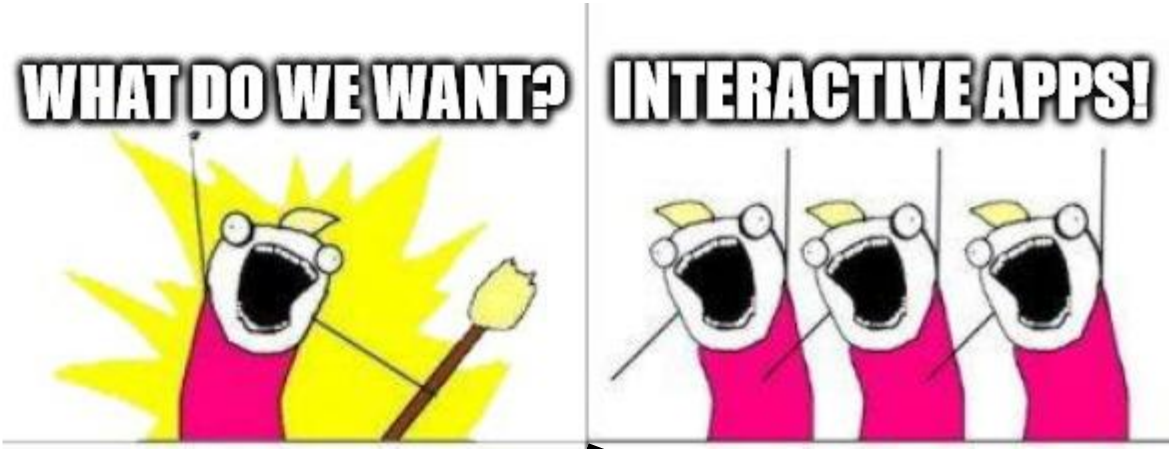
Starvation in End-to-End Congestion Control

Venkat Arun

Mohammad Alizadeh

Hari Balakrishnan

CSAIL, MIT



Loss-based CCAs don't bound delay

Delay bounding Congestion Control Algorithms (CCAs)

