 is 0 (MIN_CT).

Suppose the scheduling engine initiates a rotation timer with a time interval of 1us, i.e., CTI = 10 * RTI in this case. As shown in the figure, at T0 + 1us, the CT of queue-1 becomes 49us, the CT of queue-2 becomes 39us, etc.

At T0 + 10us, the CT of queue-6 will return to 50us (MAX_CT).

Note that the minimum D requested by a DetNet flow should not be smaller than d_1+F, where d_1 is the most urgent delay level, F is the intra-node forwarding delay. Therefore, any packets with in-time scheduling should have Q (i.e., D + E - F) that is not be smaller than d_1, and should never be inserted to a queue whose CT is negative.

4.1. Alternate Queue Allocation Rules (QAR)

There may be extreme scenarios that multiple delay levels of eligible bursts arrive sequentially, with lower priority burst arriving first and higher priority burst arriving later, and then simultaneously releasing flood. In this case, it is necessary to ensure that the higher priority burst is sent first to meet its deadline.

Therefore, it may further let a RPQ queue (act as the virtual parent queue) contain multiple subqueues, each for a delay level. The physical subqueue with small delay level (e.g., 10us) is ranked before the physical subqueue with large delay level (e.g., 20us). Packets are actually stored in the physical subqueues. That is, packets belonging to different delay levels are inserted into different subqueues and protected. In this way, for two packets with the same Q but different D, the packet with the smallest D may be scheduled first.

This alternate queue allocation rule enables eligible arrivals always have a place to store, avoiding conflicts in local positions of the RPQ queue group and causing overflow.

4.2. Scheduling Mode for RPQ

A RPQ group may be configured as either in-time or on-time scheduling mode, but cannot support both modes simultaneously.

In the in-time scheduling mode, in all non-empty queues, the packets in each queue are sequentially sent in the order of high priority queue to low priority queue. The actual bandwidth consumed by the scheduler may exceed its service rate C.

In the on-time scheduling mode, only in all non-empty queues with CT <= 0, the packets in each queue are sequentially sent in the order of high priority queue to low priority queue.

For a virtual parent queue that is allowed to be sent, for the multiple non-empty physical subqueues it contains, packets are sequentially sent from the non-empty physical subqueues in the order starting from the physical subqueues with small delay levels to the physical subqueues with large delay levels. Only when a physical subqueue is cleared, packets from the next non-empty physical subqueue be sent.

4.3. Schedulability Condition for RPQ

This section first discusses the schedulability condition based on RPQ with in-time scheduling mode.

Suppose for any delay level d_i, the corresponding accumulated constraint function is A_i(t), and let d_i < d_(i+1). Similarly, suppose for any planned residence time D_i, the corresponding constraint function is A'_i(t). For simplicity, the intra-node forwarding delay F is assumed to be 0. Then the schedulability condition is:

* A_1(t-d_1) + sum{A_i(t+CTI-d_i) for all i>=2} <= C*t, if a d_i contains only one D_i. (Equation-2)

* sum{A_i(t+CTI-d_i) for all i>=1} <= C*t, if d_i contains multiple D_i. (Equation-3)

where CTI is the CT interval between adjacency queue, and C is service rate of the EDF scheduler.

The proof is similar to [RPQ-EDF], except that the rotation interval is fine-grained and defined by RTI and the priority of each queue is within the CT range. Please refer to Appendix A.

Note that the key difference between the above two conditions (i.e., Equation-2, Equation-3) and one based on sorted queue (i.e., Equation-1) is the CTI factor.

Other common considerations are the same as in Section 3.2.

4.3.1. Schedulability Condition for Alternate QAR

According to Section 4.1, a RPQ queue may further contain multiple subqueues, each for a delay level. Under the same parent queue, all subqueues are sorted in descending order of delay level. In this case, the precise workload should exclude packets with higher delay levels than the observed packet.

In the case that d_i contains only one D_i, the schedulability condition is Equation-1.

In the case that d_i contains multiple D_i, the schedulability condition is still Equation-3.

Please refer to Appendix B.

4.3.2. Schedulability Conditions for Leaky Bucket Constraint Function

Assume that n delay levels (d_1, d_2,..., d_n) in the network needs to be supported, and the traffic arrival constraint function of each delay level d_i is the leaky bucket arrival curve A_i(t) = b_i + r_i * t. Then, Equation-2 can be expressed as:

```
b_1 <= C*d_1 - M

b_1 + b_2 + (r_1+r_2)*CTI <= C*d_2 - M

b_1 + b_2 + b_3 + (r_1+r_2)*2*CTI + r_3*CTI <= C*d_3 - M

... ...

sum(b_1+...+b_n) + (r_1+r_2)*(n-1)*CTI + r_3*(n-2)*CTI + ... +
r_n*CTI <= C*d_n - M
```

where, C is the service rate of the EDF scheduler, M is the maximum size of the interference packet.

Equation-3 can be expressed as:

```
b_1 + r_1*CTI <= C*d_1 - M

b_1 + b_2 + r_1*2*CTI + r_2*CTI <= C*d_2 - M

b_1 + b_2 + b_3 + r_1*3*CTI + r_2*2*CTI + r_3*CTI <= C*d_3 - M

... ...

sum(b_1+...+b_n) + r_1*n*CTI + r_2*(n-1)*CTI + ... + r_n*CTI <=
C*d_n - M
```

Similarly, in a specific scenario, if the ideal arrival packet interval (by the method of re-shaping or latency compensation) of all service flows are large, not less than the maximum delay level d_n, the above two schedulability conditions can be further simplified as follows:

```
b_1 <= C*d_1 - M, r_1 = b_1 / d_n;

b_1 + b_2 <= C*d_2 - M, , r_2 = b_2 / d_n;

b_1 + b_2 + b_3 <= C*d_3 - M, r_3 = b_3 / d_n;

... ...

sum(b_1+...+b_n) <= C*d_n - M, r_n = b_n / d_n;
```

4.3.3. Schedulability Condition Analysis for On-time Mode

Compared with in-time mode, on-time mode is non-work-conserving, which can be considered as the combination of damper and EDF scheduler. On-time scheduling mode applied on a flow try to maintain the packet interval between any adjacent packets of that flow to be consistent with the regulated interval on the flow entrance node. The maintenance of the packet intervals does not result in an increase in the bandwidth consumed by the flow. Furthermore, it does not cause the arrival curve to violate the traffic constraint function. So the schedulability condition (i.e., Equation-2/3) can also be applied to on-time scheduling mode. See Section 8 for more information about jitter control.

4.4. Buffer Size Design

An implementation may let all queues share the common buffer. Especially if Alternate QAR (Section 4.1) is applied, the actual buffer cost of a virtual parent queue is contributed by all the physical subqueues it contains. The actual buffer cost of each physical sub queue is dynamically allocated based on whether there is a packet inserted. According to Section 4.3, the maximum buffer cost of a physical subqueue may reach the upper limit of burst resources for the corresponding delay level.

If flows are rate-controlled (i.e., reshaping is done inside the network, or on-time scheduling mode is applied), the MAX_CT may be designed as the maximum delay level, and total necessary buffer shared by all queues should be C * d_n, where C is the service rate of the scheduler and d_n is the maximum delay level. Otherwise, MAX_CT should be larger than the maximum delay level, and with more necessary buffer, to absorb the burst accumulation. All the queues with CT larger than the maximum delay level are just used to store accumulated bursts. Please refer to Section 12 for more considerations.

5. Reshaping

Reshaping per flow inside the network, as described in [RFC2212], is done at all heterogeneous source branch points and at all source merge points to restore (possibly distorted) traffic’s shape to conform to the TSpec. Reshaping entails delaying packets until they are within conformance of the TSpec.

A network element MUST provide the necessary buffers to ensure that conforming traffic is not lost at the reshaper. Note that while large buffer makes it appear that reshapers add considerable delay, this is not the case. Given a valid TSpec that accurately describes the traffic, reshaping will cause little extra actual delay at the reshaping point (and will not affect the delay bound at all).

Maintaining a dedicated shaping queue per flow can avoid burstiness cascading between different flows with the same traffic class, but this approach goes against the design goal of packet multiplexing networks. [IR-Theory] describes a more concise approach by maintaining a small number of interleaved regulators (per traffic class and incoming port), but still maintaining the state of each flow. With this regulator, packets of multiple flows are processed in one FIFO queue and only the packet at the head of the queue is examined against the regulation constraints of its flow. However, as the number of flows increases, the IR operation may become burdensome as much as the per-flow reshaping.

For any observed EDF scheduler in the network, when the traffic arriving from all incoming ports is always reshaped, then these flows comply with their arrival constraint functions.

6. Latency Compensation

[RFC9320] presents a latency model for DetNet nodes. There are six type of delays that a packet can experience from hop to hop. The processing delay (type-4), the regulator delay (type-5) , the queueing subsystem delay (type-6), and the output delay (type-1) together contribute to the residence time in the node.

In this document, the residence time in the node is simplified into two parts: the first part is to lookup the forwarding table when the packet is received from the incoming port (or generated by the control plane) and deliver the packet to the line card where the outgoing port is located; the second part is to store the packet in the queue of the outgoing port for transmission. These two parts contribute to the actual residence time of the packet in the node. The former can be called forwarding delay (termed as F) and the latter can be called queueing delay (termed as Q). The forwarding delay is related to the chip implementation and is generally constant (with a maximum value); The queueing delay is unstable.

6.1. Accumulated Residence Time Deviation

The accumulated residence time deviation, also termed as latency deviation (E), equals accumulated planned residence time minus accumulated actual residence time. This value can be zero, positive, or negative.

The accumulated planned residence time of the packet refers to the sum of the planned residence time of all upstream nodes before the packet is transmitted to the current node. The accumulated actual residence time of the packet, refers to the sum of the actual residence time of all upstream nodes before the packet is transmitted to the current node.

In the case of in-time scheduling, E may be a very large positive value. While in the case of on-time scheduling, E may be 0, or a small value close to 0.

The setting of the latency deviation (E) of the packet needs to be friendly to the chip for reading and writing. For example, it should be designed as a fixed position in the packet. The chip may support flexible configuration for that position.

[I-D.peng-6man-delay-options] defined the method for carrying the latency deviation (E) in the IPv6 Hop-by-Hop Options Header. [I-D.pb-6man-deterministic-crh] and [I-D.p-6man-deterministic-eh] defined methods for carrying the latency deviation (E) in the IPv6 Routing Header.

6.2. Allowable Queueing Delay

When an EDF scheduler receives a packet, it can calculate allowable queueing delay (Q) for the packet. Specifically, it can first get the latency deviation (E), and add it to the planned residence time (D) of the packet at this node to obtain the adjustment residence time, and then deduct the actual forwarding delay (F) of the packet in the node. This can be expressed as:

$$Q = D + E - F$$

The scheduler selects a buffer position (e.g., queue-id, or rank) for the packet based on Q.

Note that an implementation may calculate Q at incoming port and determine the buffer position of the outgoing port. In this case, Q = D + E, and a buffer position indication may be notified from the incoming port to the outgoing port.

Assuming that the current node in a deterministic path is h, all upstream nodes are from 1 to h-1. For any node h, if the planned residence time is D[h], the actual residence time is R[h], the input latency deviation (contributed by all upstream nodes) is E[h], the forwarding delay intra-node is F[h], then the allowable queueing delay (Q) of the packet on node h, i.e., Q[h], is:

$$Q[h] = D[h] + E[h] - F[h]$$

$$E[h] = D[h-1] + E[h-1] - R[h-1]$$

$$D[0], E[0], R[0] = 0$$

6.3. Scheduled by Allowable Queueing Delay

The packet will be scheduled based on its Q that is affected by latency compensation. The earlier literature similar to the idea of latency compensation based on E can be found in [Jitter-EDF].

The core stateless latency compensation can achieve the effect of reshaping per flow to get the eligible arrivals pattern. Q can be used to sort ineligible arrivals of one delay level and prevent them from interfering with the scheduling of eligible arrivals of other delay levels.

Firstly, at the flow (e.g., flow i) entrance node, all packets (after regulation) of flow i will be released to the EDF scheduler one after another at different time (termed as ideal arrival time), but with the same allowable queueing delay (Q), with initial E = 0, i.e., Q = D, assuming no link propagation delay and intra-node forwarding delay for simplicity. This arrival pattern faced by the scheduler on the flow entrance node is denoted as arrival_pattern_0, which contains a sequence of packets with variant of intervals between adjacent packets. This arrival_pattern_0 is considered as eligible arrivals because its arrival curve is less than the constraint function $A_i(t)$. For any packet p in arrival_pattern_0, assuming its ideal arrival time is t_p_0, then, arrival_pattern_1 = arrival_pattern_0 + D is also eligible arrival, where, arrival_pattern_0 + D means that the ideal arrival time (at the scheduler of flow entrance node) of each packet in arrival_pattern_0 is added with D. In fact, arrival_pattern_1 is the eligible arrivals on the second node. That is, the second node may recover the eligible arrivals arrival_pattern_1 from the actual arrivals with the help of latency compensation, and then to schedule based on arrival_pattern_1. For instance, for any packet p, assuming it experiences an actual queuing delay q at the flow entrance node, then it will actually arrive at the second node at time t_p_0 + q, with E = D - q carried in the sending packet. The second node will recover the eligible arrival

time of packet p by, eligible arrival time = actual arrival time + E = t_p_0 + q + D - q = t_p_0 + D. Therefore, arrival_pattern_1 is recovered.

Similarly, the third node may recover arrival_pattern_2 = arrival_pattern_0 + 2*D, and the fourth node may recover arrival_pattern_3 = arrival_pattern_0 + 3*D, and so on. On any node h, packet p will be sorted in the scheduler queue based on its ideal departure time (i.e., eligible arrival time plus D) for scheduling.

