

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

B. Liu
J. Dang, Ed.
Huawei
February 22, 2021

A Queuing Mechanism with Multiple Cyclic Buffers
draft-dang-queuing-with-multiple-cyclic-buffers-00

Abstract

This document presents a queuing mechanism with multiple cyclic buffers.

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 August 26, 2021.

Copyright Notice

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

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

Table of Contents

1.	Introduction	2
2.	Requirements Language	2
3.	Terminology & Abbreviations	2
4.	Problem Statement	2
5.	Queuing with Multiple Cyclic Buffers	3
6.	Security Considerations	5
7.	IANA Considerations	5
8.	Acknowledgements	5
9.	Normative References	5
	Authors' Addresses	5

[1.](#) Introduction

Network forwarding with bounded latency and zero congestion loss is important for various industrial applications. The DetNet working group draft "DetNet Bounded Latency"[[draft-ietf-detnet-bounded-latency](#)] describes requirements for queuing mechanisms. Among the referenced queuing mechanisms, Cyclic Queuing and Forwarding (CQF) requires no per-flow dynamic state at core nodes, which is scalable when the number of flows grows. To cope with long link delay, more than two cyclic buffers can be used.

This document discusses the details of the cyclic queuing mechanisms. We propose a queuing model and mechanism with multiple cyclic buffers, which can improve bandwidth utilization without sacrificing latency and jitter.

[2.](#) Requirements Language

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

[3.](#) Terminology & Abbreviations

CQF: Cyclic Queuing and Forwarding

T_c: the length of a cycle

[4.](#) Problem Statement

IEEE 802.1Q[IEEE8021TSN] defines CQF. As described in "DetNet Bounded Latency"[[draft-ietf-detnet-bounded-latency](#)], CQF with two synchronized buffers works as follows. All nodes keep a same cycle starting time. In a cycle x , Node A sends all packets in a buffer to

Node B. In the same cycle x , Node B uses a buffer to accumulate all packets from A, and at the same time sends out the packets that have already been buffered in cycle $x-1$. In the next cycle $x+1$, Node B sends out all the packets that are received from Node A in cycle x . If a packet traverses h hops, the maximum latency is $(h+1)T_c$, and the minimum latency is $(h-1)T_c$. The jitter (latency variation) bound is $2T_c$.

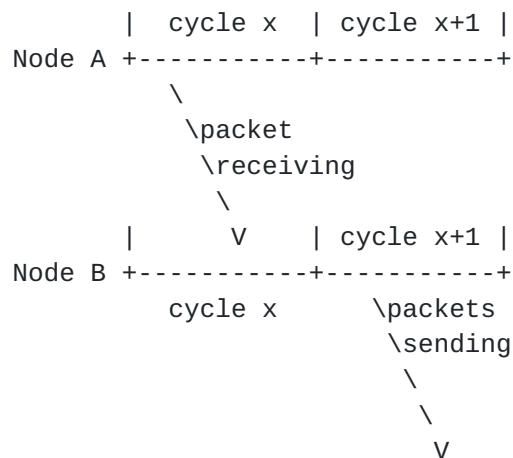


Figure-1: CQF with two synchronized buffers

The major disadvantages of CQF with two synchronized buffers are as follows. First, packets are sent and received in a same cycle by upstream and downstream nodes, respectively. So the link propagation delay must be smaller than the T_c . This prohibits the method from being used with long links, such as in WAN and MAN scenarios. Otherwise T_c must be larger than the link delay, resulting in high latency, jitter and buffer upper bound. Second, when the method is applicable, the sum of link delay, output delay, preemption delay and processing delay takes a portion of T_c , called dead time in [\[draft-ietf-detnet-bounded-latency\]](#), which cannot be used to send packets with deterministic services. This results in the conflict between good bandwidth utilization rate and good latency and jitter bound.

5. Queuing with Multiple Cyclic Buffers

This document proposes a cyclic queuing model that decouples link propagation delay with cycle length T_c . The model is shown in Figure-2.

