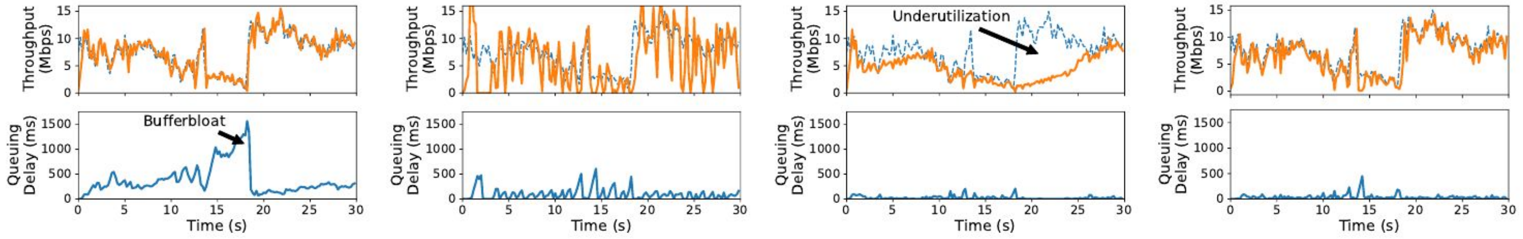
ABC: A Simple Explicit Congestion Controller for Wireless Networks

Presenter: Xuyang Cao

Background / Relevant Works

Three categories of TCP:

- E2E
  - “to estimate the link capacity, it must utilize the link fully and build up a queue” (probing)
- In-Network
  - Non-trivial modifications to hardware and protocols, and hard to deploy
  - Currently they signal overutilization, but not underutilization
- Link-Layer Informed
  - Inaccurate capacity estimation (really?)



(a) Cubic (b) Verus (c) Cubic+CoDel (d) ABC

Motivations

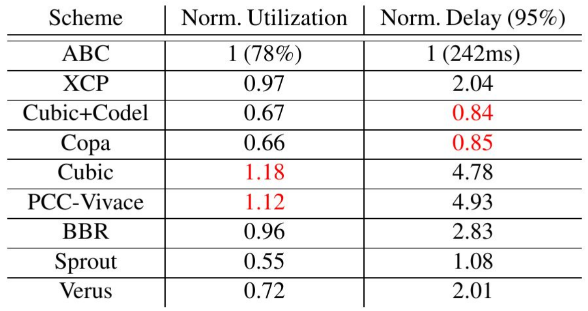
A new TCP, a combination of In-network and link-layer informed

- Build upon the ECN, no IP-header format modification: easy to deploy
  - Still need router logic modification (e.g., marking the acceleration or brake)
  - Need the sender congestion control modification
  - Need receiver ACK modification
  - Need (wireless) link capacity estimation.

- Simple in-network and end algorithm, fast feedback loop and per-packet reaction