Queuing delay
Vegas, FAST,
Copa, Verus

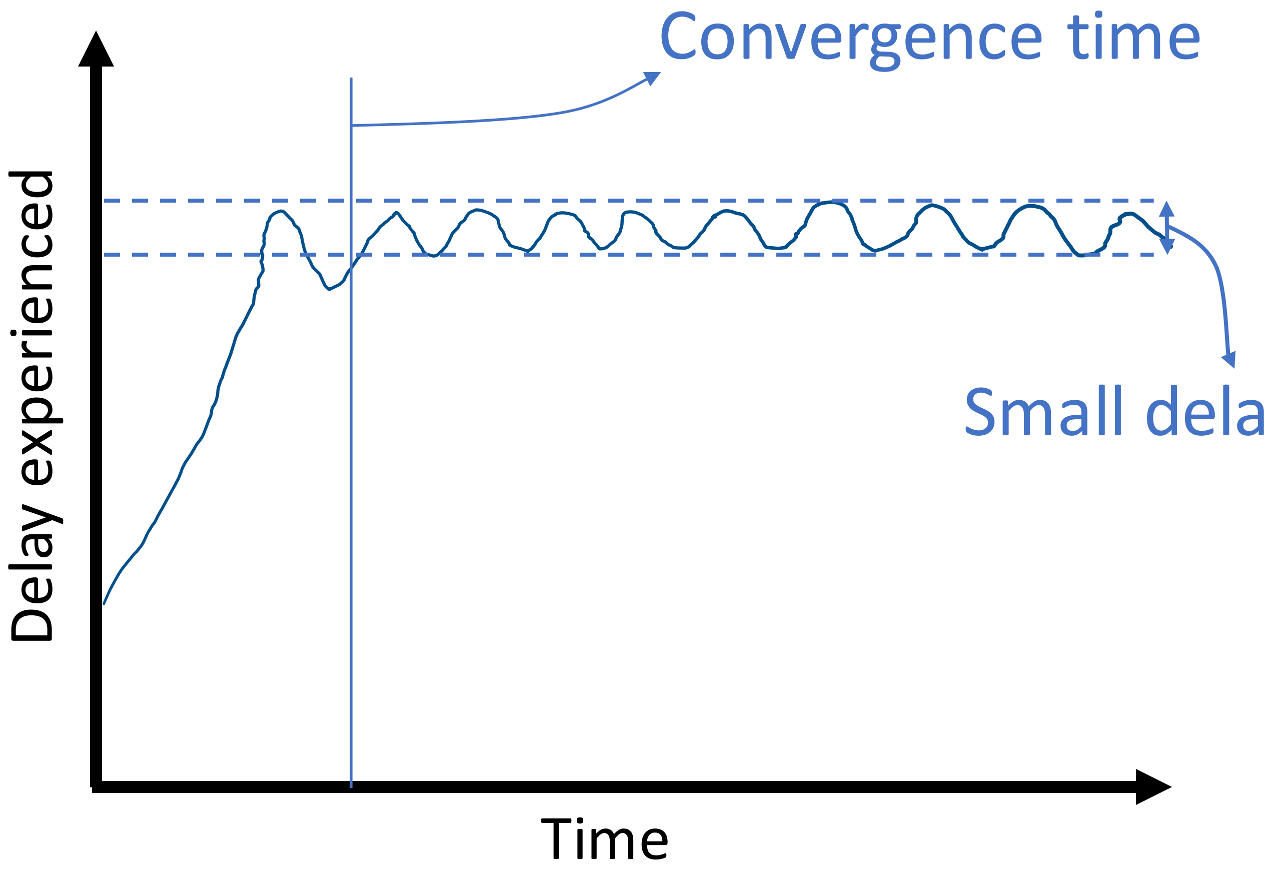
Receive rate
PCP, Sprout,
BBR

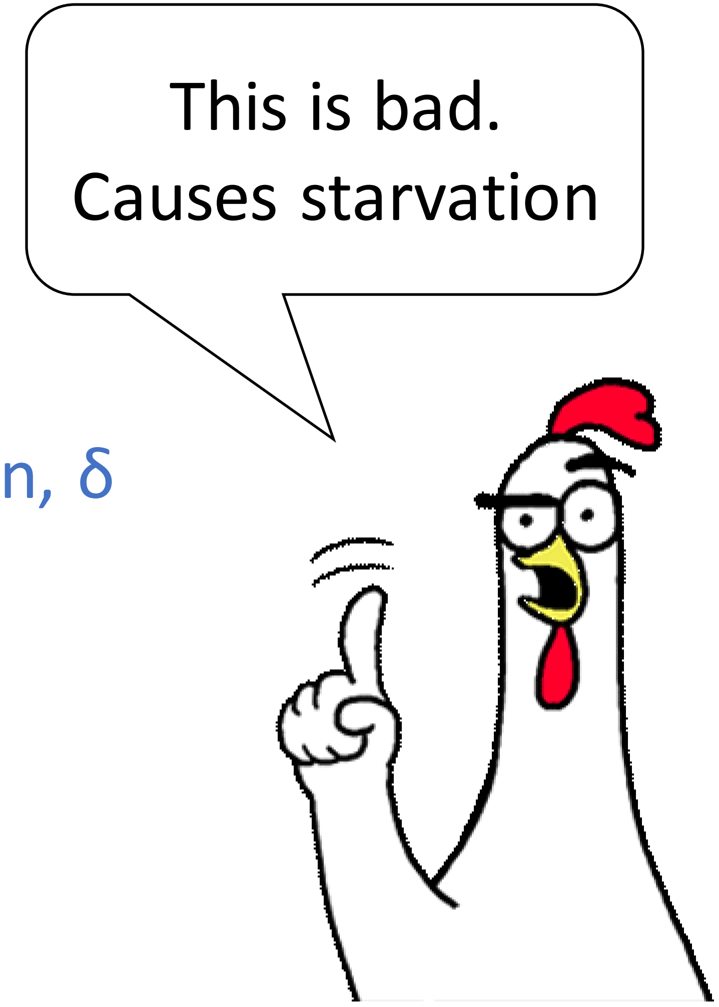
Learning based
Remy, PCC, …

Delay-convergence

Queuing delay
Learning based
Receive rate
Vegas, FAST, Copa, Verus
Remy, PCC, ...
PCP, Sprout, BBR

Delay-convergence (definition)





