ECT(1)
TSVWG,
22 April 2020

Key Goal -
What is the goal of this WG in 2020?

L4S

Consistently ultra low latency, low loss with high throughput,

… without requiring transport-layer inspection or per-flow queuing/scheduling

… usable by all applications

… creating strong incentives for deployment

Safely deploy an Internet-wide solution to the problem of congestion control scalability in TCP/QUIC/etc.
without harm to others

SCE

Retaining complete backwards compatibility with RFC-3168, provide
high-fidelity CC, to improve peak latency and/or path utilisation.

Transports use extra network information to converge to the ideal send rate (or cwnd) instead of oscillating around it.

SCE-aware AQMs will normally implement AF or FQ, which improves RTT-independence of throughput fairness for all flows, not just SCE transports.

SCE traffic should be an equal citizen.

Non-Goal: Providing a new, elevated or special class of service.

Framing the ECT(1) Codepoint Decision

L4S & SCE both propose a new usage of ECT(1)

RFC 4774 assumes DSCP as signal of alternate ECN semantics.

Decision: How ECT(1) signals alternate ECN semantics to network:

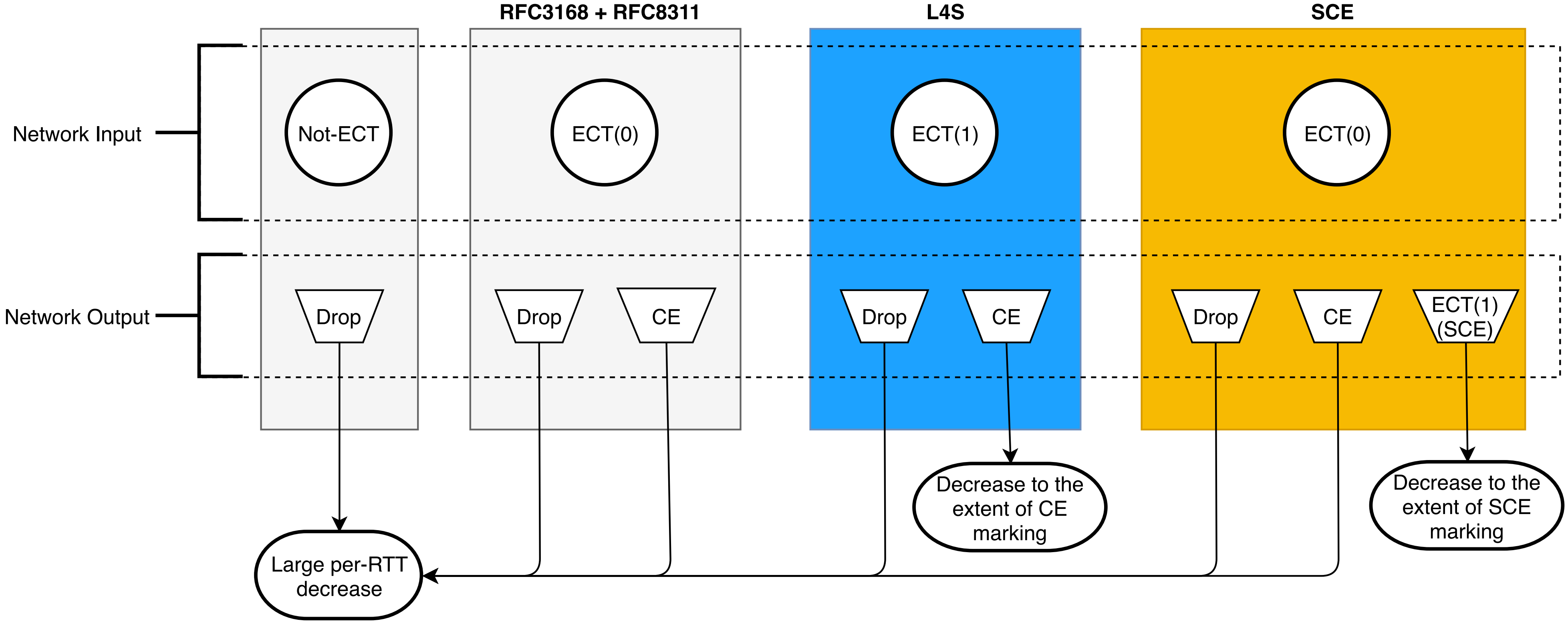
- Input, indicates transport intent and classifier for “queue” selection, [L4S]
- Output, e.g., additional indication of lesser degree of congestion [SCE]