Contributions

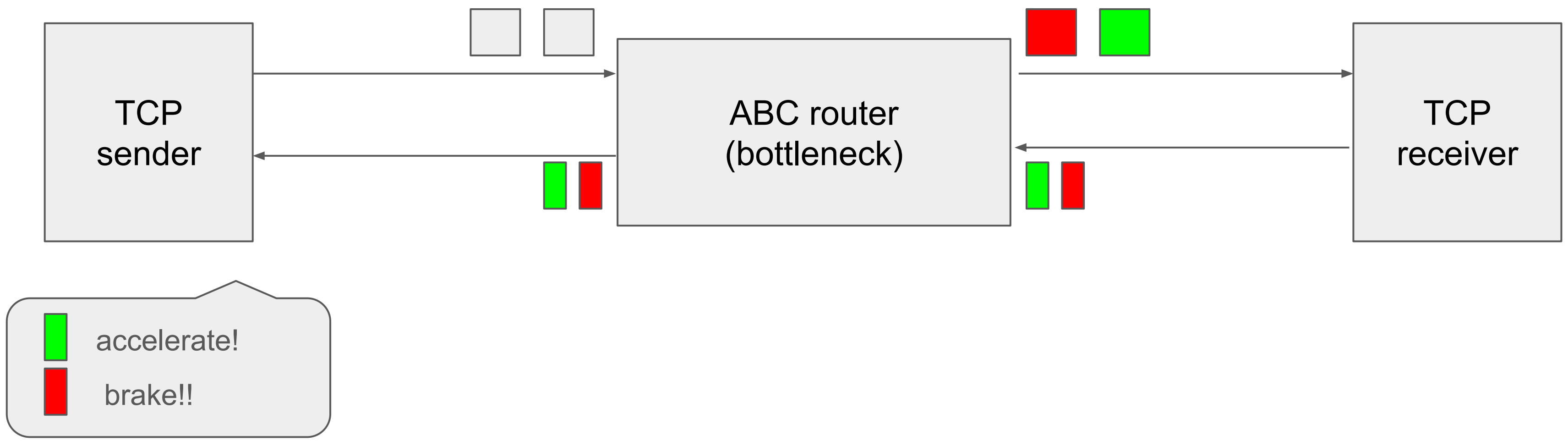
- ABC
  - Simple but robust network-assisted CC feedback loop.
- Wi-Fi telemetry and link capacity estimation
  - Real prototyping on NETGEAR and OpenWrt
- Its philosophy to handle non-ABC flows and non-ABC/multiple bottlenecks



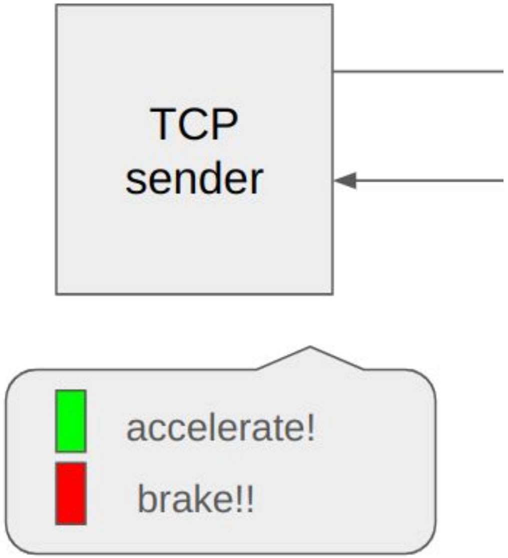
Scheme	Norm. Utilization	Norm. Delay (95%)
ABC	1 (78%)	1 (242ms)
XCP	0.97	2.04
Cubic+Codel	0.67	0.84
Copa	0.66	0.85
Cubic	1.18	4.78
PCC-Vivace	1.12	4.93
BBR	0.96	2.83
Sprout	0.55	1.08
Verus	0.72	2.01

ABC Overview

- A in-network approach, similar to driving where you sometimes tap accelerator or brake



ABC Design



On receiving an ACK,

- If accelerate ACK, CWND + 1
- If brake ACK, CWND - 1

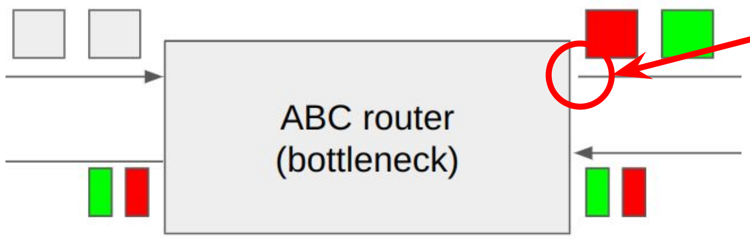
A large dynamic range of CWND size adjustment over a RTT.

Currently CWND w.

In 1 RTT, it receives w ACK, with w * f as accel and w - w * f as brake.

