

DetNet Working Group
Internet-Draft
Intended status: Standards Track
Expires: 19 October 2026

J. Joung
Sangmyung University
J. Ryoo
T. Cheung
ETRI
Y. Li
Huawei
P. Liu
China Mobile
17 April 2026

Latency Guarantee with Stateless Fair Queuing
draft-ietf-detnet-stateless-fair-queuing-00

Abstract

This document specifies the implementation details for the framework specified in ITU-T Y.3129 [Y.3129] and ITU-T Y.3148 [Y.3148]. The framework guarantees end-to-end (E2E) latency bounds to flows. The schedulers in core nodes do not need to maintain flow states. Instead, the entrance node of a flow marks an ideal service completion time according to a fluid model, called Finish Time (FT), of a packet in the packet header. The subsequent core nodes update the FT by adding a delay factor, which is a function of the flow and the nodes. The packets in the queue of the scheduler are served in the ascending order of FT. This mechanism is called the stateless fair queuing. The result is that flows are isolated from each other almost perfectly. The latency bound of a flow depends only on the flow's intrinsic parameters such as the maximum burst size and the service rate, except the link capacities and the maximum packet length among other flows sharing each output link with the flow. This document specifies the metadata, formats of metadata, the admission control procedure, and an approximation of stateless fair queuing implemented via a strict priority (SP) scheduler.

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 19 October 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Terminology	4
2.1. Terms Used in This Document	4
2.2. Abbreviations	4
3. Conventions Used in This Document	5
4. Assumptions	5
5. Framework	6
6. Operational Procedures	7
6.1. Metadata	7
6.2. Header format	7
6.2.1. IPv6 header format	7
6.2.2. MPLS label format	8
6.3. Admission control for latency guarantee	11
6.4. Compensation algorithm of time difference between nodes	13
7. Characteristics	14
7.1. Taxonomy	14
7.2. Strengths	14
8. Approximate C-SCORE via strict priority schedulers	15
8.1. General description	15
8.2. Transit node architecture	15
8.3. Algorithms	16
8.4. E2E latency bound of the approximate C-SCORE	17
9. Considerations for non-compliant nodes	17
10. Relationships to ITU-T Standards	17

11. IANA Considerations	18
12. Security Considerations	18
13. Acknowledgements	18
14. Contributor	18
15. References	18
15.1. Normative References	18
15.2. Informative References	19
Authors' Addresses	19

1. Introduction

Recommendation ITU-T Y.3129 [Y.3129] specifies the requirements and framework for deterministic networking with a set of work conserving packet schedulers that guarantees end-to-end latency bounds to flows. The schedulers in core nodes do not need to maintain flow states. Instead, the entrance node of a flow marks an ideal service completion time according to a fluid model, called finish time (FT), of a packet in the packet header. The subsequent core nodes update FT by adding a delay factor, which is a function of the flow and upstream nodes. The packets in the queue of the scheduler are served in the ascending order of FT. This mechanism is called stateless fair queuing. The result is that flows are isolated from each other almost perfectly. The latency bound of a flow depends only on the flow's intrinsic parameters, except the maximum packet length among other flows sharing each output link with the flow.

Recommendation ITU-T Y.3148 [Y.3148] further specifies the functional architecture, functional entities, operational procedures, and packet metadata to fulfil the requirements and framework specified in Recommendation ITU-T Y.3129. It enables deterministic networking with a set of work conserving packet schedulers in the data plane, which guarantees end-to-end (E2E) latency bounds to flows.

In the framework specified in [Y.3129], the edge node through which a flow enters a network is called the entrance node. The entrance node for a flow generates FT for a packet and records it in the packet. A core node, based on these records, updates FT, without per-flow state, by adding a delay factor that is a function of parameters of the node and the flow. This framework is called work conserving stateless core fair queuing (C-SCORE) [C-SCORE]. C-SCORE is work conserving and has the property that, for a certain choice of the delay factor, the expression for E2E latency bound can be found. This E2E latency bound function is the same as that of a network with stateful fair queuing schedulers in all the nodes.

This document specifies the implementation details for realizing C-SCORE. This document specifies the metadata, formats of metadata, the admission control procedures, and an approximation of stateless fair queuing implemented via a strict priority (SP) scheduler.

