To adopt or not to adopt L4S-compatible congestion control? Understanding performance in a partial L4S deployment

Fatih Berkay Sarpkaya, Fraida Fund, Shivendra Panwar

MAPRG @IETF-122 & Passive and Active Measurement (PAM) 2025

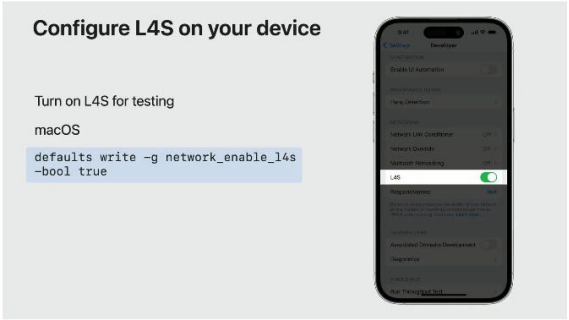
L4S

- **L**ow **L**atency
- **L**ow **L**oss
- **S**calable Throughput

Source: B. Briscoe, K. De Schepper, M. Bagnulo, and G. White, “RFC 9330: Low latency, low loss, and scalable throughput (L4S) internet service: Architecture,” USA, 2023.

L4S Gaining Industry Traction

Nokia



Apple

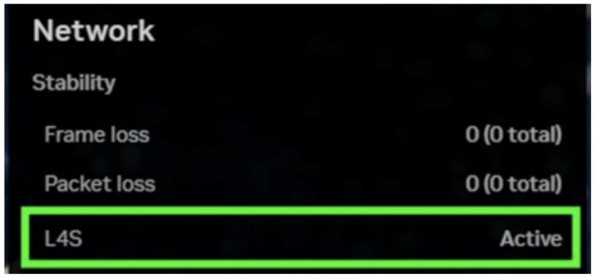
High latency peaks cause video freezes

Consistent low latency for smooth AR experience

Ericsson

Comcast

GeForce NOW

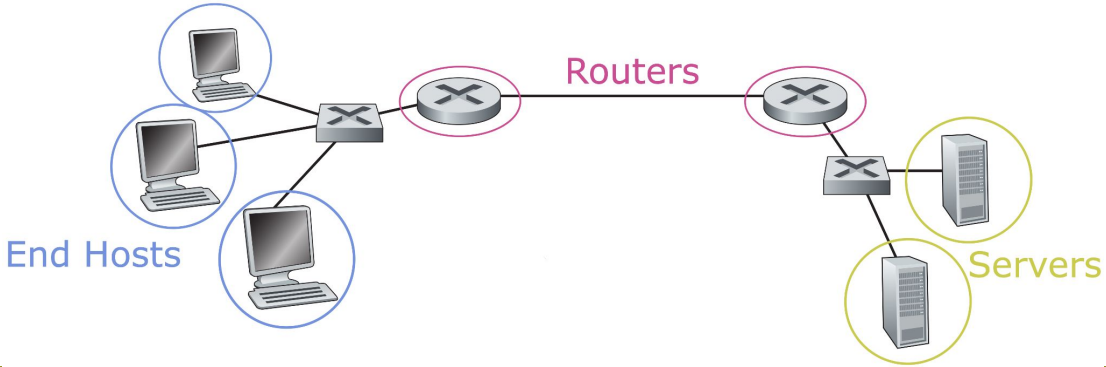


NVIDIA

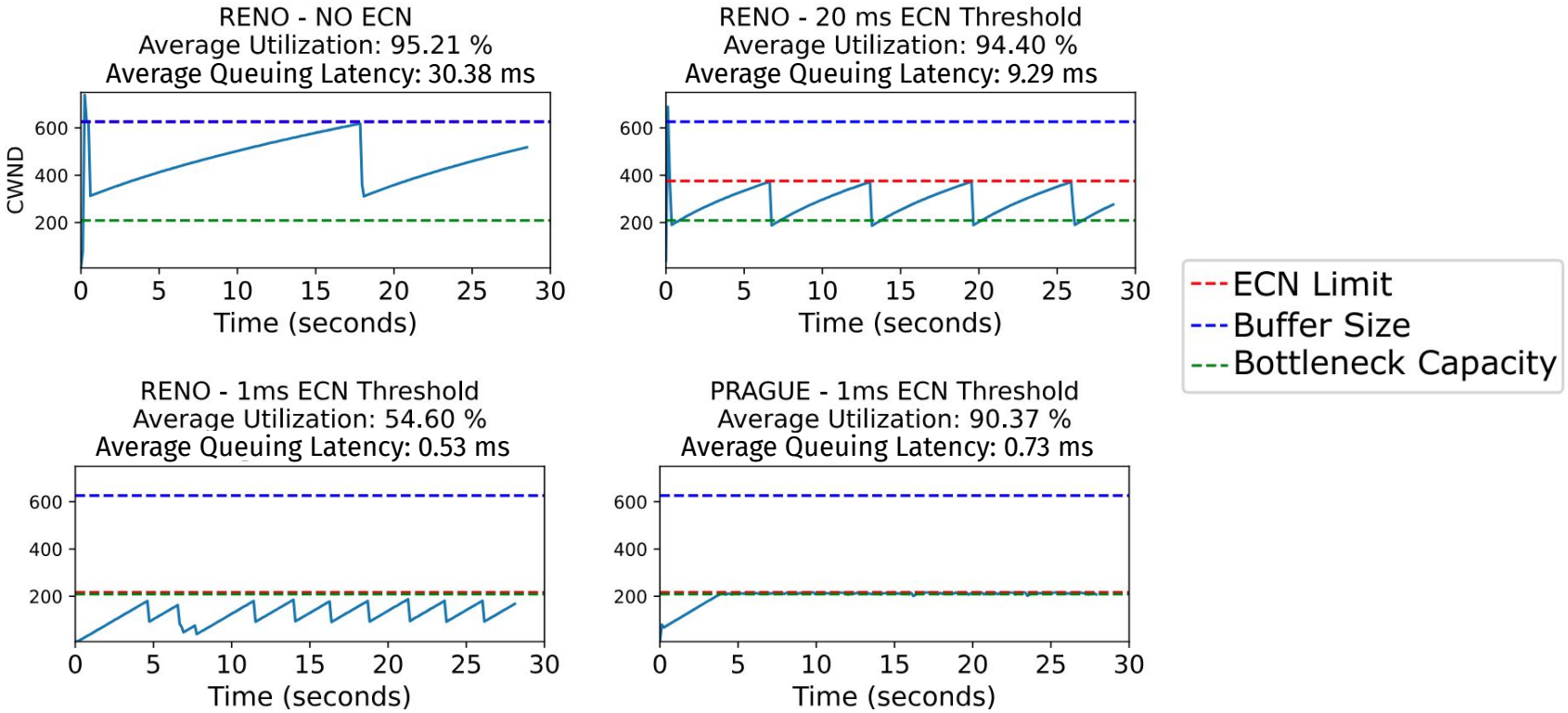
L4S: Low Latency, Low Loss and Scalable Throughput

Core Mechanisms:

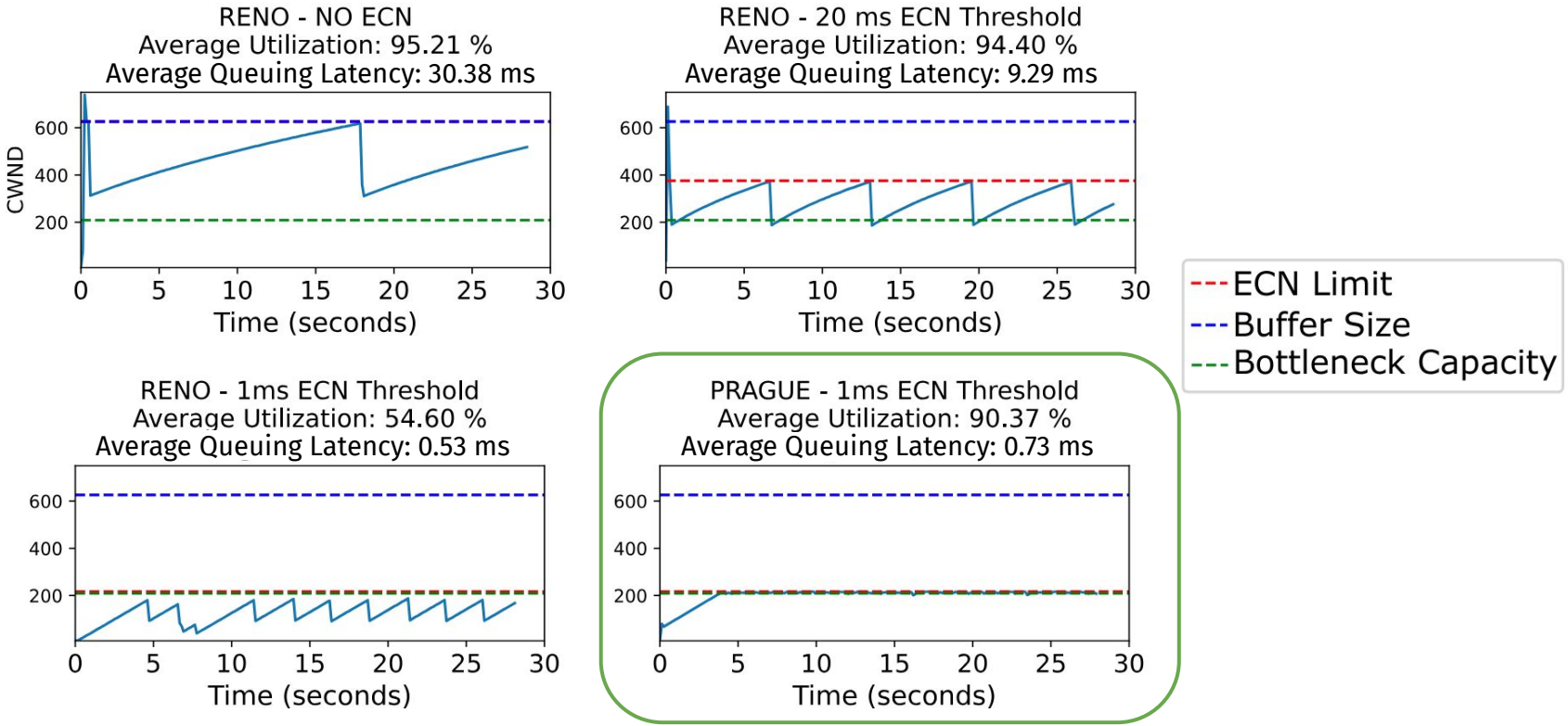
- Scalable Congestion Control (Scalable CC)
- Accurate Explicit Congestion Notification (AccECN)
- Dual Queue Active Queue Management (DualQ AQM)



