Transport Area Working Group
Internet-Draft
Intended status: Informational
Expires: May 6, 2021

G. White, Ed.
CableLabs

November 2, 2020

Operational Guidance for Deployment of L4S in the Internet
draft-white-tsvwg-l4sops-01

Abstract

This draft is intended to provide guidance to operators of end-systems, operators of networks, and researchers in order to ensure successful deployment of L4S in the Internet. It includes mechanisms that are intended to promote reasonable fairness between L4S and Classic flows sharing a single-queue [RFC3168] bottleneck link. This draft identifies opportunites to prevent and/or detect and resolve fairness problems in such networks.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In the majority of network paths, including paths where the bottleneck link utilizes packet drops (either due to buffer overrun or active queue management) in response to congestion, as well as paths that implement a 'flow-queuing' scheduler such as fq_codel [RFC8290] or CAKE, and those that implement dual-Q-coupled AQM, L4S traffic generally coexists well with classic congestion controlled traffic.

On network paths where the bottleneck link instead implements a shared-queue (FIFO) with an Active Queue Management algorithm that provides Explicit Congestion Notification signaling according to [RFC3168], it has been demonstrated that when a set of long-running flows comprising both "Classic" congestion controlled flows and L4S-compliant congestion controlled flows compete for bandwidth, the classic congestion controlled flows may achieve lower throughput when compared to the L4S congestion controlled flows. This 'unfairness' between the two classes appears to be more pronounced on longer RTT paths (e.g. 50ms and above) and/or at higher link rates (e.g. 50 Mbps and above).

The root cause of this unfairness is that [RFC3168] does not differentiate between packets marked ECT0 (used by classic senders) and those marked ECT1 (used by L4S senders), and provides an identical congestion signal (CE marks) to both classes, while the L4S architecture redefines the CE mark and congestion response in the case of ECT1 marked packets. The result is that the two classes respond differently to the CE congestion signal. The classic senders expect that CE marks are sent very rarely (e.g. approximately 1 CE mark every 200 round trips on a 50 Mbps x 50ms path) while the L4S senders expect very frequent CE marking (e.g. approximately 2 CE marks per round trip). The result is that the classic senders respond to the CE marks provided by the bottleneck by yielding capacity to the L4S flows. While this has not been demonstrated to cause starvation of the classic flows, the resulting rate imbalance can be demonstrated, and could be a cause of concern.

2. Per-Flow Fairness

There are a number of factors that influence the relative rates achieved by a set of congestion controlled flows sharing a queue in a bottleneck link.

TODO: discuss startup & convergence times, short flows, RTT-unfairness, differences in deployed CC algorithms, etc.

TODO: also mention that flow sharding is commonplace, so per-flow fairness does not imply per-application fairness

Comments received: per-end-host fairness or per-customer fairness may be more important than per-flow fairness

3. Operator of an L4S host

Support for L4S involves both endpoints: ECT1 marking & L4S-compatible congestion control on the sender, and ECN feedback on the receiver. Between these two entities, it is incumbent upon the sender to evaluate the potential for unfairness and make decisions whether or not to use L4S congestion control. The receiver is not expected to perform any testing or monitoring for unfairness, and is also not expected to invoke any active response in the case that unfairness occurs.

The responsibilities of and actions taken by a sender may strongly depend on the environment in which it is deployed. This section discusses two scenarios: a constrained environment and an unconstrained environment.

TODO: also need to discuss how/when to re-enable L4S if it becomes disabled

3.1. CDN Servers

Some hosts (such as CDN leaf nodes and servers internal to an ISP) are deployed in environments in which they serve content to a constrained set of networks or clients. The operator of such hosts may be able to determine whether there is the possibility of [RFC3168] FIFO bottlenecks being present, and utilize this information to make decisions on selectively deploying L4S. Furthermore, such an operator may be able to determine the likelihood of an L4S bottleneck being present, and use this information as well.