- New CWND w': w + wf - (w - wf) = 2wf
- (0 <= f <= 1)
- 0 <= w' <= 2w

ABC Design



tr(t): target rate

$$tr(t) = \eta\mu(t) - \frac{\mu(t)}{\delta}(x(t) - d_t)^+, \qquad (1)$$

Link capacity

Queuing delay

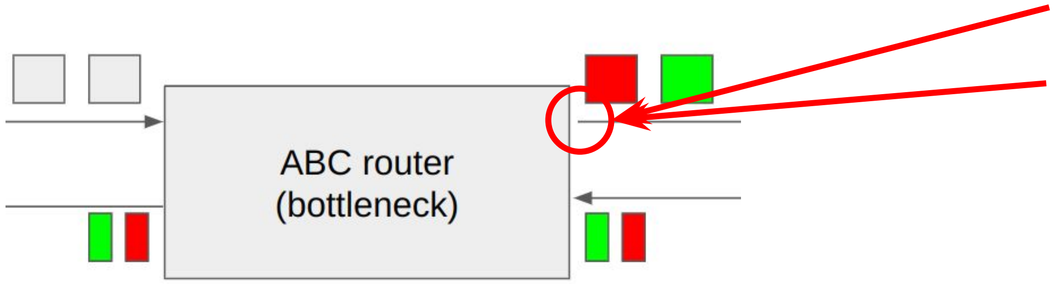
Delay tolerance threshold (e.g., for Wi-Fi aggregate TX), must > router inter-scheduling time

η, δ are tuning parameters. η (<= 1) for not saturating link; δ for how aggressive the rate adaptation will be

ABC Design

tr(t): target rate

cr(t): dequeue rate



cr(t): dequeue rate essentially tells the ACK rate.

Given an accel ratio f(t), then 2cr(t)f(t) will be arrived at router next round.

We want tr(t) = cr(t). Therefore, f(t) = tr(t)/(2cr(t))

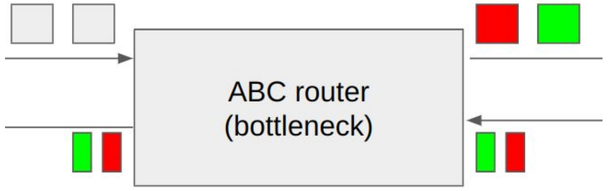
$$f(t) = \min\left\{\frac{1}{2} \cdot \frac{tr(t)}{cr(t)}, 1\right\}. \qquad (2)$$

Of course, f as ratio is capped at 1

ABC Design

tr(t): target rate

cr(t): dequeue rate



$$f(t) = \min\left\{\frac{1}{2} \cdot \frac{tr(t)}{cr(t)}, 1\right\}. \quad (2)$$

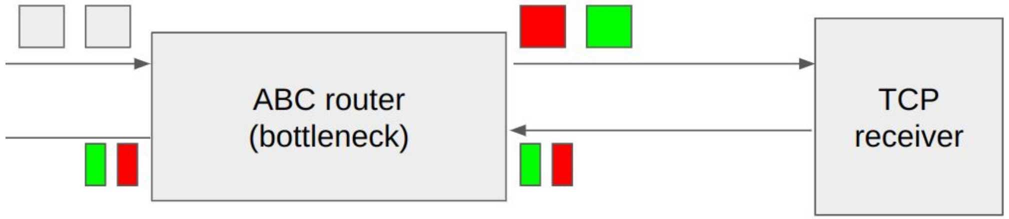
Computation of cr(t) and and tr(t) (i.e., μ) are over a sliding window T.

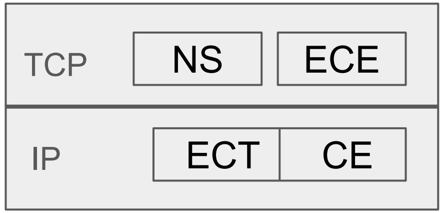
f(t) is re-computed on every de-queued packet. Given f(t), whether marking accel on a packet follows a deterministic, even pattern:

```
token = 0;
for each outgoing packet do
    calculate $f(t)$ using Equation (2);
    token = min(token + $f(t)$, tokenLimit);
    if packet marked with accelerate then
        if token > 1 then
            token = token − 1;
            mark accelerate;
        else
            mark brake;
```

Algorithm 1: Packet marking at an ABC router.