Scalable CC is necessary to maintain very low queuing delay



Scalable CC is necessary to maintain very low queuing delay



Dual Queue AQM

Classic and L4S flows do not co-exist well! We need to isolate them.

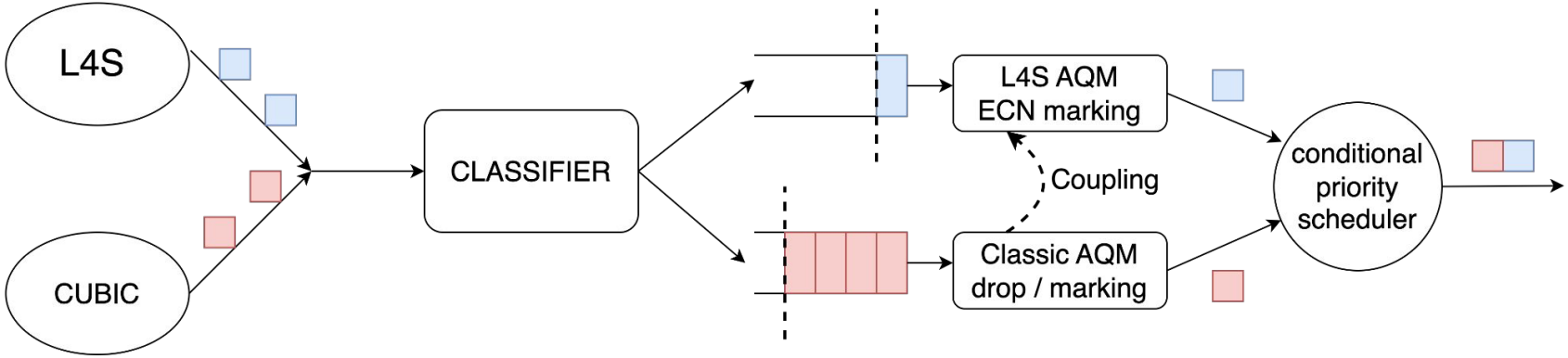
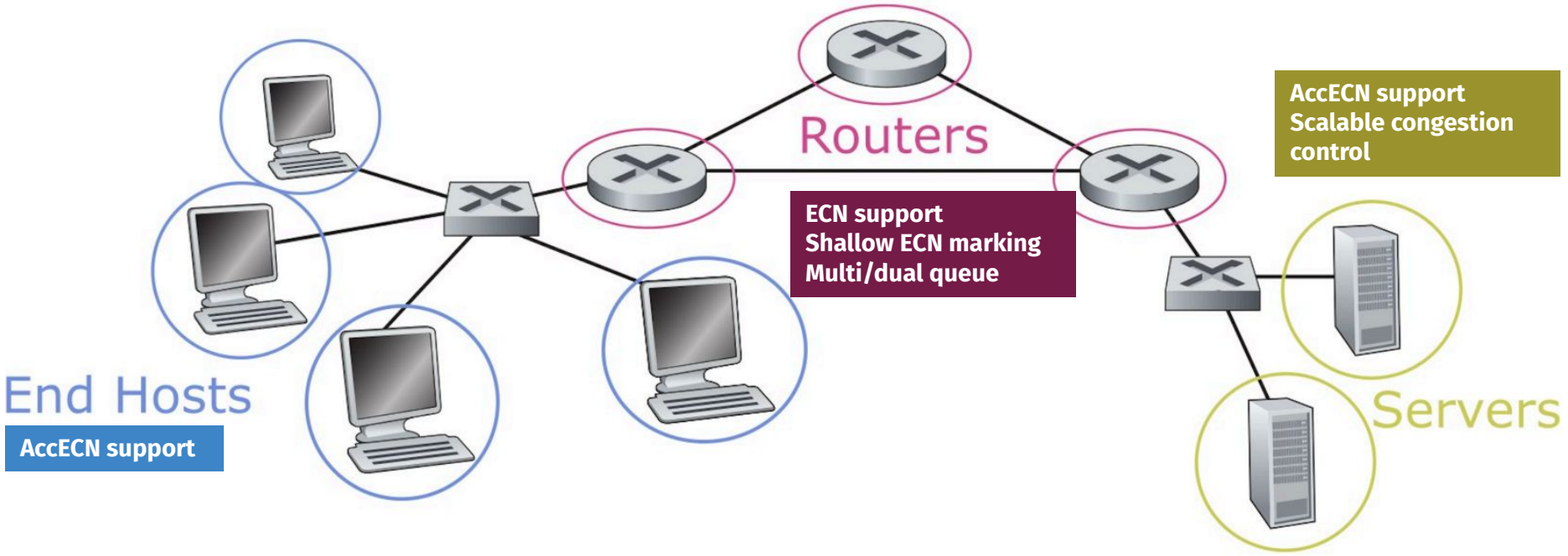