A key component of C-SCORE is the packet state that is carried as metadata. C-SCORE does not need to maintain flow states at core nodes, yet it works as one of the fair queuing schedulers, which is known to provide the best flow isolation performance. The metadata to be carried in the packet header is simple and can be updated during the stay in the queue or before joining the queue.

2. Terminology

2.1. Terms Used in This Document

2.2. Abbreviations

BE: Best Effort

C-SCORE: Work Conserving Stateless Core Fair Queuing

DetNet: Deterministic Networking

E2E: End to End

FQ: Fair Queuing

FT: Finish Time

GPS: Generalized Processor Sharing

HoQ: Head of queue

FIFO: First-In First-Out

MNA: MPLS Network Actions

NAS: Network Action Sub-Stack

PIFO: Push-In First-Out

PRPS: Packetized Rate Proportional Servers

PSD: Post-Stack Data

RSPEC: Requested Specifications

TLV: Type-Length-Value

TSpec: Traffic Specifications

VC: Virtual Clock

3. Conventions Used in This Document

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

4. Assumptions

In this document, we assume there are only two classes of traffic. The high priority or equivalently DetNet traffic requires guarantee on latency upper bounds. All the other traffic is considered to be the low priority or Best Effort (BE) traffic. High priority traffic is our only concern. However, packets of BE traffic cannot be preempted by the high priority traffic.

All the flows conform to their traffic specification (TSpec) parameters. In other words, with the maximum burst size B_i and the arrival rate a_i , the accumulated arrival from flow i in any arbitrary time interval $[t_1, t_2]$, $t_1 < t_2$, does not exceed $B_i + (t_2 - t_1)a_i$. An actual allocated service rate to a flow, r_i , can be larger than or equal to the arrival rate of the flow. As it will be shown in (6) that, by adjusting the service rate to a flow, the E2E latency bound of the flow can be adjusted. Note that r_i is used interchangeably with the symbol r to denote the service rate of a flow under observation. Total allocated service rate to all the flows in a node does not exceed the link capacity of the node. These assumptions make the resource reservation and the admission control mandatory.

A node, or equivalently a server, means an output port module of a switching device.

The entrance node for a flow is the node located at the edge of a network, from which the flow enters into the network. A core node for a flow is a node in the network, which is traversed by the flow and is not the entrance node. Note that a single node can be both an entrance node to a flow and a core node for another flow.

A packet is defined as arrived or serviced when its final bit is received by or transmitted from the node, respectively.

5. Framework

Fair queuing (FQ) schedulers utilize the concept of finish time (FT) that is used as the service order assigned to a packet. The packet with the minimum FT in a buffer is served first.

The key idea of the FQ is to calculate the ideal service completion times of packets in an imaginary fluid service model and use them as the service order in the real packet-based scheduler.

While having the excellent flow isolation property, FQ schedulers need to maintain the flow state, $F(p-1)$. Upon the arrival of each packet, the system must perform flow identification and retrieve the FT associated with the preceding packet in that flow. The system is required to update the flow state to be $F(p)$ concurrently with the packet's departure.

[Y.3129] and [Y.3148] specify a FQ scheduler, which does not need to maintain the flow state in core nodes. In the entrance node, the FTs are obtained with the following equation, where 0 denotes the entrance node of the flow under observation.

$$F0(p) = \max\{F0(p-1), A0(p)\} + L(p)/r. \quad (1)$$

In a core node h , the FT of a packet is increased by an amount, $d(h-1)(p)$, that depends on the previous node and the packet.

$$Fh(p) = F(h-1)(p) + d(h-1)(p). \quad (2)$$

The framework achieves a bounded E2E latency by defining $dh(p)$ as follows.

$$dh(p) = Lh/Rh + L/r + \text{delta_h}(p), \quad (3)$$

where Lh is the observed maximum packet length in the node h over all the flows, Rh is the link capacity of the node h , L is the maximum packet length of the flow, r is the service rate of the flow, and $\text{delta_h}(p)$ is the function that represents the time difference between node h and $h+1$.

The time difference function of packet p , $\text{delta_h}(p)$, is defined as the difference between the service completion time at node h and the arrival time at node $h+1$ of the packet. It includes the clock discrepancy and the propagation delay between nodes. Note that this function is relatively stable over packets, thus can be approximated as a constant value delta_h .