ABC Design





ECT	CE	Interpretation
0	0	Non-ECN-Capable Transport
0	1	ECN-Capable Transport ECT(1)
1	0	ECN-Capable Transport ECT(0)
1	1	ECN set

For ECN,

- Router set 11 on ECT + CE to indicate congestion, if ECN is enabled (i.e., initially marked as 01 or 10).
- Then receiver marks ACK ECE to 1, if the data ECT+CE is 11.

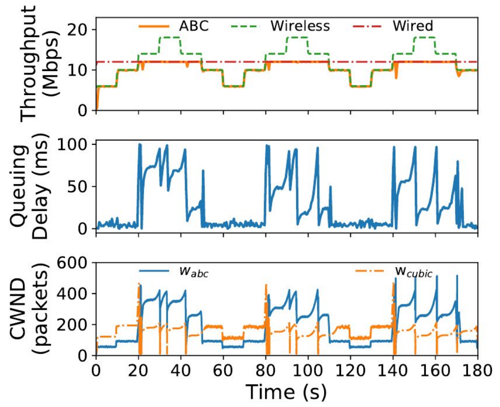
For ABC,

- Router set ECT+CE as 01 for accel and 10 for brake.
- Then ABC receiver marks the NS to 1/0 to indicate the accel or brake.

"we repurpose the NS (nonce sum) bit, which was originally proposed to ensure ECN feedback integrity [17] but has been reclassified as historic"

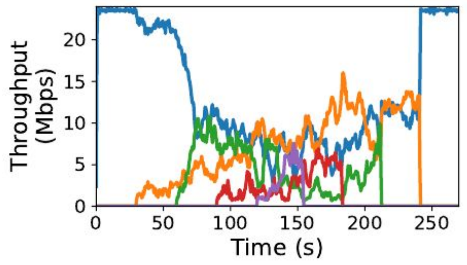
ABC Design (bottlenecks and fairness)

- For multiple ABC routers and to detect the bottleneck,
  - each packet is initially marked accelerate by the sender. ABC routers may change a packet marked accelerate to a brake, but not vice versa.
- For co-existence with non-ABC routers and to detect the bottleneck,
  - Maintain two CWNDs and two CC mechanisms. One is W_abc and uses ABC, the other is a traditional E2E like W_cubic and uses CUBIC. Adopt the min(W_abc, W_cubic).

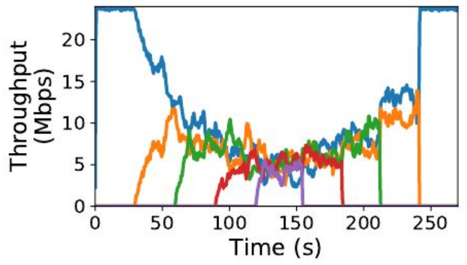


- To ensure fairness,
  - Add an additive increase.

$$w \leftarrow \begin{cases} w+1+1/w & \text{if accelerate} \\ w-1+1/w & \text{if brake} \end{cases} \quad (3)$$



(a) ABC w/o AI



(b) ABC with AI

ABC Design

- ABC router will isolate non-ABC and ABC flows into different queues to ensure fairness.
- In practice, ABC sender updates on bytes (not per packet on CWND size) to account for delayed ACK.
- No problem for lost ACK. It only slow down the CWND changes.
- ABC router will not affect ECN router, but ECN router will affect ABC router.

Wi-Fi Capacity Estimation (single-user)

Wi-Fi will send packet in batch with size M. If app-limited (e.g., b < M long enough), it will also send after wait for a while.

Wi-Fi has layer-2 ACK. Thus we have inter-ACK time T_IA(b, t).