Fig: Dual Queue Coupled AQM

Binary Codepoint	Codepoint Name	Meaning
00	Not-ECT	Not ECN-capable transport
01	ECT(1)	L4S-capable transport
10	ECT(0)	Not L4S-capable transport
11	CE	Congestion Experienced

Table I: L4S codepoints and meaning

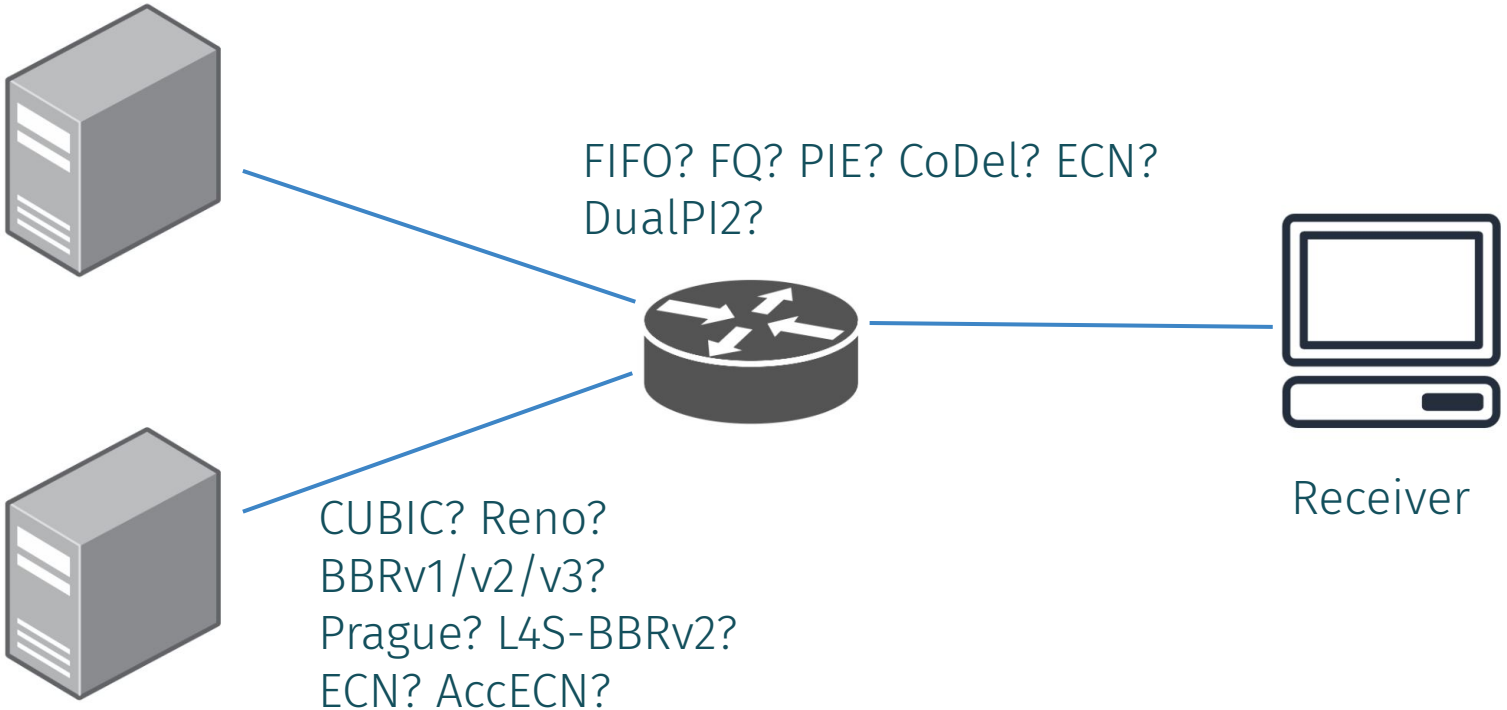
L4S involves both endpoints and routers!