When $dh(p)$ is decided by (3), then it is proven that

$$Dh(p) \leq (B-L)/r + \sum_{j=0}^h \{L_j/R_j + L/r\} + \sum_{j=0}^{(h-1)} \{\delta_j(p)\}, (4)$$

where $Dh(p)$ is the latency experienced by p from the arrival at the node 0 to the departure from node h [KAUR], [C-SCORE]. B is the maximum burst size of the flow under observation that p belongs to. The term $\sum_{j=0}^{(h-1)} \{\delta_j(p)\}$ includes the clock discrepancies and the propagation delays between the nodes. Considering that the clock discrepancies can be neglected in an absolute time reference, this term effectively represents the E2E propagation delay.

Note that the latency bound in (4) is the same to the network where every node has a stateful FQ scheduler. The parameters in the latency bound are all intrinsic to the flow, except L_h/R_h and propagation delays.

6. Operational Procedures

6.1. Metadata

[Y.3129] and [Y.3148] specifies the metadata to be carried by a packet, which are $Fh(p)$, L , and r . $Fh(p)$ is dynamic and needs to be updated every hop, in accordance with (2). L and r are static values for a flow.

As a packet arrives at a core node h , it carries metadata $Fh(p)$, L , and r . $Fh(p)$ is pre-calculated at node $h-1$ with $d(h-1)(p)$, which is a function of node $h-1$. $dh(p)$ can be obtained by the summation of L/r , and the node specific parameters L_h/R_h and δ_h . L_h/R_h and δ_h values should be maintained by the node h . At node h , $F(h+1)(p)$ value is calculated and replaces $Fh(p)$.

The detailed operations regarding metadata creation and updates in the entrance node and core nodes are specified in [Y.3129] and [Y.3148]. The considerations for the mitigation of the complexity at the entrance node can also be found in these references.

6.2. Header format

6.2.1. IPv6 header format

The IPv6 Hop-by-Hop (HbH) Options Header [RFC8200] should be used for carrying the metadata. It is a specific type of Extension Header designed for information that must be examined and processed by all the transit nodes along a packet's delivery path. This header is identified by a Next Header value of 0 in the IPv6 main header. The HbH Options header consists of a sequence of variable-length options. These options are encoded in a Type-Length-Value (TLV) format. In this transformation, the metadata (Finish Time, L , and r) are stored

in the Option Data fields of three separate Options within a single HbH Options header. Length of the HbH Options header should be specified by the Hdr Ext Len field. Option Type and Option Data Len fields should specify the C-SCORE metadata identifier and the length of the metadata, which are to be determined.

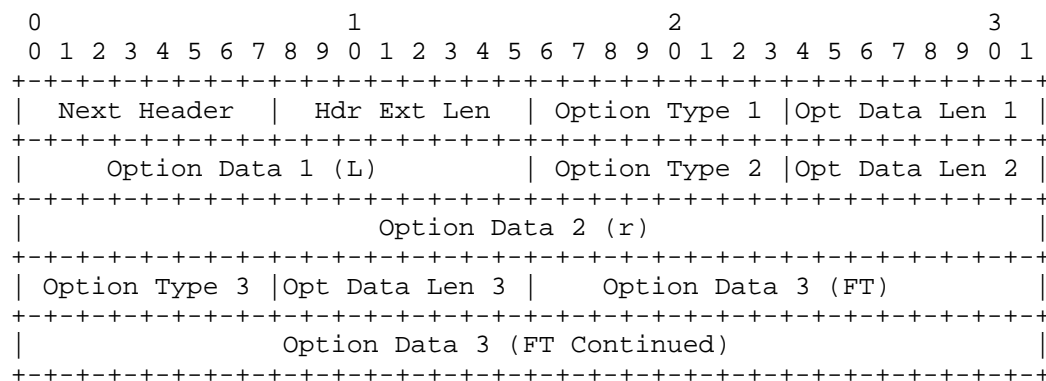


Figure 1: IPv6 HbH Options Header example for two metadata for C-SCORE

The following is the operational procedure in a transit node. The hardware parser sees Next Header 0 in the main IPv6 header and directs the packet to the metadata management function. This functional entity scans for the Option Type corresponding to C-SCORE. The Finish Time (48 bit, TBD), L (16 bit, TBD), and r (32 bit, TBD) are extracted and processed. The new FT is calculated. Since the offset is fixed relative to the start of the HbH Options header, the functional entity performs a single cycle write to update the metadata field before recalculating any necessary checksums (if applicable) and forwarding.

