

DetNet
Internet-Draft
Intended status: Standards Track
Expires: 28 October 2023

Y. Li
S. Ren
G. Li
F. Yang
Huawei Technologies
J. Ryoo
ETRI
P. Liu
China Mobile
26 April 2023

IPv6 Options for Cyclic Queuing and Forwarding Variants
draft-yizhou-detnet-ipv6-options-for-cqf-variant-02

Abstract

The fundamental Cyclic Queuing and Forwarding (CQF) defined by Time-Sensitive Networking (TSN) requires no per-stream per-hop state maintenance and at the same time its end-to-end bounded latency and jitter can be easily computed. Such features are attractive and therefore CQF is being considered in wider deployments. To accommodate the different deployments, there are variants of CQF enhancement. This document introduces a new IPv6 option to include the cycle identification to help leverage CQF variants in DetNet network to facilitate the deployments.

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 28 October 2023.

Copyright Notice

Copyright (c) 2023 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	2
2.	Terminologies	3
3.	Features of CQF Variants	3
3.1.	Desire to support higher speed links in DetNet	4
3.2.	Desire to support longer links in DetNet	5
4.	CQF Variants	7
5.	The CQF-Variant Option	11
5.1.	CQF-Variant Option Format	12
5.2.	CQF-Variant Option Processing	13
6.	Encapsulation of CQF-Variant Option	13
7.	Collaboration Considerations	14
8.	Security Considerations	15
9.	IANA Considerations	15
10.	Contributors	16
11.	References	16
11.1.	Normative References	16
11.2.	Informative References	16
	Authors' Addresses	17

[1.](#) Introduction

The latency guarantee is the one of the most important features to be provided by Deterministic Networking (DetNet).

[\[I-D.ietf-detnet-bounded-latency\]](#) presents some examples of queuing mechanisms to show how each or the combination of them can be used to achieve the different levels of end-to-end latency bound requirements. Among them, Cyclic Queuing and Forwarding (CQF) shows a great simplicity in calculation and configuration of latency bounds.

Based on the two-buffer CQF specified in annex T of [\[IEEE8021Q\]](#), the latency bound is between $(h-1) * T_c + DT$ and $(h+1) * T_c$ where h is the number of hops, T_c is the cycle interval of buffer swapping and DT , called dead time is a time interval at the end of a cycle, during which no frames can be transmitted from the buffer to ensure the last byte of the cycle to be received earlier than the end of the same cycle by the next downstream node [\[I-D.ietf-detnet-bounded-latency\]](#).

DT is usually the sum of output delay, link delay, frame preemption delay, and processing delay. There can also be other factors contributing to DT such as the time synchronization drifting. Normally the DT is much lesser than cycle interval T_c so that it is negligible. Hence the minimum latency becomes $(h-1) * T_c$ approximately. That is to say the end-to-end jitter is bounded by $2 * T_c$ approximately for two-buffer CQF.

The latency and jitter bound of CQF is relatively intuitive and almost fully depends on cycle interval T_c and number of hops. The lesser T_c is, the smaller the end-to-end latency is. At the same time, it requires no per-stream per-hop status at the intermediate nodes as long as the cycle interval T_c is carefully selected to make sure the overall bandwidth allowed in T_c is large enough to accommodate all the traffic volume requiring CQF scheduling. Therefore, CQF offers very attractive features to users in terms of simplicity and intuition and is getting wider acceptance in DetNet.

When employing CQF in networks to support higher speed long links, there are variants to enhance CQF for those specific network characteristics, including more buffers (greater than two buffers) and cycle identification. The CQF variants do not change the basic concept of fundamental CQF's en-queue and de-queue logic, while a new data plane option is required. This document introduces the new IPv6 [\[RFC8200\]](#) options to help such CQF variants unambiguously identify the cycle in which the upstream node sends the data packet when the large number of buffers is employed so that CQF variants can adapt to the wider deterministic networking requirements.

2. Terminologies

CQF: Cyclic Queuing and Forwarding. A queuing mechanism defined by annex T of [\[IEEE8021Q\]](#). This document also uses CQF to refer to the variants enhanced from it using time cycle based en-queuing and draining.

3. Features of CQF Variants

The fundamental CQF uses two buffers to swap the packet receiving and transmission. Swapping is executed every cycle interval T_c . Two buffer generally maps to two traffic class queues.

When CQF is used in higher speed and/or longer links in wider area networking, new features like larger number (≥ 3) of buffers and cycle identification are introduced. [Section 3](#) and [Section 4](#) illustrate why these features are required and how they work.