A new protocol won't be deployed universally, all at once - there will be some incremental deployment.

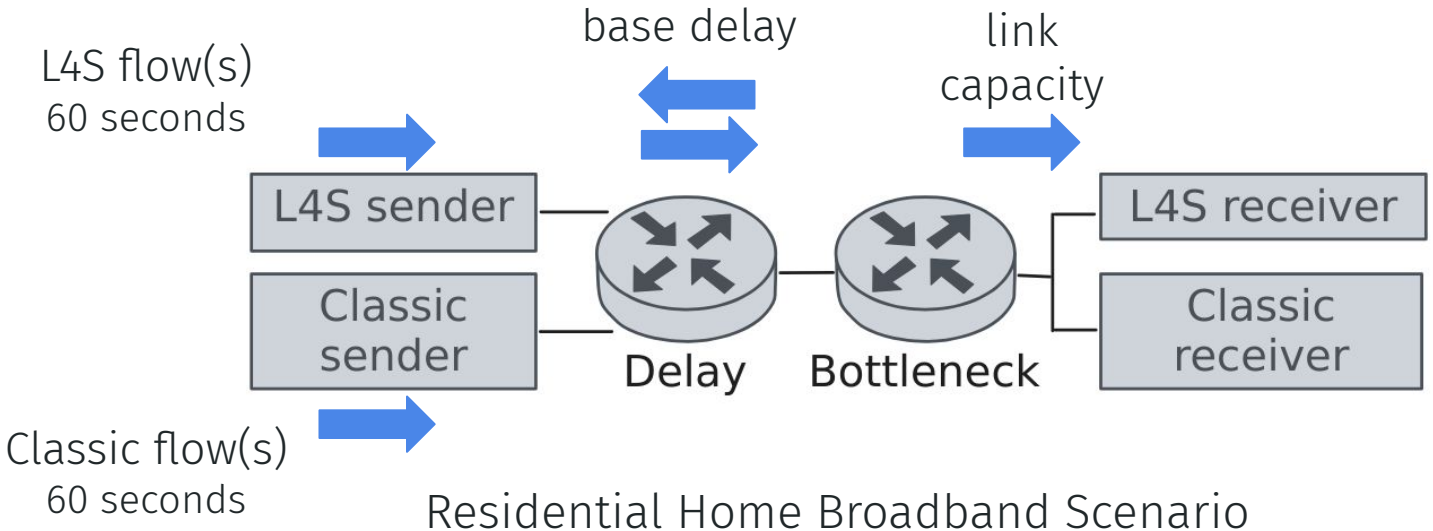
As a sender, should I use L4S?

Sender is not aware of the bottleneck type and competing flows!



Experiment design on FABRIC testbed: topology

A flow is most likely to encounter a bottleneck either at a peering point, or at the access link. We emulate network conditions that are representative of an access link: **10 ms base RTT**, **100 Mbps bottleneck link capacity**.