Note that L, the maximum packet length of the flow, does not need to be precise at the byte level. Similarly r, the service rate of the flow, can be of a coarse granularity or even several discrete pre-defined levels. The encoding of these metadata will be specified in a later version of this draft.

6.2.2. MPLS label format

[I-D.ietf-mpls-mna-detnet] specifies formats and mechanisms for using MPLS Network Actions (MNA) to support DetNet services, including bounded latency, low loss and in-order delivery. It specifies three information elements of DetNet packets, which are Flow identifier (Flow-ID), Sequence information (SeqNum), Latency information (LatencyInfo). The C-SCORE metadata Fh(p), L, and r are the Latency

Information, according to this specification.

Two approaches for carrying information elements are specified: In-Stack and Post-Stack MNAs. With In-Stack MNA, the DetNet-specific information is embedded directly within the MPLS label stack, as part of a Network Action Sub-stack (NAS). The information elements reside before the Bottom of Stack (BOS) bit. It uses a Network Action Indicator (NAI) to signal that the subsequent labels in the sub-stack are actually ancillary data (Flow-ID, etc.) rather than traditional switching labels.

With Post-Stack MNA, the DetNet-specific information is carried after the label stack. The data resides between the BOS bit and the start of the user payload. An indicator within the label stack (the NAI) points to the presence of Ancillary Data located immediately after the stack. Post-stack data is better suited for large or variable-sized data that would otherwise make the label stack prohibitively deep.

This document follows the Post-Stack encoding approach, but the In-Stack approach is not excluded. In the Post-Stack approach, the MNA sub-stack is usually placed immediately after the bottom of the MPLS label stack. This allows for large amounts of ancillary data to be carried without making the label stack excessively deep.

The Post-Stack MNA solution contains two components:

- 1) Post-Stack MPLS Header Presence Bit carried in In-Stack MNA Sub-Stack
- 2) Post-Stack MPLS Header that includes Post-Stack MPLS Header Type (PSMHT) and Post-Stack Network Actions (PSNA)

Bit 20 in Label Stack Entry (LSE) Format B carried in the In-Stack NAS is defined as the P bit to indicate the presence of the Post-Stack MPLS Header in the packet after the BOS bit.

[I-D.ietf-mpls-mna-ps-hdr] LSE Format B refers to a specialized structure for a LSE that carries ancillary data instead of a traditional switching label.

PSMHT can be located immediately following the BOS label, which includes PS-HDR-LEN and Version/Type. PS-HDR-LEN specifies the total length of the post-stack metadata. Version/Type identifies the MNA-POST-STACK-HDR. The metadata for C-SCORE are FT, L and r. The number of bits required for these metadata are for further study and to be specified. FT, L and r are carried across three separate PSNAs to support hop-by-hop scheduling. If FT is of 48 bit length, then the PSNA for FT should have Post-Stack Network Action Length (PS-NAL) = 1, as in Figure 2.