For example, if a particular network is known to have deployed [RFC3168] FIFO bottlenecks, deployment of L4S should be delayed until those bottlenecks can be upgraded to mitigate any potential issues as discussed in the next section.

If a particular network offers connectivity to other networks (e.g. in the case of an ISP offering service to their customer's networks), the lack of RFC3168 FIFO bottleneck deployment in the ISP network can't be taken as evidence that RFC3168 FIFO bottlenecks don't exist end-to-end (because one may have been deployed by the end-user network). In these cases, deployment of L4S will need to take appropriate steps to detect the presence of such bottlenecks. At present, it is believed that the vast majority of RFC3168 bottlenecks in end-user networks are implementations that utilize fq_codel or Cake, where the unfairness problem does not exist. While this doesn't completely eliminate the possibility that a [RFC3168] FIFO bottleneck could exist, it nonetheless provides useful information that can be utilized in the decision making around the potential risk for any unfairness to be experienced by end users.

- Prior to deploying L4S on servers:
  - Consult with network operators on presence of [RFC3168] FIFO bottlenecks
  - Consult with network operators on presence of L4S bottlenecks
  - Perform downstream tests per access network
    - Tests (TBD) to detect absence of RFC 3168 (TODO: need more discussion about test methodologies and their implications (complexity, accuracy, etc.)).

    + Enable AccECN feedback for TCP, but enable/disable L4S per access network

- In-band [RFC3168] detection and monitoring: (cite: Fallback Tech Report)

  * Real-time response (fallback)

  * Non-real-time response (disable for future connections)

3.2. Other hosts

Hosts that are deployed in locations that serve a wide variety of networks face a more difficult prospect in terms of identifying the presence of RFC3168 FIFO bottlenecks. Nonetheless, steps can be taken to minimize the risk of unfairness.

Methods that can be deployed include:

- In-band [RFC3168] detection (and possibly fallback)

- Per-dst path test:

  * For a connection capable of L4S feedback

  * If CE feedback, perform active test (TBD) for [RFC3168] presence

Since existing studies have hinted that RFC3168 FIFO bottlenecks are rare, detections using these techniques may also prove to be rare. Therefore, it may be possible for a host to cache a list of end host ip addresses where a RFC3168 bottleneck has been detected. Entries in such a cache would need to age-out after a period of time to account for IP address changes, path changes, equipment upgrades, etc.

It has been suggested that a public blacklist of domains that implement RFC3168 FIFO bottlenecks or a public whitelist of domains that are participating in L4S experiment could be maintained. While this may be possible, a number of significant issues would need to be addressed, not the least of which is the fact that presence of RFC3168 FIFO bottlenecks or L4S bottlenecks is not a property of a domain, it is the property of a path between two endpoints.

4. Operator of a Network

While it is, of course, preferred for networks to deploy L4S-capable high fidelity congestion signaling, and while it is more preferable for L4S senders to detect problems themselves, a network operator who has deployed equipment in a likely bottleneck link location (i.e. a link that is expected to be fully saturated) that is configured with an [RFC3168] FIFO AQM can take certain steps in order to improve rate fairness between classic traffic and L4S traffic, and thus enable L4S to be deployed in a greater number of paths.

4.1. Configure AQM to treat ECT1 as NotECT

If equipment is configurable in such as way as to only supply CE marks to ECT0 packets, and treat ECT1 packets identically to NotECT, or is upgradable to support this capability, doing so will eliminate the risk of unfairness.

4.2. Configure Non-Coupled Dual Queue

Equipment supporting [RFC3168] may be configurable to enable two parallel queues for the same traffic class, with classification done based on the ECN field.

