DetNet
Internet-Draft
Intended status: Standards Track
Expires: 21 July 2024

Shaofu. Peng
ZTE
Peng. Liu
China Mobile
Kashinath. Basu
Oxford Brookes University

18 January 2024

Policing Caused Jitter Control Mechanism
draft-peng-detnet-policing-jitter-control-00

Abstract

A mechanism to eliminate jitter caused by policing delay is described. It needs to be used in combination with a scheduling mechanism that provides low jitter for the transport path, and ultimately provides a low jitter guarantee for the application flow.

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 21 July 2024.

Table of Contents

1. Introduction 2
2. Requirements Language 3
3. Overview of the Solution 3
4. Set Edge Policing Delay Budget 5
5. Multi-domain considerations 5
6. Encoding Considerations 6
7. IANA Considerations 7
8. Security Considerations 7
9. Acknowledgements 7
10. References 7
10.1. Normative References 7
Authors' Addresses 8

1. Introduction

A policing function, as defined in RFC2216, differentiates those packets in a traffic flow which conform to a particular token bucket specification from those packets which do not. A Token Bucket is a particular form of traffic specification consisting of a "token rate" r and a "bucket size" b. More specifically, the traffic must obey the rule that over all time periods, the amount of data sent cannot exceed r*T+b, where T is the length of the time period. The common treatment accorded nonconforming packets may be relegating the packet to best effort service, discarding the packet, or marking the packet in some fashion.

A flow with conforming packets released to the network is the condition that must be followed to obtain deterministic forwarding services. This is also consitent with the problem statement of flow characterization in [RFC8557]. We assume that there is enough buffer space at the network entry to store nonconforming packets, that is important for deterministic flows that expect zero packet loss. A conforming packet may expierence zero policing delay, while a nonconforming packet may expierence non-zero policing delay. After policing on the network entry, the packet will be guaranteed a bounded delay or jitter by the applied scheduling mechanisms in the network. Generally, the scheduling mechanism guarantees the delay performance provided by the transport path in the network, and does not pay attention to the runtime policing delay at the network entry.

Although some application flows may declare in the contract with the network service provider that they will accept the introduction of policing delay for illegally arrived packets (i.e., nonconforming packets), other application flows, especially those that are extremely sensitive to jitter, hope not to get larger jitter due to the policing delay.

This document describes a mechanism to eliminate jitter caused by policing delay. It needs to be used in combination with a scheduling mechanism that provides low jitter for the transport path, and ultimately provides a low jitter guarantee for the application flow.

2. 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.

3. Overview of the Solution

The end-to-end delay expierenced by the application flow can be considered to consist of two parts: edge policing delay, and transport path delay. According to flow's end-to-end delay requirement and edge policing delay budget, a transport path with expected delay performance (i.e., end-to-end delay requirement minus edge policing delay budget) can be calculated and setup. A propriate edge policing delay budget can be configured for the application flow according to its TSpec and actual possible arrival pattern, and generally be the maximum policing delay that may be possibly experienced at the network entry.

The edge policing delay budget is consumed by the headend and endpoint of the transport path. Meet:

edge policing delay budget = headend policing delay + endpoint damping delay

The headend policing delay is the actual policing delay experienced at the network entry. The endpoint damping delay is obtained by subtracting the headend policing delay from the edge policing delay budget .

Figure 1 shows that the relationship between these delay elements.