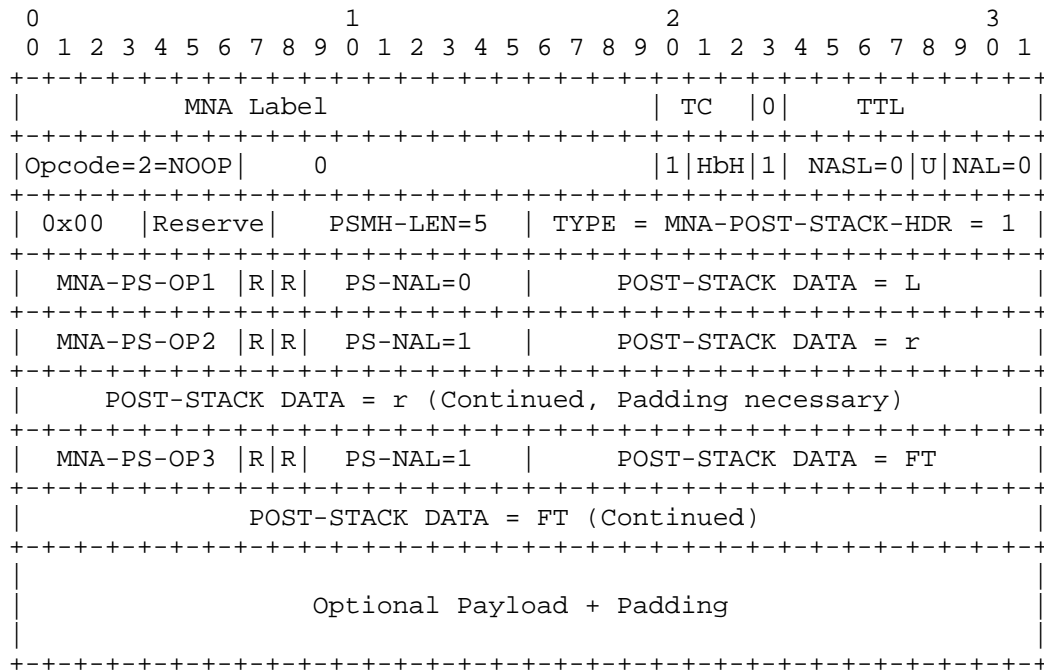


Figure 2: Post Stack MNA Sub-Stack example with three PSNAs for three metadata for C-SCORE

Figure 2 is an example where the Post-Stack MNA Sub-Stack encodes three different PSNAs for three metadata. Their details are as follows:

- The offset of the Hop-By-Hop scoped PSNA is 0.
- PSMH-LEN=5: This is the total length of the Post-Stack MPLS Header (PSMH).
- MNA-PS-OP1: Post-Stack MNA Opcode (TBD) for L

- PS-NAL=0: PSNA does not contain any additional data.
- MNA-PS-OP2: Post-Stack MNA Opcode (TBD) for r
- PS-NAL=1: PSNA contains one additional 4-octet Ancillary Data.
- MNA-PS-OP3: Post-Stack MNA Opcode (TBD) for Finish Time
- PS-NAL=1: PSNA contains one additional 4-octet Ancillary Data.

The post-stack data r can be of 32 bit length, which can make the padding necessary. The rule for padding will be specified by [I-D.ietf-mpis-mna-detnet] in its future version.

Note that C-SCORE does not require Flow-ID to be carried in the packet. This is because of its stateless nature.

6.3. Admission control for latency guarantee

The following is a recommended procedure for admission of a flow in the C-SCORE framework.

- 1) An application requests a flow with T-Spec (Traffic Specification) and R-Spec (Requested Specification). T-Spec includes service rate r , maximum packet size L , and maximum burst size B . R-Spec includes the E2E latency and jitter bounds.
- 2) The entrance node sends a PATH message toward the egress. This message contains the T-Spec and R-Spec.
- 3) The entrance node acts as the admission controller. It maintains per-flow state and performs the initial shaping, if necessary.
- 4) Core nodes check if the requested service rate r is within the aggregate bandwidth threshold of the outgoing interface.
- 5) Each core node adds its per-hop max latency value ($L/r + L_h/R_h$) to a field in the PATH message. This allows the egress node to calculate the E2E latency bound.
- 6) The egress node receives the PATH message. If the E2E latency bound meets the application's requirement, it generates a RESV message.
- 7) The RESV message travels back to the entrance node. The core nodes in the path confirms the admission of the flow.
- 8) If the RESV message reaches the entrance node, the flow is

admitted.

In a C-SCORE architecture utilizing the RSVP-like admission process described above, it is not only possible but highly efficient to find out and negotiate the optimal available service rate for a flow. The admission process can probe the network to determine the maximum supportable rate without violating the deterministic latency bounds of existing flows. The discovery of the available rate occurs during the PATH message traversal. The following is an alternative admission process, which identifies the tightest capacity link in the path.

- 1) An application requests a flow with T-Spec (Traffic Specification) and R-Spec (Requested Specification).
- 2) The entrance node sends a PATH message with a Desired Rate (r_{desired}) and a Minimum Acceptable Rate (r_{min}), along with T-Spec and R-Spec.
- 3) Each core node on the path calculates its residual capacity (r_{avail}). This is the total capacity of the link minus the sum of the service rates (r) of all already admitted flows.
- 4) As the PATH message moves hop-by-hop, it maintains a field called Path-Available-Rate (r_{path}). At each hop h , the router performs: $r_{\text{path}} = \min(r_{\text{path}}, r_{\text{avail}_h})$, while the subscript h denotes the node.
- 5) By the time the PATH message reaches the egress, r_{path} represents the maximum service rate the entire end-to-end path can support at that specific moment.
- 6) The optimal rate is not always the highest possible rate. The egress node can select the optimal rate, which can be less than r_{path} .
- 7) Once the optimal service rate ($r = r_{\text{opt}}$) is determined, the egress sends the RESV message back to the entrance node, carrying the r_{opt} value.
- 8) The core nodes in the path confirms the r_{opt} value for the flow.
- 9) If the RESV message reaches the entrance node, the flow is admitted.
- 10) The entrance node commits this rate to its per-flow state.