Starvation is caused by non-congestive delay

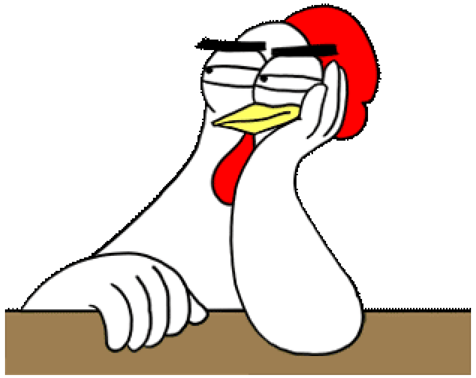
Total delay = Propagation delay

+

Congestive (bottleneck) delay

+

Non-congestive delay



Hard to distinguish between these

Sources of non-congestive delay

Wi-Fi sends TCP ACKs in bursts of tens of ms

Cellular base stations have a complex service process

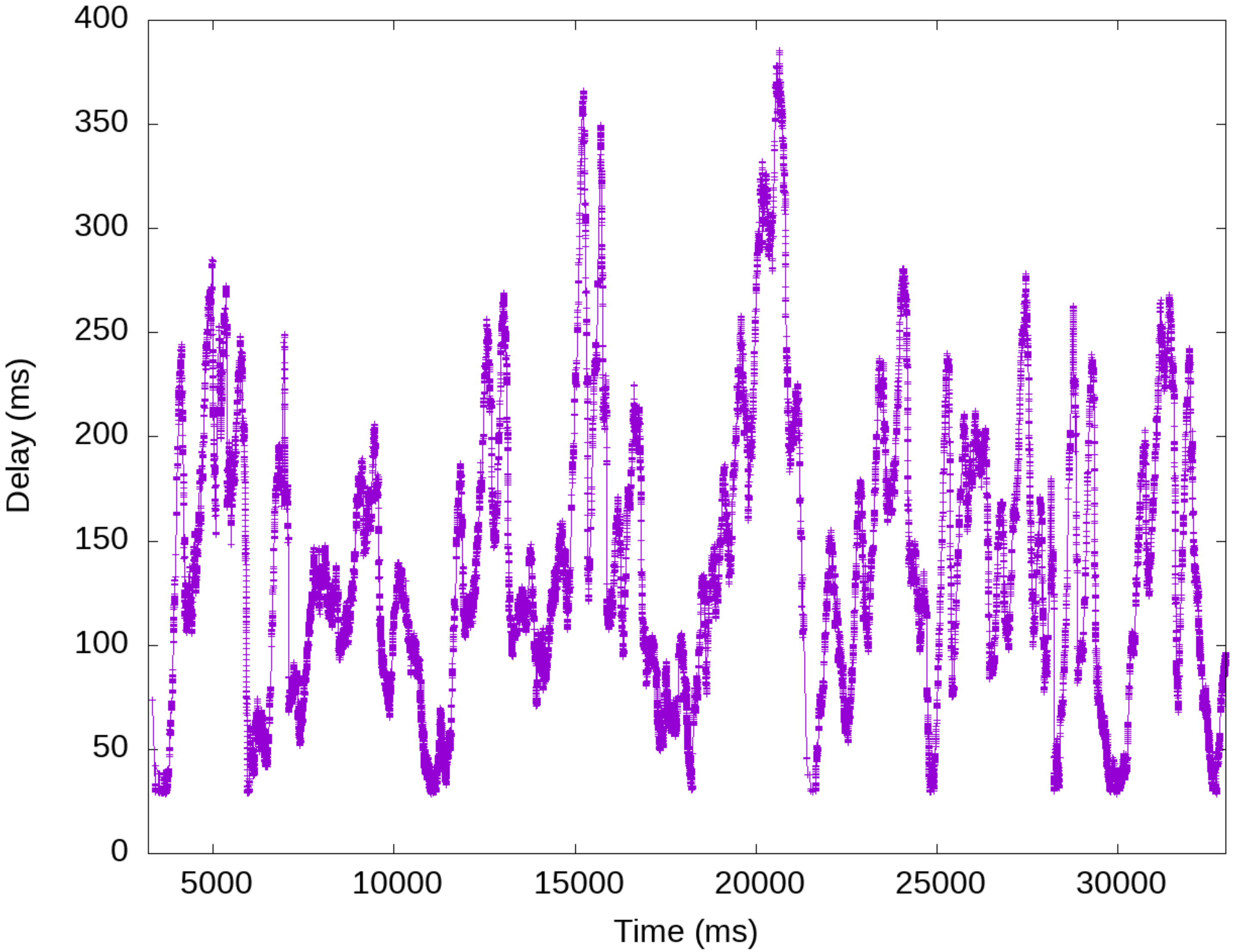