3.1. Desire to support higher speed links in DetNet

The fundamental CQF typically uses no less than hundreds of microseconds as a cycle interval. In a network with a small diameter, say less than 8 hops, it is sufficiently good to provide an end-to-end latency bound in the order of several milliseconds.

With the increasing of link speed from 100Mbps to 1Gbps, 10Gbps or even higher in larger networks, either more bytes can be transmitted within the same cycle interval or the smaller cycle interval is required to transmit the same amount of bytes in a cycle as that in low speed networks.

Figure 1 shows a simple calculation on the number of bytes that can be transmitted in a cycle with different cycle intervals and link speeds. 1500 bytes is labeled with * as a baseline because a typical maximum Ethernet frame is 1500 bytes and a selected cycle interval should at least allow one such frame size to be transmitted unless otherwise specified.

Bytes Transmitted in a Cycle				
Cycle Time	Link Speed			
(us)	100Mbps	1Gbps	10Gbps	
1	12.5	125	1250	
1.2	15	150	1500*	
2	25	250	2500	
4	50	500	5000	
10	125	1250	12500	
12	150	1500*	15000	
120	1500*	15000	150000	

Figure 1: Bytes transmitted within one cycle interval

When the link speed is at 10Gbps, the cycle interval could be as small as 1.2 us if a 1500 byte frame needs to be transmitted in one cycle interval. It is not an accurate calculation because there are certainly other factors to determine the cycle interval. However, it shows that as the link speed increases, cycle interval can be greatly

reduced in practice while satisfying the minimum amount of data transmitted in a single cycle. The end-to-end latency bound when applying CQF is determined by cycle interval and number of hops. That is to say, CQFs with a smaller cycle interval have the potential to meet more strict end-to-end latency requirements in higher link speed networks or meet the same end-to-end latency requirement in networks with much larger network diameter (number of hops).

Industry automation has some typical application period requirement, e.g. 100 us to 2 ms for isochronous traffic, 500 us to 1 ms for cyclic-synchronous and 2 to 20 ms for cyclic-asynchronous traffic. The network cycle interval is usually a fraction of the application period. When the cycle interval is in the order of tens of microseconds, CQF can be used to meet the most strict end-to-end latency requirements. For instance, if we assume the number of hops is 24, when cycle interval is set to 10us, the end-to-end latency bound can be around $(24+1)*10 = 250$ us which has the potential to meet the latency bound requirement for isochronous traffic.

In summary a higher speed network makes the shorter cycle interval feasible because sufficiently large traffic volume can be transmitted within one cycle interval. A shorter cycle interval further offers shorter end-to-end latency and jitter bounds which provide CQF with the potentials to meet more strict latency requirements in wider deployments while preserving its simplicity of latency calculation and provisioning. Therefore there is a strong motivation to leverage CQF and at the same time to make cycle interval as short as possible.

3.2. Desire to support longer links in DetNet

As shown in Figure 2 for fundamental two buffer CQF, the last byte sent by node A in cycle (i-1) has to be ready for sending at node B before the start of cycle i. To realize it, DT or dead time is imposed. It is a time interval usually at the end of a cycle so that a node should not send the scheduled CQF packets.

Dead time is at least the sum of the maximum propagation delay to the next node, the maximum processing delay at the next node and the maximum other time variations. Therefore either the longer propagation or longer processing delay makes dead time larger. Packets from DetNet service is likely to be propagated over long links in the wider area. It takes around 5us per kilometer to propagate, i.e. 0.5ms every hundred kilometers. Hence the dead time can be as large as milliseconds or tens of milliseconds in case of hundred kilometers of longer links and larger processing delays. That would make the dead time eat up most of the cycle interval when cycle interval is short (e.g., at the same order or one order higher of magnitude in time as dead time). Then the useful time in a cycle

will be much reduced. In some extreme cases, when the link is long and the cycle interval is set to extremely short, the first packet sent in a cycle by a node will not be possibly received in the same cycle interval at the next node. That makes the useful time in a cycle reaches zero in two buffer CQF. Then two buffer CQF will be no longer suitable.