Option 1:

o Configure 2 queues, both with ECN; 50:50 WRR scheduler

o Queue #1: ECT1 & CE packets - Shallow immediate AQM target

o Queue #2: ECT0 & NotECT packets - Classic AQM target

o Outcome

  * n L4S flows and m long-running Classic flows

  * if m & n are non-zero, get 1/2n and 1/2m of the capacity, otherwise 1/n or 1/m

  * never < 1/2 each flow's rate if all had been Classic

This option would allow L4S flows to achieve low latency, low loss and scalable throughput, but would sacrifice the more precise flow balance offered by [I-D.ietf-tsvwg-aqm-dualq-coupled]. This option would be expected to result in some reordering of previously CE marked packets sent by Classic ECN senders, which is a trait shared with [I-D.ietf-tsvwg-aqm-dualq-coupled]. As is discussed in

[I-D.ietf-tsvwg-ecn-l4s-id], this reordering would be of very low risk.

Option 2:

- Configure 2 queues, both with AQM; 50:50 WRR scheduler

- Queue #1: ECT1 & NotECT packets - ECN disabled

- Queue #2: ECT0 & CE packets - ECN enabled

- Outcome

  - ECT1 treated as NotECT

  - Flow balance for the 2 queues the same as in option 1

This option would not allow L4S flows to achieve low latency, low loss and scalable throughput in this bottleneck link. As a result it is a less prefered option.

4.3. WRED with ECT1 Differentation

This configuration is similar to Option 2 in the previous section, but uses a single queue with WRED functionality.

- Configure the queue with two WRED classes

- Class #1: ECT1 & NotECT packets - ECN disabled

- Class #2: ECT0 & CE packets - ECN enabled

4.4. ECT1 Tunnel Bypass

Using an RFC6040 compatibility mode tunnel, tunnel ECT1 traffic through the [RFC3168] bottleneck with the outer header indicating Not-ECT.

Two variants

1. per-domain: tunnel ECT1 pkts to domain edge towards dst

2. per-dst: tunnel ECT1 pkts to dst

4.5. Disable RFC3168 ECN Marking

While not a recommended alternative, disabling [RFC3168] ECN marking eliminates the unfairness issue. Clearly a downside to this approach is that classic senders will no longer get the benefits of Explict Congestion Notification.

4.6. Re-mark ECT1 to NotECT Prior to AQM

While not a recommended alternative, remarking ECT1 packets as NotECT ensures that they are treated identically to classic NotECT senders. However, this also eliminates the possibility of downstream L4S bottlenecks providing high fidelity congestion signals.

5. Researchers

5.1. Detection of Classic ECN FIFO Bottlenecks

TODO: Describe active testing methods, in-band or out-of-band, that can distinguish FIFO from FQ.

5.2. End-to-end measurement of L4S vs. Classic performance

TBD

6. Contributors

Thanks to Bob Briscoe, Jake Holland, Koen De Schepper, Olivier Tilmans, Tom Henderson, Asad Ahmed, and members of the TSVWG mailing list for their contributions to this document.

7. IANA Considerations

None.

8. Security Considerations

None.

9. Informative References

[I-D.ietf-tsvwg-aqm-dualq-coupled]
Schepper, K., Briscoe, B., and G. White, "DualQ Coupled AQMs for Low Latency, Low Loss and Scalable Throughput (L4S)", draft-ietf-tsvwg-aqm-dualq-coupled-12 (work in progress), July 2020.

[I-D.ietf-tsvwg-ecn-l4s-id]
Schepper, K. and B. Briscoe, "Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay (L4S)", draft-ietf-tsvwg-ecn-l4s-id-10 (work in progress), March 2020.

[I-D.ietf-tsvwg-l4s-arch]
Briscoe, B., Schepper, K., Bagnulo, M., and G. White, "Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture", draft-ietf-tsvwg-l4s-arch-07 (work in progress), October 2020.

[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>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[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>.

[RFC8290] Hoeiland-Joergensen, T., McKenney, P., Taht, D., Gettys, J., and E. Dumazet, "The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm", RFC 8290, DOI 10.17487/RFC8290, January 2018, <https://www.rfc-editor.org/info/rfc8290>.

Author's Address

Greg White (editor)
CableLabs

Email: g.white@cablelabs.com