Residential Home Broadband Scenario

FABRIC

Experiment design on FABRIC testbed: single queue types

	ECN	AQM	Multi/Dual Queue	Shallow ECN marking for L4S
★ FIFO: single drop tail queue				
★ Drop-based PIE (+AQM): single queue with PIE AQM		✅		
Drop-based CoDel (+AQM): single queue with CoDel AQM				
FIFO (+ECN): single drop tail queue with ECN	✅			
CoDel (ECN + AQM): single queue with CoDel AQM	✅	✅		

★ prevalent in the Internet

Source: G. White, Operational Guidance on Coexistence with Classic ECN during L4S Deployment, IETF, Mar. 17, 2024.
G. White, personal communication, Feb, 2025.

Experiment design on FABRIC testbed: multi-queue types

	ECN	AQM	Multi/Dual Queue	Shallow ECN marking for L4S
★ DualPI2 (ECN + Dual-Queue + AQM): proposed L4S bottleneck	✅	✅	✅ (Dual)	✅
★ L4S-aware FQ-CoDel (ECN + Multi-Queue + AQM): fair queuing with the CoDel AQM	✅	✅	✅ (Multi)	✅
★ FQ-CoDel (ECN + Multi-Queue + AQM): fair queuing with the CoDel AQM	✅	✅	✅ (Multi)	
FQ (ECN + Multi-Queue): fair queue with flow isolation and ECN	✅		✅ (Multi)	

★ prevalent in the Internet

- DualPI2 is not prevalent but it is getting deployed as an enabler of L4S.
- L4S-aware FQ-Codel is not prevalent but perhaps prevalent in the future due to its L4S support.

Source: G. White, Operational Guidance on Coexistence with Classic ECN during L4S Deployment, IETF, Mar. 17, 2024. & G. White, personal communication, Feb, 2025.

Experiment design on FABRIC testbed: L4S congestion controls

	Loss-Based	Rate Based	ECN Support
Prague (main L4S CC)	✅		AccECN
L4S-compatible BBRv2*		✅	AccECN

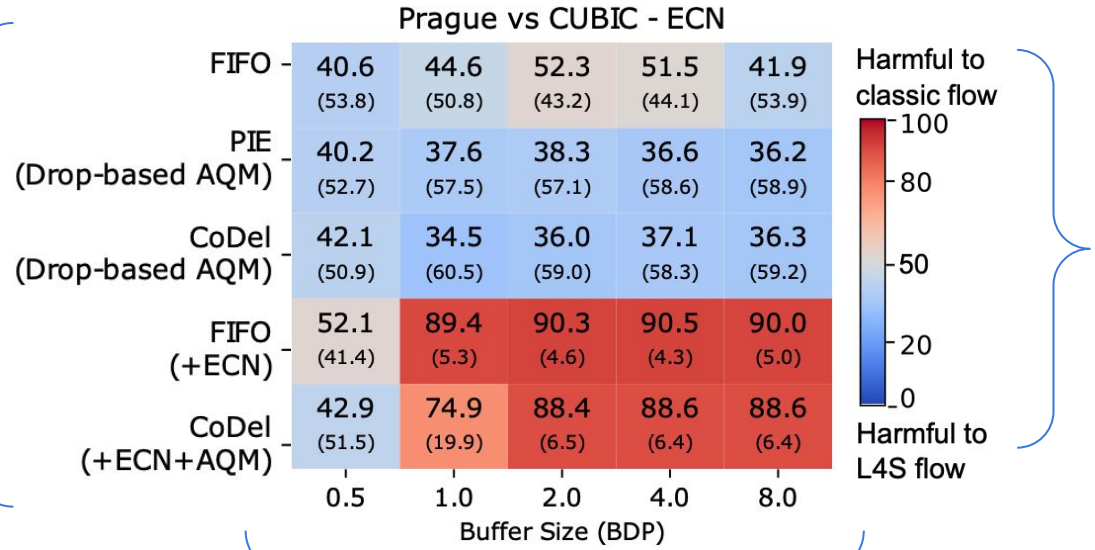
* BBRv2 with L4S marking and AccECN.

Experiment design on FABRIC testbed: non-L4S congestion controls

	Loss-Based	Rate Based	ECN Support
Cubic	✅		✅ (Classic)
BBRv1		✅	Does not support ECN
BBRv2		✅	✅ (DCTCP-style ECN)
BBRv3		✅	✅ (DCTCP-style ECN) (minRTT needs to be <5ms.)