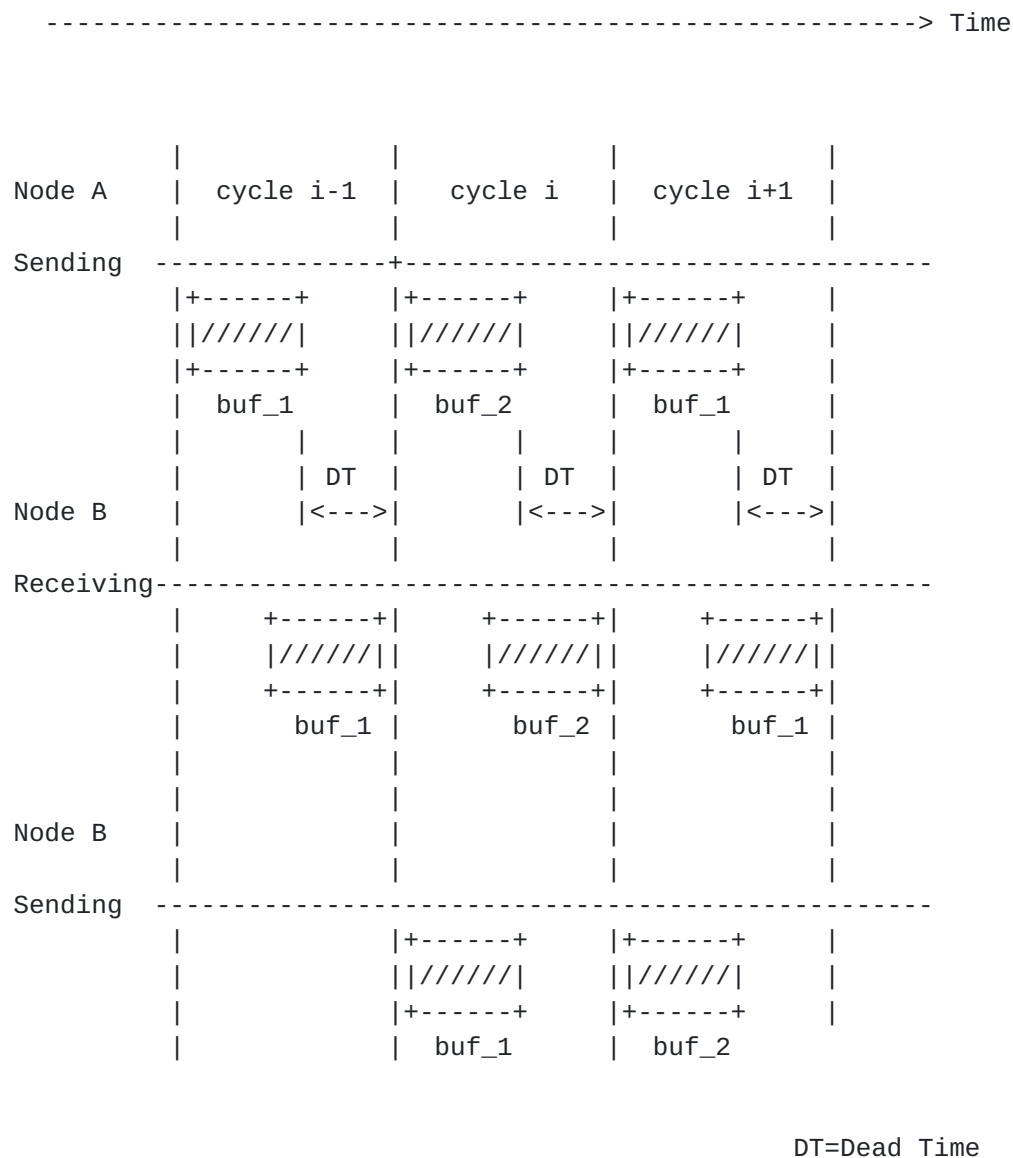


Figure 2: Fundamental Two Buffer CQF

In summary it is hard to achieve the followings at the same time in the fundamental two buffer CQF.

1. Shorter cycle interval to support lower end to end latency as shown in [Section 3.1](#).

2. Larger dead time to support longer link between neighbours.
3. Smaller ratio of dead time to cycle interval to improve utilization.

4. CQF Variants

CQF variants are used to solve the dilemma aforementioned with minor changes to the fundamental two buffer CQF. This section introduces the large number of buffers and cycle identification variants to CQF.

CQF can use more than two buffers to minimize the dead time and increase the useful time in a cycle so as to support long link delay. Figure 3 shows how a three buffer CQF works in a rotating manner in general. Node A sends packets in cycle (i-1). The time interval over which node B receives these packet spans two cycles, cycle (i-1) and cycle i. Hence a method is needed to make node B send them all at once in cycle (i+1) in order to ensure packets in a single cycle from the previous node always being sent out in one cycle at the current node.