End hosts send packets/acks in bursts

OS will only process packets when it gets the chance

One path can have multiple of these

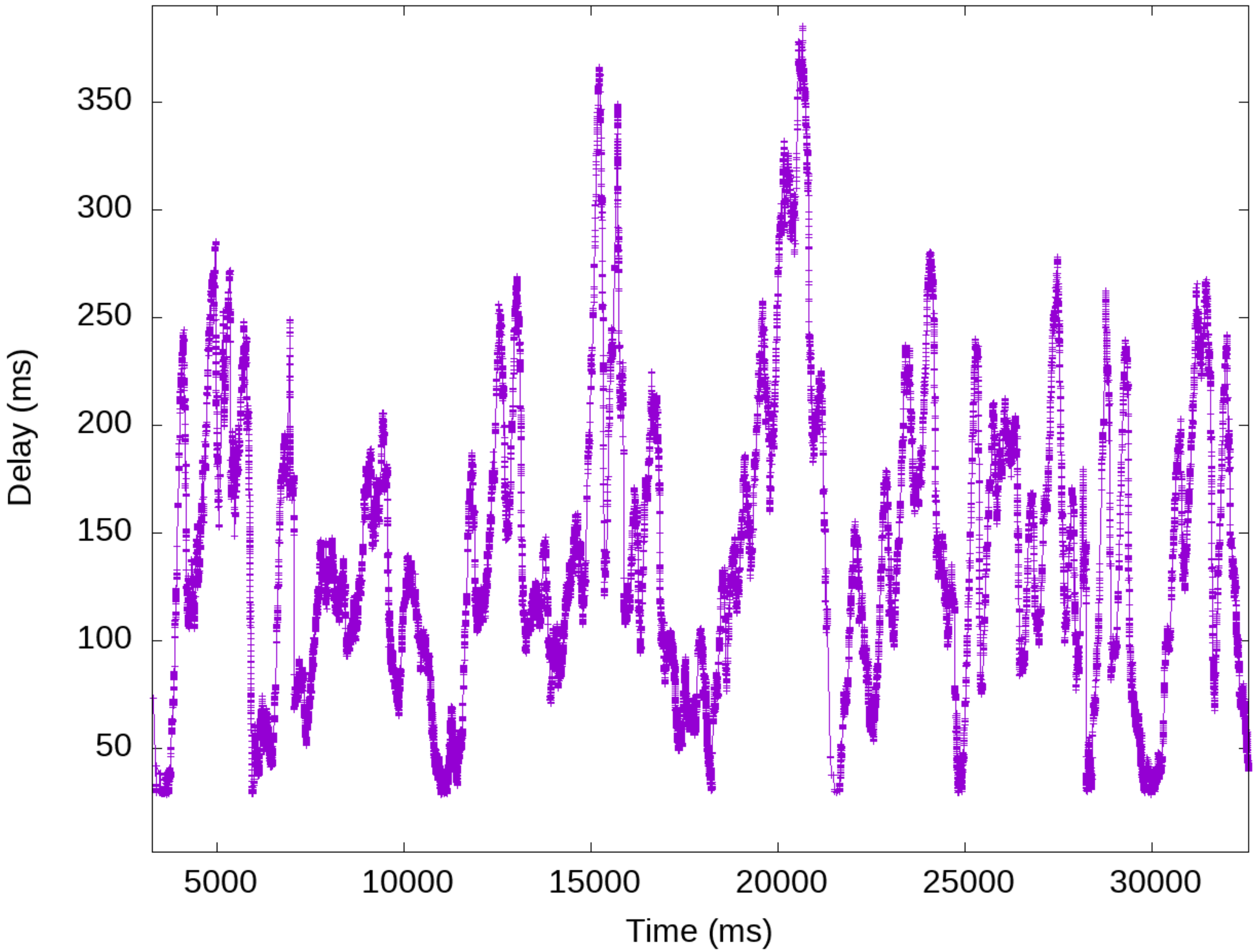
How large is this delay (cellular)?

Pantheon: the training ground for Internet congestion-control research, USENIX ATC'18, Francis Yan et al.



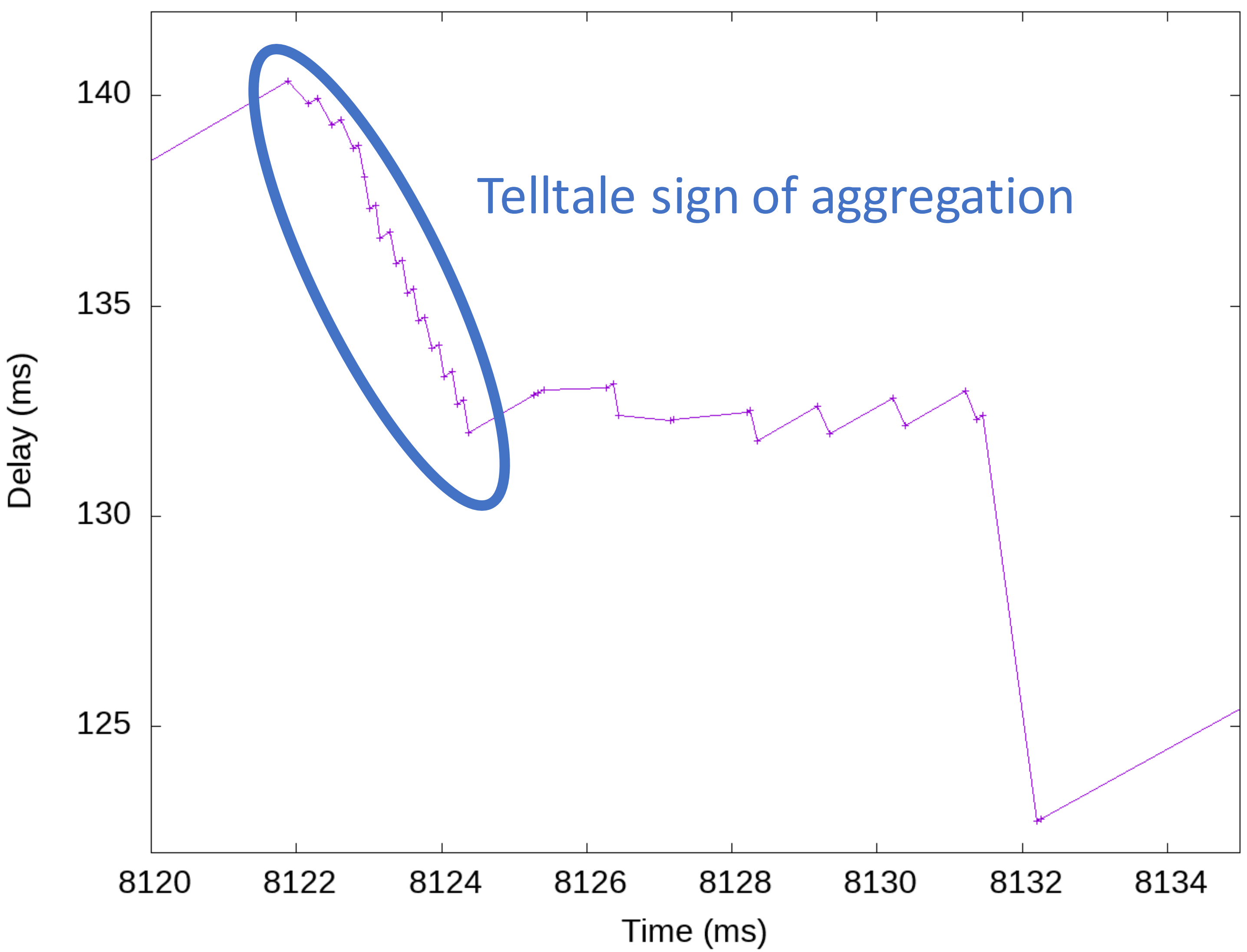
How large is this delay (cellular)?