Given frame size S, then the current dequeue rate is $cr(t) = \frac{b \cdot S}{T_{IA}(b,t)}.$

We want the link capacity (limit): $\hat{\mu}(t) = \frac{M \cdot S}{\hat{T}_{IA}(M,t)}.$

Note: $T_{IA}(b,t) = \frac{b \cdot S}{R} + h(t).$

h is overhead, independent of transmission size b * S

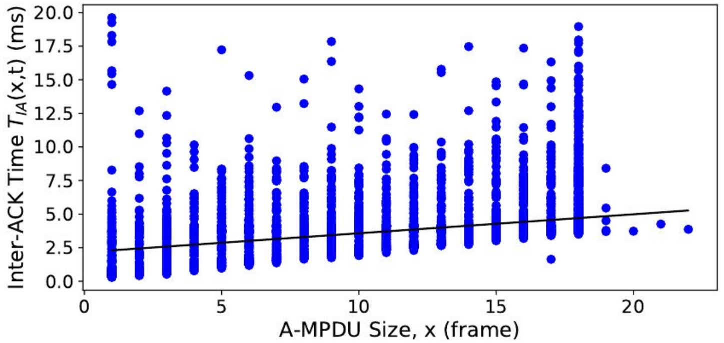
Wi-Fi Capacity Estimation (Continued)

Therefore, given S, R (bitrate), M can be observed,

$$T_{IA}(b,t) = \frac{b \cdot S}{R} + h(t).$$

Also T_IA(b, t) and b can be observed,

$$\hat{T_{IA}}(M,t) = \frac{M \cdot S}{R} + h(t)$$
$$= T_{IA}(b,t) + \frac{(M-b) \cdot S}{R}.$$



We have the estimated T_IA(M, t) and hence estimated link capacity.

Multiple users in Wi-Fi

- Per-user queues case:
  - *“Fairness among different users is ensured via scheduling users out of separate queues”*
  - Aforementioned link estimation for each queue.

- Single queue case:
  - Streams are aggregated and the whole is view as one flow for ABC. Based on probability, all users’ packets will experience the same accel/brake ratio.

Cellular capacity estimation

“the 3GPP cellular standard describes how scheduling information at the cellular base station can be used to calculate per-user link rates.”

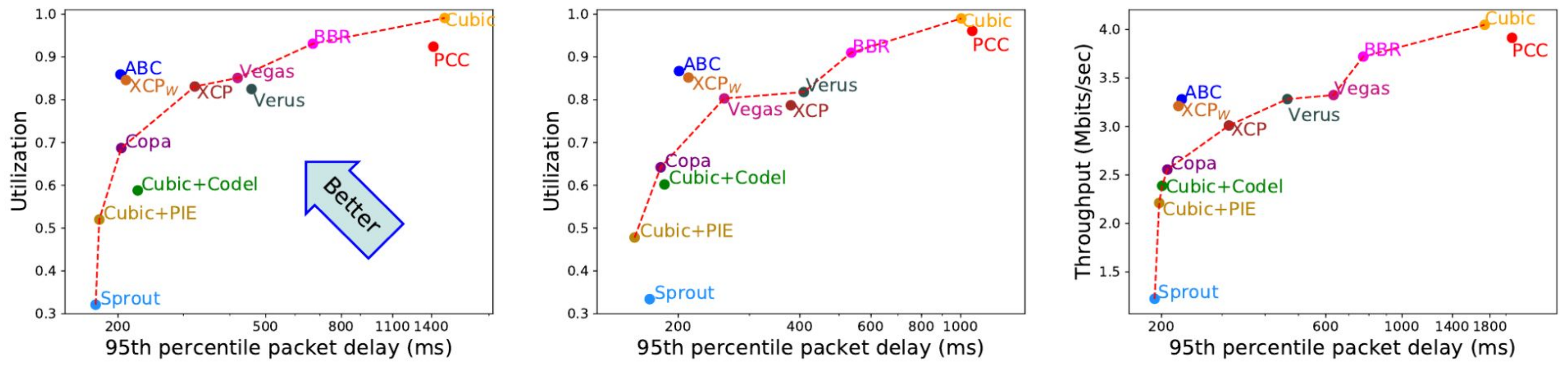
Indeed, but you need non-trivial modification or telemetry to achieve such estimation.