Example Results to Illustrate Our Methodology

Different bottleneck types



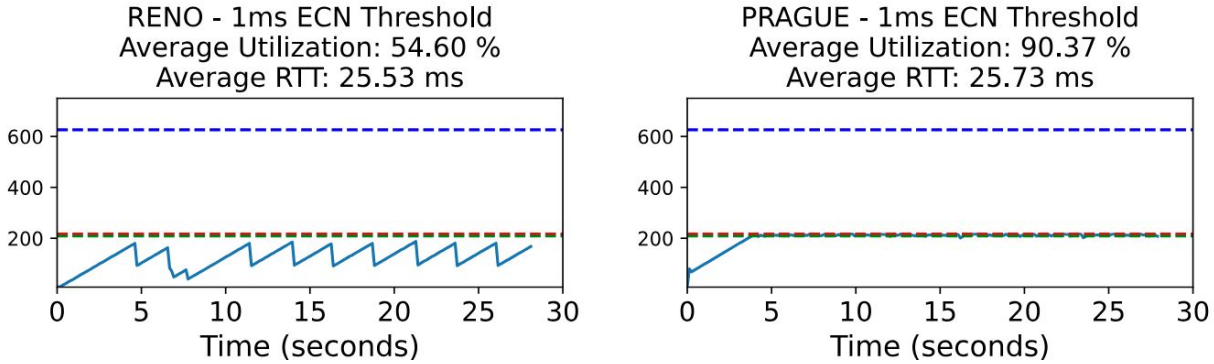
Throughput share

Different buffer sizes

Single Queue with ECN Support - L4S flow dominates!

🔴 Single queue + ECN: L4S flow captures 75-90% of capacity (leaves classic flow with as little as 10%).

🟡 But it is not prevalent on the Internet!



DualPI2 (★) does not work well with BBR!

🔴 DualPI2 (L4S-compatible queue): L4S flow gets less than 5-15% of capacity against BBR (v1/v2/v3) flow.

🔴 This finding is concerning because of the (increasing) prevalence of BBR on the Internet.

Fair queuing (★) allows safe coexistence

🟢 With a fair queue that enforces flow isolation, any variety of L4S congestion control can coexist with any variety of classic congestion control.

🟢 L4S and Classic flows share the capacity fairly.

🟡 Only L4S-aware FQ-CoDel(★) also supports ultra-low latency.

Low-latency requires full L4S support

- Low-latency benefits are realized only when all of the ingredients are in place (endpoints + middlebox).
- DualPI2 or L4S-aware FQ bottlenecks with L4S flow
- L4S flows achieve sub-millisecond queuing latency.

Our finding: L4S is not “Internet-safe” yet!

→ Early results from controlled deployments have been promising but...

→ Given that the sender of an L4S flow cannot be sure what type of queue is at the bottleneck router or what other flows will share it, safe coexistence across the Internet cannot be guaranteed right now.

→ A flow using L4S congestion control may be "starved" by or "starve" other flows at a shared bottleneck.

Thank You for Listening !

Experiment Artifacts:
https://github.com/fatihsarpkaya/L4S-PAM2025

Contact: fbs6417@nyu.edu

Extra Slides

Fatih Berkay Sarpkaya, Fraida Fund, Shivendra Panwar

Passive and Active Measurement (PAM) International Conference, 2025

Summary

AQM	CUBIC	BBRv1	BBRv2 (one-sided DCTCP ECN)	BBRv2 (two-sided DCTCP ECN)	BBRv3
SQ (Drop)	✓ (Fig. 3, 15)	✗ (Fig. 13)	✗ Prague ✓ L4S-BBRv2 (Fig. 7, 11, 19, 25, 27)		
SQ (ECN)	✗ (Fig. 3, 15)	✗ (Fig. 13)	✗ (Fig. 7, 19)	✓ (Fig. 11, 19)	✗ (Fig. 25, 27)
DualPI2 (ϟ)	✓ (Fig. 4, 16)	✗ (Fig. 13)	✓ (Fig. 8, 20)	✗ (Fig. 11, 20)	✗ (Fig. 26, 28)
FQ (ϟ if L4S-aware)	✓ (Fig. 4, 13, 8, 26, 11, 16, 20, 28)				

Table 1: Can a sender turn on L4S (✓) or would it harm/be harmed if it did (✗)? (ϟ indicates settings where ultra-low latency is achieved, otherwise it is not.)

Research Questions

1. TCP Prague is a scalable congestion control prototype developed by the L4S team. Can senders be assured that TCP Prague will not cause harm to, or be harmed by, another flow at a shared bottleneck?

We have ***partially*** investigated this question in our previous paper [2].

2 F. B. Sarpkaya, A. Srivastava, F. Fund, and S. Panwar, "To switch or not to switch to TCP Prague? Incentives for adoption in a partial L4S deployment," ANRW '24, Vancouver, Canada, Jul. 2024.