Pantheon: the training ground for Internet congestion-control research, USENIX ATC’18, Francis Yan et al.



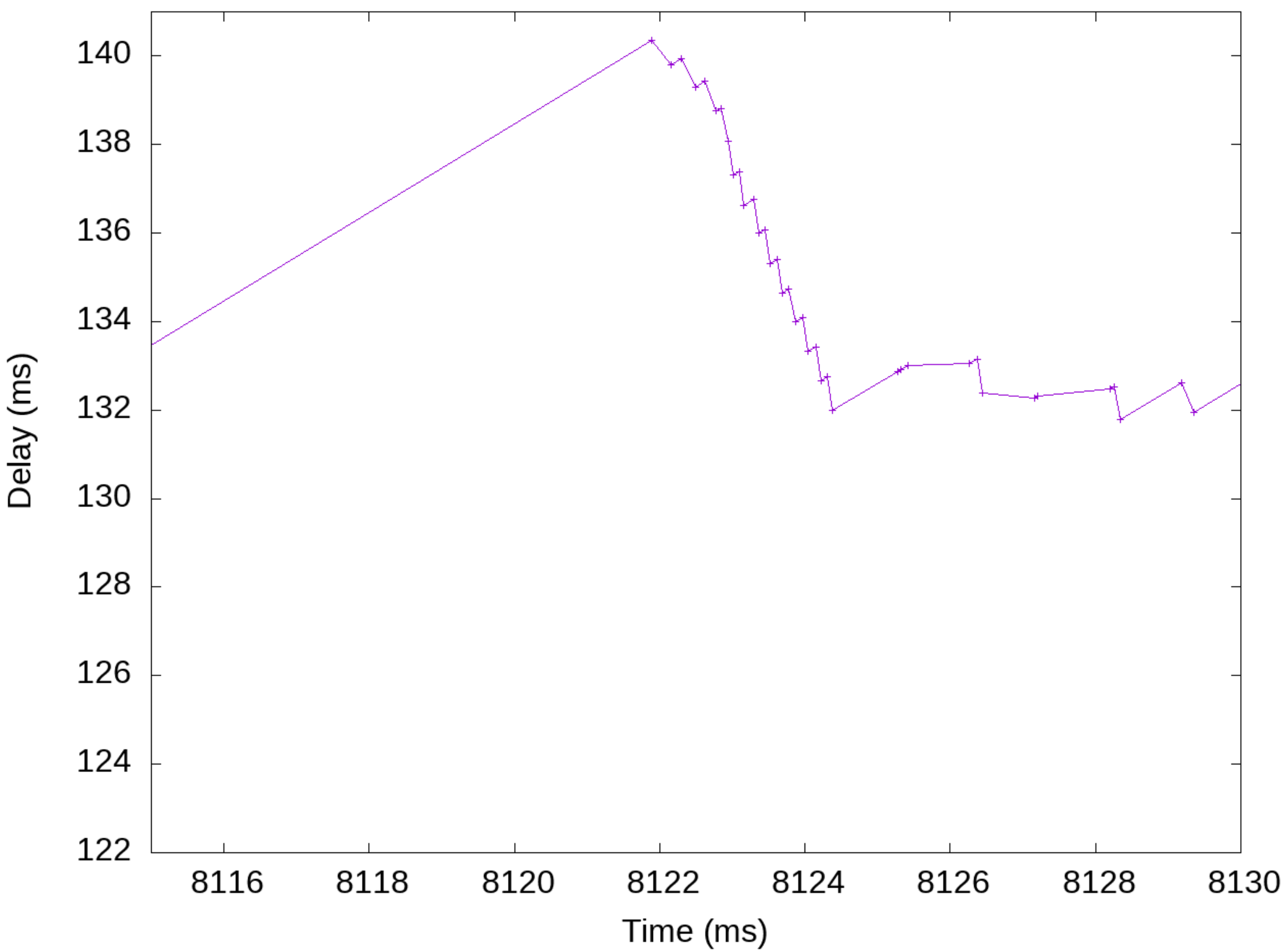
How large is this delay (cellular)?

Pantheon: the training ground for Internet congestion-control research, USENIX ATC’18, Francis Yan et al.