The optimal rate should be between r_{path} and r_{min} . The egress node can select the optimal rate based on the following criteria:

- Latency requirements: Higher service rates result in smaller Finish Time (FT) increments, reducing the per-hop queuing delay.
- Buffer constraints: The rate must be balanced with the max burst (B) parameter to ensure the core nodes' buffers do not overflow. Note that the burst accumulates linearly with the service rate.
- Network availability: The egress may choose a rate lower than r_{path} to leave room for other flows that can join later.

Since the core nodes only need to know the final committed rate r in the packet header, the complex discovery logic is restricted to the control plane and edge nodes. Large-scale networks can use this mechanism to perform the aggregate rate discovery, where the r_{path} represents the available capacity for an entire bundle of flows of the same path. By using this approach, the admission process also effectively maps the current congestion state of the network onto a single service rate parameter.

6.4. Compensation algorithm of time difference between nodes

There are time differences between nodes, including the clock discrepancies and the propagation delays. This time difference can be defined as the difference between the service completion time of a packet measured at the upstream node and the arrival time of the packet measured at the current node. In other words,

$$\text{delta_h}(p) = A(h+1)(p) - Ch(p),$$

where $\text{delta_h}(p)$ is the time difference between node h and $h+1$, and $Ch(p)$ is the service completion time measured at node h , for packet p respectively.

FT does not need to be precise. It is used just to indicate the packet service order. Therefore, if we can assume that the propagation delay is constant and the clocks do not drift, then $\text{delta_h}(p)$ can be simplified to a constant value, delta_h .

The time difference delta_h may be updated only once in a while.

The procedure for obtaining the departure time from node h , $Ch(p)$, at node $h+1$ will be elaborated in a later version of this draft. The $Ch(p)$ information can be obtained on demand, or be reported periodically. This information exchange can be based on a piggybacking mechanism with packet metadata.

Alternatively, the explicit messages can be exchanged. In this case the message format will follow that of Network Time Protocol (NTP), but the procedure will be simpler.

Note that the time difference between non-adjacent nodes can also be obtained similarly. This feature is useful when there are non-compliant nodes in between. In this case, however, the variable queuing delay from the non-compliant nodes should be taken into account. One possible solution is to sample the time difference values over an enough interval, and take the maximum value.

7. Characteristics

7.1. Taxonomy

The framework in this document, C-SCORE, is a flow level, rate based, work conserving, asynchronous, non-periodic, and in-time solution, according to the taxonomy suggested by [I-D.ietf-detnet-dataplane-taxonomy].

[I-D.ietf-detnet-dataplane-taxonomy] also defines seven suitable categories for deterministic networking. A category is defined to be a set of solutions that is put together by one or more criteria, where a criterion is a principle or standard by which a solution can be judged or decided to be put into a certain category.

C-SCORE belongs to the "flow level rate based unbounded category", which is one of the seven suitable categories, according to this categorization.

7.2. Strengths

C-SCORE's per hop latency dominant factor is the maximum packet length divided by the service rate of the flow. This is independent of other flows' parameters. As such, its most distinguishable strength is the flow isolation capability. It can assign a fine tuned E2E latency bound to a flow, by controlling the flow's own parameters such as the service rate. Once the latency bound is assigned to the flow, then it remains almost the same in spite of the network situation changes, such as other flows' join and leave.

It is work conserving, thus enjoys the statistical multiplexing gain without wasting bandwidth, which is the key to the Internet's success. The consequence is a smaller average latency. The observable maximum latency is also much smaller than the theoretical latency bound. Note that, with a work conserving solution, observing the theoretical latency bound is extremely difficult in real situations. It is because the worst latency is an outcome of a

combination of multiple rare events, e.g. a maximum burst from a flow collides with the maximum bursts from all other flows at every node. In contrast, non-work conserving solutions make it common to observe their latency bounds.