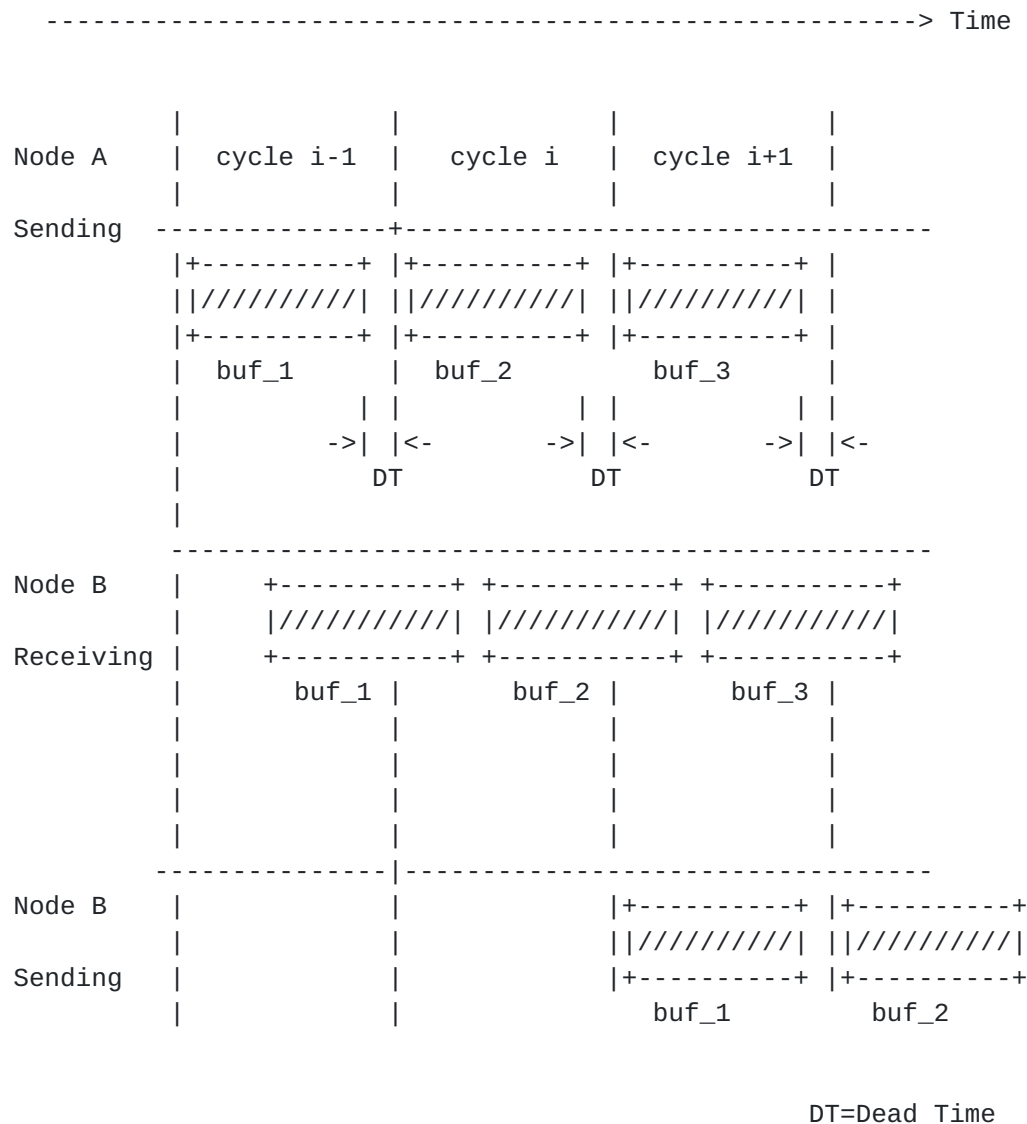


Figure 3: Three Buffer CQF

More than three buffers will be required when the receiving interval at node B for packets sent in a single cycle interval from node A spans over more than two cycle interval boundaries. This can happen when the time variance including propagation, processing, regulation and other factors between two neighbouring DetNet nodes is large. Note that due to the variance in time, the receiving interval at the downstream node can be much larger than one cycle interval in which the upstream node transmits. When time variance is large and cycle interval and dead time are set small, the possible receiving time of the last few packets from node A's cycle (i-1) at node B can overlap with the possible receiving time of the first few packets from node A's cycle i in different rounds of buffer rotations. Hence, when the buffer number is larger than two, if the receiving side still uses

the traditional CQF implicit time borderline to demarcate the receiving packets from the consecutive cycles of the upstream node, it may cause the ambiguity in identifying the right sending cycle at the upstream node and further affect the correctness of the decision of which output buffer to put the received packets at the current node.

