

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2020

L. Geng
China Mobile
P. Willis
BT
S. Homma
NTT
L. Qiang
Huawei
July 8, 2019

Requirements of Layer 3 Deterministic Latency Service draft-geng-detnet-requirements-bounded-latency-03

Abstract

This document specifies some technical, operational and management requirements of deploying deterministic latency service on layer 3 networks from the perspective of the service provider.

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 January 9, 2020.

Copyright Notice

Copyright (c) 2019 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](#)
- [1.1. Requirements Language](#) [3](#)
- [1.2. Terminology & Abbreviations](#) [3](#)
- [2. Technical Requirements](#) [3](#)
- 2.1. Requirement 1: Must support the dynamic creation, modification and deletion of deterministic services . . . [3](#)
- 2.2. Requirement 2: Should tolerate a certain degree of time variance [4](#)
- 2.2.1. Sub-requirement 2.1: Should support asynchronous clocks across domains [4](#)
- 2.2.2. Sub-requirement 2.2: Should tolerate clock jitter & wander within a clock synchronous domain [4](#)
- 2.3. Requirement 3: Must support Inter-Continental propagation delay [5](#)
- [3. Operational and Management Requirements](#) [6](#)
- [3.1. Requirement 4: Should have self-monitoring capability](#) . . [6](#)
- 3.2. Requirement 5: Should be robust against denial of service attacks [6](#)
- 3.3. Requirement 6: Must tolerate failures of links or nodes and topology changes [7](#)
- [3.4. Requirement 7: Must be scalable](#) [7](#)
- 3.4.1. Sub-requirement 7.1: Must be scalable to numerous network devices [7](#)
- 3.4.2. Sub-requirement 7.2: Must be scalable to massive traffic flows [7](#)
- [4. IANA Considerations](#) [7](#)
- [5. Security Considerations](#) [8](#)
- [6. Acknowledgements](#) [8](#)
- [7. Normative References](#) [8](#)
- Authors' Addresses [9](#)

1. Introduction

DetNet is chartered to provide bounds on latency, jitter (delay variation) and packet loss [[draft-ietf-detnet-problem-statement](#)]. Where latency and jitter are two closely related performance characteristics, this document refers to bounded latency and bounded jitter collectively as deterministic latency.

This document does not intend to define any specific mechanism, but specifies some technical, operational and management requirements

that must be considered when deploying deterministic latency service at layer 3.

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.

1.2. Terminology & Abbreviations

This document uses the terminology defined in [\[draft-ietf-detnet-architecture\]](#).

TSN: Time Sensitive Network

2. Technical Requirements

2.1. Requirement 1: Must support the dynamic creation, modification and deletion of deterministic services

There are at least two modes to provide deterministic service over an operator's network: 1) deterministic VPN; 2) point-to-point deterministic tunnel. No matter in which mode, the layer 3 deterministic latency mechanisms must be able to support the dynamic creation, modification and deletion of deterministic services without affecting any deterministic services that are already running in the network.

In a local area network such as a factory, the information about when a deterministic service will start, how long the service will last, can be known in advance, or can even be planned. Based on this information, the local area network can adopt a global programming mechanism to calculate the accurate processing behaviors for each device, and achieve a global optimal performance. However, such global programming mechanisms are unsuitable for service providers' networks. Many deterministic applications are expected to running on a service provider's network simultaneously. Different deterministic applications may have different lifecycles and SLA requirements, hence the network state changes dynamically. If a mechanism relies on a stable network state for global computing, any change in network state (e.g., new application starts, or an application finishes, or SLA requirement changes) will lead to re-computing, even worse if all devices need to stop work and install the recomputed results, then this mechanism is hard to be deployed on service provider's network.

2.2. Requirement 2: Should tolerate a certain degree of time variance

2.2.1. Sub-requirement 2.1: Should support asynchronous clocks across domains

One of DetNet's objectives is to stitch TSN islands together. All devices inside a TSN domain are time-synchronized, and most of TSN technologies rely on precise time synchronization. However, different TSN islands may have different clocks which are not synchronized as shown in Figure 1, where the time difference of two TSN domain is D. DetNet needs to connect these two TSN domains together and provide end-to-end deterministic latency service. The mechanism adopted by DetNet should be able to support the interaction across time domains by using a fine controlled buffer (the time difference 'D' may be us-level) to absorb the time difference, or relying on the mechanism itself to implement cross domain time mapping.