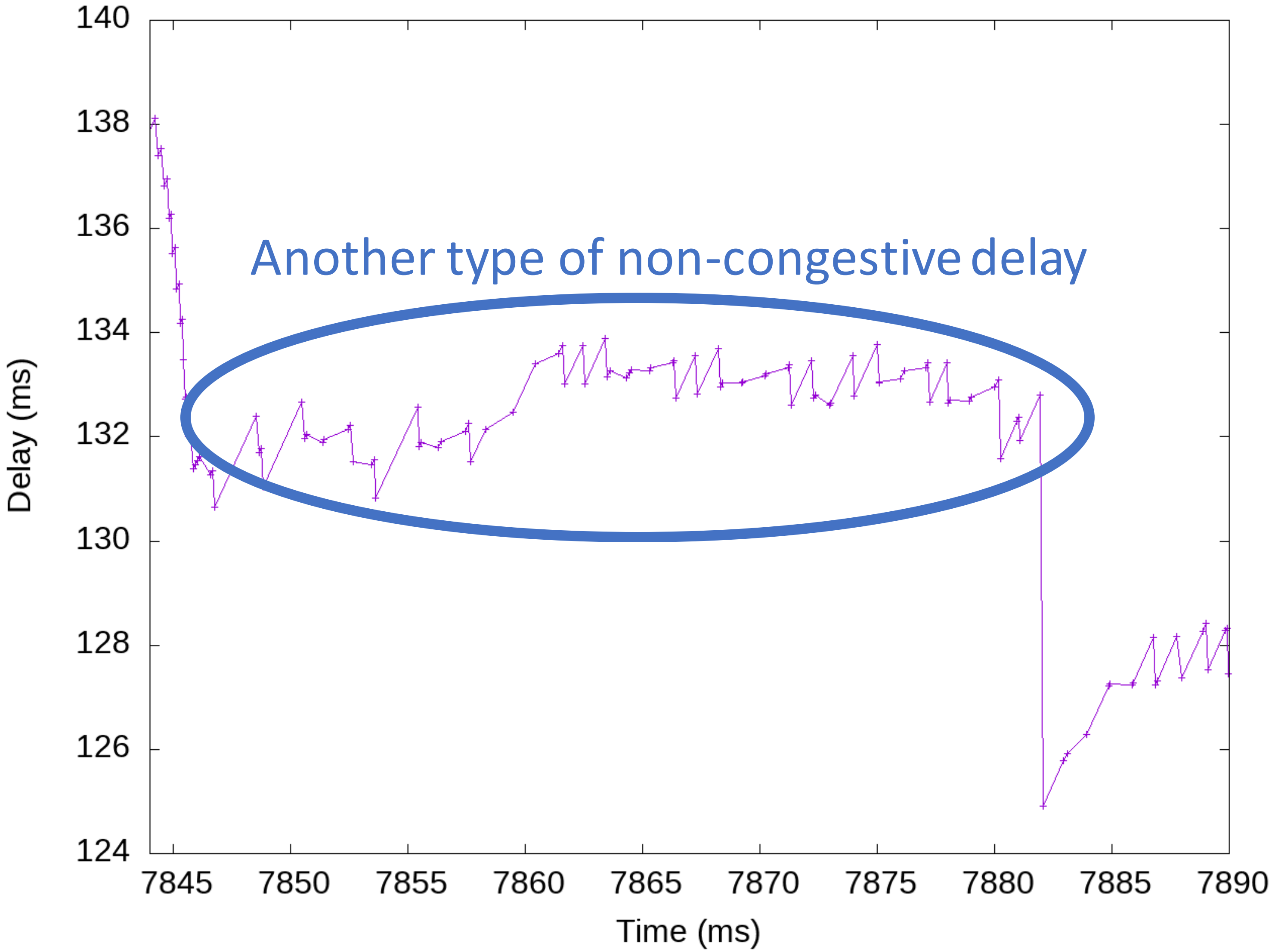
How large is this delay (cellular)?

Pantheon: the training ground for Internet congestion-control research, USENIX ATC'18, Francis Yan et al.



How large is this delay (cellular)?

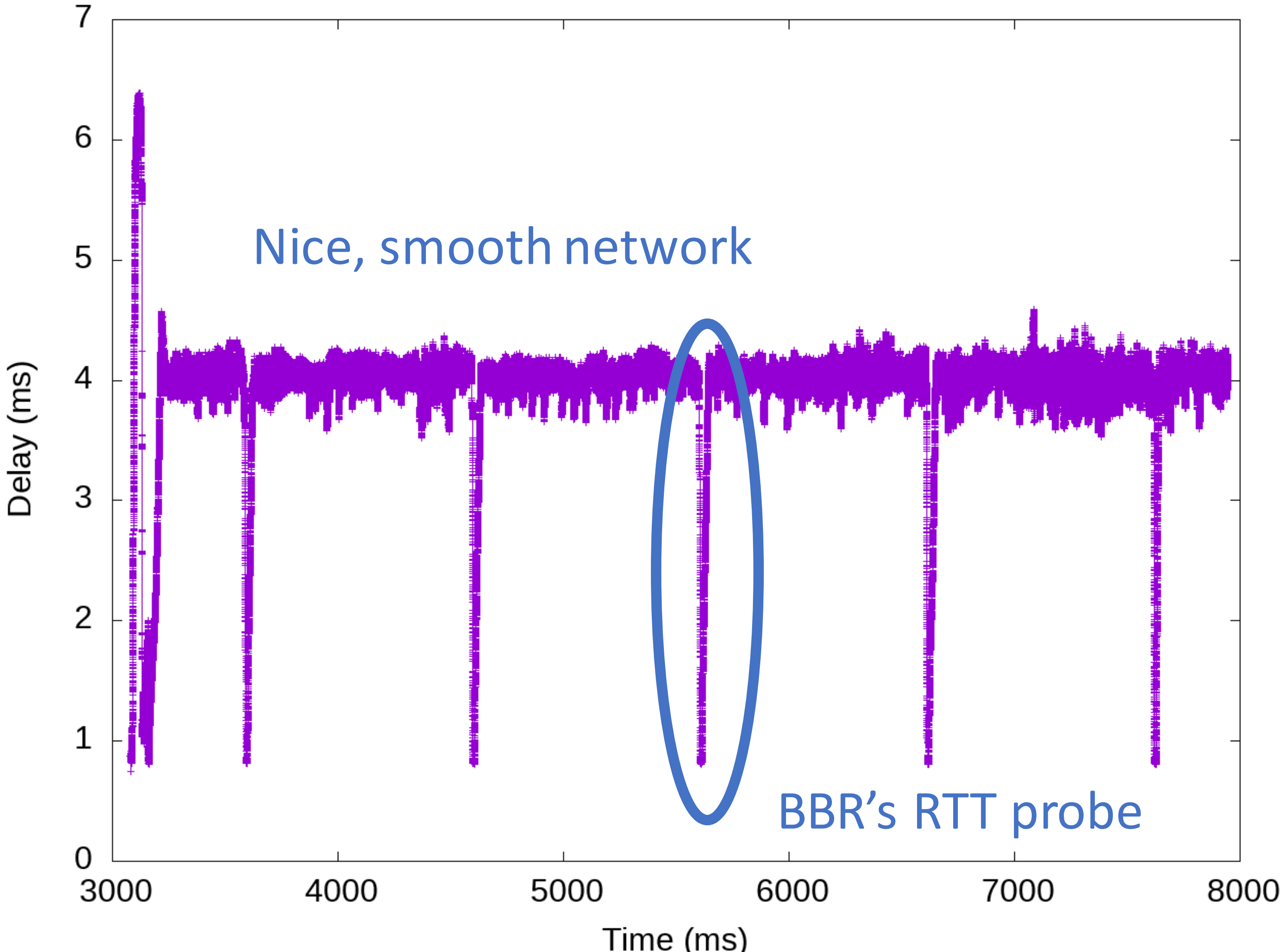
Pantheon: the training ground for Internet congestion-control research, USENIX ATC'18, F. Yan et al.