At Internet scope: Choose at most one usage, not both.

Framing the ECT(1) Codepoint Decision

Two proposals that use ECT(1) as that signal [L4S, SCE]

Decision: How ECT(1) signals alternate ECN semantics to network:



L4S

More than 50% of all Internet traffic is encrypted and/or tunneled (mobile and VPNs)

ECT(1) as an input was chosen in L4S because:

- It doesn’t require access to transport headers
- It can be supported by network bottlenecks that implement FQ or a simpler dual-queue scheme
- It is supported by existing tunnels and encapsulations
- It works in the presence of TCP ACK thinning
- It is compatible with existing data center network deployments
- It eventually frees up ECT(0)

ECT(1) as an output (SCE) fails on all the above aspects
SCE would force & ossify FQ dependence and L4 header inspection throughout the Internet

All TSVWG open issues have been addressed

- There are no remaining IPR concerns
- Safe coexistence in RFC3168 ECN bottlenecks
- RTT dependence
- CE-ambiguity & Reordering

L4S has been demonstrated both in simulation and in testbed implementations across a wide range of conditions and multiple applications

If L4S is standardized, consistent ultra-low latency for all applications including web, game streaming, cloud VR/AR and high-fidelity interactive media can rapidly be supported on hundreds of millions of broadband connections worldwide, with zero-configuration. This opens up an important new space for application innovation.

Because of all of these factors, the L4S architecture has broad industry support: in network equipment, network operators and end-systems.

SCE

SCE's signalling method and control loop are simple and robust, requiring only man-weeks to implement.

On-wire protocol minimal: one ECN codepoint & one TCP header flag; statelessly interpretable.

Hardware implementation is feasible both at low cost (CPE) and at high speed (datacentre, backhaul).

Full RFC-3168 backwards compatibility, via separate, unambiguous CE & SCE signals.

CE (with RFC-8511 response) acts as a safety valve for SCE absence/erasure & large capacity reductions.

Safe, decoupled system design permits future innovation in algorithm space, both AQM and CC.

Working examples in the public reference implementation, including simple 1/p scheme.

Framing the ECT(1) Codepoint Decision

Two proposals that use ECT(1) as that signal [L4S, SCE]

Decision: How ECT(1) signals alternate ECN semantics to network:

L4S:

L4S chose ECT(1) as a second input:

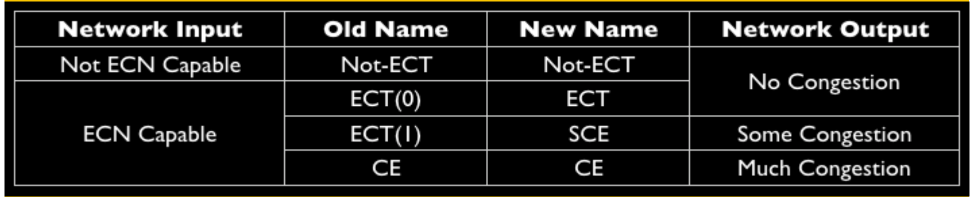
- to isolate low latency packets without requiring Transport header inspection and per-flow queuing/scheduling
- to isolate Not-ECT and ECT(0) packets from the ECT(1) experiment

(SCE alters ECN-capable packets whether in the experiment or not)

L4S avoided a second output (SCE) because:

- it wastes a codepoint - the extent of marking already gives multi-level output
- RFC3168 and RFC4301 (IPsec) encapsulations strip it off

SCE:
SCE's input to the network is ECT(0), as in RFC-3168.
Changing ECT(0) to ECT(1) is the network's new output.