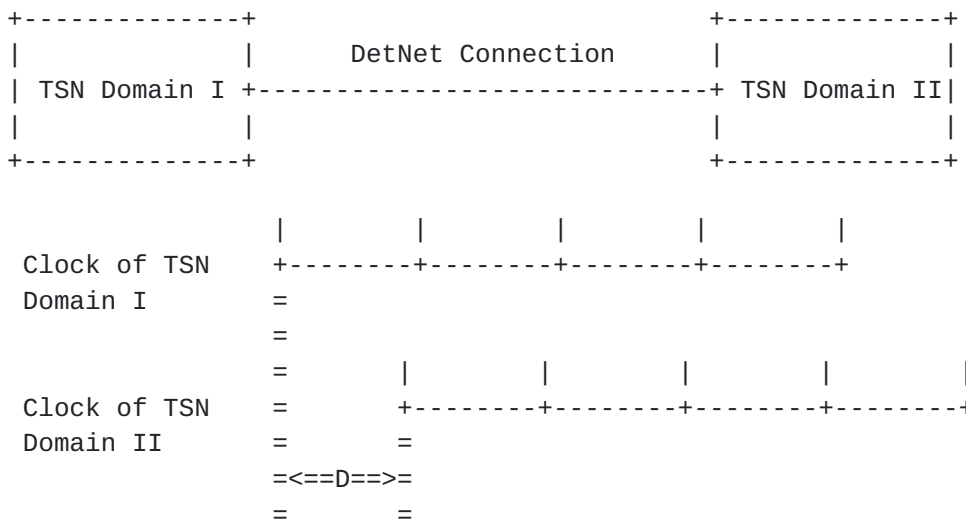


Figure 1: TSN islands interconnecting

2.2.2. Sub-requirement 2.2: Should tolerate clock jitter & wander within a clock synchronous domain

DetNet domain itself can be time synchronous or asynchronous, depending on actual deployment, application, use cases, etc. Even within a time synchronous domain, the synchronized clocks may also experience jitter & wander. Different areas have different clock accuracy, for example the crystal oscillator in Ethernet is specified at 100 ppm [[Fast-Ethernet-MII-clock](#)], SyncE can achieve 50 ppb [[G.8262](#)], and more precise time synchronization [[G.8273](#)] is expected in 5G mobile backhaul. Hence the mechanisms adopted by DetNet should

be able to tolerate a certain degree of clock jitter & wander accordingly.

2.3. Requirement 3: Must support Inter-Continental propagation delay

In contrast to Layer 2 TSN that is deployed on a LAN [[IEEE-TSN](#)], DetNet is expected to be deployed in a larger scale carrier networks that have long link propagation delay which means that DetNet must work on network links that have inter-continental propagation delays. Long link propagation delay can cause some troubles to cyclic forwarding type mechanisms. In order to guarantee deterministic latency, cyclic forwarding type mechanisms usually require a packet be sent out (or received) at a particular cycle, rather than be sent out (or received) randomly. There is a mapping between the sending cycle of upstream node and the receiving cycle of downstream node. In a local area network that with short link propagation delay, the cycle mapping relationship could be very simple.

As an example shown in Figure 2, where the length of a cycle is 10 us, and the sending cycle of upstream node X and the receiving cycle of downstream node Y correspond to the same period of time (e.g., 0~10 us). Packets sent from X at sending cycle will arrive at downstream node Y at receiving cycle, and the link propagation delay between X and Y is smaller than the length of a cycle (i.e., 10 us).

Suppose a large scale network wants to keep using this simple cycle mapping relationship, however the link distance between two nodes is longer. Moreover, a downstream node may have many upstream nodes each with different link propagation delays (e.g., 9 us, 10 us, 11 us, 15 us, 20 us). In order to absorb the longest link propagation delay, then the length of cycle must be set to at least 20 us. However since packet's arrival time varies within the receiving cycle, larger cycle length means larger delay variance.

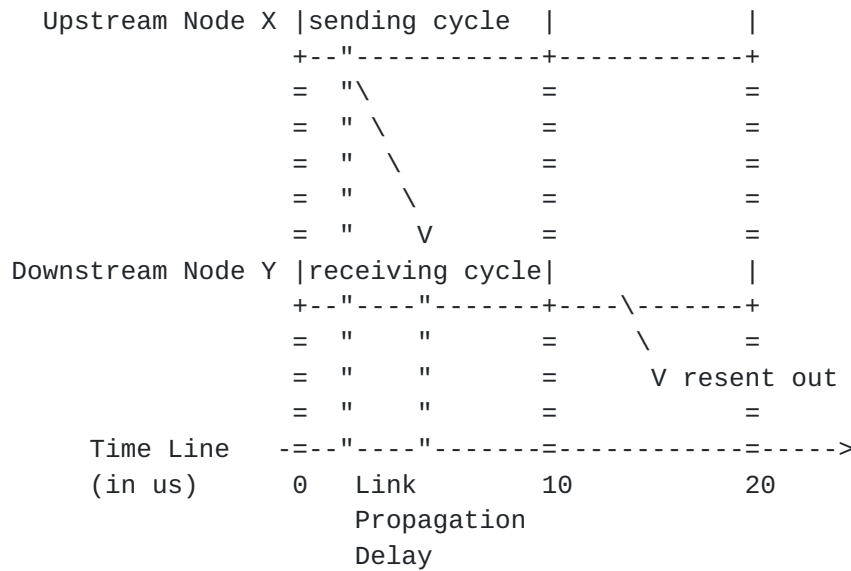


Figure 2: An example of limited link

3. Operational and Management Requirements

[Authors note: more explanations for each requirement need to be added in later versions.]