Another type of non-congestive delay

How large is this delay (wired)?

Pantheon: the training ground for Internet congestion-control research, USENIX ATC’18, Francis Yan et al.

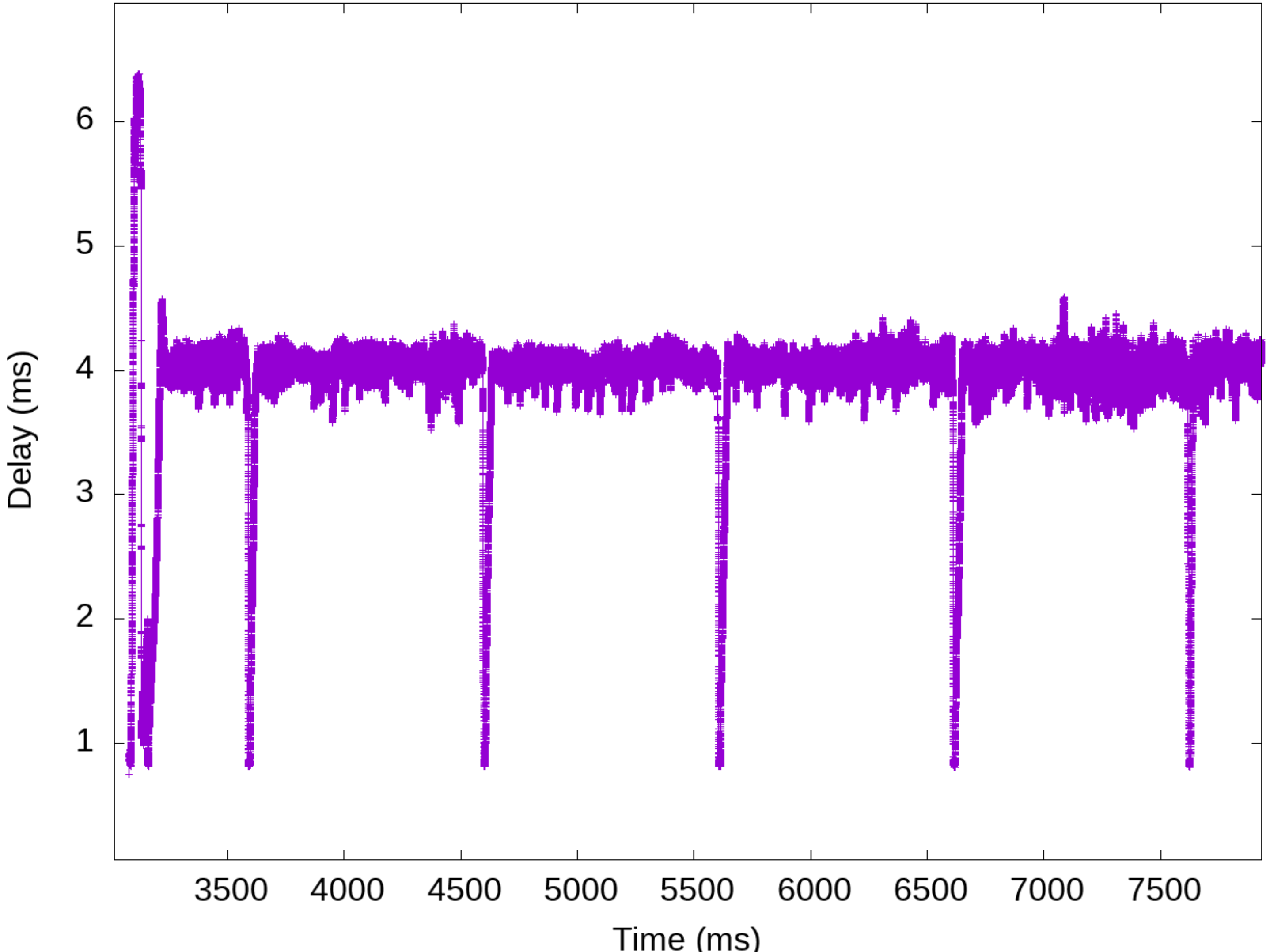


Nice, smooth network

BBR’s RTT probe

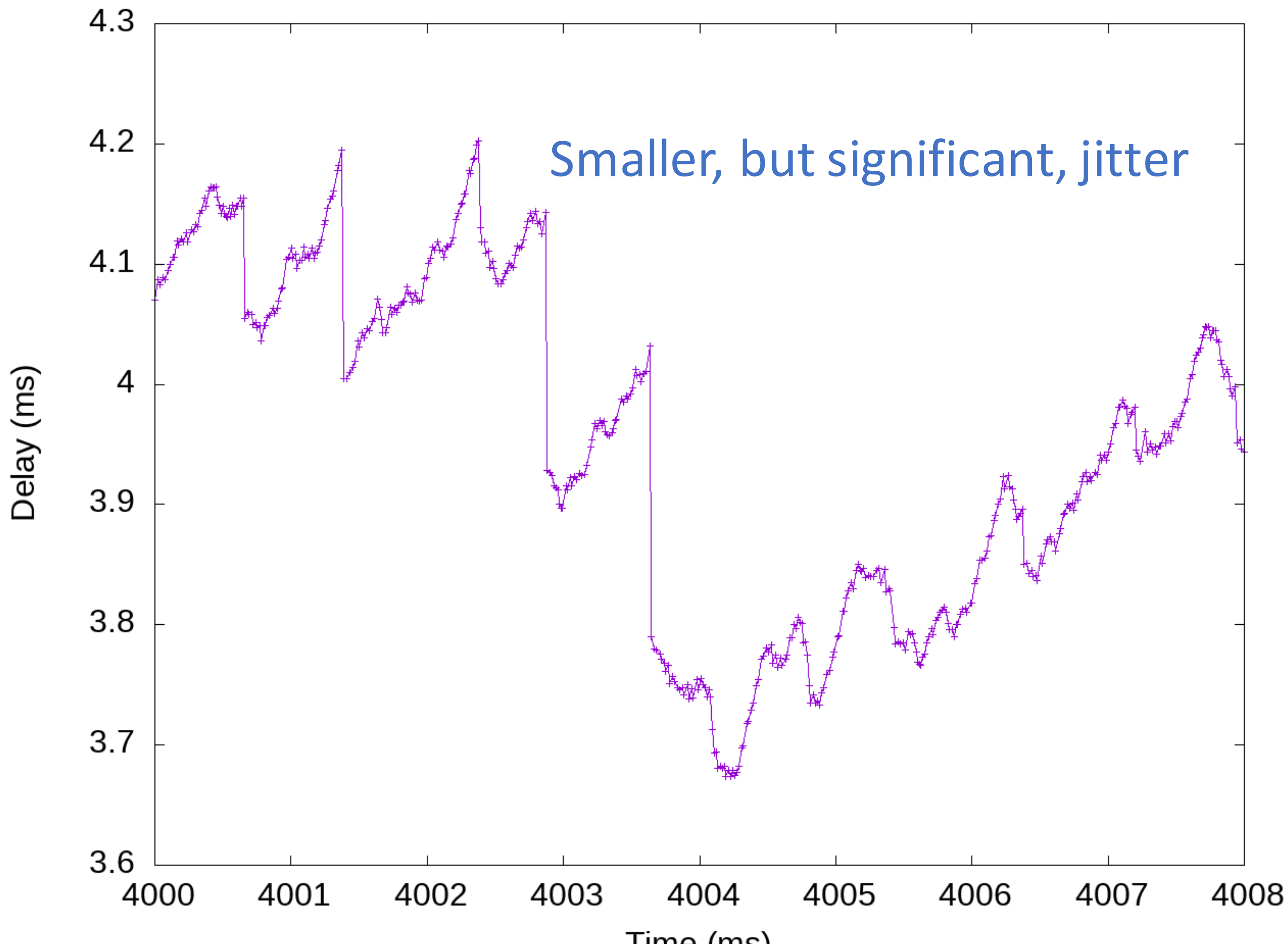
How large is this delay (wired)?

Pantheon: the training ground for Internet congestion-control research, USENIX ATC’18, Francis Yan et al.



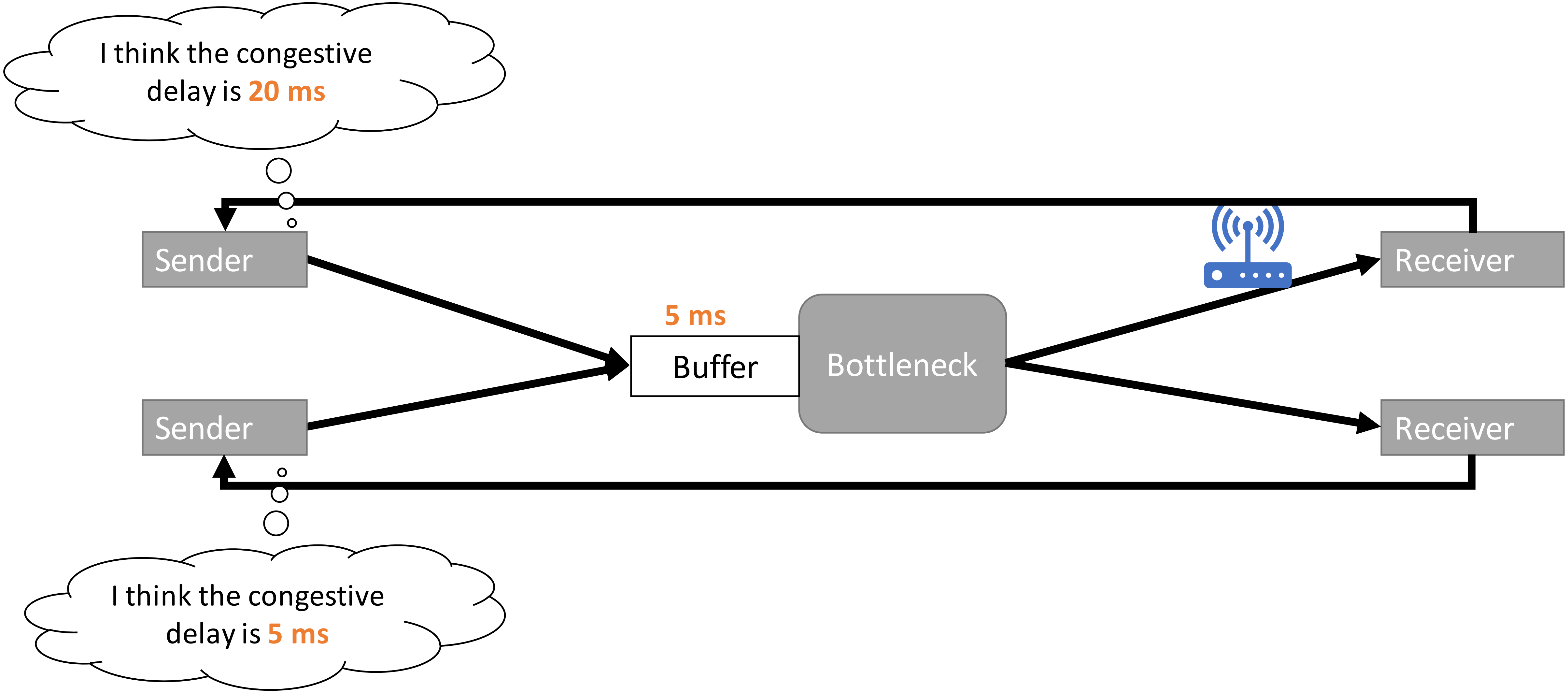
How large is this delay (wired)?