Figure 4 shows such an ambiguity when time based cycle demarcation is used. The packet sent by node A in its cycle (i-1) can be received at any time in the receiving interval indicated as "receiving window for A's buf_1" in Figure 4. The receiving window refers to the time interval between the earliest time that the first packet sent in a given cycle from an upstream node is processed and enqueued in an output buffer and the latest time that the last packet of the cycle is processed and enqueued in an output buffer. Network operators may configure the size of the receiving window, taking the time variance of their networks into account. It can be seen that the spanning time period of receiving window is longer than the cycle interval. This is because there is a large time variance experienced between A and B, e.g. varying processing time for different packets in different cycles. It does not mean the receiving interval for every cycle always constantly span over such a large receiving window. The receiving window time interval indeed is determined by the worst case time variance value and that should be used for regular time cycle demarcation.

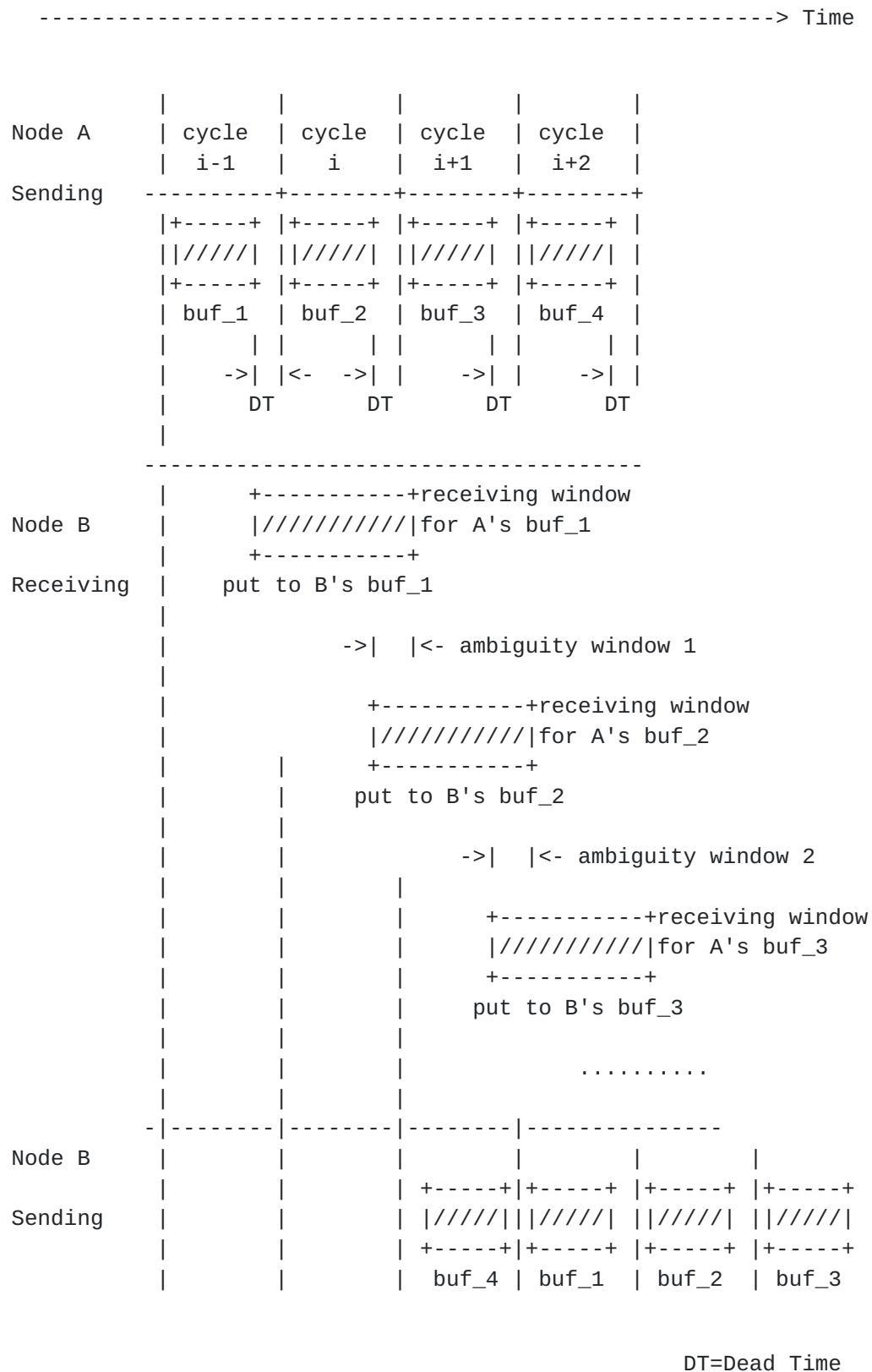


Figure 4: Ambiguity of time based cycle demarcation in CQF

When a packet is received in ambiguity window 1 in Figure 4, node B is not able to use the receiving time to determine which buffer is the correct one to put the packet in because it cannot tell if the packet is sent from cycle (i-1) or cycle i on node A. If node B puts the packet to the wrong output buffer, the packet may experience the unexpected delay. At the same time, the packet occupying the non-designated buffer may break the contracts between the end hosts and DetNet networks and then cause the unpredictable consequences.

