Prague requirements / L4S-ID

draft-ietf-tsvwg-ecn-l4s-id-16

Koen De Schepper, Nokia Bell Labs

Bob Briscoe, independent

TSVWG @ IETF110-111 interim meeting

March 10, 2021

Lots of activity, good discussions and progress

- Prague Requirements Survey
  - targeting CC developers
  - feasible/realizable?
  - supported by a broad community (allows several different CCs)?
- Chair’s detailed review
- List discussions and recommendations
- Focus on Normative text (editorials later)

➔ Resulting in clearer and to the point requirements

Prague Requirements Survey

Multiple responses received

4 were publicly shared:

- Linux TCP-Prague by L4Steam
- SCReAM by Ingemar Johansson
- GeforceNow by NVIDIA
- Apple by Vidhi Goel (NEW)

→ Listed in https://l4steam.github.io/#prague-requirements-compliance

Other responses shared privately:

→ consolidated summary available at:

https://l4steam.github.io/PragueReqs/Prague_requirements_consolidated.pdf

Objections on:
SHOULD detect loss by counting in time-based units …

Updated based on Apple's and Google's feedback

- objection that time-based is only/sufficient way to allow scalable reordering mechanism for L4S traffic
- express requirement, not the mechanism → allows alternative/more robust implementations
- RACK/RFC8985 is actually doing more than time-based only

→ Not only time-based
→ Adaptive interval
→ Refer to RACK/RFC8985

Strong objections on: documentation-only reqs

- The specification MUST describe in detail …
- The specification MUST define, quantify and justify burst limit approach …

- Are these documentation requirements really needed?
- How can it be enforced?
- May not be possible (proprietary).

Actions on the draft:

→ These requirements have been removed and **advised** in general

Clarifications on:
Coexist safely with Reno congestion control

- MUST react to packet loss in a way that will coexist safely with a TCP Reno congestion control [RFC5681] …

- Not clear what it means "coexist safely with a TCP Reno congestion control“
- Don't want to be as degraded as Reno for long RTTs

Balance between openness to innovations and guidance/recommendations
→ keep open during experiment, not the mechanism but the result is important

Consensus on:
MUST react to packet loss in a way that will coexist safely
… with a TCP Reno congestion control [RFC5681]…
→ … with a Classic congestion controls such as standard Reno [RFC5681], as required by [RFC5033]…

Comments on:
SHOULD scale down to fractional congestion window …

- SHOULD scale down to fractional congestion window …

- Not all convinced if it will be a problem on the Internet, and might not implement
- Multiple research implementations exist; others support it or plan to implement

→ If it occurs on the Internet, those that implement this will back-off while others not

Final call to keep SHOULD
Output of the experiment: is this important to make MUST

Comments on:
monitoring; fallback; replacement

- MUST implement monitoring to detect non_L4S ECN AQM…
- SHOULD be capable to automatically fall back …
- MUST be capable of being replaced (operator action) by a Classic congestion control …

- Is detection itself required?
- Robust detection scheme needs real deployment experience.
- Combination with delay-based control could minimize potential issues
- Develop during experiment as needed.
- Is “replace” required or can it disable L4S part to reduce to Classic response only
- On active flows or new flows

→ Clarified monitoring: on live traffic unless on path by external/alternate monitor
→ Clarified adaptation / replacement conditions
→ Clarified adaptation from ECT(1) to ECT(0) (no need if transient / can fall back)
→ Aligned requirements with L4S Operational guidelines draft

Comments on:
MUST reduce RTT bias

- Conflict between MUST and “as much as possible”

- RTT bias is meant only for Rate convergence
  - Not for slow start, getting up to speed, reduction on strong marking signal, which will still need to scale with RTT to preserve stability and efficiency

- Is rate fairness absolute or more gradual (still “as much as possible”, compromising between stable throughput for LL services and optimally using short periods of available BW)

- Discussion? Outcome of experiment?

Unchanged:

Requirements:

- An L4S sender MUST set the ECN field to ECT(1) → OS APIs and Kernels need to support it
- MUST NOT set ECT(1) unless it complies with …
- A sender that sets ECT(1) SHOULD implement a scalable congestion control
- MUST provide feedback of the extent of CE marking … → Some remaining concerns with Accurate ECN → tcpm

Non-Normative performance suggestions:

- Setting ECT(1) in TCP Control Packets and Retransmissions → Make normative?
- Faster than Additive Increase
- Faster Convergence at Flow Start

Actions on the draft:

→OK after minor clarifications

Other open topics

Guard DSCP

- Stops deployment flexibility
- Adoption level is the success criterium of this experiment
- Stopping its adoptability from the start is in contradiction of the experiment (also stops experimenting with real RFC3168 detection)
- Is there a robust Guard scheme defined yet?