Because the scheduler always schedules based on eligible arrivals, its scheduling power will not be overwhelmed by actual arrivals that may include burst accumulation.

It may be assumed that the packets are sorted in the queue with the ideal departure time as a virtual regulation, because the rank distance between the adjacent packets of the flow i is maintained consistently with the corresponding regulated interval between these two adjacent packets on the flow entrance node. There is no requirement for this virtual regulation and a real regulation component to have exactly the same pattern, as long as each pattern is less than the arrival constraint function A_i(t).

Although latency compensation has the effect of reshaping, but it is not equivalent to reshaping. Considering an accumulated bursts that violates the traffic constraint function and arrives at a node, if reshaping is used, it will substantially introduce shaping delay for the ineligible bursts, which will then enter the queueing subsystem. However, if latency compensation is used, this ineligible bursts will only be penalized with a larger Q and tolerated to be placed in the queueing sub-system, and in the case of in-time mode it may be immediately sent if higher priority queues are empty.

Note that the premise of latency compensation is that a flow must be based on a fixed explicit path. If multiple packets from the same flow arrive at the intermediate node via multiple paths with different propagation lengths, even if these packets are all eligible, bursts accumulation may still form and cannot even be punished.

7. Solution Options

7.1. Option-1: Reshaping plus Sorted Queue

As shown in Figure 2, a received packet is inserted to the PIFO queue according to rank = A + D - F, where, A is the time that packet arrived at the scheduler, i.e., arrive_time_S in the figure. Depending on the situation of the accumulated burst arrived at the input port, different packets may face different shaping delays. The shaper will convert the input ineligible arrivals pattern (if possible) into an eligible arrivals pattern. Here, D - F may be denoted as the allowable queueing delay Q.

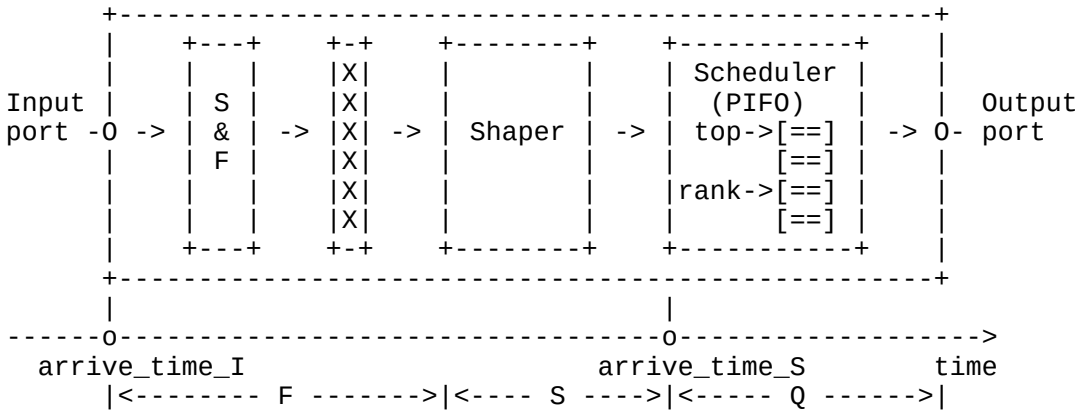

Figure 2: Reshaping plus Sorted Queue

Enqueue rule:

* For two packets with different rank, the packet with a smaller rank is closer to the head of the queue.

* For two packets with the same rank, the packet with a smaller D is closer to the head of the queue.

* For two packets with the same rank and D, the packet that arrive at the scheduler first is closer to the head of the queue.

The planned residence time (D) should be carried in the packet.

The scheduling mode (in-time or on-time) should also be carried in the packet, and used to insert packet into PIFO with the corresponding scheduling mode.

Dequeue rule:

* As mentioned in Section 3.1, for a PIFO with in-time scheduling mode, as long as the queue is not empty, packets always departure from the HoQ for transmission; while for PIFO with on-time scheduling mode, only if the queue is not empty and the rank of the HoQ packet is equal to or earlier than the current system time, the HoQ packet can be sent.

However, in this option the dequeue rule of on-time mode can not guarantee jitter, due to lack of factor E to absorb jitter per hop. The dequeue rule of on-time mode only controls the starting time when packets are allowed to be scheduled, but cannot guarantee that different packets experience the same queuing delay.

7.2. Option-2: Reshaping plus RPQ

As shown in Figure 3, a received packet is inserted to the appropriate RPQ queue with specific CT to meet CT <= Q < CT+CTI when the packet arrived at the scheduler, where Q = D - F. Depending on the situation of the accumulated burst arrived at the input port, different packets may face different shaping delays. The shaper will convert the input ineligible arrivals pattern (if possible) into an eligible arrivals pattern.

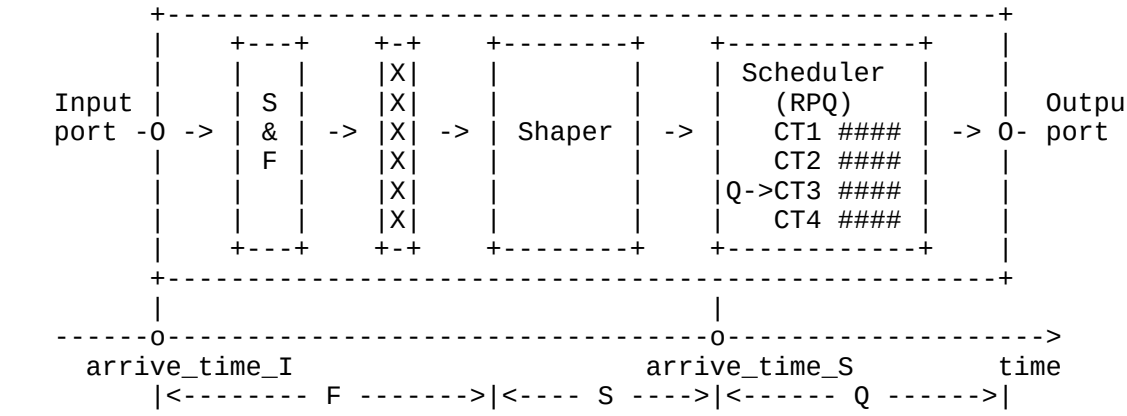

Figure 3: Reshaping plus RPQ

Enqueue rule:

* For a packet with Q, select the target RPQ queue (i.e., the virtual parent queue) with corresponding CT, that meet CT <= Q < CT+CTI.

* Under the selected virtual parent queue, select the target physical subqueue with corresponding delay level d_i, which is closest to D-F and not greater than D-F.

The planned residence time (D) should be carried in the packet.

The scheduling mode (in-time or on-time) should also be carried in the packet, and used to insert packet into RPQ with the corresponding scheduling mode.

Dequeue rule:

* As mentioned in Section 4.2, for a RPQ group with in-time scheduling mode, in all non-empty queues, the packets in each queue are sequentially sent in the order of high priority queue to low priority queue; while for a RPQ group with on-time scheduling mode, only in all non-empty queues with CT <= 0, the packets in each queue are sequentially sent in the order of high priority queue to low priority queue.

However, in this option the dequeue rule of on-time mode can not guarantee jitter, due to lack of factor E to absorb jitter per hop. The dequeue rule of on-time mode only controls the starting time when packets are allowed to be scheduled, but cannot guarantee that different packets experience the same queuing delay.

7.3. Option-3: Latency Compensation plus Sorted Queue

As shown in Figure 4, a received packet is inserted to the PIFO queue according to rank = A1 + E + D, or rank = A2 + E + D - F, where, A1 is the time that packet arrived at the input port (i.e., arrive_time_I in the figure), A2 is the time that packet arrived at the scheduler (i.e., arrive_time_S in the figure). Note that E is initially 0 on the flow entrance node, and generally not 0 on other nodes and will update per hop. Depending on the situation of the accumulated burst arrived at the input port, different packets may have different input latency deviation E. Latency compensation will convert the input ineligible arrivals pattern (if possible) into an eligible arrivals pattern. Here, E + D - F may be denoted as the allowable queueing delay Q.

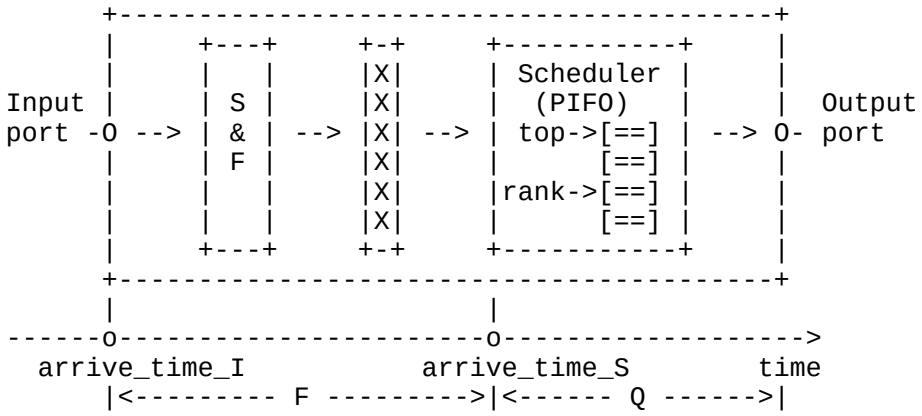

Figure 4: Latency Compensation plus Sorted Queue

The planned residence time (D) and latency deviation (E) should be carried in the packet.

The enqueue and dequeue operations are the same as Section 7.1.

In this option the dequeue rule of on-time mode can guarantee jitter with the help of factor E to absorb jitter per hop. See Section 8 for more information.

7.3.1. Packet Disorder Considerations

Suppose that two packets, P1, P2, are generated instantaneously from a specific flow at the source, and the two packets have the same planned residence time. P1 may face less interference delay than P2 in their journey. When they arrive at an intermediate node in turn, P2 will have less allowable queueing delay (Q) than P1 to try to stay close to P1 again. It should be noted that to compare who is earlier is based on the time arriving at the scheduler plus packet's Q. The time difference between the arrival of two packets at the scheduler may not be consistent with the difference between their Q. It is possible to get an unexpected comparison result.

As shown in Figure 5, P1 and P2 are two back-to-back packets belonging to the same flow. The arrival time when they are received on the scheduler is shown in the figure. Suppose that the Q values of two adjacent packets P1 and P2 are 40us and 39us, and arrive at the scheduler at time T1 and T2 respectively. P1 will be sorted based on T1 + 40us, while P2 will be sorted based on T2 + 39us. Ideally, T2 should be T1 + 1us. However, this may be not the case. For example, it is possible that T2 = T1 + 0.9us, Q1 = 40, Q2 = 39.1, but just because the calculation accuracy of Q1 and Q2 is

microseconds, so they are, e.g., with half-adjust, approximately 40 us and 39 us, respectively. This means that P2 will be sorted before P1 in the PIFO, resulting in disorder.

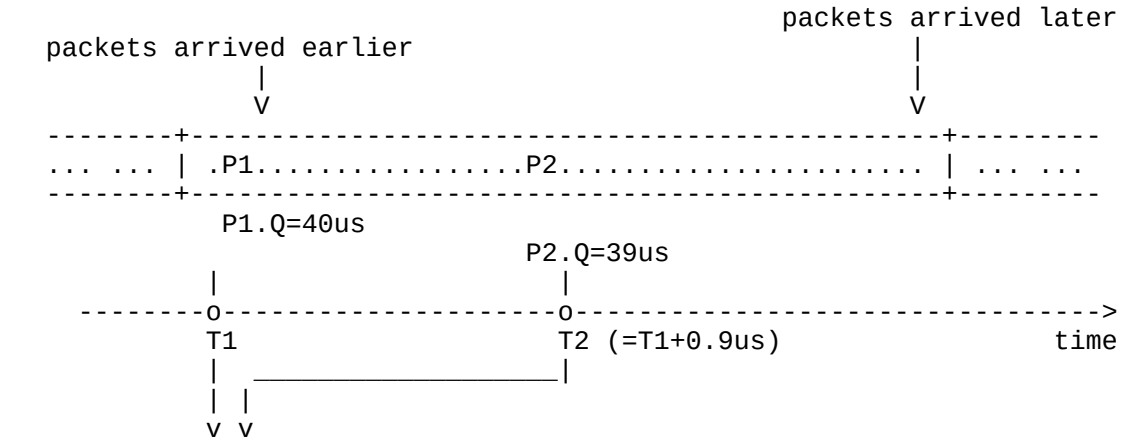

Figure 5: Disorder Illustration of PIFO

DetNet architecture [RFC8655] provides Packet Ordering Function (POF), that can be used to solve the above disorder problem caused by the latency compensation.

Alternatively, Section 8 provides E|D decoupling method to firstly absorb latency deviation E by the pre-scheduler which may maintain FIFO queue per incoming port plus delay level. In this case, packets from the same flow will only determine the damping delay, but not the position, in the FIFO based on latency deviation E, to avoid disorder. Latency deviation E no longer works in the post-scheduler.

Note that in practical situations, two back-to-back packets of the same flow are generally evenly distributed within the burst interval by regulation, which means that the distance between these two packets is generally much greater than the calculation accuracy mentioned above, meaning that the disordered phenomenon will not really occur. For example, the regulated result meets a Length Rate Quotient (LRQ) constraint, and the time interval between two consecutive packets of size l_i and l_j should be at least l_i/r, where r is the flow rate (i.e., the reserved bandwidth of the flow). This can be done by LRQ based regulation, or enhanced leaky bucket based regulation, depending on implementation.