It has been noted that the DT can be greatly increased to beat the time variance in order to make the receiving windows do not overlap so as to remove such ambiguity. However, it is not always practical and usually not desired because large DT will eat useful cycle time and bring the low utilization issue as illustrated in [Section 3](#). Therefore, it would be desired to keep DT as small as possible and at the same time identify the cycle interval correctly.

CQF variant carries the cycle identification in the data plane to help determine the correct output port buffer to place the data packets. It requires the IPv6 data plane enhancement which is introduced in next section. The configuration and forwarding logic makes no change from the fundamental CQF. The mapping relation from the upstream node A's output port cycle to the downstream node B's output port cycle should be determined in advance and stored on node B. Such CQF variants can be deployed in networks supporting frequency synchronization only, which alleviate the dependency on network-wide strict time synchronization. When calculating and determining the mapping relation, the time phase difference between neighboring nodes should be taken into consideration if only frequency synchronization is in use. Optionally the mapping relation can be dynamically changed when the network condition changes.

This document does not specify the mechanisms to determine the mapping relation since it can be easily deduced from fundamental CQF and does not require additional standardization. Some example can be found in section 5.1 in [[multipleCQF](#)].

5. The CQF-Variant Option

This document defines a new IPv6 option for DetNet to leverage the CQF variant queuing mechanism. This option is to be placed in the IPv6 HbH (Hop-by-Hop) Options or DOH (Destination Option Header) header.

5.1. CQF-Variant Option Format

The CQF-Variant Option helps the receiving port to identify in which time cycle interval the packet is sent from the upstream node. It can be used to determine the output port buffer to enqueue the packet.

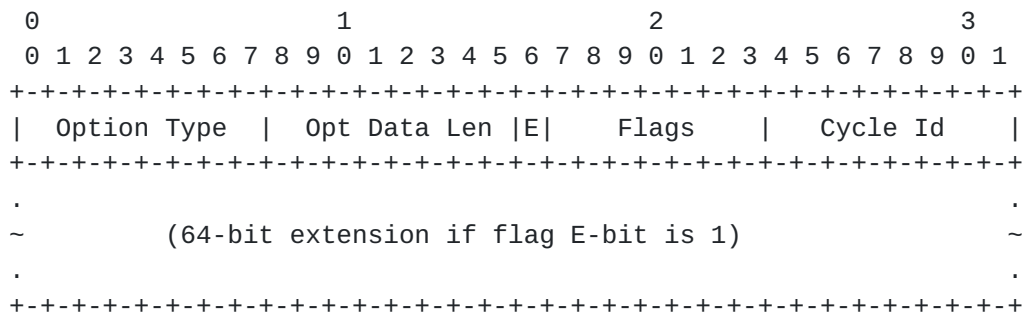


Figure 5: CQF-Variant Option Format Example

CQF-Variant Option fields:

- * Option Type: 8-bit identifier of the type of option. Value TBD by IANA. If the processing IPv6 node does not recognize the Option Type it must discard the packet and return an ICMP message (the highest-order 2 bits = 10). The Option Data of this option may change en route to the packet's final destination (the third-highest-order bit=1).
- * Opt Data Len: 8-bit length of the option data.
- * Flags: 8-bit field to indicate what CQF-Variant information followed. The leftmost bit is called E-bit. When E-bit set to 1, there is a 64-bit extension in length after Cycle Id.
- * Cycle Id: 8-bit field to indicate the time cycle ID at output port of the upstream node when the packet is sent out.
- * 64-bit extension: This field contains values required for a particular CQF variant, such as timestamp. This field exists only when E-bit in Flags field is set to one. [Editor's Note: Text will be modified or added as specific uses for this field are identified]

5.2. CQF-Variant Option Processing

A packet carrying CQF-Variant Option with Cycle Id does not change the fundamental cyclic queuing and forwarding behaviors of CQF. At the data plane, the packet transmitted from an output port carries an unambiguous cycle Id. There are different ways to manage the cycle id and output buffer. The simplest one is to map a buffer to a cycle id in a rotating manner. When the buffers rotate for transmission, cycle id also rotates. Packets in one buffer are sent all at once within a cycle interval and carry the same cycle id.