Network Input	Old Name	New Name	Network Output
Not ECN Capable	Not-ECT	Not-ECT	No Congestion
ECN Capable	ECT(0)	ECT	
	ECT(1)	SCE	Some Congestion
	CE	CE	Much Congestion

Existing AQMs treat ECT(1) like ECT(0), so can still upgrade SCE marks to CE marks.

At Internet scope: Choose at most one usage, not both.

Friendly Coexistence with Competing Traffic*

Scheduling (e.g. FQ) not a significant cause of coexistence problems

Coexistence Focus: Traffic competition at a shared bottleneck with ECN

- Competing Traffic with TCP congestion control, e.g., Reno, Cubic, etc.
- Competing Traffic drives bottleneck queue occupancy.
- Starvation of one class of traffic is not an acceptable outcome.

Any solution needs to deal with this scenario.

*RFC 4774 Option 3 (section 4.3) Incremental Deployment

Friendly Coexistence

L4S provides friendly coexistence across all known bottleneck types

Sender falls back to classic over drop-tail and RFC3168 bottlenecks

Simple Coupled DualQ and updated FQ enable ultra-low delay with coexistence

On existing networks, SCE transports behave identically to conventional RFC-3168 transports. They interpret CE in the same way and respond according to RFC-8511.

When SCE signalling is added, network nodes are responsible for maintaining fair competition. e.g.: Fair Queuing, or Approximate Fairness, or Controlled Environment w/o non-SCE traffic.

L4S	Fair-ness	Ultra-Low Delay	L4 Unaware
DropTail	Y	N	Y
RFC3168 FIFO	Y	N	Y
Dual Queue (coupld)	Y	Y	Y
Per-Flow Scheduling	Y	Y	N*

SCE	Fair-ness	Ultra-Low Delay	L4 Unaware
DropTail	Y	N	Y
RFC3168 FIFO	Y	N	Y
Dual Queue (Codel AF)	Y	Y	N*
Per-Flow Scheduling	Y	Y	N*

*** Flow identifiers only**

Experimental Deployment: Operations & Management

Does the WG believe proposed deployment is manageable?

- How does an operator detect L4S/SCE traffic?
- What is the impact of ECT(1) propagation? (current tunnels, lower layers).
- How to re-classify in a domain not a part of this experiment?

WG needs to decide how to manage the experiment

ECT(1)
Propagation through the current Internet

L4S Propagation

- Minimal requirements for NW deployment (works without FQ)
- Designed to get at least equal performance under all conditions
- Full support through existing NWs (tunnels, lower layers)
- L4S Applications benefit from Classic NW ECN marking due to fallback

➔ Strong incentive for ALL applications to use L4S by default

SCE Propagation

- ECT(1) traverses the Internet well, due to foresight in RFC-3168
- Mixtures of ECT(0) and ECT(1) fragments are underspecified on reassembly; this may disrupt SCE signal applied on tunnel paths
- SCE transports use CE in absence/ erasure of SCE, and for large capacity reductions; this ensures effective congestion control

➔ Update to RFC-3168 desirable to obtain full benefits on tunnel paths

Experiment End

What if the experiment with ECT(1) were to fail to conclude:
Will the IETF be able to re-use ECT(1) in future?

If L4S succeeds:
ECT(0) could eventually be reclaimed

L4S Un-deployment:
Endpoint behaviour is confined to ECT(1) which can be blocked by network.
ECT(1) could be reclaimed in the longer term.

If SCE success:
ECT(0) & ECT(1) assigned to SCE

SCE Un-deploymemt:
SCE endpoint behaviour is harmless at IP layer.
SCE AQMs can be detected by their ECT(1) emissions, or eliminated by observing CE without ECT(1).
ECT(1) can be reclaimed by locating and removing SCE AQMs.