Evaluation

Implementation and setup

- ABC endpoints: implemented as kernel modules using pluggable TCP API (i.e., as a congestion control flavor).
- ABC router: OpenWrt and NETGEAR WNDR 3800 with modified qdisc kernel module and modified Wi-Fi driver to achieve ABC marking and link estimation.
- Mahimahi emulation for cellular setting.
- For emulation, min RTT = 100ms, 250 MTU-sized packet buffer, $\eta = 0.98$ and $\delta = 133$ ms.

Evaluation (cellular)



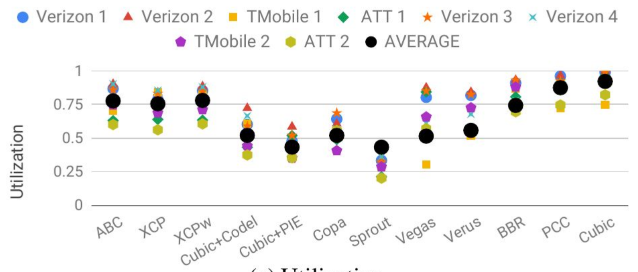
(a) Downlink (b) Uplink (c) Uplink+Downlink

Figure 7: **ABC vs. previous schemes on three Verizon cellular network traces —** In each case, ABC outperforms all other schemes and sits well outside the Pareto frontier of previous schemes (denoted by the dashed lines).

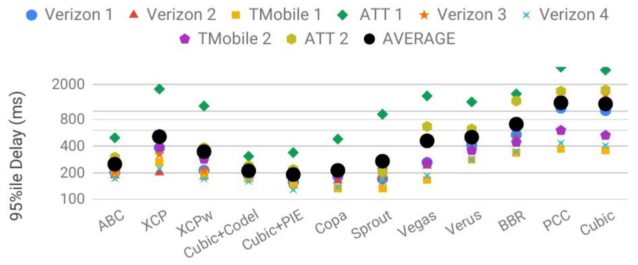
When only Downlink, Uplink, or both uses cellular network.

XCP_w is XCP with per-packet level CWND update.

Evaluation (cellular)



(a) Utilization



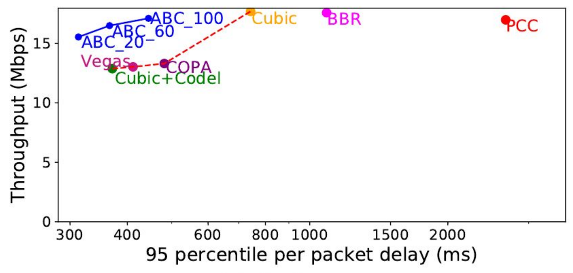
(b) 95th percentile per-packet delay

Figure 8: **95th percentile per-packet delay across 8 cellular link traces —** On average, ABC achieves similar delays and 50% higher utilization than Copa and Cubic+Codel. PCC and Cubic achieve slightly higher throughput than ABC, but incur 380% higher 95th percentile delay than ABC.

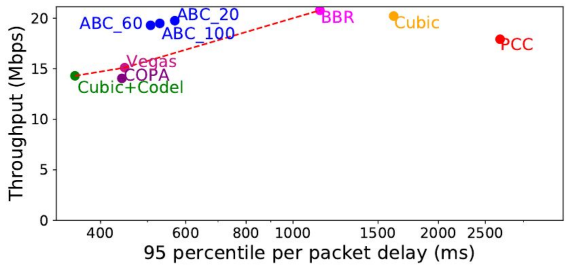
Across a variety of operators.

Essentially ABC achieves the optimal throughput+latency combination, beyond the existing Pareto frontier.

