Workgroup: DetNet
Internet-Draft:
draft-mohammadpour-detnet-bounded-delay-variation-00
Published: 10 September 2021
Intended Status: Informational
Expires: 14 March 2022

Authors: E. Mohammadpour   J-Y. Le Boudec
         EPFL   EPFL

DetNet Bounded Packet-Delay-Variation

Abstract

Some DetNet use-cases (applications) require guaranteed bounds on packet delay-variation, not just on latency. This document gives a methodology to derive guaranteed packet delay-variation bounds in DetNet and apply it to a number of proposed mechanisms. When the required packet delay-variations is very low, clock non-idealities affect the bounds, even in a synchronized DetNet networks. This document also gives a methodology to account for such an effect.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Some applications that use DetNet networks, such as such as industrial Internet of Things [ITU-Y3000] and electrical utilities [RFC8578], require not just guaranteed bounds on the worst-case packet delay, but also on packet delay variation (PDV), defined as the difference between worst-case and best-case delays.

A general framework to compute latency bounds is presented in [I-D.ietf-detnet-bounded-latency]. In this document, we extend this framework to compute guaranteed bounds on PDV.

When the packet-delay-variation requirement is very low, even in a time-synchronized DetNet network, clock non-idealities affect the bounds, as seen, e.g., in [MohammadpourDamper]. This document gives a methodology, derived from [ThomasTime], to incorporate such effects in the computation of packet-delay-variation bounds within DetNet.

This document also applies the presented framework to compute packet-delay-variation bounds on a number of packet scheduling mechanisms, some of which are taken from [RFC8655] and [I-D.ietf-detnet-bounded-latency], while others are specifically targetting low PDV. Finally, this document gives an application of the framework to compute end-to-end packet-delay-variation bounds on a sample DetNet network with a combination of various packet scheduling mechanisms.

2. Terminology and Definitions

This document uses the terms defined in [RFC8655]. This document also uses the following terms:.

PDV

Packet Delay-Variation as in [RFC3393]. It is also called "latency variation" or "jitter" in [RFC8655].

3. Clock Model

We call H_TAI the perfect clock, i.e. the international atomic time (Temps Atomique International). In practice, the local clock of a system deviates from the perfect clock [ThomasTime]. In time-sensitive networks, clocks can be synchronized or non-synchronized. Non-synchronized clocks are independently configured and do not interact with each other; this corresponds to the free-running mode in Section 4.4.1 of [g810]. When clocks are synchronized, using methods like Network Time Protocol (NTP), Precision Time Protocol (PTP), WhiteRabbit, Global Positioning System (GPS), the occurrence of an event, when measured with different clocks, is bounded by the time error bound (~1us or less in PTP, WhiteRabbit, and GPS; ~100ms in NTP).

This document follows the clock model in [ThomasTime], which applies to time-sensitive networks. Consider a clock H_i that is either synchronized with time error bound &#969;, or not synchronized (in which case we set &#969;=+&#8734;). Let d^H_i [resp. d^H_TAI] be a delay measurement done with clock H_i [resp. in TAI], then [ThomasTime]:

d^H_TAI - d^H_i <= min((&#961;-1) * d^H_i + &#951;, 2&#969;),

d^H_TAI - d^H_i >= - min((1- 1/&#961;) * d^H_i+ &#951;/&#961;, 2&#969;),

where &#961; is the stability bound and &#951; the timing-jitter bound of the clock H_i. Note that this set of bounds is symmetric, i.e. we can exchange the roles of H_i and H_TAI in the above equation. We assume that the parameters &#969;, &#961;,&#951; are valid for all clocks in the network, i.e. we consider network-wide time-error, stability and time-jitter bounds.

4. Computing End-to-end Packet-Delay-Variation Bound

Computation of end-to-end PDV bound requires a time model that includes all the sources of latency within a flow path. In this document we use the existing time model presented in [I-D.ietf-detnet-bounded-latency].