3.1. Requirement 4: Should have self-monitoring capability

Both network operators and end-users need to be able to measure if the deterministic latency service is working correctly and meeting its SLA. Once the monitored results exceed the expected bounds, network operators and end-users should be notified, and some service protection mechanisms should also be triggered accordingly. In addition, network operators can input the collected results into a reporting system and produce the latency (and jitter) distribution over a period of time, which would be helpful for operators in understanding their networks performance.

However, such fine-granularity monitoring is costly. Hence although the self-monitoring is an important capability to both operators and customers, it is not recommended as a mandatory requirement until the use cases are clear.

3.2. Requirement 5: Should be robust against denial of service attacks

To protect the services requiring deterministic latency, the mechanisms implemented by DetNet should be robust to denial of service attacks. This includes robustness against attacks on the mechanisms to generate clock synchronization.

[[draft-ietf-detnet-architecture](#)] has discussed using the traffic

admission control at the ingress or through the fault mitigation methods, to prevent (or mitigate) the excess traffic caused by malicious or malfunction devices. DetNet also considers using authentication and authorization to mitigate man-in-the middle attacks

3.3. Requirement 6: Must tolerate failures of links or nodes and topology changes

A step change in latency due to a single network topology change is inevitable. However if the topology keeps changing many times, then DetNet may not deliver on its SLA. The topology changes alone may not be sufficient on a traditional IP network to raise any alarms, but the mechanism's self-monitoring should detect this, and keep working in flapping network topologies.

3.4. Requirement 7: Must be scalable

Further to the requirement to work on inter-continental links, the deterministic latency forwarding mechanisms must scale to networks of significant size with numerous network devices and massive traffic flows.

3.4.1. Sub-requirement 7.1: Must be scalable to numerous network devices

A simple use case to understand is ultra-low-latency (public) 5G transport networks, which would require DetNet extend to every 5G base station. For some network operators, their network may need to connect to ~100 K base stations (serving multiple MNOs'), and this number will only increase with 5G.

3.4.2. Sub-requirement 7.2: Must be scalable to massive traffic flows

If each ultra-low-latency slice or MNO is treated as a separate deterministic latency traffic flow (or tunnel), then even if each base station has a limited number of ultra-low latency slices or MNOs (e.g. ~10), there will still be a lot of, ~1M, deterministic latency traffic flows on one network simultaneously.

4. IANA Considerations

This document makes no request of IANA.

5. Security Considerations

This document will not introduce new security problems.

6. Acknowledgements

The Authors would like to thank David Black, Stewart Bryant for their review, suggestion and comments to this document.

7. Normative References

[[draft-ietf-detnet-architecture](#)]

"DetNet Architecture", <<https://datatracker.ietf.org/doc/draft-ietf-detnet-architecture/>>.

[[draft-ietf-detnet-problem-statement](#)]

G.8262 : Timing characteristics of a synchronous Ethernet equipment slave clock
G.8262 : Timing characteristics of a synchronous Ethernet equipment slave clock, "DetNet Problem Statement", <<https://datatracker.ietf.org/doc/draft-ietf-detnet-problem-statement/>>.

[Fast-Ethernet-MII-clock]

"Fast Ethernet MII clock",
<<http://www.ti.com/lit/ds/symlink/dp83865.pdf>>.

[G.8262] "G.8262 : Timing characteristics of a synchronous Ethernet equipment slave clock",
<<https://www.itu.int/rec/T-REC-G.8262>>.

[G.8273] "G.8273: Framework of phase and time clocks",
<<https://www.itu.int/rec/T-REC-G.8273/en>>.

[IEEE-TSN]

"IEEE TSN Task Group",
<<http://www.ieee802.org/1/pages/tsn.html>>.

[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>>.

Authors' Addresses

Liang Geng
China Mobile
Beijing
China

Email: gengliang@chinamobile.com

Peter Willis
BT
BT Adastral Park
Ipswich IP5 3RE
UK

Email: peter.j.willis@bt.com

Shunsuke Homma
NTT
Tokyo
Japan

Email: shunsuke.homma.fp@hco.ntt.co.jp

Li Qiang
Huawei
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: qiangli3@huawei.com