7.4. Option-4: Latency Compensation plus RPQ

As shown in Figure 6, a received packet is inserted to the appropriate RPQ queue with specific CT to meet CT <= Q < CT+CTI when the packet arrived at the scheduler, where Q = D + E - F. Depending on the situation of the accumulated burst arrived at the input port, different packets may have different input latency deviation E. Latency compensation will convert the input ineligible arrivals pattern (if possible) into an eligible arrivals pattern.

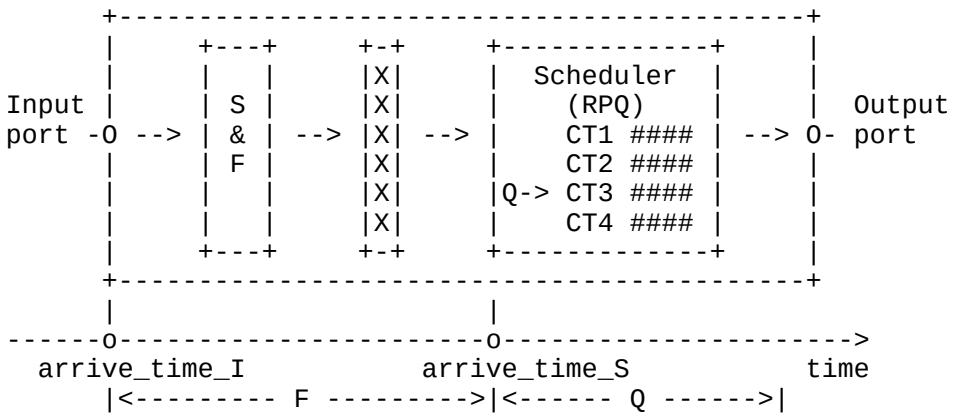

Figure 6: Latency Compensation plus RPQ

The planned residence time (D) and latency deviation (E) should be carried in the packet.

The enqueue and dequeue operations are the same as Section 7.2.

In this option the dequeue rule of on-time mode can guarantee jitter with the help of factor E to absorb jitter per hop. See Section 8 for more information.

Figure 7 depicts an example of packets inserted to the RPQ queues.

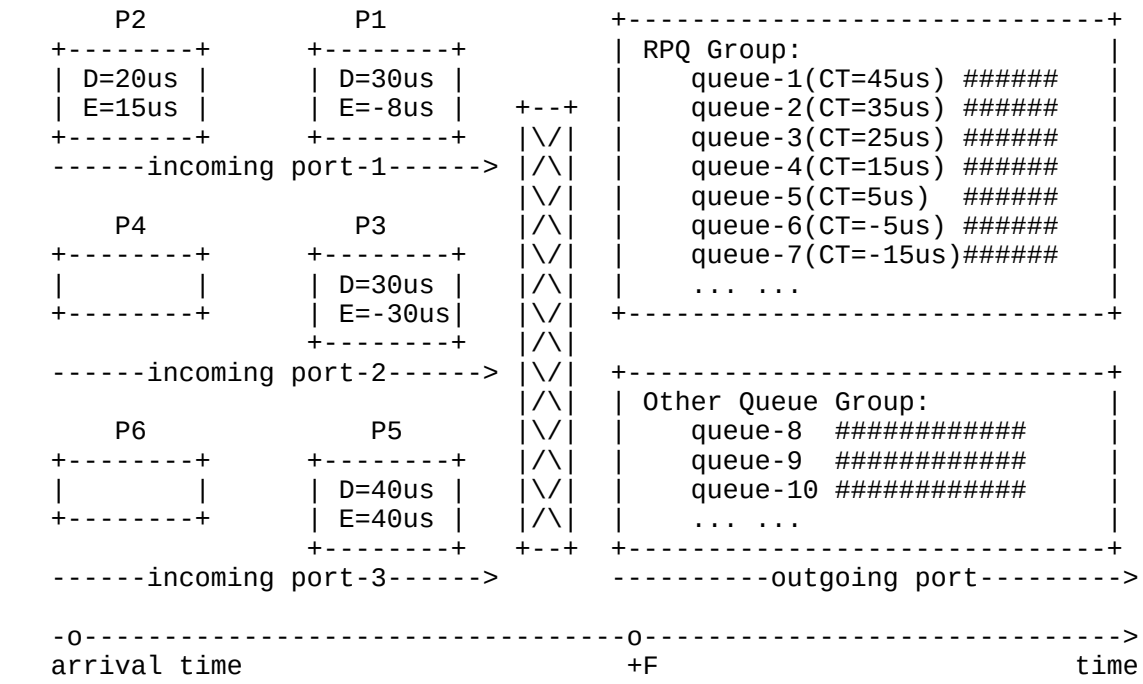

Figure 7: Time Sensitive Packets Inserted to RPQ

As shown in Figure 7, the node successively receives six packets from three incoming ports, among which packet 1, 2, 3 and 5 have corresponding deadline information, while packet 4 and 6 are best-effort packets. These packets need to be forwarded to the same outgoing port. It is assumed that they arrive at the line card where the outgoing port is located at almost the same time after the forwarding delay (F = 5us). At this time, the queue status of the outgoing port is shown in the figure. Then:

* The allowable queueing delay (Q) of packet 1 is 30 - 8 - 5 = 17us, and it will be put into queue-4 (its CT is 15us), meeting the condition that Q is in the range [15, 25).

* The allowable queueing delay (Q) of packet 2 is 20 + 15 - 5 = 30us, and it will be put into queue-3 (its CT is 25us), meeting the condition that Q is in the range [25, 35).

* The allowable queueing delay (Q) of packet 3 is 30 - 30 - 5 = -5us, and it will be put into queue-6 (its CT is -5us), meeting the condition that Q is in the range [-5, 5).

* The allowable queueing delay (Q) of packet 5 in the node is 40 + 40 - 5 = 75us, and the queue it is placed on is not shown in the figure (such as a hierarchical queue).

* Packets 4 and 6 will be put into the non-deadline queue in the traditional way.

According to Section 4.3, An eligible packet (i.e., E = 0) from a specific delay level, even at the end of the inserted queue, can ensure that it does not exceed its deadline, which is the key role of the CTI factor in the condition equation. Now, assuming that a packet is penalized to a lower priority queue based on its positive E, this penalty will not result in more than expected delay, apart from potential delay E.

For example, when a packet is inserted queue based on

$$CT_x <= Q < CT_x + CTI$$

even if it is at the end of the queue, according to D = Q - E, i.e., after time E (the penalty time), then

$$CT_x - E <= Q - E < CT_x - E + CTI$$

That is

$$CT_y <= D < CT_y + CTI$$

So, in essence, it is still equivalent to an eligible packet entering the corresponding queue based on its delay level, and apply the schedulability condition.

7.4.1. Packet Disorder Considerations

Suppose that two packets, P1, P2, are generated instantaneously from a specific flow at the source, and the two packets have the same planned residence time. P1 may face less interference delay than P2 in their journey. When they arrive at an intermediate node in turn, P2 will have less allowable queueing delay (Q) than P1 to try to stay close to P1 again. It should be noted that to compare who is earlier is based on queue’s CT and packet’s Q, according to the above queueing rule (CT <= Q < CT+CTI), and the CT of the queue is not changed in real-time, but gradually with the decreasing step RTI. It is possible to get an unexpected comparison result.

As shown in Figure 8, P1 and P2 are two packets belonging to the same flow. The arrival time when they are received on the scheduler is shown in the figure. Suppose that CTI is 10us, the decreasing step

RTI is 1us, and the transmission time of each packet is 0.01us. Also suppose that the Q values of two adjacent packets P1 and P2 are 40us and 39us respectively, and they are both received in the window from T0 to T0+1us. P1 will enter queue-B with CT range [40, 50), while P2 will enter queue-A with CT range [30, 40) just before the rotation event occurred. This means that P2 will be scheduled before P1, resulting in disorder.

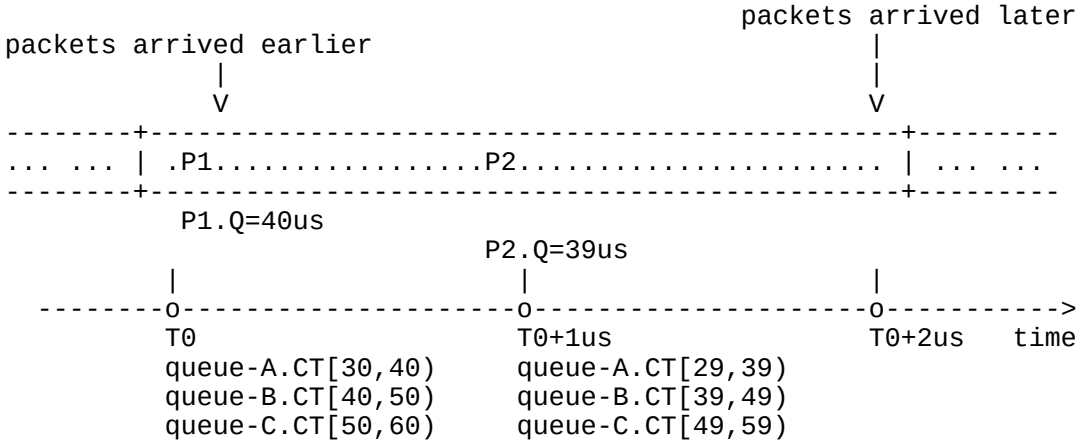

Figure 8: Disorder Illustration of RPQ

DetNet architecture [RFC8655] provides Packet Ordering Function (POF), that can be used to solve the above disorder problem caused by the latency compensation.

Alternatively, Section 8 provides E|D decoupling method to firstly absorb latency deviation E by the pre-scheduler which may maintain FIFO queue per incoming port plus delay level. In this case, packets from the same flow will only determine the damping delay, but not the position, in the FIFO based on latency deviation E, to avoid disorder. Latency deviation E no longer works in the post-scheduler.

Note that in practical situations, two back-to-back packets of the same flow are generally evenly distributed within the burst interval by policing, which means that the distance between these two packets is generally much greater than the calculation accuracy mentioned above, meaning that the disordered phenomenon will not really occur. For example, the regulated result meets a Length Rate Quotient (LRQ) constraint.

8. Jitter Performance by On-time Scheduling

The enqueue and dequeue rule of on-time mode described in Section 7.3 and Section 7.4 will absorb latency deviation E on each hop, and achieve a low jitter. The ultimate E2E jitter depends on the delay experienced on the last node of the flow, which may be from 0 to the delay bound, i.e., the corresponding delay level d_i.

Depending on different methods of absorbing E, there are slight differences in scheduling behavior.

* E+D integration: E will be added to D to get the adjusted D’; The packet is scheduled by the EDF scheduler configured with on-time mode based on D’.

* E|D decoupling: There are 2-tier schedulers; The packet is scheduled by pre-scheduler configured with on-time mode based on E, then scheduled by post-scheduler configured with in-time mode based on D.

In the case of E+D integration, it may explicitly introduce the mandatory hold time, and cause that the actual departure time of the packet may be after its deadline. Assuming that the eligible arrivals pattern of all delay levels causes the scheduler to work at full speed (i.e., service rate C), then for in-time mode, the worst case is that there may be a packet of a specific delay level to be sent just before its deadline during the busy period; While for E+D integration case, the busy period may just start at its deadline and cause the sending time of the packet to exceed its deadline. However, as mentioned above, the worst case of this exceeding value will not exceed the delay level value, which is intuitive because it is equivalent to the situation where the observed packet arrives asynchronously after the delay level value. Note that this exceeding deadline does not accumulate with the number of hops. The E2E latency is in the range [D*hops, D*hops+d_i].

In the case of E|D decoupling, the explicitly mandatory hold time is only contributed by E ensured by the pre-scheduler (configured with on-time mode), and the actual departure time (from the post-scheduler) of the packet will always be before its deadline. Assuming that the eligible arrivals pattern of all delay levels cause the post-scheduler (configured with in-time mode) to work at full speed, for E|D decoupling case, the worst case is that there may be a packet of a specific delay level to be sent just before its deadline during the busy period. The E2E latency is in the range [D*(hops-1), D*(hops-1)+d_i]. Note that the pre-scheduler may maintain a PIFO, an RPQ, or several FIFO queues each for particular "incoming port + delay level". Figure 9 shows the functional entities inside the node.

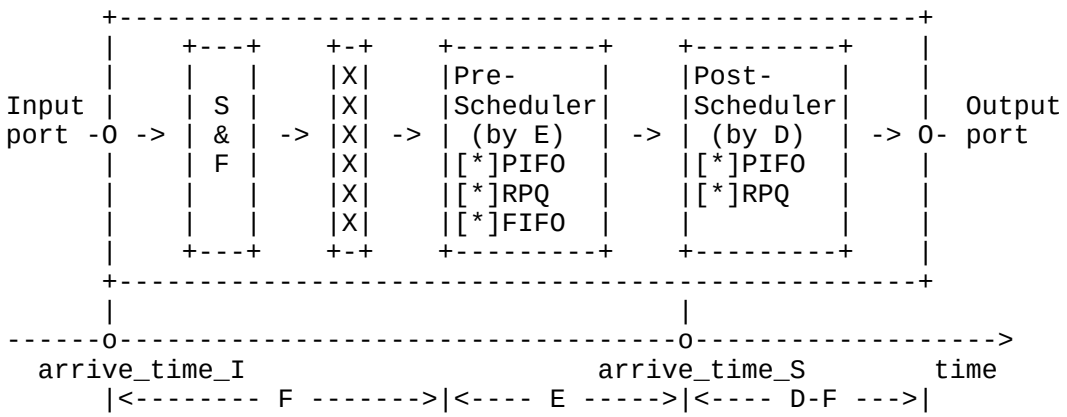

Figure 9: E|D decoupling with 2-tier Schedulers

The following Figure 10 shows the difference between on-time scheduling and in-time scheduling.