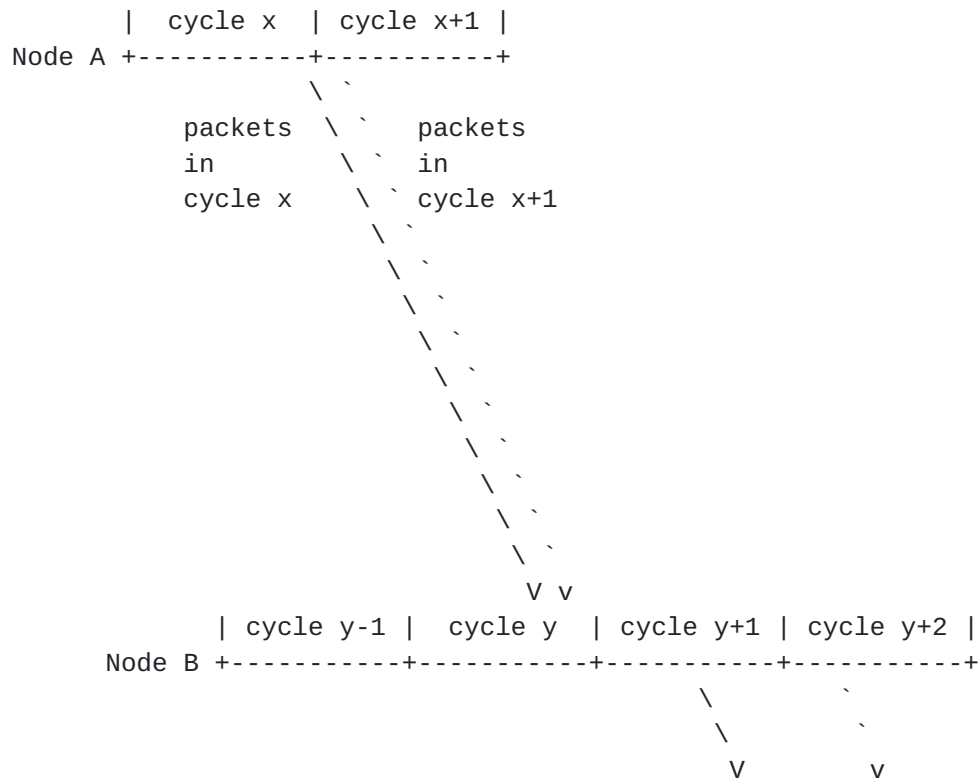


Figure-2: Proposed model for cyclic queuing

In this model, the cycle starting time points of different nodes can be either synchronized or not. However, the "phase difference" between neighbor nodes should be stable, and the variation should be bounded.

Since link delay and T_c are decoupled, a small T_c can be used with arbitrary link length without sacrificing bandwidth utilization. Any time range in a cycle can be used for deterministic traffic. In Figure-2, the last packet of the deterministic packets sent in cycle x by Node A is received by Node B in its local cycle y (all variations and jitters are considered). Node B stores those packets in a buffer associated with cycle y , and transmits the packets in this buffer in cycle $y+1$. The first packet of cycle $x+1$ may also be received by Node B in cycle y . So two or even more receiving buffers are needed simultaneously.

Since there is variation in link delay and output delay, the last packet in cycle x and the first packet in cycle $x+1$ may be difficult to distinguish to Node B, especially when the bandwidth utilization is high. To resolve the ambiguity, a cycle label can be put in a packet, which identifies which cycle the packet belongs with. Packets in different cycles carry different cycle labels. So Node B

can unambiguously distinguish the packet's sending cycle and map it in a correct local buffer even when the bandwidth is fully utilized.

There can be multiple ways to map a cycle label in a packet to a local cyclic buffer. For example, an ordered pair of neighboring nodes can learn a cycle mapping table. Node B may receive packets from different upstream nodes that carry different cycle labels. Node B can use the mapping table to swap the labels to a same local cycle label, and put the packets into a same local buffer. Or, an upstream node can swap a local label to a downstream label before transmitting the packet. Another way is to put a label stack in the packet, so every hop just pops a label and maps it to a local buffer. Further study is required to define how cycle labels are formatted and processed and how mapping tables are learned.

6. Security Considerations

TBD

7. IANA Considerations

TBD

8. Acknowledgements

TBD

9. Normative References

[[draft-ietf-detnet-bounded-latency](#)]

"DetNet Bounded Latency", <<https://tools.ietf.org/html/draft-ietf-detnet-bounded-latency>>.

[IEEE8021TSN]

"IEEE 802.1 Time-Sensitive Networking (TSN) Task Group", <<http://www.ieee802.org/1/>>.

[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>>.

Authors' Addresses

Bingyang Liu
Huawei
No.156 Beiqing Road
Beijing, P.R. China 100095
China

Email: liubingyang@huawei.com

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

Email: dangjuanna@huawei.com