Evaluation (Wi-Fi)



(a) Single user



(b) Two users, shared queue

Figure 9: **Throughout and mean delay on Wi-Fi —** For the multi-user scenario, we report the sum of achieved throughputs and the average of observed 95th percentile delay across both users. We consider three versions of ABC (denoted ABC _*) for different delay thresholds. All versions of ABC outperform all prior schemes and sit outside the pareto frontier.

For two users, they share the same FIFO queue.

Try delay threshold 20ms, 60ms, 100ms dt delay threshold for ABC.

To mimic common Wi-Fi dynamics, vary the MCS between 1 and 7 every 2 seconds.

Coexistence with non-ABC bottleneck (emulation)

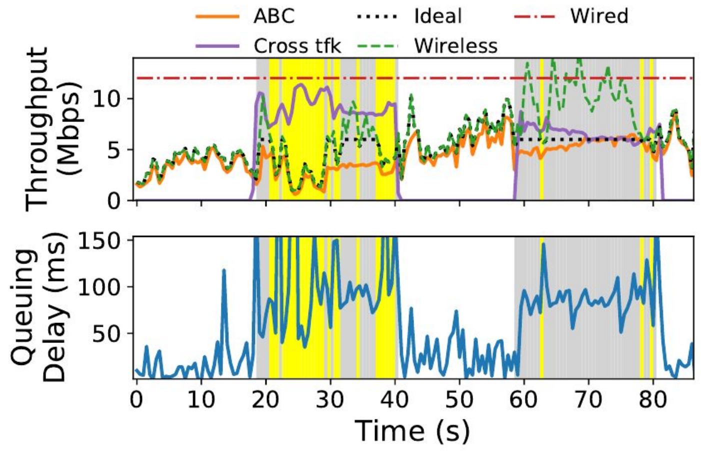
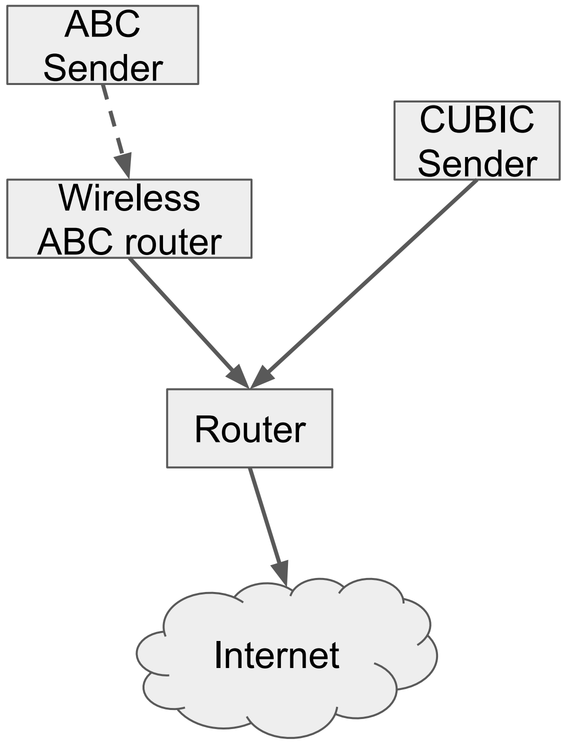


Figure 10: **Coexistence with non-ABC bottlenecks** — ABC tracks the ideal rate closely (fair share) and reduces queuing delays in the absence of cross traffic (white region).

- Grey region: wired is bottleneck, should be fair share
- Yellow region: wireless is bottleneck