Who Plans to Deploy L4S?

Akamai
Apple
Broadcom Corp.
Casa Systems
Commscope Inc.
Nokia
Valve
Vodafone
Comcast (USA)
Vodafone GmBH (Germany)
Liberty Global
Cox Communications (USA)
Izzi Telecom (Mexico)
Telecom (Argentina)
Vodafone España (Spain)
Rogers Communications (Canada)
Shaw Communications (Canada)
Mediacom Communications (USA)
Tele2 AB (Sweden)
Cogeco Communications (Canada)

Additional Statements of Support from:

CableLabs
Ericsson
Google
Microsoft

Support in 3GPP:

Apple
AT&T
CableLabs
Ericsson
Google
Nokia
Nokia Shanghai Bell
Sony
Sprint
T-Mobile
Vodafone

Who Plans to Deploy SCE?

Akamai and HP-Enterprise have expressed specific interest in SCE and contributed to our research. HPE is focused on datacentre applications.

SCE could be deployed at scale in CPE, in a similar way to the Cake FQ-AQM - indeed by simply updating existing deployments of Cake to an SCE-enabled version. This would primarily involve the Linux kernel, OpenWRT and its derivatives. The new Linux wifi stack also implements FQ-AQM and would be another good candidate for widespread SCE deployment. We have a good working relationship with the relevant parties, and the means to perform controlled experimental deployments.

Additionally, if ECT(1) is defined as an output from the network (as SCE describes), then we believe much of the existing L4S work could be straightforwardly translated into this paradigm, bringing **most potential L4S deployment scenarios** with it. This translation would confer some benefits on L4S, namely seamless compatibility with existing networks, and removing opposition from some quarters over a perceived "fast lane" mechanism.

And if no decision …

If no decision is made on ECT(1) usage (input vs. output):

- The L4S identifier draft will be unable proceed in its current form that uses ECT(1) by itself to identify low latency traffic.