4.1. DetNet Time Model

Figure 1 is a breakdown of the per-hop latency experienced by a packet passing through a DetNet transit node, taken from [I-D.ietf-detnet-bounded-latency].

```
      DetNet transit node A            DetNet transit node B
   +-------------------------+       +------------------------+
   |              Queuing    |       |              Queuing   |
   |   Regulator subsystem   |       |   Regulator subsystem  |
   |   +-+-+-+-+ +-+-+-+-+   |       |   +-+-+-+-+ +-+-+-+-+  |
-->+   | | | | | | | | | +   +------>+   | | | | | | | | | +  +--->
   |   +-+-+-+-+ +-+-+-+-+   |       |   +-+-+-+-+ +-+-+-+-+  |
   |                         |       |                        |
   +-------------------------+       +------------------------+
   |<->|<------>|<------->|<->|<---->|<->|<------>|<------>|<->|<--
2,3  4      5        6      1    2,3   4      5        6     1   2,3
                1: Output delay             4: Processing delay
                2: Link delay               5: Regulation delay
                3: Frame preemption delay   6: Queuing delay
```

Figure 1: Timing model for DetNet or TSN

4.2. Methodology

Consider a DetNet flow (or an aggregate of DetNet flows). The end-to-end delay-variation for the flow is defined as difference between the end-to-end worst-case and best-case latencies of its packets, measured in TAI,

```
e2e_PDV = e2e_worst_case_latency - e2e_best_case_latency.
```

"V" is an upper-bound on the end-to-end PDV of the flow if and only if for any two packets n and m with end-to-end latencies "d_n" and "d_m"

```
|d_n - d_m| <= V.
```

Then V is computed as:

V = e2e_latency_upper_bound - e2e_latency_lower_bound.

A general framework to compute end_to_end_latency_upper_bound is described in [I-D.ietf-detnet-bounded-latency]. The same framework can be used to compute e2e_latency_lower_bound; in Section 5, we provide the bound for a set of queuing mechanisms.

5. Packet Scheduling Techniques

This section provides formulas to compute PDV bounds for a number of packet scheduling mechanisms within DetNet networks.

5.1. Guaranteed-Service IntServ

TBD from [I-D.ietf-detnet-bounded-latency] and [RFC2212].

5.2. Differentiated Services

TBD from [RFC2475] and [RFC7657].

5.3. Credit-Based Shaper with Asynchronous Traffic Shaping

TBD from [I-D.ietf-detnet-bounded-latency]

5.4. Cyclic Queuing and Forwarding (CQF)

TBD from [IEEE8021Q]

5.5. Dampers

Dampers are proposed to reduce packet delay-variation in time-sensitive networks [VermaJitter],[ZhangRCSP],[CruzScedPlus]. A damper delays every DetNet packet by an amount written in a packet header field, called damper header, which carries an estimate of the earliness of this packet with respect to a known latency upper-bound of upstream systems. This ideally leads to zero PDV; in practice, there is still some small residual PDV, due to errors in acquiring timestamps and in computing and implementing delays. As a positive side effect, dampers create packet timings that are almost the same as at the source, with small errors due to residual PDV, and thus cancel most of the burstiness increase imposed by the network. The residual burstiness increase that remains when dampers are used is not influenced by the burstiness of cross-traffic. Thus, dampers solve the burstiness cascade issue [CharnyDelay]: individual flows that share a resource dedicated to a class may see their burstiness increase, which may in turn increase the burstiness of other downstream flows. Furthermore, dampers are stateless; hence, solving the burstiness cascade in a stateless manner makes the dampers of interest for DetNet networks.

We call jitter-compensated system (JCS) any delay element or aggregate of delay elements with known latency and PDV bounds, for which we want to compensate PDV by means of dampers. This is typically the queuing system on the output port of a switch or router used in DetNet transit nodes. It can also be a switching fabric or an input port processing unit, or even a larger system. For DetNet flows, a JCS should be able to time stamp packet arrivals and departures using the available local times. It should also increment the damper header field in every DetNet packet (if one is present) by an amount equal to an estimate of the earliness of this packet with respect to the known latency upper-bound &#948; at the JCS. If no damper header is present, it inserts one, with a value equal to the estimated earliness. The earliness is computed as:

```
   earliness = &#948; - actual_delay_in_the_JCS.
```

When a DetNet flow crosses a JCS, for actual PDV removal to occur, there must be a downstream damper on the path of the flow [MohammadpourDamper] (e.g. if the JCS is a switch output port, the next downstream damper is typically located on the output port of the next downstream switch). The damper also resets the damper header, so that the next downstream damper will see only the earliness accumulated downstream of this damper. Designing a stand-alone damper is a challenge, because such a damper may need to release a large number of packets instantly or within a very short time, which might not be feasible. This is why damper implementations are often associated with queuing systems; then, the time at which a damper releases a packet is simply the time at which the packet becomes visible to the queuing system.