Research Questions

2. There is an implementation of BBRv2 that adopts some of the principles of scalable congestion control used by TCP Prague. ***Does L4S-compatible BBRv2 have more favorable properties for adoption than TCP Prague?***

3. Is the harm caused by or to the L4S-compatible flow mitigated when the bottleneck is shared by a large number of flows?

4. In BBRv3, ECN is switched ON or OFF automatically, depending on whether or not a constant path delay threshold is exceeded. ***Is the BBRv3 approach more favorable for the deployment of L4S-compatible congestion control?***

“Ready for Adoption” – Evaluation Logic

There are various methods to evaluate the extent to which a CCA achieves fairness:

- Jain’s Fairness Index (JFI)
- Max-min fairness (MmF)
- Normalizing the throughput difference between competing flows.
- Ratio-based fairness measure.
- Harm-based fairness

- Our evaluations aim to inform senders that “if you use scalable congestion control in this type of network, you may experience this behavior.”

- We focus on the results where a flow gets less than 20% of bottleneck link bandwidth or more than 80% (when it should get 50%). This heuristic helps us identify the most problematic settings.

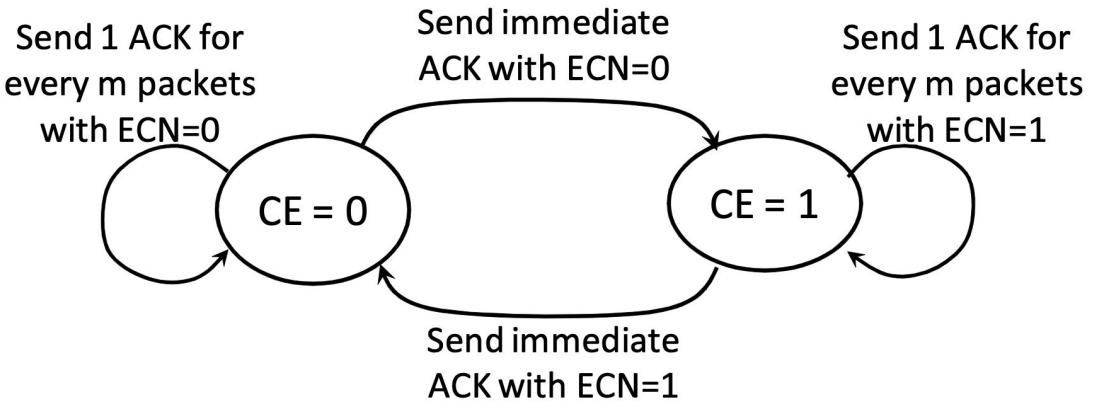
DCTCP Style ECN Marking vs Classic Marking

- Classic Marking:

- After the receiving a CE marked packet, the receiver persistently marks the ECE flag in the TCP header until the sender acknowledges it with a CWR flag.

- DCTCP style Marking

- It sets ECE flag if and only if a packet has CE codepoint.



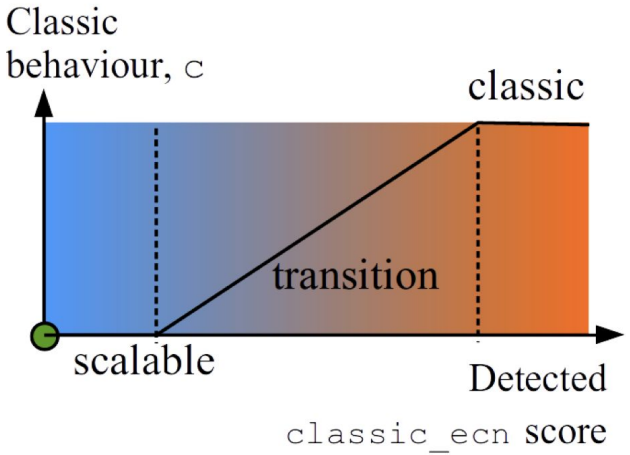
DCTCP Style Marking on the Receiver

ECN Fallback Algorithm

- This algorithm determines the AQM type, primarily by measuring

  - RTT variation
  - queue depth
  - self-limiting factors (i.e. application or receiver window size.)

- The algorithm's decision, whether for a pure classic AQM, a pure L4S AQM, or a transition between them, influences the congestion window reduction.




```
// original DCTCP reduction within prague_ssthresh()
reduction = cwnd * alpha / 2;

// reduction alternative #2
reduction = cwnd * max(alpha, c * ALPHA_ABE) / 2;
```