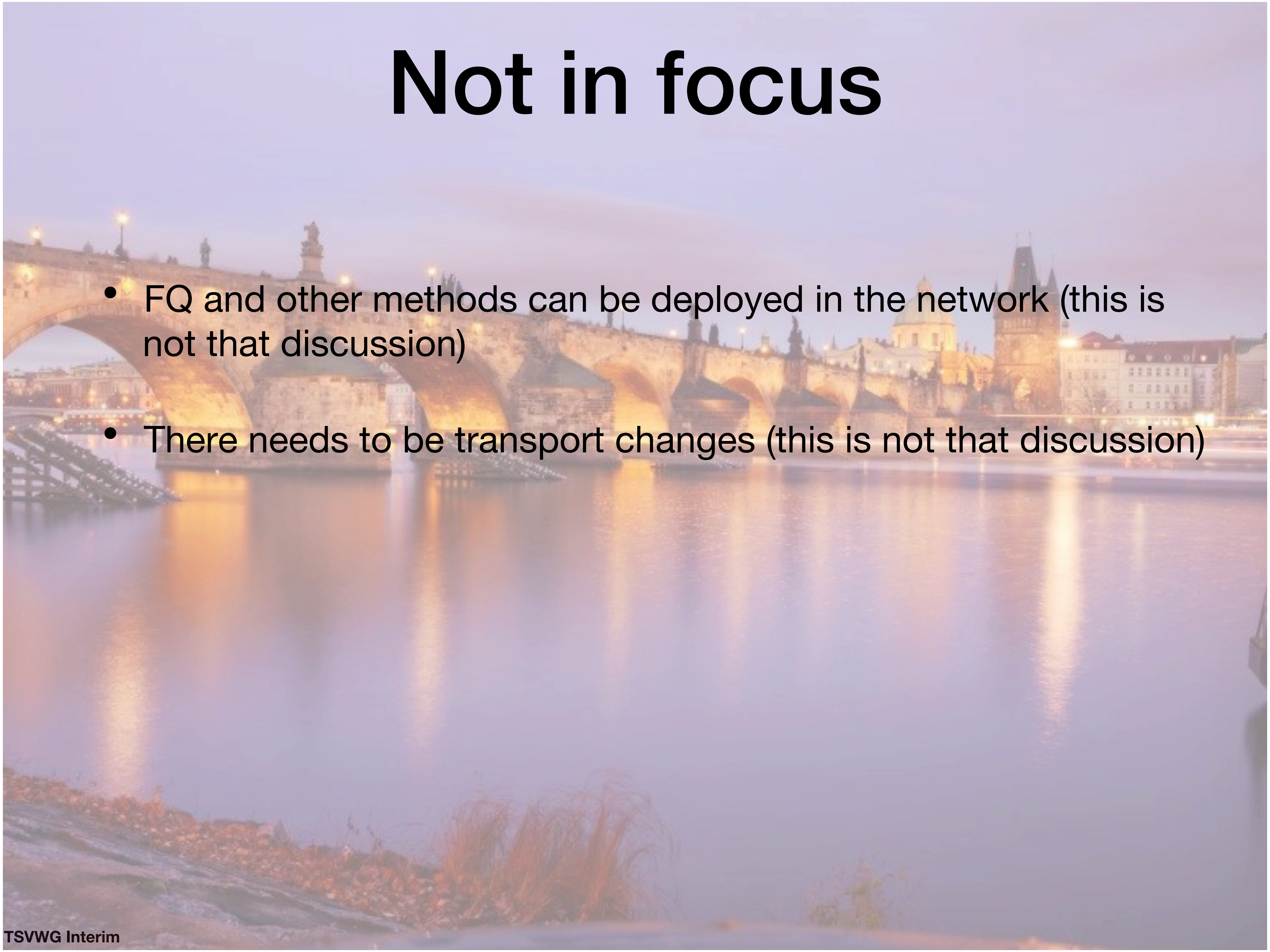
- Any other proposal, including SCE, that uses ECT(1) for a non-RFC 3168 purpose will be similarly unable to use ECT(1) by itself.

RFC 4774 (BCP 124) “Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field”, including its recommendation for DSCP usage, becomes relevant in this scenario.

If an approach is found to be unworkable before we publish it, the WG reserves the right to revisit the ECT(1) decision.

Additional Slides

Not in focus



- FQ and other methods can be deployed in the network (this is not that discussion)
- There needs to be transport changes (this is not that discussion)

Implementing L4S

Sender:

Implement Congestion Control per Prague Requirements & Mark as ECT(1)
Implement Accurate ECN (TCP-only)

Network Bottleneck:

Implement Dual Q Coupled AQM or FQ AQM with shallow threshold / ramp

Tunnels/Encapsulations:

Nothing needed

Fragmentation and Reassembly:

Nothing needed

Receiver:

QUIC/DCCP: nothing needed; TCP: implement Accurate ECN

Current Implementations:

Linux & NS3 TCP Prague sender
Linux, FreeBSD & NS3 Accurate ECN sender/receiver
SCREAM & BBRv2 senders
Initial QUIC Prague sender
Linux & NS3 Dual Queue Coupled AQM qdisc
Linux & NS3 L4S fq_codel AQM qdisc
DOCSIS cable modems & CMTS
Nokia WiFi AP

Drafts:
draft-ietf-tsvwg-l4s-arch
draft-ietf-tsvwg-ecn-l4s-id
draft-ietf-tsvwg-aqm-dualq-coupled
draft-ietf-tcpm-accurate-ecn

Implementing SCE

Implement AQM eg. by duplicating Codel:

100ms interval, 5.0ms target for CE/drop - (takes precedence)

25ms interval, 2.5ms target for SCE - (without drop fallback**)**

SCE AQM should also include FQ or AF features, unless in controlled environment where this is shown to be unnecessary.

See draft-morton-tsvwg-codel-approx-fair.
See also draft-morton-tsvwg-sce for transport-layer implementation aspects.