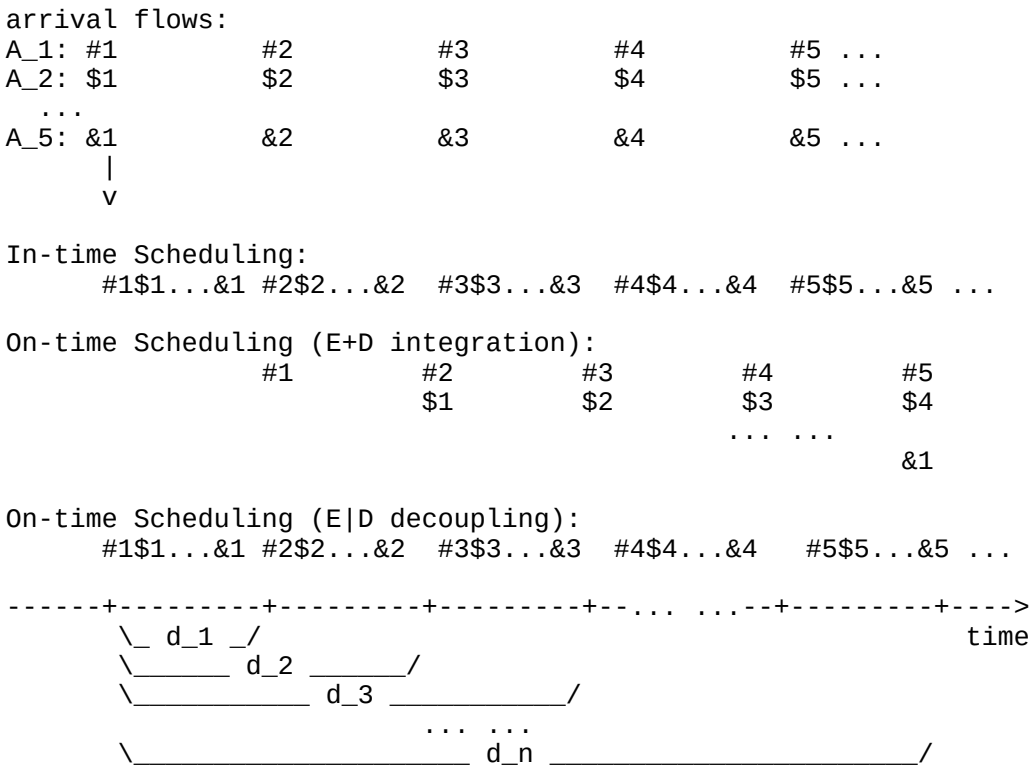

Figure 10: Difference between In-time and On-time Scheduling

As shown in the figure, each burst of A_1 (corresponding to delay level d_1) is termed as #num, each burst of A_2 (corresponding to delay level d_2) as $num, and each burst of A_5 (corresponding to delay level d_5) as &num. A single burst may contain multiple packets. For example, burst #1 may contain several packets, and the actual time interval between #1 and #2 may be small. Although the figure shows the example that the burst interval of multiple flows is the same and the phase is aligned, the actual situation is far from that. However, this example depicts the typical scheduling behavior.

In the in-time scheduling, all concurrent traffic of multiple levels will be scheduled as soon as possible according to priority, to construct a busy period. For example, in the duration d_1, in addition to the burst #1 that must be sent, the burst $1~&1 may also be sent, but the latter is not necessarily scheduled to be sent before the burst #2 as shown in the figure. Here, it can be clearly seen that in-time scheduling cannot guarantee jitter.

While in the case of on-time scheduling with E+D integration option, each burst is scheduled at its deadline, which may just be the begin of the busy period. Because of the scheduling delay, the transmission of the burst will exceed its deadline. The last packet of the burst will face more delay than the first packet. For example, when burst #5 enters the PIFO, it may have the same deadline with bursts from $4 of A_2 to &1 of A_5. When the deadlines of multiple packets are the same, use planned residence time (D) as tiebreaker, i.e., the smaller the D, the smaller the rank. So, #5 send first and may exceed the deadline by one d_1; Then send $4 and may exceed the deadline by one d_2; ...; Finally, send &1 and may exceed the deadline by one d_5.

In the case of on-time scheduling with E|D decoupling, assuming that the latency deviation E for each burst is 0 in the above figure, all concurrent traffic of multiple levels, similar to in-time, will also be scheduled as soon as possible according to priority, to construct a busy period. For example, in the duration d_1, in addition to the burst #1 that must be sent, the burst $1~&1 may also be sent. However, $1~&1 will obtain punishment based on their E on the next node (not shown in the figure).

9. Resource Reservation

Generally, a path may carry multiple DetNet flows with different delay levels. For a certain delay level d_i, the path will reserve some resources from the delay resource pool of the link. The delay resource pool here, as leaky bucket constraint function shown in Section 3.2.1 or Section 4.3.2, is a set of preset parameters that meet the schedulability conditions. For example, the level d_1 has a burst upper limit of b_1 and a bandwidth upper limit of r_1. A path j may allocate partial resources (b_i_j, r_i_j) from the resource pool (b_i, r_i) of the link's delay level d_i. A DetNet flow k that carried in path j, may use resources (b_i_j_k, r_i_j_k) according to its T_SPEC. It can be seen that the values of b_i_j and r_i_j determine the scale of the number of paths that can be supported, while the values of b_i_j_k and r_i_j_k determine the scale of the number of flows that can be supported. The following expression exists.

* sum(b_i_j_k) <= b_i_j, for all flow k over the path j.

* sum(r_i_j_k) <= r_i_j, for all flow k over the path j.

* sum(b_i_j) <= b_i, for all path j through the specific link.

* sum(r_i_j) <= r_i, for all path j through the specific link.

9.1. Delay Resource Definition

The delay resources of a link can be represented as the corresponding burst and bandwidth resources for each delay level. Basically, what delay levels (e.g., 10us, 20us, 30us, etc) are supported by a link should be included in the link capability.

Figure 11 shows the delay resource model of the link. The resource information of each delay level includes the following attributes:

* Delay Bound: Refers to the delay bound intra-node corresponding to this delay level. It is a pre-configuration value.

* Maximum Reservable Bursts: Refers to the maximum amount of bit quota corresponding to this delay level. It is a pre-allocated value or resource limit set based on the schedulability condition.

* Utilized Bursts: Refers to the burst utilization of this delay level.

* Maximum Reservable Bandwidth: Refers to the maximum amount bandwidth corresponding to this delay level. It is a pre-allocated value or resource limit set based on the schedulability condition.

* Utilized Bandwidth: Refers to the bandwidth utilization of this delay level.

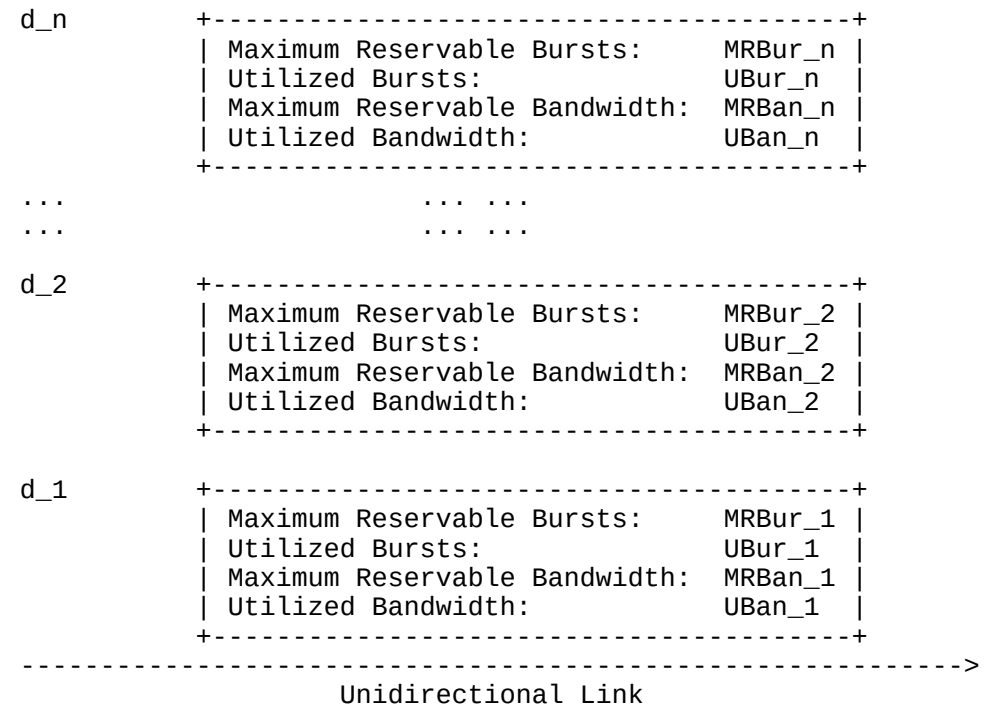

Figure 11: Delay Resource of the Link

For a specific link:

* If Maximum Reservable Bursts and Maximum Reservable Bandwidth are used for schedulability condition checking, they need to set reasonable values at the beginning to meet the schedulability condition, and in the future, there is no need to execute schedulability condition checking during the setup procedure of any flow passing through this link, but only need to check that the aggregated burst and bandwidth of all flows belonging to the same delay level do not exceed Maximum Reservable Bursts and Maximum Reservable Bandwidth, respectively.

* If Utilized Bursts and Utilized Bandwidth are used for schedulability condition checking, there is necessary to execute schedulability condition checking during the setup procedure of any new flows passing through this link.

The IGP/BGP extensions to advertise the link’s capability and delay resource is defined in [I-D.peng-lsr-deterministic-traffic-engineering].

9.2. Traffic Engineering Path Calculation

A candidate path may be selected according to the end-to-end delay requirement of the flow. Subtract the accumulated link propagation delay from the end-to-end delay requirement, and then divide it by the number of hops to obtain the average planned residence time (D) for each node. Or, different nodes may have different planned residence time (D). By default, select the appropriate delay level d_i (d_i <= D-F) closest to the planned residence time (D), and then reserve resources from delay level d_i on each hop. A local policy may allow more larger D to consume resources with smaller delay levels on the control plane. If so, the data packet should also carry a corresponding smaller D.

Note that it is planned residence time (D), not delay level (d_i), carried in the data packet.

9.3. Overprovision Analysis

For each delay level d_i, the delay resource of the specific link is (b_i, r_i). A path j may allocate partial resources (b_i_j, r_i_j) from the resource pool (b_i, r_i). In order to support more d_i flows in the network, it is necessary to set larger b_i and r_i. However, as mentioned earlier, the values of b_i and r_i are set according to schedulability conditions and cannot be set at will.

For bandwidth resource reservation case, the upper limit of the total bandwidth that can be reserved for all aggregated flows of delay level d_i is r_i, which is the same as the behavior of traditional bandwidth resource reservation. There is no special requirement for the measurement interval of calculating bandwidth value.

For the burst resource reservation case, the upper limit of the total burst that can be reserved for all aggregated flows of delay level d_i is b_i. If the burst of each flow of level d_i is b_k, then the number of flows can be supported is b_i/b_k, which is the worst case considering the concurrent arrival of these flows. However, the burst resource reservation is independent of bandwidth resource, i.e., it does not take the calculation result of b_k/d_i to get an overprovision bandwidth and then to affect the reservable bandwidth resources.

By providing multiple delay levels, 100% of the link bandwidth can be allocated to DetNet flows, as can be seen from the schedulability condition equation.

10. Policing on the Ingress

On the ingress PE node, policing must be performed on the incoming port, so that DetNet flow does not violate its T-SPEC. This kind of traffic regulation is usually the shaping using leaky bucket. After policing, the shaped pattern of the DetNet flow may contain discrete multiple bursts evenly distributed within its periodic service burst interval (SBI). For example, An arriving elephant flow will be diluted and released to the EDF scheduler.

According to [RFC9016], the values of Burst Interval, MaxPacketsPerInterval, MaxPayloadSize of the DetNet flow will be written in the SLA between the customer and the network provider, and the flow entrance node will set the corresponding bucket depth according to MaxPayloadSize to forcibly delay the excess bursts. The flow entrance node also sets the corresponding bucket rate according to the promised arrival rate.

The shaped pattern is generally inconsistent with the original arrival pattern of the DetNet flow, and some bursts of the original arrival pattern may experience more shaping delay than others. The shaped pattern and the original arrival pattern can be as consistent as possible by increasing the bucket depth, but this means that the flow will occupy more burst resources, and reduce the service scale that the network can support according to the schedulability conditions.

On the flow entrance node, for the burst with applied shaping delay, shaping delay cannot be included in the latency compensation equation, otherwise, it will make that burst catch up with the previous burst, resulting in damage to the policing result and violation of the arrival constraint function. Please refer to [I-D.peng-detnet-policing-jitter-control] for the elimination of jitter caused by shaping delay on the flow entrance node.

Then, the regulated traffic arrives at the EDF scheduler on the outgoing port. Since the traffic of each delay level meets the leaky bucket arrival constraint function and the parameters of the shaping curve do not exceed the limits of the parameters provided by the schedulability conditions, the traffic can be successfully scheduled based on deadline.

Figure 12 depicts an example of policing and deadline based scheduling on the ingress PE node in the case of option-4 with on-time mode. In the figure, the shaping delay of each burst is termed as S#.