When a router receives a data packet with Cycle Id, it uses the cycle id to enqueue the packet to the correct output buffer that implicitly maps to a local cycle id for output transmission. Therefore the cycle id changes every hop.

Each node has the pre-computed and maintained mapping relation between the incoming cycle id of each input port and the output buffer number of the outgoing port. Such relation is determined in advance by measuring and/or configure the various delay components of the serviced DetNet flows as indicated in Figure 1 in [\[I-D.ietf-detnet-bounded-latency\]](#). Once the output buffer to place an incoming packet is determined, the cycle id value carried in later packet transmission is decided implicitly.

Cycle Id can be used as an implicit aggregated flow id and also is a quick way to identify the streams requiring DetNet service which provides an alternative to use 6-tuple to identify an IPv6 flow shown in [\[RFC8938\]](#).

6. Encapsulation of CQF-Variant Option

When used in IPv6 networks, the CQF-Variant option can be placed in an HbH extension header or DOH.

Figure 6 shows the encapsulation of CQF-Variant option in HbH extension header. When every DetNet forwarding node along the path is provisioned to use CQF variants as the queuing mechanism, this option should be placed here. If a router does not support this option, it discards the packet and returns an ICMP message.

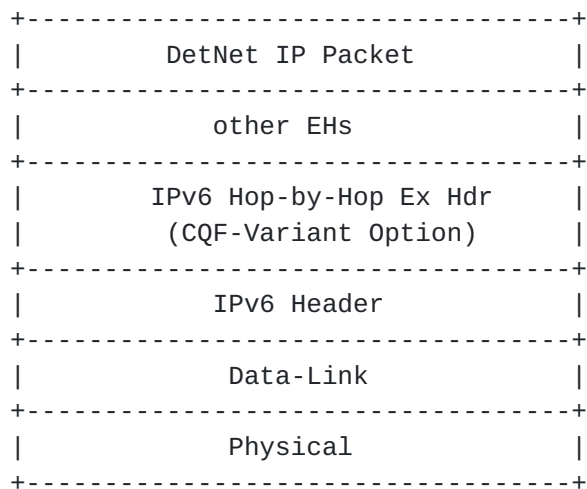


Figure 6: Example 1: CQF-Variant Option Encapsulated in HbH

In some deployments the path selection is indicated using IPv6 routing header (RH) by specifying a set of nodes that must be traversed by the packet along its path to the destination. When such a source routing mechanism is used, CQF-Variant Option is placed in DOH (Destination Option Header) as shown in Figure 7. Then CQF-Variant Option will be processed by the specified in-path routers.

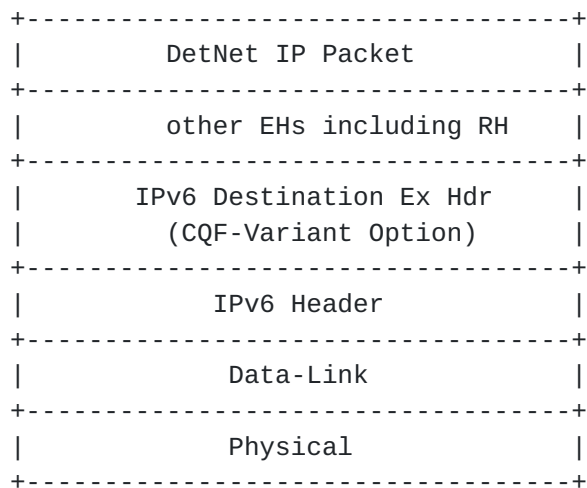


Figure 7: Example 2: CQF-Variant Option Encapsulated in DOH

(To be discussed: Should and how CQF-Variant Option to be placed in SRv6.)

7. Collaboration Considerations

[Editor's Note: This chapter will be deleted when the WG forms consent on final solution.]

From the comments and presentations in detnet session of IETF 114, the authors paid much attention on related solutions, e.g., [[I-D.yzz-detnet-enhanced-data-plane](#)] and [[I-D.eckert-detnet-tcqf](#)]. The former document introduced a general IPv6 Option of BLI(Bounded Latency Information) that can be carried in IPv6 extension header. introduced a general IPv6 Option of BLI(Bounded Latency Information) that can be carried in IPv6 extension header. To encapsulate cycle id and potential timestamp information by the way introduced in BLI, it will takes 5 Bytes and 12 Bytes correspectively see Figure 8. In this document, it will takes 2 Bytes and 10 Bytes for the option data. From the latter TCQF document, it will reuse DSCP code points in a single administrative domain. Only 4 bits of DSCP is available, it is unable to acommodate cycle id or timestamp information. Full discussion should be taken in detnet WG and final decision is significant to future proposal of solution of detnet.