It is generally not possible, or required, to remove PDV in all network elements, because time stamping and damper-header update come with a cost. Therefore, it is required, for our timing analysis, to consider what we call bounded-delay systems (BDSs), defined as any delay element or aggregate of delay elements with known latency and PDV bounds, and for which we do not compensate PDV. Constant latency elements (e.g. an output link propagation delay), variable delay elements with very low jitter (e.g., very high speed backbone network) are examples of BDSs.

We classify and model existing designs of dampers in next subsections and give formulas for the computation of PDV and latency bounds, taken from [MohammadpourDamper].

5.5.1. Damper Classification

An ideal damper delays a packet by exactly the amount required by the damping header. Consider a packet n with damper header H_n that arrives at local time Q_n to a damper. Then an ideal damper releases the packet at time E_n:

E_n = Q_n + H_n.

Jitter-control Earliest-Deadline-First [VermaJitter] is an ideal damper, used in combination with an Earliest-Deadline-First scheduler.

Many other implementations of dampers use some tolerance for the packet release times, due to the difficulty of implementing exact timings. We call damper with tolerances &#916;^L,&#916;^U, a damper such that the eligibility time E_n, of packet n, in local time, satisfies:

Q_n + H_n - &#916;^L >= E_n <= Q_n + H_n + &#916;^U.

The tolerances can vary from hundreds of nanosecond to a few microsecond based on implementation. RCSP [ZhangRCSP] is an instance of dampers with tolerance. Since the definition of damper with tolerance does not preclude packet misordering, re-sequencing and head-of-line dampers avoid packet misordering.

Re-sequencing damper with tolerances &#916;^L,&#916;^U is a system that behaves as the concatenation of a damper with same tolerances and a re-sequencing buffer that, if needed, re-orders packets based on the packet order at the entrance of the damper. The packet order is with respect to a flow of interest. SCED+ [CruzScedPlus] is an instance of re-sequencing dampers.

Head-of-line damper is introduced in [GrigorjewJCATS] and is implemented as a FIFO queue. It has tolerance parameters &#916;^L,&#916;^U as well as processing bounds &#966;^min,&#966;^max. A head-of-line damper behaves as re-sequencing damper with with tolerances &#916;^L,&#916;^U followed by a single-server FIFO queue with processing bounds &#966;^min,&#966;^max. When a packet arrives, its arrival time is collected and the packet is stored at the tail of the queue. Only the packet at the head of the queue is examined; if its eligibility time is passed, it is immediately released, otherwise it is delayed and released at its eligibility time. When the head packet is released, it is removed from the damper queue and the next packet (if any) becomes the head of the queue and is examined. When an arriving packet finds an empty queue, it is immediately examined. By construction, packet ordering is preserved.

5.5.2. Bound Computations

In this subsection, we provide latency and PDV bounds for a simple case (general case is available in [MohammadpourDamper]) where we want to compensate the PDV imposed by the queuing delay (6) in Figure 1 by the mean of dampers. Therefore the queuing subsystem is a JCS with a known latency upper-bound (the latency upper-bound includes the delay from first-bit-in to last-bit-in hidden in the link delay (2) of Figure 1). A damper is placed before the queuing subsystems (replaced the regulator in Figure 1). The delays (1), (2) only the first-bit-out to first-bit-in, (3), (4) in Figure 1 are assumed to be the BDSs.

Assume that the queuing subsystem has latency upper-bound &#948; and PDV J, and the latency upper-bound, lower-bound and PDV bound on the BDSs are &#960;, &#960;' and v. Also, assume that a damper with tolerances &#916;^L,&#916;^U is placed before the queuing subsystem in the DetNet transit node B. Then, the latency lower-bound, upper-bound and the PDV bound from the entrance of queuing subsystem in transit node A to the entrance of queuing subsytem in transit node B, in TAI, are computed as follows.

If damper with tolerances &#916;^L,&#916;^U is used:

latency_lower-bound = &#948; + &#960;' - &#916;^L - &#949; - &#968;' ,

latency_upper-bound = &#948; + &#960; + &#916;^U + &#949; + &#968; ,

PDV_bound = v + 2 * &#949; + &#968; + &#968;' ,

where

&#968;' = min((&#961;-1) * (&#948; + &#916;^U + &#949;) + 2 * &#951; , 4 * &#969;),

&#968; = min((1-1\&#961;) * (&#948; - &#916;^L + -&#949;)) + 2 * &#951;/&#961; , 4 * &#969;).

If re-sequencing damper with tolerances &#916;^L,&#916;^U is used and all the other elements are FIFO, the same bounds as mentioned above is obtained.