New recent Replay protection interaction

- Is limited reordering resilient which can be caused by ECT(0) → CE marking on second DualQ
- Is a problem for ECT(0) users only
- OTOH:
- Impact limited to drops within the same RTT, only when CE is sent for that RTT anyway
- Similar issues with DiffServ in single Secure tunnel
- Solutions need to be in making reordering resilience scalable or replay mechanism

Other open topics

New end of experiment requirements:

- In case unforeseen problems arise with the L4S experiment, it MUST be possible to configure an L4S implementation to disable the L4S treatment. Once disabled, all packets of all ECN codepoints will receive Classic treatment and ECT(1) packets MUST be treated as if they were {ToDo: Not-ECT / ECT(0) ?}

Conclusion

- Good progress

- Other inputs are still welcome (public or private)

Backup

- Non-L4S ECN AQM

- A scalable congestion control MUST implement monitoring in order to detect a likely non-L4S but ECN-capable AQM at the bottleneck. On detection of a likely ECN-capable bottleneck it SHOULD be capable (dependent on configuration) of automatically adapting its congestion response to coexist with TCP Reno congestion controls [RFC5681] (see Appendix A.1.4 for rationale and a referenced algorithm).

  Note that a scalable congestion control is not expected to change to setting ECT(0) while it falls back to coexist with Reno.

- In uncontrolled environments, monitoring MUST be implemented to support detection of problems with an ECN-capable AQM at the path bottleneck that appears not to support L4S and might be in a shared queue. Such monitoring SHOULD be applied to live traffic that is using Scalable congestion control. Alternatively, monitoring need not be applied to live traffic, if monitoring has been arranged to cover the paths that live traffic takes through uncontrolled environments.

  The detection function SHOULD be capable of making the congestion control adapt its ECN-marking response to coexist safely with Classic congestion controls such as standard Reno [RFC5681], as required by [RFC5033]. Alternatively, if adaptation is not implemented and problems with such an AQM are detected, the scalable congestion control MUST be replaced by a Classic congestion control.

  Note that a scalable congestion control is not expected to change to setting ECT(0) while it transiently adapts to coexist with Reno.

  See Appendix A.1.5 and [I-D.ietf-tsvwg-l4sops] for rationale.

- RTT bias
- Scaling to low RTT

o A scalable congestion control MUST eliminate RTT bias as much as possible in the range between the minimum likely RTT and typical RTTs expected in the intended deployment scenario (see Appendix A.1.5 for rationale).	o A scalable congestion control MUST reduce RTT bias as much as possible in the range between the minimum likely RTT and typical RTTs expected in the intended deployment scenario (see Appendix A.1.6 for rationale).
o A scalable congestion control SHOULD remain responsive to congestion when typical RTTs over the public Internet are significantly smaller because they are no longer inflated by queuing delay. It would be preferable for the minimum window of a scalable congestion control to be lower than 1 segment rather than use the timeout approach described for TCP in S.6.1.2 of [RFC3168] (or an equivalent for other transports). However, a lower minimum is not set as a formal requirement for L4S experiments (see Appendix A.1.6 for rationale).	o A scalable congestion control SHOULD remain responsive to congestion when typical RTTs over the public Internet are significantly smaller because they are no longer inflated by queuing delay. It would be preferable for the minimum window of a scalable congestion control to be lower than 1 segment rather than use the timeout approach described for TCP in S.6.1.2 of [RFC3168] (or an equivalent for other transports). However, a lower minimum is not set as a formal requirement for L4S experiments (see Appendix A.1.7 for rationale).

- Scaling loss detection with throughput
- Limiting bursts

o A scalable congestion control SHOULD detect loss by counting in time-based units, which is scalable, as opposed to counting in units of packets (as in the 3 DupACK rule of RFC 5681 TCP), which is not scalable. As packet rates increase (e.g., due to new and/or improved technology), congestion controls that detect loss by counting in units of packets become more likely to incorrectly treat reordering events as congestion-caused loss events (see Appendix A.1.7 for further rationale). This requirement does not apply to congestion controls that are solely used in controlled environments where the network introduces hardly any reordering.

o A scalable congestion control is expected to limit the queue caused by bursts of packets. It would not seem necessary to set the limit any lower than 10% of the minimum RTT expected in a typical deployment (e.g. additional queuing of roughly 250 us for the public Internet). This would be converted to a number of packets under the worst-case assumption that the bottleneck link capacity equals the current flow rate. No normative requirement to limit bursts is given here and, until there is more industry experience from the L4S experiment, it is not even known whether one is needed - it seems to be in an L4S sender's self-interest to limit bursts.