Coexistence with other flows

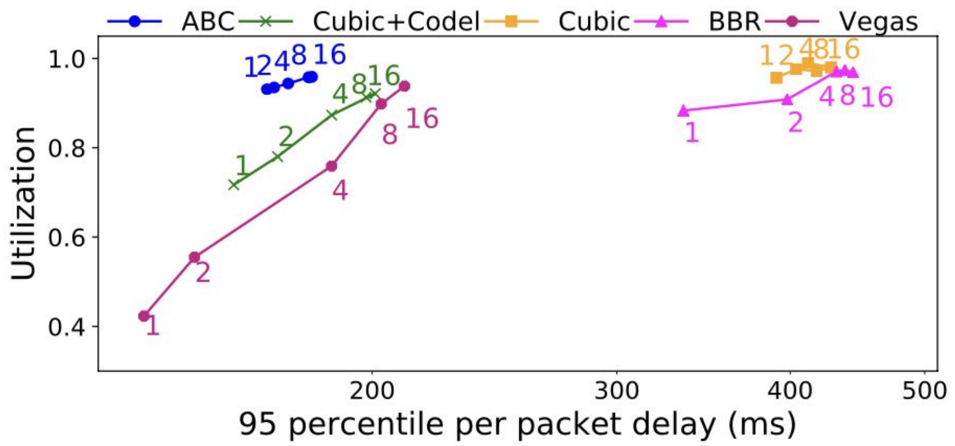
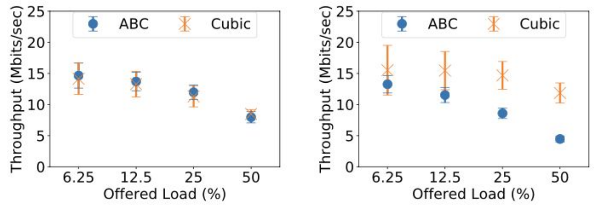


Figure 11: **Coexistence among ABC flows** — ABC achieves similar aggregate utilization and delay irrespective of the number of connections. ABC outperforms all previous schemes.



(a) ABC (b) RCP's Zombie List

Figure 13: **Coexistence with non-ABC flows** — Across all scenarios, the standard deviation for ABC flows is small and the flows are fair to each other. Compared to RCP's Zombie List strategy, ABC's max-min allocation provides better fairness between ABC and non-ABC flows. With ABC's strategy, the difference in average throughput of ABC and Cubic flows is under 5%.

ABC router's fair sharing mechanism for ABC + non-ABC flows

Barely any latency increases and high utilization regardless of flow numbers.

RTT Unfairness is also reasonable

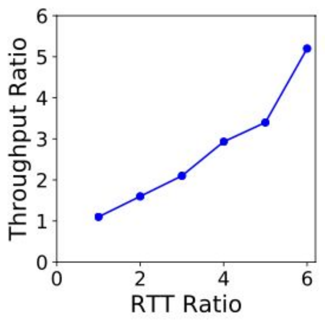
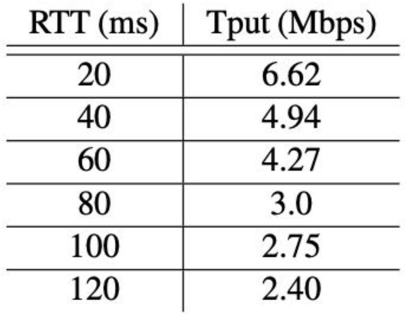


Figure 12: **RTT unfairness**



RTT (ms)	Tput (Mbps)
20	6.62
40	4.94
60	4.27
80	3.0
100	2.75
120	2.40

Table 1: **RTT unfairness**

Left: two flows, one varies RTT from 20 to 120ms, one fixes at 20ms

Right: six flows with different RTTs.

My opinion

- Simple but very robust and versatile in terms of design.

- Albeit claimed easy actualization, to me it’s challenging.
  - Need to modify routers’ logics.
  - The link capacity estimation requires modification or telemetry to Wi-Fi and RANs.
  - Need to modify TCP endpoints.

- If you can estimate the link capacity with high accuracy, why not have the direct link-capacity-informed TCP?

- The evaluation is detailed and throughout.

Discussion!

Let’s look at Perusall.

Conclusion

ABC is a network-assisted simple congestion control, where it mimics the car acceleration/brake, marking the ACK packet with accel/brake signal to tune the CWND. It’s intuitive but also has rigorous mathematical proof to ensure its robustness.