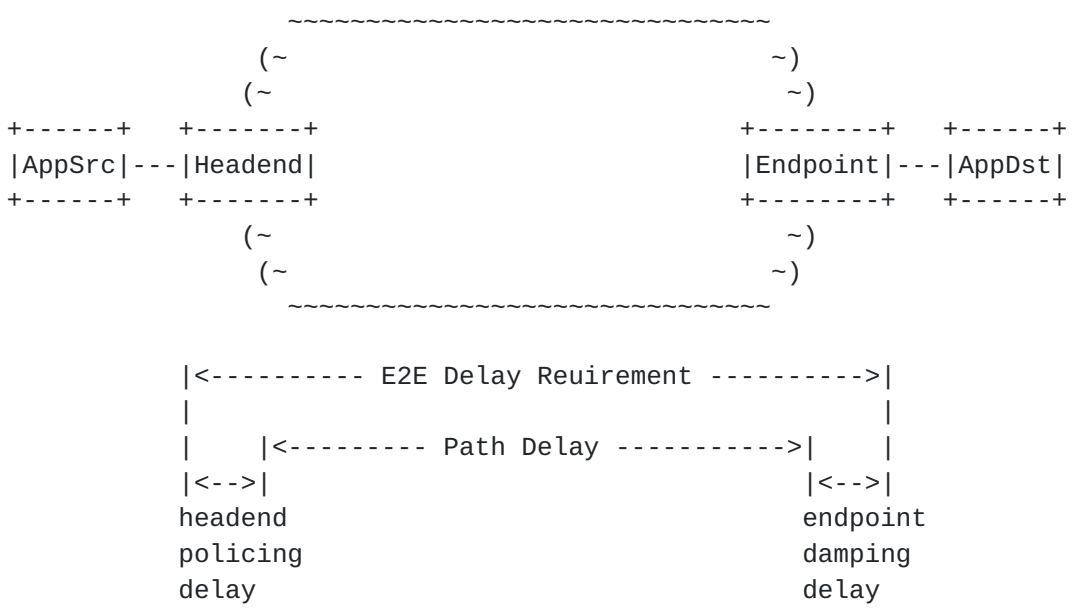

Figure 1: Relationship Between Delay Elements

When the headend of the transport path receives the packet from the application source, it may carry the endpoint damping delay in the encoded packet sent to the endpoint, and the endpoint damping delay will be used as the holding time imposed on the endpoint before the packet is delivered to the application destination.

Note that some scheduling mechanisms, such as [I-D.peng-detnet-deadline-based-forwarding] and [I-D.peng-detnet-packet-timeslot-mechanism], also provide the latency deviation (E) information of the transport path. Depending on the implementation, the endpoint can use different buffers to firstly implement jitter control based on path latency deviation (E) and secondly jitter control based on endpoint damping delay, or use the same buffer to uniformly implement jitter control based on the sum of path latency deviation (E) and endpoint damping delay.

Therefore, a flow with possible nonconforming packets will be regulated and changed to be conforming at the network entry, then get deterministic forwarding service within the network, and finally reverts to the nonconforming pattern at the network exit and deliver to the application destination.

4. Set Edge Policing Delay Budget

The edge policing delay budget can be configured for the application flow according to its TSpec and actual possible arrival pattern.

For example, an application flow has service burst interval (SBI) 100 us, and three packets P1, P2, P3 per SBI. Assuming the maximum packet size is 1000 bits. It can be seen that the bandwidth required by the application flow is 30 Mbps. The packet size 1000 bits and required bandwidth 30 Mbps are also the leaky bucket policing parameters at the network entry.

In the ideal case, a conforming pattern of the application flow is that the three packets arrive evenly in the SBI at the network entry. However, a extremely case of nonconforming pattern may be that P1, P2, P3 are closely arrived together. After leaky bucket regulation, P1, P2 and P3 will get different policing delays, of which P3 has the largest policing delay which may be 2/3 of SBI and can be used as edge policing delay budget.

There may be other settings of edge policing delay budget that are not based on the above extreme case, such as sampling the most likely actual arrival pattern of flow to set a smaller edge policing delay budget. Especially, for the timeslot based scheduling mechanism, the edge policing delay budget may be set based on the maximum deviation between the actual arrival timeslot and ideal arrival timeslot.

The network entry should maintain the edge policing delay budget for each application flow, and when it receives a packet from the application source, it identifies the flow to which the packet belongs and applies the corresponding edge policing delay budget. If the edge policing delay budget is M, and the actual policing delay of the packet is S, then the endpoint damping delay for that packet equals to M - S.

5. Multi-domain considerations

In the case of multi-domain, each domain may apply different scheduling machanisms. For each transit domain and egress domain, the input traffic should be conforming and then get the deterministic forwarding services. However, the output traffic from upstream domain may not always be conforming.