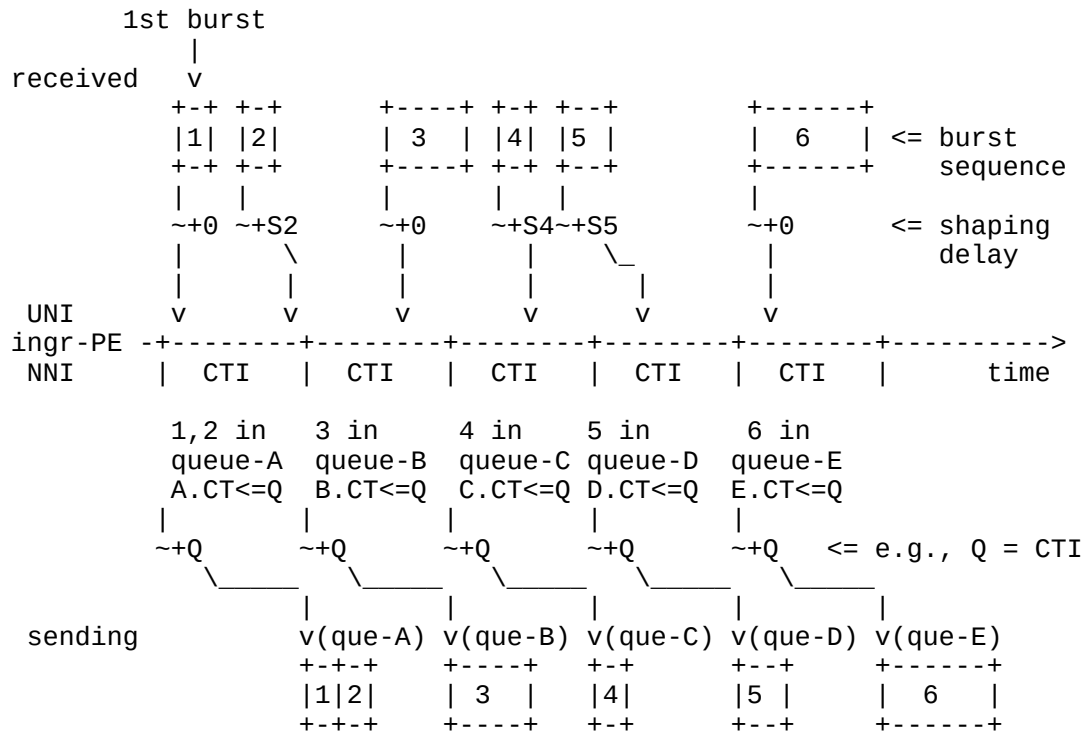

Figure 12: Deadline Based Packets Orchestrating

There are 6 bursts received from the client. The burst-2, 4, 5 has policing delay S2, S4, S5 respectively, due to the consumption of tokens by previous burst. While burst-1, 3, 6 has zero policing delay because the number of tokens is sufficient. The policing makes 6 bursts roughly distributed within the service burst interval.

Assuming that the forwarding delay F experienced by all bursts is 0. In the case of latency compensation plus RPQ, they will have the same allowable queueing delay (Q), regardless of whether they have experienced policing delay before. When the packets of burst-1, 2 arrive at the scheduler, according to CT <= Q < CT+CTI, they will be placed in Queue-A with matched CT and waiting to be sent. Similarly, when the packets of burst-3/4/5/6 arrive at the scheduler, they will be placed in Queue-B/C/D/E respectively and waiting to be sent according to the de-queue rules of on-time mode. Note that each sending burst may get a latency deviation E, especially for burst-2, which is sent closely adjacent to burst-1 in the sending pattern.

11. Compatibility with Legacy Device

Deadline is suitable for end-to-end and interconnection between different networks. A large-scale network may span multiple networks, and one of the goals of DetNet is to connect each network domain to provide end-to-end deterministic delay service. The adoption techniques and capabilities of each network are different, and the corresponding topology models are either piecewise or nested.

For a particular path, if only some nodes in the path upgrade support the deadline based mechanism defined in this document, the end-to-end deterministic delay/jitter target will only be partially achieved. Those legacy devices may adopt the existing SP or WFQ mechanisms, and ignore the possible deadline information carried in the packet, thus the residence delay produced by them cannot be perceived by the adjacent upgraded node. The more upgraded nodes included in the path, the closer to the delay/jitter target. Although, the legacy devices may not support the data plane mechanism described in this document, but they can be freely programmed (such as P4 language) to measure and insert the deadline information into packets, in this case the delay/jitter target may be achieved.

Only a few key nodes are upgraded to support deadline mechanism, which is low-cost, but can meet a flow with relatively loose time sensitive. Figure 13 shows an example of upgrading only several network border nodes. In the figure, only R1, R2, R3 and R4 are upgraded to support deadline based mechanism. A deterministic path across domain 1, 2, and 3 is established, which contains nodes R1, R2, R3, and R4, as well as explicit nodes in each domain. Domain 1, 2 and 3 use the traditional SP mechanism. The encoding of the packet sent by R1 includes the planned residence time and the latency deviation E. Especially, DS filed in IP header ([RFC2474]) are also set to appropriate values. The basic principle of setting is that the less the planned residence time, the higher the priority, to avoid the interference by non DetNet flows.

The delay analysis based on strict priority without re-shaping in each domain can be found in [SP-LATENCY], which gives the equation to evaluate the worst-case delay of each hop. The worst-case delay per hop depends on the number of hops and the burst size of interference flows that may be faced on each hop. [EF-FIFO] also shows that, for FIFO packet scheduling be used to support the EF (expedited forwarding) per-hop behavior (PHB), if the network utilization level alpha < l/(H-l), the worst-case delay bound is inversely proportional to 1-alpha*(H-1), where H is the number of hops in the longest path of the network. Having fewer hops in the SP domain is better.

Although the EDF scheduling with in-time mode, the SP scheduling and EF FIFO scheduling are all work-conserving, the EDF scheduling can further distinguish between urgent and non urgent packets according to deadline information other than traffic class. The operation of dynamically modifying the key fields, i.e., the latency deviation (E), of the packet can avoid always overestimating worst-case latency on all hops just like SP.

For a specific DetNet flow, if it experiences too much latency in the SP domain (due to unreasonable setting of DS field and the inability to distinguish between DetNet and non DetNet flows), even if the border node accelerates the transmission, it may not be able to achieve the target of low E2E latency. If the traffic experiences less latency within the SP domain, the on-time scheduling mode applied on the border node can help achieve the end-to-end jitter target.

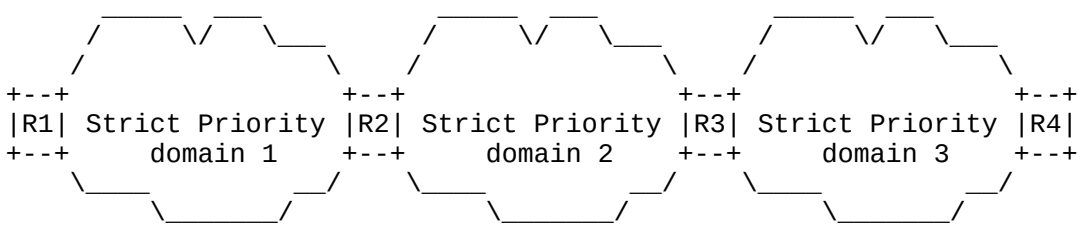

Figure 13: Example of partial upgrade

12. Deployment Considerations

According to the above schedulability conditions, each delay level d_i has dedicated delay resources, and the smaller d_i, the more valuable it is. The operator needs to match the corresponding d_i for each flow. It should be noted that the per-hop latency provided by EDF for the flow is based on flow’s RSpec, not TSpec.

In the case of option-3 and 4 with in-time scheduling behavior, more buffer is required to absorb burst accumulation.

For a specific flow, the accumulated bursts on a intermediate node consists of multiple rounds of burst interval. For example, the packets generated by the source within the first round of burst interval (always experiencing the worst case delay along the path) is caught up by the packets generated within the second round of burst

interval (always experiencing the best case delay along the path). For delay level d_i, the worst case delay is d_i, the best case delay is l/R, where l is the smallest packet size of the flow, R is the port rate. For simplicity to get the estimate size of accumulated bursts, here the best case delay is considered as 0. Drawing on the method provided in [SP-LATENCY], the accumulated bursts of d_i is:

* ACC_BUR_i = ((d_i * h) / burst_interval) * b_i

For example, d_i is 10 us, burst_interval is 250 us, this means that within the 25th hop, there will only be one b_10 burst in the queue. If it exceeds 25 hops and is within 50 hops, there may be two b_10 burst simutaneously in the queue.

The accumulated bursts of other delay levels can be similarly estimated. Operators need to evaluate the required buffer size based on network hops and the supported delay levels. The benefit of in-time scheduling is to obtain an E2E latency of no more than D*hops as small as possible.

Operators may also apply on-time scheduling per hop to simplify the design of buffers. On-time scheduling absorbed latency deviation E on each hop and can get a jitter for each delay level to the value of delay level in theory (i.e., the worst case is that on the last node there are full flows contributed by all delay levels that are discharging floodwater at the same time, however, in reality, the DetNet flow of the output port facing the destination customer side may only involve one delay level, then the jitter may be only one CTI (e.g., 10us)).

In summary, the in-time scheduling with latency compensation, can suffer from the uncertainty caused by burst accumulation, and it is recommended only deployed in small networks, i.e., a limited domain with a small number of hops, where the burst accumulation issue is not serious; The on-time scheduling per hop is recommended to be used in large networks.

On-time scheduling has an additional cost of pre-scheduler component compared to in-time scheduling. Operators may enable in-time scheduling on intermediate devices and enable on-time scheduling on network exit devices to achieve the goal of low jitter of EDF path. In this case, the local policy of the intermediate device should allow the use of in-time scheduling for the packets that actually require on-time service.

13. Evaluations

The following summarizes how the deadline-based mechanism ensures bounded latency and jitter:

1) Partition delay resource for each delay level on the outgoing port according to the schedulability condition, i.e., preset parameters of the arrival constraint function.

2) Reserve delay resource on each link of the calculated path for the flow to be set up. This step is also known as admission condition check.

3) Execute policing on the flow entrance node, to let the admitted flow obey its constraint function (i.e., TSpec).

4) Execute reshaping or latency compensation (recommended) in the network core for each flow, to convert the ineligible arrivals to eligible arrivals that still obey the constraint function of each flow.

5) Guarantee bounded delay by in-time scheduling mode; Guarantee bounded delay and jitter by on-time scheduling mode.

13.1. Large Scaling Requirements Matching Degree

The following table is the evaluation results based on the requirements that is defined in [I-D.ietf-detnet-scaling-requirements]. Note that all asynchronous mechanisms (such as EDF, ATS) do not require complete synchronization of crystal oscillator frequencies between devices. The latency error caused by the deviations of clocks from their nominal rates, e.g., +100ppm, is generally in the nanosecond range and can be ignored.

requirements	Evaluation	Notes
3.1 Tolerate Time Asynchrony	Yes	No time sync needed; No frequency sync needed.
3.2 Support Large Single-hop Propagation Latency	Yes	The eligible arrival of flows is independent with the link propagation delay.
3.3 Accommodate the Higher Link	Partial	The higher service rate, the more buffer needed for each

Speed		delay level. And, extra instructions to calculate E.
3.4 Be Scalable to the Large Number of Flows and Tolerate High Utilization	Yes	Limited number of delay levels are mapped by lots of flows. No overprovision in the resource reservation. Utilization may reach 100% link bandwidth. The unused bandwidth of the high delay level can be used by the low levels or BE flows.
3.5 Tolerate Failures of Links or Nodes and Topology Changes	N/A	Independent of queueing mechanism.
3.6 Prevent Flow Fluctuation	Yes	Flows are permitted based on the resources reservation of delay levels, and isolated from each other.
3.7 Be scalable to a Large Number of Hops with Complex Topology	Yes	E2E latency is liner with hops, from ultra-low to low latency by multiple delay levels. E2E jitter is low by on-time scheduling.
3.8 Support Multi-Mechanisms in Single Domain and Multi-Domains	N/A	Independent of queueing mechanism.

Figure 14: Evaluation for Large Scaling Requirements

13.2. Taxonomy Considerations

[I-D.ietf-detnet-dataplane-taxonomy] provides criteria for classifying data plane solutions.

For performance, the per hop latency dominant factor of EDF is the delay levels that is defined according to schedulability condition.

For functional characteristics, EDF is non-periodic, class level for traffic granularity, and right-bounded or bounded for time Bounds.

* Non-periodic: The scheduling power of an EDF is measured over an arbitrarily long non repetitive time range, scheduling in an orderly manner according to the urgency of the packets, and there is no defined periodic quantification unit of scheduling power.

* Class level: DetNet Flows can be grouped by similar service requirements, i.e., delay levels provided in the network. Packets will be provided EDF service based on delay level, without checking flow identification.

* Right-bounded/bounded: A packet’s deadline is defined as its maximum time bound. So EDF with in-time mode is right-bounded. While EDF with on-time mode, due to further limiting the minimum time bound, is bounded.

[I-D.ietf-detnet-dataplane-taxonomy] also specifies the suitable categories of solutions for DetNet. According to the above functional characteristics, EDF with in-time mode will map to right-bounded category, and EDF with on-time mode will map to class level non-periodic bounded category.

13.3. Examples

This section describes the example of how the deadline mechanism supports DetNet flows with different latency requirements.

13.3.1. Heavyweight Loading Example

This example observes the service scale and different latency bound supported by the deadline mechanism in the heavyweight loading case.