It is rate based, thus the admission condition check process is simple, which is dependent only on the service rates of flows. This process aligns well with existing protocols.

Overall, C-SCORE suits large scale networks, at any utilization level, with various types of flows join and leave dynamically.

8. Approximate C-SCORE via strict priority schedulers

8.1. General description

C-SCORE requires a priority queue that sorts the packets in a queue, in accordance with their finish times. This may restrict the overall maximum throughput of a system, when compared to the strict priority (SP) schedulers used in the current practices of switching nodes. SP schedulers are usually composed of 8 to 32 queues, and schedule the packets of higher priority queue first whenever they are present. The packets in the same queue are served on a first in first out (FIFO) basis. It is common to find the SP schedulers in hardware chips for current switching nodes.

It would be desirable if C-SCORE can be implemented with such an SP scheduler. In this section, the architecture and algorithms for the Approximate C-SCORE with rotating SP schedulers are specified. The E2E latency bound of the network of the approximate C-SCOREs is also specified.

8.2. Transit node architecture

The architecture of an approximate C-SCORE transit node, in which the SP scheduler with a limited number of queues behaves as an approximate priority queue, is specified in this section.

The input port module classifies the deterministic flows and best effort (BE) flows. The deterministic flows are put into one of N queues that work as rotating SP scheduler queues. If the queue k , $0 \leq k \leq N-1$, is the highest priority queue at time t , then the queue $(k+N-1) \bmod N$ has the lowest priority at t . BE flows are put into its own queue in the output port module. BE queue is served only when there is no packet in queues 0 to $N-1$.

Assume that the time is divided into fixed-length slots. Let us say that a slot is allocated to a certain queue. Further, let T_i denote the terminal boundary of Slot i . An arriving packet p is assigned to Slot i if its finish time, $F(p)$, falls within the interval $(T_{i-1}, T_i]$. This mapping corresponds to a discrete set of hardware queues where packets are buffered and processed according to a First-In-First-Out (FIFO) discipline. The system utilizes a strict priority (SP) scheduler to arbitrate across these queues, granting precedence to those representing the earliest finish-time. The scheduler operates in a work-conserving manner, providing service at line rate whenever the system is backlogged.

8.3. Algorithms

To provide differentiated quality-of-service, the per-hop latency must be modulated according to the specific service rates of traversing flows. By reducing the scheduling granularity, or slot length (S), the system can provide tiered latency bounds based on the number of slots occupied by a flow's maximum virtual service interval (L/r). Specifically, for a flow f where $(n-1)S < L/r \leq nS$, the per-hop latency is bounded by $(n+1)S$. This mechanism ensures fairness by granting lower latency to flows with higher service rates (smaller L/r), thereby aligning temporal performance with bandwidth allocation. The scheduling granularity, or slot duration S , is defined such that $S \geq \min_p [L(p)/r(p)]$. While a finer granularity generally minimizes end-to-end latency, reducing S below this lower bound, which is determined by the minimum packet transmission time relative to the flow rate, yields no additional scheduling benefit. In practical implementations, N hardware queues are allocated and managed in a cyclic manner to accommodate these slots. Consequently, the selection of S involves a fundamental trade-off: a coarser slot duration allows for a reduced number of physical queues, N , at the expense of decreased scheduling precision.

In an idealized preemptive system, the scheduler ensures that all packets mapped to a specific slot are fully serviced before its terminal boundary, T_i . Consequently, the service interval for any packet p is strictly contained within its assigned slot. However, in a realistic non-preemptive environment, the completion time is extended by a factor of L_h/R_h , where L_h is the maximum packet length and R_h is the link rate of the output link h , respectively. Thus, the service is guaranteed to complete no later than $T_i + L_h/R_h$.