One option is to implement jitter control based on endpoint damping delay in each domian independently. That is, for each domain entry, it maintain the edge policing delay budget for each application flow, identify the application flow, regulate the nonconforming arrived packet and calculate the endpoint damping delay for the packet. In this option, each domain contribute a edge policing delay budget repeatedly, which may lead to a large end-to-end delay (but not necessarily, such as selecting a transport path with a smaller delay in each domain). When the packet leaves the upstream domain, the encapsulation metadata related to the scheduling mechanism of the upstream domain and the endpoint damping delay information are removed, and then the current domain entry will re-encapsulate the scheduling metadata related to the scheduling mechanism of the current domain and the endpoint damping delay information. The limitation for this option is the states maintained at each domain entry.

Another optoin is to implement jitter control based on endpoint damping delay only once for the entire multi-domian. The key operation in this option is to treat each transit domain entry (or egress domain entry) as a transit node of the scheduling mechanism of that domain, that means that the legacy parameters of the scheduling mechanism of the upstream domain (such as path latency deviation, scheduling sorting information) need to continue to be input and used as basic parameters in the current domain. The upstream domain exit should be aware that the scheduling mechanism has switched, and is responsible for mapping the metadata of the old scheduling mechanism to the metadata of the new scheduling mechanism, and especially, the metadata of the new scheduling mechanism contains the legacy scheduling results of the previous domain. The current domian entry should be aware of the legacy scheduling result (which can be treated as initial scheduling parameters) carried in the received packet. When the packet leaves the upstream domain, the endpoint damping delay information is always persisted and unchanged. This option seems less cost than the first option, e.g, no states per application flow maintained on each domain entry. However, the the mapping of scheduling metadata and processing initial scheduling parameters should be excecuted on several border nodes.

6. Encoding Considerations

A new IPv6 option for DOH Options header ([RFC8200]), or a new ancillary data for MPLS MNA header ([I-D.ietf-mpls-mna-hdr]), can be defined to carry endpoint damping delay. It is also possible to carry endpoint damping delay in an IPv6 Routing Header, such as in the segment field, to flexibly control which nodes (e.g, each domain exit, or only egress domain exit) need to implement jitter control based on endpoint damping delay.

The related encoding format and its usage will be defined in separate documents.

7. IANA Considerations

This document need not require IANA allocations.

8. Security Considerations

TBD.

9. Acknowledgements

TBD.

10. References

10.1. Normative References

[I-D.ietf-mpls-mna-hdr]
Rajamanickam, J., Gandhi, R., Zigler, R., Song, H., and K. Kompella, "MPLS Network Action (MNA) Sub-Stack Solution", Work in Progress, Internet-Draft, draft-ietf-mpls-mna-hdr-04, 21 October 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-mpls-mna-hdr-04>.

[I-D.peng-detnet-deadline-based-forwarding]
Peng, S., Du, Z., Basu, K., cheng, Yang, D., and C. Liu, "Deadline Based Deterministic Forwarding", Work in Progress, Internet-Draft, draft-peng-detnet-deadline-based-forwarding-08, 14 December 2023, <https://datatracker.ietf.org/doc/html/draft-peng-detnet-deadline-based-forwarding-08>.

[I-D.peng-detnet-packet-timeslot-mechanism]
Peng, S., Liu, P., Basu, K., Liu, A., Yang, D., and G. Peng, "Timeslot Queueing and Forwarding Mechanism", Work in Progress, Internet-Draft, draft-peng-detnet-packet-timeslot-mechanism-05, 14 December 2023, <https://datatracker.ietf.org/doc/html/draft-peng-detnet-packet-timeslot-mechanism-05>.

[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>.

[RFC8557] Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", RFC 8557, DOI 10.17487/RFC8557, May 2019, <https://www.rfc-editor.org/info/rfc8557>.

Authors' Addresses

Shaofu Peng
ZTE
China
Email: peng.shaofu@zte.com.cn

Peng Liu
China Mobile
China
Email: liupengyjy@chinamobile.com

Kashinath Basu
Oxford Brookes University
United Kingdom
Email: kbasu@brookes.ac.uk