Figure 15 provides a typical reference topology that serves to represent or measure the multihop jitter and latency experience of a single "flow i" across N hops (in the figure, N=10). On each of the N outgoing interfaces (represented by circles in the figure), "flow i" has to compete with different flows (represented by different symbols on each hop). Especially, the competed flows arrive simultaneously at multiple incoming ports, with the same starting time when measuring their respective residence time. The characteristic of this reference topology is that every link that "flow i" passes through may be a bottleneck link with 100% network utilization, causing "flow i" to achieve the worst-case latency on each hop.

As shown in Figure 15:

* Network transmission capacity: each link has rate 10 Gbps. Assuming the service rate of EDF scheduler allocate the total port bandwidth.

* TSpec of each flow, maybe:

  - burst size 1000 bits, and average arrival rate 1 Mbps.

  - or, burst size 1000 bits, and average arrival rate 10 Mbps.

  - or, burst size 1000 bits, and average arrival rate 100 Mbps.

* RSpec of each flow, maybe:

  - E2E latency 100us, and E2E jitter less than 10us or 100us.

  - or, E2E latency 200us, and E2E jitter less than 20us or 200us.

  - or, E2E latency 300us, and E2E jitter less than 30us or 300us.

  - or, E2E latency 400us, and E2E jitter less than 40us or 400us.

  - or, E2E latency 500us, and E2E jitter less than 50us or 500us.

  - or, E2E latency 600us, and E2E jitter less than 60us or 600us.

  - or, E2E latency 700us, and E2E jitter less than 70us or 700us.

  - or, E2E latency 800us, and E2E jitter less than 80us or 800us.

  - or, E2E latency 900us, and E2E jitter less than 90us or 900us.

  - or, E2E latency 1ms, and E2E jitter less than 100us or 1ms.

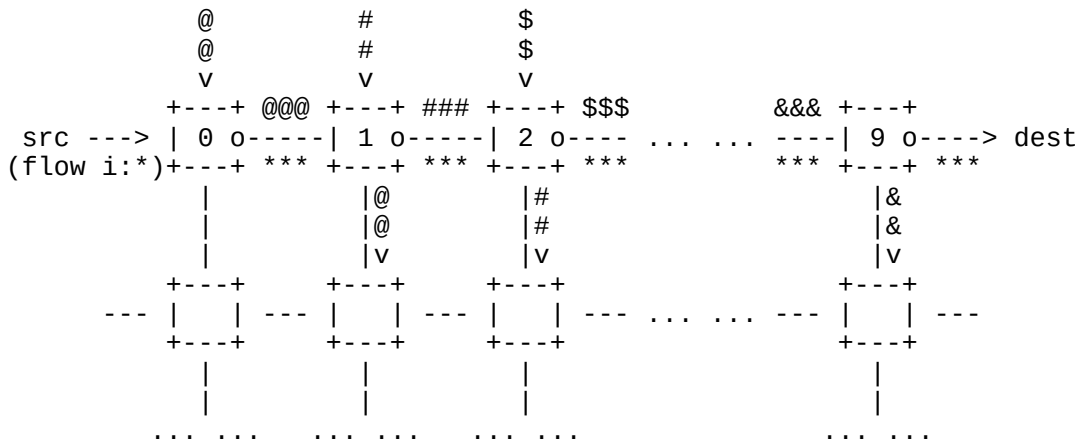

Figure 15: Heavyweight Loading Topology Example

For the observed flow i (marked with *), its TSpec and RSpec may be any of the above. Assuming that the path calculated by the controller for the flow i passes through 10 nodes (i.e., node 0~9). Especially, at each hop, flow i may conflict with other DetNet flows, also with similar TSpec and RSpec as above, originated from other sources, e.g., conflicts with flow-set "@" at node 0, conflicts with flow-set "#" at node 1, and so on.

For each link along the path, it may provide multiple delay levels, e.g., d1 (10us), d2 (20us), ..., d10 (100us). Assuming no link propagation delay and intra-node forwarding delay. If flow i uses d1 resources, it can ensure an E2E latency of 100us (i.e., d1 * 10 hops), and jitter of 10us(on-time mode) or 100us (in-time mode). The results of using resources of other delay levels are similar.

The table below shows the possible tight allocation of delay resources on each link based on Equation-1, as well as the corresponding service scale supported, where, b = utilized burst resource (K bits), r = utilized bandwidth resource (Mbps), s = service scale (number), assuming that the resource limit of each delay level is b_limit = 100000 bits, r_limit = 1 Gbps.

Note that in the table each row only shows the data where all flows served by all delay levels have the same TSpec (e.g., in row-1, TSpec per flow is burst size 1000 bits and arrival rate 1 Mbps), while in reality, flows served by different delay levels generally have different TSpec. It is easy to add rows to describe various combinations.

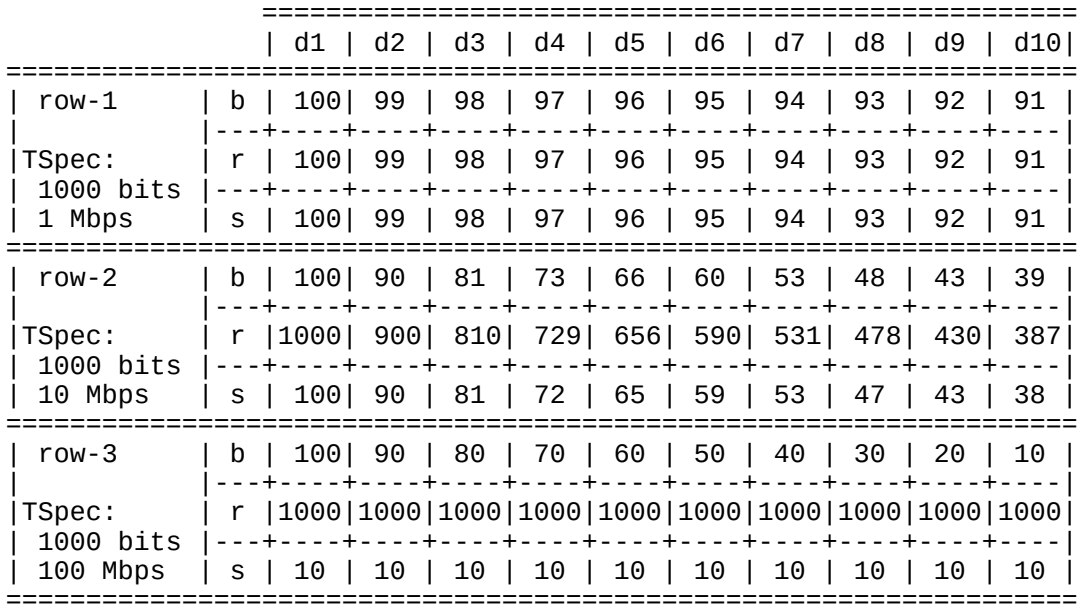

Figure 16: Delay Resource Pool and Service Scale Example

13.3.2. Lightweight Loading Examples

The following examples observe how the preset service scale is supported and mapped to different delay levels by the deadline mechanism in the lightweight loading case.

In these examples, the network only contains a small number of bottleneck links with low network utilization, and it can be considered as the lightweight loading case of Figure 15. Lightweight loading usually means having a smaller calculated worst-case latency per hop, or the actual latency experienced doesn’t reach the worst-case latency.

13.3.2.1. Grid Reference Topology

[I-D.ietf-detnet-dataplane-taxonomy] describes a Grid topology (Figure 17) with partial mesh. Three flow types, i.e., audio, video, and CC (Command and Control) are considered to require deterministic networking services. Among them, audio and CC flows consume less bandwidth (1.6 Mbps per flow and 0.48 Mbps per flow respectively) but both require lower E2E latency (5ms), while video flows consume more bandwidth (11 Mbps per flow) but can tolerate larger E2E latency (10ms).

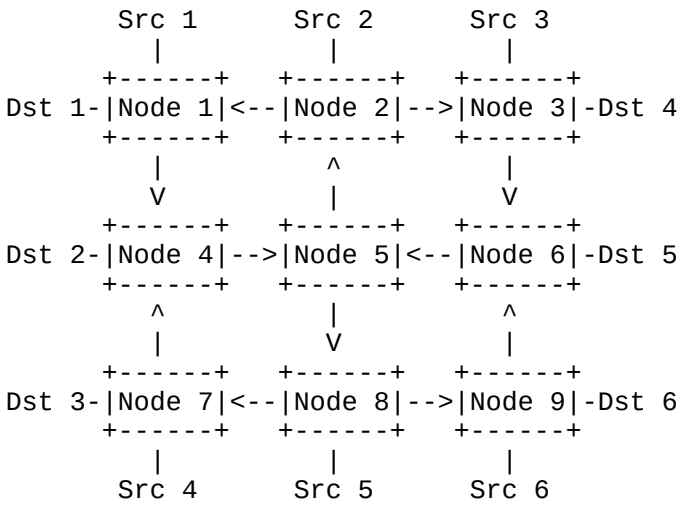

Figure 17: Grid Reference Topology

According to the preset rules that generate a unique route for every source and destination pair, the details of all paths are as follows:

Src1-1-Dst1
Src1-1-4-Dst2
Src1-1-4-5-8-7-Dst3
Src1-1-4-5-2-3-Dst4
Src1-1-4-5-8-9-6-Dst5
Src1-1-4-5-8-9-Dst6

Src2-2-1-Dst1
Src2-2-1-4-Dst2
Src2-2-3-6-5-8-7-Dst3
Src2-2-3-Dst4
Src2-2-3-6-Dst5
Src2-2-3-6-5-8-9-Dst6

Src3-3-6-5-2-1-Dst1
Src3-3-6-5-2-1-4-Dst2
Src3-3-6-5-8-7-Dst3
Src3-3-Dst4
Src3-3-6-Dst5
Src3-3-6-5-8-9-Dst6

```
Src4-7-4-5-2-1-Dst1
Src4-7-4-Dst2
Src4-7-Dst3
Src4-7-4-5-2-3-Dst4
Src4-7-4-5-8-9-6-Dst5
Src4-7-4-5-8-9-Dst6

Src5-8-7-4-5-2-1-Dst1
Src5-8-7-4-Dst2
Src5-8-7-Dst3
Src5-8-7-4-5-2-3-Dst4
Src5-8-9-6-Dst5
Src5-8-9-Dst6

Src6-9-6-5-2-1-Dst1
Src6-9-6-5-2-1-4-Dst2
Src6-9-6-5-8-7-Dst3
Src6-9-6-5-2-3-Dst4
Src6-9-6-Dst5
Src6-9-Dst6
```

Where, flows to destination Dst1 and Dst6 are audio flows, flows to destination Dst2 and Dst5 are CC flows, and flows to destination Dst3 and Dst4 are video flows. Each path carries 10 flows, e.g., the path "Src1-1-Dst1" carries 10 audio flows. It can be seen that the longest path contains 7 hops, and the bottleneck link involves link (2-3) and link (8-7), both of which have 10 audio flows, 60 video flows, and 10 CC flows.

According to the longest path and the expected E2E latency, the per-hop latency bound for each type of flow can be estimated, i.e., 700us for audio and CC flows, 1400us for video flows. This means that the deadline mechanism needs to provide appropriate delay levels, and the delay level mapped by audio and CC flows cannot be larger than 700us, and the delay level mapped by video flows cannot be larger than 1400us.

For simplicity, a unified delay resource pool is configured on each link in the network, although different links can indeed be configured differently. This unified delay resource pool must meet the resource allocation requirements on both bottleneck and non-bottleneck links, so the load is slightly increased and it is assumed that the number of each type of flows on a link reached 60. Figure 18 shows a possible delay resource pool and the corresponding delay levels mapped by flows. Note that there are other possible resource pool designs as long as they meet schedulability conditions.

Delay Levels	Bursts (Kbits)	Bandwidth (Mbps)	Services Mapped
d1 (100 us)	b1 = 40	r1 = 10	
d2 (200 us)	b2 = 144	r2 = 30	CC
d3 (300 us)	b3 = 0	r3 = 0	
d4 (400 us)	b4 = 0	r4 = 0	
d5 (500 us)	b5 = 0	r5 = 0	
d6 (600 us)	b6 = 0	r6 = 0	
d7 (700 us)	b7 = 120	r7 = 96	Audio
d8 (800 us)	b8 = 0	r8 = 0	
d9 (900 us)	b9 = 0	r9 = 0	
d10 (1000 us)	b10 = 0	r10 = 0	
d11 (1100 us)	b11 = 720	r11 = 660	Video

Figure 18: Delay Resource Pool and Service Mapped

Where, the granularity of delay level is chosen at the level of 100 us based on the link capability (1 Gbps) and the concurrent bursts (984 Kbits) of three type of flows. Intuitively, if the link capability is larger, such as 10 Gbps, the granularity can be chosen to be smaller, such as at the level of 10us.

The maximum delay level d11 (1100 us) is selected according to the minimum regulated packet interval of any flow, i.e., d11 is not larger than any regulated packet interval of any flow. In this example, the regulated packet interval (i.e., packet_size / service_rate) for flows audio, video, and CC are 1.25 ms, 1.1 ms, and 5 ms, respectively.

All delay levels consume approximately 800 Mbps bandwidth due to slightly increasing the loading. 60 CC flows are mapped to delay level d2, 60 audio flows are mapped to delay level d7, and 60 video flows are mapped to delay level d11. The delay level d1 may be used for more urgent flows other than the three types of flows considered.

For example, on the bottleneck link (2-3), 10 audio flows will allocate <burst = 20 Kbits, bandwidth = 16 Mbps> from d7, 60 video flows will allocate <burst = 720 Kbits, bandwidth = 660 Mbps> from d11, and 10 CC flows will allocate <burst = 24 Kbits, bandwidth = 5 Mbps> from d2.