The approximation follows the architecture of C-SCORE. In other words, equations (1) and (2) are still used. However (3) is replaced by the following equation (5).

$$d_h(p) = (L_h^{\max})/R_h + (n_{(f,h)}+1)S_h + \delta_h(p), \quad (5)$$

8.4. E2E latency bound of the approximate C-SCORE

The E2E latency of a network with the approximate C-SCORE schedulers is upper bounded by

$$B/r + \sum_{h=0}^H \{ (n_{(f,h)}+1)S_h + L_h/R_h \} + \sum_{h=0}^{H-1} \{ \delta_h(p) \},$$

where S_h is the slot length at node h , $n_{(f,h)}$ is an integer specific to flow f at node h , which meets $(n_{(f,h)}-1)S_h < L_f/r \leq n_{(f,h)}S_h$. In other words, $n_{(f,h)} = \lceil L_f/(r*S_h) \rceil$. The term $\sum_{h=0}^{H-1} \{ \delta_h(p) \}$ includes the clock discrepancies and the propagation delays between the nodes. Considering that the clock discrepancies can be neglected in an absolute time, this term actually represents the E2E propagation delay.

9. Considerations for non-compliant nodes

There can be non-compliant end nodes and relay nodes in the network. There can be naturally end nodes without necessary signalling capabilities for admission control. There also can be legacy nodes or the nodes with dataplane enhancement solutions other than C-SCORE. How these can be compensated, or how much these nodes degrade the performance of C-SCORE will be discussed in a later version.

10. Relationships to ITU-T Standards

This document is based on two ITU-T standards, Y.3129 [Y.3129] and Y.3148 [Y.3148].

Recommendation ITU-T Y.3129 specifies the requirements and framework for C-SCORE. The requirements are for generation, update and properties of FT. The framework describes a mechanism to guarantee E2E latency bounds, while meeting the requirements. It specifies how to obtain FT in core nodes, select the delay factor, and configure a network for latency guarantee.

Recommendation ITU-T Y.3148 specifies the functional architecture, functional entities, operational procedures, and packet metadata to fulfil the requirements and framework specified in Recommendation ITU-T Y.3129. The operational procedures are for the stateless FQ in relay nodes, including entrance node and core nodes with creation and update of FT values of packets.

This document bridges the gap between the high-level ITU-T standards and deployable protocols. It achieves this by defining the metadata structures and the corresponding header formats for IPv6 and MPLS.

Furthermore, it specifies the procedures required for admission control and time-difference compensation, alongside the technical methodology for C-SCORE approximation.

11. IANA Considerations

There might be matters that require IANA considerations associated with metadata. If necessary, relevant text will be added in a later version.

12. Security Considerations

This section will be described later.

13. Acknowledgements

14. Contributor

15. References

15.1. Normative References

[I-D.ietf-mppls-mna-detnet]

Song, X., Mirsky, G., Varga, B., Gandhi, R., and Q. Xiong, "MPLS Network Action for Deterministic Networking", Work in Progress, Internet-Draft, draft-ietf-mppls-mna-detnet-00, 7 January 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-mppls-mna-detnet-00>>.

[I-D.ietf-mppls-mna-ps-hdr]

Rajamanickam, J., Gandhi, R., Zigler, R., Dong, J., and J. Bhattacharya, "Post-Stack MPLS Network Action (MNA) Header Specification", Work in Progress, Internet-Draft, draft-ietf-mppls-mna-ps-hdr-08, 2 April 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-mppls-mna-ps-hdr-08>>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [Y.3129] International Telecommunication Union, "Requirements and framework for stateless fair queuing in large scale networks including IMT-2020 and beyond", ITU-T Recommendation Y.3129, April 2024.
- [Y.3148] International Telecommunication Union, "Functional architecture for stateless fair queuing in large scale networks including IMT-2020 and beyond", ITU-T Recommendation Y.3148, August 2025.

15.2. Informative References

- [C-SCORE] Joung, J., Kwon, J., Ryoo, J., and T. Cheung, "Scalable flow isolation with work conserving stateless core fair queuing for deterministic networking", IEEE Access, vol. 11, pp. 105225 - 105247, doi:10.1109/ACCESS.2023.3318479, 2023.
- [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-05, 8 January 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-detnet-dataplane-taxonomy-05>>.
- [KAUR] Kaur, J. and H.M. Vin, "Core-stateless guaranteed rate scheduling algorithms", in Proc. INFOCOM, vol.3, pp. 1484-1492, 2001.

Authors' Addresses

Jinoo Joung
Sangmyung University
Email: jjoung@smu.ac.kr

Jeong-dong Ryoo
ETRI
Email: ryoo@etri.re.kr

Taesik Cheung
ETRI

Email: cts@etri.re.kr

Yizhou Li
Huawei
Email: liyizhou@huawei.com

Peng Liu
China Mobile
Email: liupengyjy@chinamobile.com