o A scalable congestion control's loss detection SHOULD be resilient to reordering over an adaptive time interval that scales with throughput and adapts to reordering (as in [RFC8985]), as opposed to counting only in fixed units of packets (as in the 3 DupACK rule of [RFC5681] and [RFC6675], which is not scalable). As packet rates increase (e.g., due to new and/or improved technology), congestion controls that detect loss by counting in units of packets become more likely to incorrectly treat reordering events as congestion-caused loss events (see Appendix A.1.8 for further rationale). This requirement does not apply to congestion controls that are solely used in controlled environments where the network introduces hardly any reordering.

o A scalable congestion control is expected to limit the queue caused by bursts of packets. It would not seem necessary to set the limit any lower than 10% of the minimum RTT expected in a typical deployment (e.g. additional queuing of roughly 250 us for the public Internet). This would be converted to a number of packets under the worst-case assumption that the bottleneck link capacity equals the current flow rate. No normative requirement to limit bursts is given here and, until there is more industry experience from the L4S experiment, it is not even known whether one is needed - it seems to be in an L4S sender's self-interest to limit bursts.

- Replaceability
- Fall-back on Loss

In order to coexist safely with other Internet traffic, a scalable congestion control MUST NOT tag its packets with the ECT(1) codepoint unless it complies with the following bulleted requirements:	In order to coexist safely with other Internet traffic, a scalable congestion control MUST NOT tag its packets with the ECT(1) codepoint unless it complies with the following bulleted requirements:
	o A scalable congestion control MUST be capable of being replaced by a Classic congestion control (by application and/or by administrative control). If a Classic congestion control is activated, it will not tag its packets with the ECT(1) codepoint (see Appendix A.1.3 for rationale).
o As well as responding to ECN markings, a scalable congestion control MUST react to packet loss in a way that will coexist safely with a TCP Reno congestion control [RFC5681] (see Section 1.2 on Terminology for definition of Reno-Friendly and Appendix A.1.3 for rationale).	o As well as responding to ECN markings, a scalable congestion control MUST react to packet loss in a way that will coexist safely with Classic congestion controls such as standard Reno [RFC5681], as required by [RFC5033] (see Appendix A.1.4 for rationale).

...

To participate in the L4S experiment, a scalable congestion control MUST be capable of being replaced by a Classic congestion control (by application and by administrative control). A purely Classic congestion control will not tag its packets with the ECT(1) codepoint.	

All agreed: Compliant or planned

An L4S sender MUST set the ECN field to ECT(1)	- Compliant or planned - OS APIs and Kernels need to support it (can RFC8311 be used to justify API updates)	None, OK as is
A sender that sets ECT(1) SHOULD implement a scalable congestion control	- Compliant or planned - More clarification needed to align marking rate to throughput	Improve informative text for rate convergence of long flows
MUST eliminate RTT …	- Compliant or planned - Also for longer RTTs more throughput is planned	None, OK as is
SHOULD detect loss by counting in time-based units …	- Compliant or planned	None, OK as is
MUST NOT set ECT(1) unless it complies with following …	- Compliant to this requirement - Comments were on referred requirements	None, OK as is

All agreed (non-normative): Supported or planned

Setting ECT(1) in TCP Control Packets and Retransmissions	- Supported or planned	RTP/RTCP clarifications will be added
Faster than Additive Increase	- Supported or planned	None, OK as is
Faster Convergence at Flow Start	- Research code exists and planned	None, OK as is

Questioned and Strong objections

The specification MUST describe in detail …	- Is this requirement really needed? - How can it be enforced? - May not be possible (propriatary).	This requirement is removed
SHOULD scale down to fractional congestion window …	- Multiple research codes exist - Not all convinced if this is needed, others support it and plan to implement - Develop during experiment as needed.	Keep SHOULD. The need for this requirement should be observed during the experiment
limit bursts … The specification MUST define, quantify and justify its approach …	- Normative requirement is mainly documentation related, see above - Can more clear guidelines be given?	The normative MUST is removed. Warning text still present.

Clarification needed

MUST provide feedback of the extent of CE marking …	- Compliant - Clarification needed for feedback timing and RTT requirements - Some remaining concerns with Accuate ECN	- Appropriate feedback timing depends on the proprietary protocol and needs to be tuned to it - Remaining concerns about Accurate ECN needs to be dealt with in tcpm.
MUST react to packet loss in a way that will coexist safely with a TCP Reno congestion control [RFC5681] …	- Compliant to the intent - Not clear what it means "coexist safely with a TCP Reno congestion control" - Don't want to be as degraded as Reno for long RTTs	- Seeking input from WG on clarification to this requirement e.g. RFC5033
MUST implement monitoring to detect non_L4S ECN AQM… SHOULD be capable to automatically fall back … MUST be capable of being replaced by a Classic congestion control …	- Robust detection scheme needs real deployment experience. - Develop during experiment as needed. - Combination with delay-based control could minimize potential issues - Clarification: is detection itself required?	- If L4S Operational guidelines draft is adopted, these requirements will need to be aligned with it