For example on the non-bottleneck link (8-9), 50 audio flows will allocate <burst = 100 Kbits, bandwidth = 80 Mbps> from d7, zero video flows will not allocate resources from d11, and 30 CC flows will allocate <burst = 72 Kbits, bandwidth = 15 Mbps> from d2.

If on-time mode is applied, each packet of the audio flow may experience per-hop latency 700 us, and each packet of the CC flow may experience per-hop latency 200 us, and each packet of the video flow may experience per-hop latency 1100 us.

If in-time mode is applied, the best per-hop latency experienced by a packet in any flow may be 0 (without considering intra-node forwarding delay F), and the theoretical worst-case latency may be the same as that in the on-time mode in the case of heavyweight loading. However, due to lightweight loading in this example, smaller worst-case latency can be achieved. For example, on the bottleneck link (2-3), the admitted burst aggregation is composed of CC 24 Kbits, audio 20 kbits, and video 720 Kbits in descending order of transmission priority, therefore, the worst-case per-hop latency experienced by the last packet of flows CC, audio, and video, is 24 us, 44 us, and 764 us, respectively, which are much smaller than the values in the on-time mode. Similarly, on the non-bottleneck link (8-9), the admitted burst aggregation is composed of CC 72 Kbits and audio 100 kbits, in descending order of transmission priority, therefore, the worst-case per-hop latency experienced by the last packet of flows CC and audio is 72 us and 172 us respectively, which are also much smaller than the values in the on-time mode.

NOTE:

* In the above process of resource allocation, the 10 flows carried on each path are individually allocated burst resources. This is the most general case, that is, although the 10 flows share the same path, they are assumed to be independent of each other. However, in some cases, if these 10 flows are treated as a macro flow and policing is executed at the flow entrance node for the macro flow (the leaky bucket depth is still the maximum packet size, but the leaky bucket rate is the aggregation rate), and resources are reserved for the macro flow instead of the member flow, then less burst resources will be consumed and larger service scales can be supported.

* This example conforms to the scenarios described in Section 3.2.1 and Section 4.3.2 for the application of simplified schedulability condition where d_n is not larger than any regulated packet interval of any flow. Therefore, in Figure 18 there are remaining 76 Kbits bursts available for any other delay level d_i, to support more flows.

* Video flows have 30 back-to-back packets per single burst, and are being regulated on the flow entrance node, to support 60 video flows on each link. As discussed in Section 10, operators may increase the bucket depth for video flows to make the shaped pattern and the original arrival pattern as consistent as possible, but this will be harmful to service scale. There is a trade-off between burstiness, policing, and service scale.

13.3.2.2. Ring-Mesh Reference Topology

[I-D.ietf-detnet-dataplane-taxonomy] describes another hierarchical Ring-Mesh topology (Figure 19), where, node 1~9 are core routers, and each leaf group consists of 10 ring networks. Each ring network (Figure 20) has 8 nodes, with one node connected to the core by a separate inter-domain link.

The capacity of all the links in the core network is 10 Gbps. The capacity of all the links in the leaf network, including the inter-domain link, is 1 Gbps.

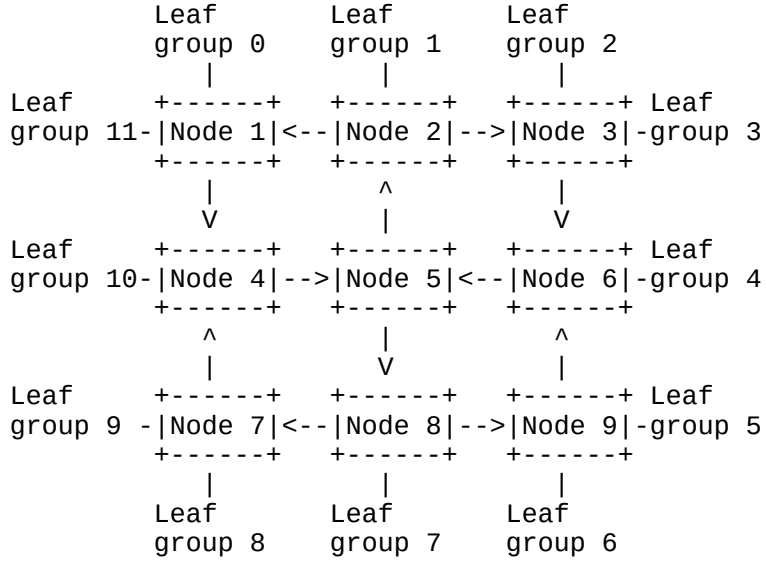

Figure 19: Hierarchical Ring-Mesh Topology

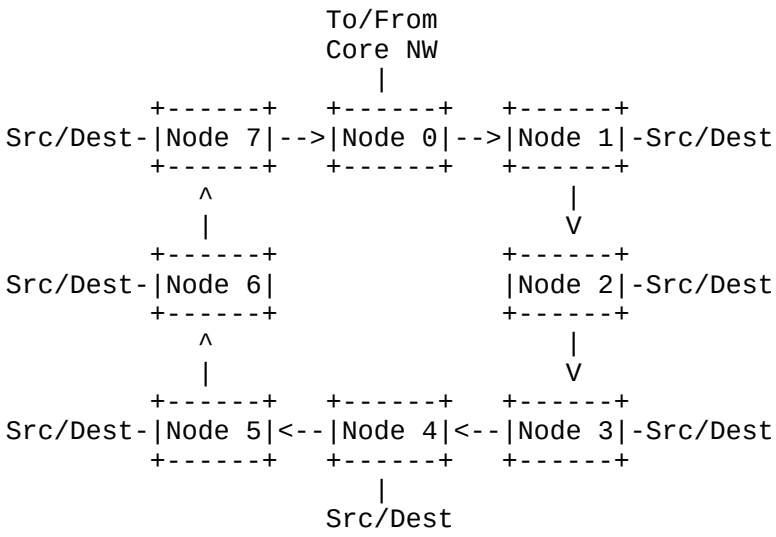

Figure 20: Ring Network

Again, three flow types, i.e., audio, video, and CC (Command and Control) are considered to require deterministic networking services, whose TSpec and RSpec are consistent with the above Grid example.

A flow-set is defined that includes 7 audio flows, 7 video flows, and 32 CC flows, all of which share the same DetNet path. For example, node 1 in the source ring may send a flow-set to node 7 in the destination ring, and the DetNet path may be, 1-2-3-4-5-6-7-0 (source ring), inter-domain link, core, inter-domain link, 0-1-2-3-4-5-6-7 (destination ring).

The longest DetNet path may be 20 hops, where, 7 hops in the source ring, 7 hops in the destination ring, 2 inter-domain hops, and 4 hops in the core.

The preset routing of DetNet path is that, every flow-set in a ring network travels from node i to node (i+7)mod8, and each leaf group i sends n flow-sets (e.g., n = 10, if a leaf group contains 10 ring networks) to the leaf group (i+6)mod12.

Take a flow-set from the source ring to the destination ring as the observed flow-set. The observed flow-set will compete with other 6 flow-sets in the ring, and compete with more flow-sets (coming from other leaf groups) in the core. Note that there is no competition on the inter-domain link.

In this example, it is no longer assumes that every packet of all flow-sets, including the observed flow-set and the competed flow-sets, arrives simultaneously. Although assuming extremely high concurrency can accommodate any topology with some actual concurrency, it underestimate the service scale that can be admitted. In fact, in the ring network, the concurrency at each hop is that only two input interfaces compete for one output interface. For inter-domain links, concurrency is even zero. In the core network, concurrency is also limited. By utilizing the knowledge of concurrency, more reasonable delay levels can be chosen to serve all flows.

In the ring network, on each hop, a bad flow interleaving is that there are two bursts competing for the outgoing interface. Their sizes are 1 flow-set and 6 flow-sets, respectively. The resolved size is 1 flow-set. Assign audio, video and CC to a single delay level d1. The resolved size of d1 is 174800 bits by 7 audio, 7 video, and 32 CC packets, introducing a maximum queueing delay of 174.8 us. The chosen d1 must not be less than 174.8 us. Considering that the transmission time of all bursts (i.e., all audio, video, and CC packets of 7 flow-sets) is 1.2236 ms, and the upcoming next round of 7 flow-sets will be the video packets after 1.1 ms and the audio packets after 1.25ms (with the resolved size 98000 bits and queueing delay 98 us), it can be seen that the transmission of current round of bursts will postpone the transmission of the upcoming next round of bursts, resulting in a postponement delay of 123.6 us (i.e., 1.2236 ms minus 1.1 ms), introducing a maximum queueing delay of 221.6 us (i.e., 123.6 us plus 98 us) for the next round of bursts. Similarly, walking through the following periodic rounds, it can be seen that the queuing delay will not exceed the above maximum value. Therefore, d1 (225 us) can be chosen for all flows in the ring network, with burst resource 1223600 bits and bandwidth 725 Mbps by 49 audio, 49 video, and 224 CC flows. Note that according to schedulability condition, the burst resource of d1 will affect the bounded delay of other delay levels with lower priority (if have).

For simplicity, a unified delay resource pool is configured on each link in the ring network, as shown in the following Figure 21.

Delay Levels	Bursts (Kbits)	Bandwidth (Mbps)	Services Mapped
d1 (225 us)	b1 = 1223.6	r1 = 725	CC/Audio/Video

Figure 21: Delay Resource Pool and Service Mapped in the Ring

In the core network, the details of all DetNet paths are as follows:

group0 -1-4-5-8-9- group6
group1 -2-1-4-5-8- group7
group2 -3-6-5-8-7- group8
group3 -3-6-5-8-7- group9
group4 -6-5-2-1-4- group10
group5 -9-6-5-2-1- group11
group6 -9-6-5-2-1- group0
group7 -8-7-4-5-2- group1
group8 -7-4-5-2-3- group2
group9 -7-4-5-2-3- group3
group10 -4-5-2-3-6- group4
group11 -1-4-5-8-9- group5

Where, the bottleneck link-4-5 will carry 70 flow-sets, in which, 10 flow-sets each from separate inter-domain link, 30 flow-sets from node 1, and 30 flow-sets from node 7.

Another bottleneck link-5-2 will also carry 70 flow-sets, in which, 30 flow-set from node 6, and 40 flow-sets from node 4.

On the bottleneck link-4-5, a bad flow interleaving is that there are 12 bursts competing for the outgoing interface. Their sizes are 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 30, and 30 flow-sets respectively. The resolved size is 40 flow-sets. Assign audio and CC to delay level d1 with high priority, and video to delay level d2 with low priority. The resolved size of d1 is 36320000 bits by 280 audio and 1280 CC packets, introducing a maximum queueing delay of 363.2 us. The resolved size of d2 is 69920000 bits by 280 audio, 1280 CC and 280 video packets, introducing a maximum queueing delay of 699.2 us. The chosen d1 and d2 must be larger than 363.2 and 699.2 us respectively. Considering that the transmission time of 12 incoming bursts (i.e., all audio, video, and CC packets of 70 flow-sets) is 1.2236 ms, and the upcoming next round of 70 flow-sets will be the video packets after 1.1 ms (with the resolved size 3360000 bits and queueing delay 336 us) and the audio packets after 1.25ms (with the resolved size 560000 bits and queueing delay 56 us), it can be seen that the transmission of current round of bursts will postpone the upcoming next round of bursts, resulting in a postponement delay of 123.6 us

(i.e., 1.2236 ms minus 1.1 ms), introducing a maximum queueing delay of 179.6 us (i.e., 123.6 us plus 56 us) for the next round of audio, and a maximum queueing delay of 557.6 us (i.e., 123.6 us plus 336 us, and plus 98 us by 490 audio packets) for the next round of video. Similarly, walking through the following periodic rounds, it can be seen that the queuing delay will not exceed the above maximum value. Therefore, in the core network, d1 (370 us) can be chosen for all audio and CC flows, with burst resource 6356000 bits and bandwidth 1860 Mbps by 490 audio and 2240 CC flows, and, d2 (700 us) can be chosen for all video flows, with burst resource 5880000 bits and bandwidth 5390 Mbps by 490 video flows.

The above chosen d1, d2 can also work for bottleneck link-5-2. For simplicity, a unified delay resource pool is configured on each link in the core network, with slightly increasing the loading and assuming that each link will carry 70 flow-sets, although different links can indeed be configured differently.

Figure 22 shows the delay resource pool and the corresponding delay levels mapped by flows in the core network.

Delay Levels	Bursts (Kbits)	Bandwidth (Mbps)	Services Mapped
d1 (400 us)	b1 = 6356	r1 = 1860	CC/Audio
d2 (700 us)	b2 = 5880	r2 = 5390	Video

Figure 22: Delay Resource Pool and Service Mapped in the Core

Other explanations are similar to the previous example of Grid reference topology. A noteworthy difference from the previous example is that the same flow is mapped to different delay levels between the ring domain and core domain to flexibly adapt to the large difference of service scale in these two domains.

14. IANA Considerations

There is no IANA requestion for this document.

15. Security Considerations

Security considerations for DetNet are described in detail in [RFC9055]. General security considerations for the DetNet architecture are described in [RFC8655]. Considerations specific to the DetNet data plane are summarized in [RFC8938].

Adequate admission control policies should be configured in the edge of the DetNet domain to control access to specific delay resources. Access to classification and mapping tables must be controlled to prevent misbehaviors, e.g., an unauthorized entity may modify the table to map traffic to an expensive delay resource, and competes and interferes with normal traffic.