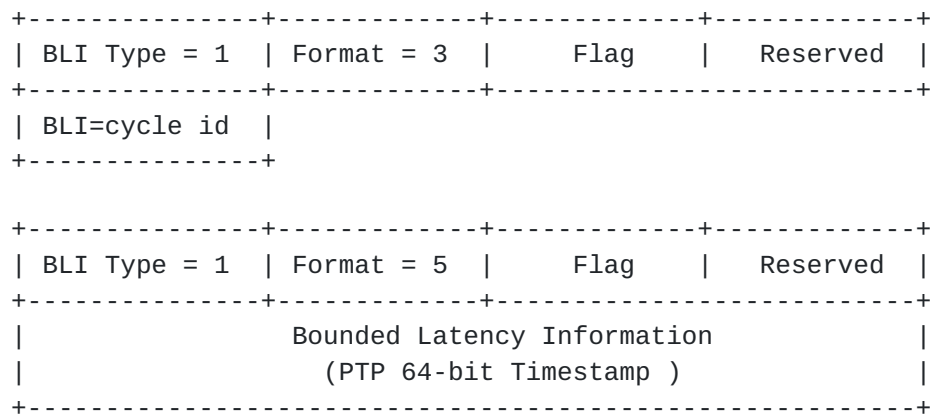


Figure 8: CQF-Variant Information in Format of BLI Options

8. Security Considerations

9. IANA Considerations

This document defines a new CQF-Variant Option for the “Destination Options and Hop-by-Hop Options” under the “Internet Protocol Version 6 (IPv6) Parameters” registry [[IPV6-PARMS](#)] with the suggested values in Figure 9.

Hexa	act	chg	rest	Description	Reference
0xB1	10	1	10001	CQF-Variant Option	this document

Figure 9: CQF-Variant Option Code in Destination Options and Hop-by-Hop Options

10. Contributors

The following co-authors have contributed to this document:

Xiaoliang Zheng Huawei Email: zhengxiaoliang@huawei.com

11. References

11.1. Normative References

- [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>>.

11.2. Informative References

- [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>>.

- [I-D.ietf-detnet-bounded-latency]
Finn, N., Le Boudec, J., Mohammadpour, E., Zhang, J., and B. Varga, "Deterministic Networking (DetNet) Bounded Latency", Work in Progress, Internet-Draft, [draft-ietf-detnet-bounded-latency-10](#), 8 April 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-detnet-bounded-latency-10>>.

- [I-D.yzz-detnet-enhanced-data-plane]
Geng, X., Zhou, T., Zhang, L., and Z. Du, "DetNet Enhanced Data Plane", Work in Progress, Internet-Draft, [draft-yzz-detnet-enhanced-data-plane-02](#), 24 December 2022, <<https://datatracker.ietf.org/doc/html/draft-yzz-detnet-enhanced-data-plane-02>>.

- [I-D.eckert-detnet-tcqf]
Eckert, T. T., Bryant, S., Malis, A. G., and G. Li, "Deterministic Networking (DetNet) Data Plane - Tagged Cyclic Queuing and Forwarding (TCQF) for bounded latency with low jitter in large scale DetNets", Work in Progress, Internet-Draft, [draft-eckert-detnet-tcqf-02](#), 13 March 2023, <<https://datatracker.ietf.org/doc/html/draft-eckert-detnet-tcqf-02>>.

[IEEE8021Q]

"IEEE Standard for Local and Metropolitan Area Network – Bridges and Bridged Networks", IEEE Std 802.1Q-2018, DOI 10.1109/ieeestd.2018.8403927, 2018, <<https://doi.org/10.1109/ieeestd.2018.8403927>>.

[multipleCQF]

Finn, N., "Multiple Cyclic Queuing and Forwarding", October 2021, <<https://www.ieee802.org/1/files/public/docs2021/new-finn-multiple-CQF-0921-v02.pdf>>.

[IPV6-PARMS]

IANA, ., "Internet Protocol Version 6 (IPv6) Parameters", n.d., <<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml>>.

Authors' Addresses

Yizhou Li
Huawei Technologies
Email: liyizhou@huawei.com

Shoushou Ren
Huawei Technologies
Email: renshoushou@huawei.com

Guangpeng Li
Huawei Technologies
Email: liguangpeng@huawei.com

Fan Yang
Huawei Technologies
Email: shirley.yangfan@huawei.com

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

Peng Liu
China Mobile
Email: liupengyjy@chinamobile.com