If head-of-line damper with tolerances &#916;^L,&#916;^U and processing bounds &#966;^min,&#966;^max is used and all the other elements are FIFO, the latency_upper-bound and PDV_bound are increased by &#952; where

&#952; = ((b + r * PDV_bound) * &#966;^max ; if &#966;^max <= 1/r,

&#952; = +&#8734; ; if &#966;^max > 1/r, where the DetNet flow has per-packet leaky-bucket arrival curve at the entrance of queuing subsystem at DetNet transit node A, i.e., the number of packets that can be emitted by the flow within any period of time t is not larger than r * t + b where r is the rate of packets and b is bucket size in packets.

When an element is not FIFO in Figure 1, comparing to dampers with tolerance, using a re-sequencing damper worsens the PDV bound by J and a head-of-line damper worsens the PDV bound by 2 * J; further discussion is available in [MohammadpourDamper].

5.6. Mechanism XXX

TBD

5.7. Mechanism XXX

TBD

5.8. Mechanism XXX

TBD

6. Example application on DetNet IP network

TBD

7. Security considerations

Detailed security considerations for DetNet are cataloged in [RFC9055], and more general security considerations are described in [RFC8655].

8. IANA considerations

This document has no IANA actions.

9. References

9.1. Normative References

[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>.

9.2. Informative References

[CharnyDelay]

A. Charny and J.-Y. Le Boudec, "Delay Bounds in a Network with Aggregate Scheduling", 2002, <https://link.springer.com/chapter/10.1007/3-540-39939-9_1>.

[CruzScedPlus] R. L. Cruz, "SCED+: efficient management of quality of service guarantees", 1998, <https://ieeexplore.ieee.org/document/665083>.

[g810] L. Thomas and J.-Y. Le Boudec, "G.810 : Definitions and terminology for synchronization networks", <https://www.itu.int/rec/T-REC-G.810-199608-I/en>.

[GrigorjewJCATS] A. Grigorjew, F. Metzger, T. Hossfeld, J. Specht, F.-J. Goetz, F. Chen, and J. Schmitt, "Asynchronous Traffic Shaping with Jitter Control", 2020, <https://opus.bibliothek.uni-wuerzburg.de/frontdoor/index/index/docId/20582>.

[I-D.ietf-detnet-bounded-latency] N. Finn, J-Y. Le Boudec, E. Mohammadpour, J. Zhang, B. Varga, and J. Farkas, "DetNet Bounded Latency", <https://www.ietf.org/archive/id/draft-ietf-detnet-bounded-latency-07.html>.

[IEEE8021Q] IEEE 802.1, "IEEE Std 802.1Q-2018: IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks", 2018, <http://ieeexplore.ieee.org/document/8403927>.

[ITU-Y3000] ITU-T, "ITU-T Y.3000-series - Representative use cases and key network requirements for Network 2030", 2020, <https://www.itu.int/rec/T-REC-Y.Sup67-202007-I>.

[MohammadpourDamper] E. Mohammadpour and J.-Y. Le Boudec, "Analysis of Dampers in Time-Sensitive Networks with Non-ideal Clocks", 2021, <https://arxiv.org/abs/2109.02757>.

[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>.

[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>.

[RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <https://www.rfc-editor.org/info/rfc3393>.

[RFC7657]
Black, D., Ed. and P. Jones, "Differentiated Services (Diffserv) and Real-Time Communication", RFC 7657, DOI 10.17487/RFC7657, November 2015, <https://www.rfc-editor.org/info/rfc7657>.

[RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <https://www.rfc-editor.org/info/rfc8578>.

[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>.

[ThomasTime] L. Thomas and J.-Y. Le Boudec, "On Time Synchronization Issues in Time-Sensitive Networks with Regulators and Nonideal Clocks", 2020, <https://dl.acm.org/doi/10.1145/3393691.3394206>.

[VermaJitter] D.C. Verma, H. Zhang, and D. Ferrari, "Delay jitter control for real-time communication in a packet switching network", 1991, <https://ieeexplore.ieee.org/abstract/document/152873>.

[ZhangRCSP] H. Zhang and D. Ferrari, "Rate-controlled static-priority queueing", 1993, <https://ieeexplore.ieee.org/abstract/document/253355>.

Authors' Addresses

Ehsan Mohammadpour
EPFL
IC Station 14
CH-1015 Lausanne EPFL
Switzerland

Email: ehsan.mohammadpour@epfl.ch

Jean-Yves Le Boudec
EPFL
IC Station 14
CH-1015 Lausanne EPFL
Switzerland

Email: jean-yves.leboudec@epfl.ch