16. Acknowledgements

The authors appreciate Alexej Grigorjew for his guidance on queueing algorithms, and appreciate David Black, Jinoo Joung, Toerless Eckert, Xuesong Geng, Bin Tan, and Aihua Liu for their insightful comments and productive discussion that helped to improve the document.

17. References

17.1. Normative References

[I-D.ietf-detnet-dataplane-taxonomy]
Joung, J., Geng, X., Peng, S., and T. T. Eckert, "Dataplane Enhancement Taxonomy", Work in Progress, Internet-Draft, draft-ietf-detnet-dataplane-taxonomy-03, 2 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-detnet-dataplane-taxonomy-03>.

[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-08, 1 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-detnet-scaling-requirements-08>.

[I-D.p-6man-deterministic-eh]
Peng, S., "Deterministic Source Route Header", Work in Progress, Internet-Draft, draft-p-6man-deterministic-eh-01, 10 October 2024, <https://datatracker.ietf.org/doc/html/draft-p-6man-deterministic-eh-01>.

[I-D.pb-6man-deterministic-crh]
Peng, S. and R. Bonica, "Deterministic Routing Header", Work in Progress, Internet-Draft, draft-pb-6man-deterministic-crh-01, 10 October 2024, <https://datatracker.ietf.org/doc/html/draft-pb-6man-deterministic-crh-01>.

[I-D.peng-6man-deadline-option]
Peng, S., Tan, B., and P. Liu, "Deadline Option", Work in Progress, Internet-Draft, draft-peng-6man-deadline-option-01, 11 July 2022, <https://datatracker.ietf.org/doc/html/draft-peng-6man-deadline-option-01>.

[I-D.peng-6man-delay-options]
Peng, S., "Delay Options", Work in Progress, Internet-Draft, draft-peng-6man-delay-options-00, 18 January 2024, <https://datatracker.ietf.org/doc/html/draft-peng-6man-delay-options-00>.

[I-D.peng-detnet-policing-jitter-control]
Peng, S., Liu, P., and K. Basu, "Mechanism to control jitter caused by policing in Detnet", Work in Progress, Internet-Draft, draft-peng-detnet-policing-jitter-control-01, 8 October 2024, <https://datatracker.ietf.org/doc/html/draft-peng-detnet-policing-jitter-control-01>.

[I-D.peng-lsr-deterministic-traffic-engineering]
Peng, S., "IGP Extensions for Deterministic Traffic Engineering", Work in Progress, Internet-Draft, draft-peng-lsr-deterministic-traffic-engineering-03, 23 December 2024, <https://datatracker.ietf.org/doc/html/draft-peng-lsr-deterministic-traffic-engineering-03>.

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

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

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[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>.

[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>.

[RFC9016] Varga, B., Farkas, J., Cummings, R., Jiang, Y., and D. Fedyk, "Flow and Service Information Model for Deterministic Networking (DetNet)", RFC 9016, DOI 10.17487/RFC9016, March 2021, <https://www.rfc-editor.org/info/rfc9016>.

[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>.

[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>.

17.2. Informative References

[CQ-EDF] "Programmable Calendar Queues for High-speed Packet Scheduling", 2020, <https://dl.acm.org/doi/10.5555/3388242.3388292>.

[EDF-algorithm] "A framework for achieving inter-application isolation in multiprogrammed, hard real-time environments", 1996, <https://ieeexplore.ieee.org/document/896011>.

[EF-FIFO] "Fundamental Trade-Offs in Aggregate Packet Scheduling", 2001, <https://ieeexplore.ieee.org/document/992892>.

[IR-Theory] "A Theory of Traffic Regulators for Deterministic Networks with Application to Interleaved Regulators", 2018, <https://ieeexplore.ieee.org/document/8519761>.

[Jitter-EDF]
"Delay Jitter Control for Real-Time Communication in a Packet Switching Network", 1991, <https://ieeexplore.ieee.org/document/152873>.

[Net-Calculus]
"Network Calculus: A Theory of Deterministic Queuing Systems for the Internet", 2001, <https://leboudec.github.io/netcal/latex/netCalBook.pdf>.

[P802.1DC] "Quality of Service Provision by Network Systems", 2023, <https://1.ieee802.org/tsn/802-1dc/>.

[PIFO] "Programmable Packet Scheduling at Line Rate", 2016, <https://dl.acm.org/doi/pdf/10.1145/2934872.2934899>.

[RC-EDF] "Efficient Network QoS Provisioning Based on per Node Traffic Shaping", 1996, <https://ieeexplore.ieee.org/document/532860>.

[RC-EDF-para]
"Traffic Shaping for End-to-End Delay Guarantees with EDF Scheduling", 2000, <https://ieeexplore.ieee.org/document/847934>.

[RPQ-EDF] "Exact Admission Control for Networks with a Bounded Delay Service", 1996, <https://ieeexplore.ieee.org/document/556345/>.

[SCED] "SCED: A Generalized Scheduling Policy for Guaranteeing Quality-of-Service", 1999, <https://ieeexplore.ieee.org/document/803382>.

[SP-LATENCY]
"Guaranteed Latency with SP", 2020, <https://ieeexplore.ieee.org/document/9249224>.

Appendix A. Proof of Schedulability Condition for RPQ

Figure 23 below gives the proof of schedulability condition for RPQ.

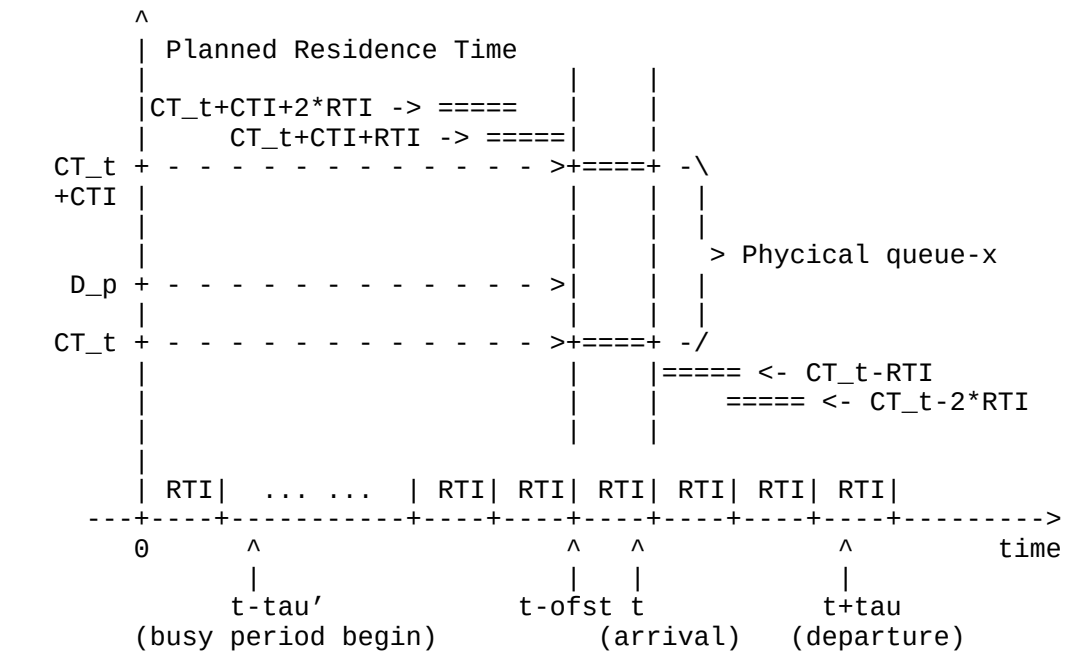

Figure 23: RPQ Based Scheduling

Suppose that the observed packet, with planned residence time D_p, arrives at the scheduler at time t and leaves the scheduler at time t+tau. It will be inserted to physical queue-x with count-down time CT_t at the current timer interval RTI with starting time t-ofst and end time t-ofst+RTI. According to the above packet queueing rules, CT_t <= D_p < CT_t+CTI. Also suppose that t-tau' is the beginning of the busy period closest to t. Then, the number of packets within time interval [t-tau', t+tau] that must be scheduled before the observed packet can then be determined. In detailed:

* For all flow i with planned residence time D_i meeting CT_t <= D_i < CT_t+CTI, the workload is sum{A'_i[t-tau', t]}.

  Explanation: since the packets with planned residence time D_i in the range [CT_t, CT_t+CTI) arrived at time t will be sent before the observed packet, the packets with the same D_i before time t will become more urgent at time t, and must also be sent before the observed packet.

* For all flow i with planned residence time D_i meeting D_i >= CT_t+CTI, the workload is sum{A'_i[t-tau', t-ofst-(D_i-CT_t-CTI)]}.

Explanation: although the packets with planned residence time D_i larger than CT_t+CTI arrived at time t will be sent after the observed packet, but the packets with the same D_i before time t, especially before time t-ofst-(D_i-CT_t-CTI), will become more urgent at time t, and must be sent before the observed packet.

* For all flow i with planned residence time D_i meeting D_i < CT_t, the workload is sum{A'_i[t-tau', t+(CT_t-D_i)]}.

Explanation: the packets with planned residence time D_i less than CT_t at time t will certainly be sent before the observed packet, at a future time t+(CT_t-D_i) the packets with the same D_i will still be urgent than the observed packet (even the observed packet also become urgent), and must be sent before the observed packet.

* Then deduct the traffic that has been sent during the busy period, i.e., C*(tau+tau').

Let tau as D_p, and remember that CT_t <= D_p, the above workload is less than

sum{A'_i(tau'+CT_t+CTI-D_i) for all D_i >= CT_t} +
sum{A'_i(tau'+CT_t-D_i) for all D_i < CT_t} - C*(tau'+D_p)

It is further less than

sum{A'_i(tau'+D_p+CTI-D_i) for all D_i >= D_2} + A'_1(tau'+D_p-D_1) - C*(tau'+D_p)

Then, denote x as tau'+D_p, we have

sum{A'_i(x+CTI-D_i) for all D_i >= D_2} + A'_1(x-D_1) - C*(x)

In the case that d_i contains only one D_i, A_i = A'_i, d_i = D_i, so the above workload is

sum{A_i(x+CTI-d_i) for all d_i >= d_2} + A_1(x-d_1) - C*(x)

If the above workload is less than zero, then Equation-2 is obtained.

In the case that d_i contains multiple D_i, e.g., d_1 is the minimum delay level with 10us, D_1 ~ D_10 is 10 ~ 19us respectively, d_2 is 20us, D_11 ~ D_20 iS 20 ~29us respectively, etc. Let D_1 ~ D_10 consume the resources of d_1, and D_11 ~ D_20 consume the resources of d_2, etc. Then, the above workload is less than

sum{A'_i(x+CTI-d_i) for all D_i belonging to d_i} - C*(x)

That is sum{A_i(x+CTI-d_i) for all d_i} - C*(x), and if it is less than zero, then Equation-3 is obtained.

Appendix B. Proof of Schedulability Condition for Alternate QAR of RPQ

In the case that d_i contains only one D_i, the schedulability condition is Equation-1. This is because, in the workload, for all D_i meeting D_i >= CT_t+CTI, their contributed workload is changed to sum{A'_i[t-tau', t-ofst-(D_i-CT_t)]} based on the analysis of Equation-2, that is, the amount of workload A'_i(CTI) (that is placed in queue-x) is excluded.

In the case that d_i contains multiple D_i, the schedulability condition is still Equation-3. This is because multiple D_i may belong to the same delay level as D_p. Assuming that within time zone [t-ofst, t-ofst+I] the list of all arrived D_i in the same parent queue-x with [CT_t, CT_t+CTI) as the observed packet (with D_p) is:

* D_a1~D_am, where D_a1 is closer to CT_t+CTI, they are larger than D_p (but smaller than CT_t+CTI) and belongs to a larger delay level than d_p (corresponding delay level of D_p).

* D_b1~D_bm, they are larger than D_p and belongs to the same delay level as d_p.

* D_p.

* D_c1~D_cm, they are smaller than D_p, and may belongs to the same delay level as d_p or a lower delay level than d_p.

So that both D_b1~D_bm and D_c1~D_cm should be scheduled before the observed packet. This is also true for these set of packets that have arrived in history.

Strictly, for D_a1, the contributed workload is sum{A'_i[t-tau', t-ofst+I-CTI]}, that is, only before time t-ofst+I-CTI the arrived packets of D_a1 will be placed in a more urgent queue-y with [CT_t, CT_t + CTI) than queue-x (at this history time its CT is [CT_t+CTI, CT_t + 2AT)) and should be shceduled before the observed packet. Similarly, for D_a2, the contributed workload is sum{A'_i[t-tau', t-ofst+I-CTI+I]}, for D_am, the contributed workload is sum{A'_i[t-tau', t-ofst+I-CTI+(m-1)*I]}.

Note that queue-x also contains packets with D_i (e.g., D_a0, larger than D_a1) that have arrived in history. For D_a0, the contributed workload is sum{A'_i[t-tau', t-ofst+I-CTI-(D_a0-D_a1)]}.

However, the number of m is not fixed. For safety, the workload time zone of D_a1~D_am can be overestimated to time instant t and considered that they need to be scheduled before the observed packet. Based on this, Equation-3 can be obtained.

Authors' Addresses

Shaofu Peng
ZTE Corporation
China
Email: peng.shaofu@zte.com.cn

Zongpeng Du
China Mobile
China
Email: duzongpeng@foxmail.com

Kashinath Basu
Oxford Brookes University
United Kingdom
Email: kbasu@brookes.ac.uk

Zuopin Cheng
Zhejiang P&T College
China
Email: chengzp@zptc.edu.cn

Dong Yang
Beijing Jiaotong University
China
Email: dyang@bjtu.edu.cn

Chang Liu
China Unicom
China
Email: liuc131@chinaunicom.cn