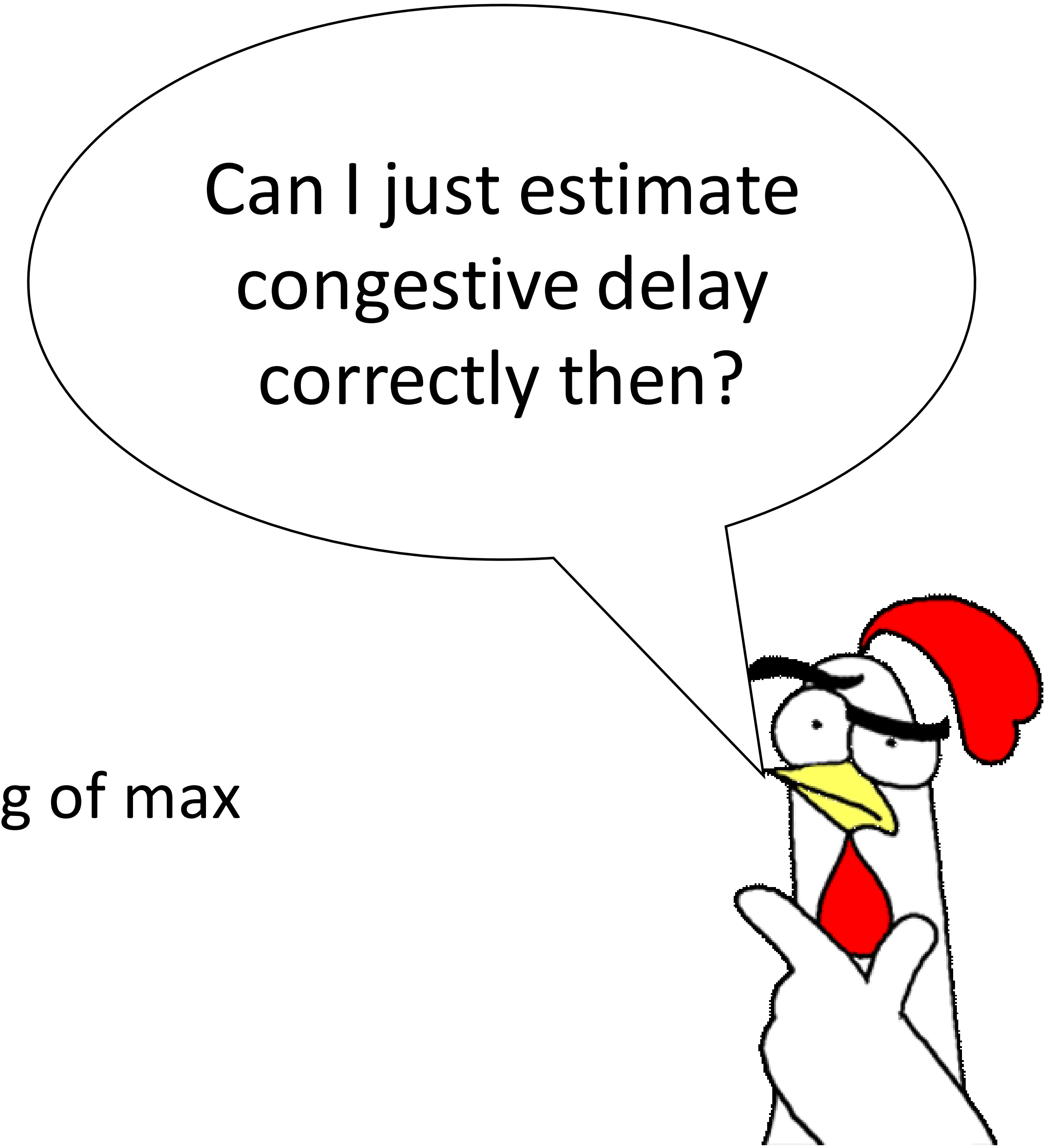
Pantheon: the training ground for Internet congestion-control research, USENIX ATC’18, Francis Yan et al.



Smaller, but significant, jitter

Non-congestive delays confuse congestion estimation


I think the congestive delay is 20 ms
Sender
Sender
5 ms
Buffer
Bottleneck
Receiver
Receiver
I think the congestive delay is 5 ms



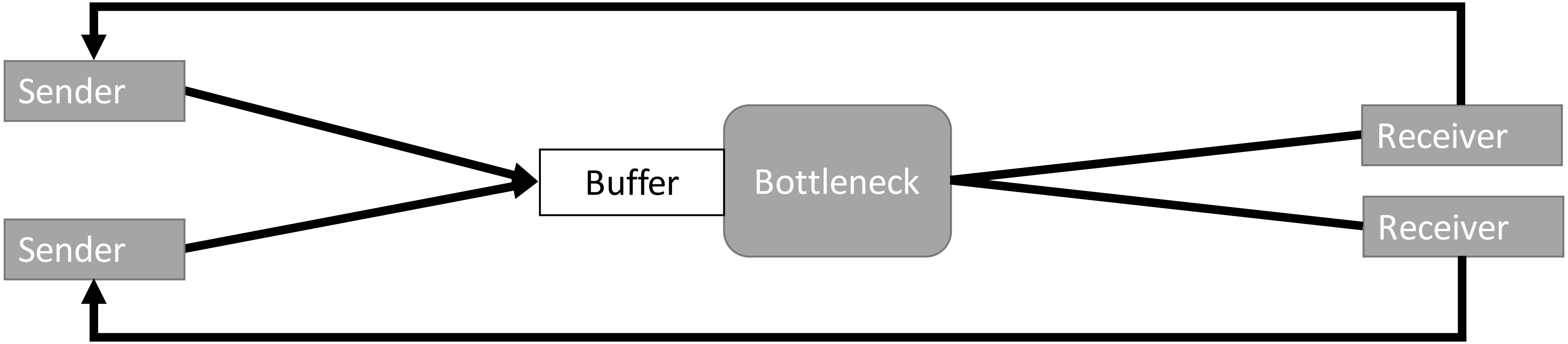
Every estimator we are aware of has failure modes:

Delay

Instantaneous, average, median, min, avg of max

Rate

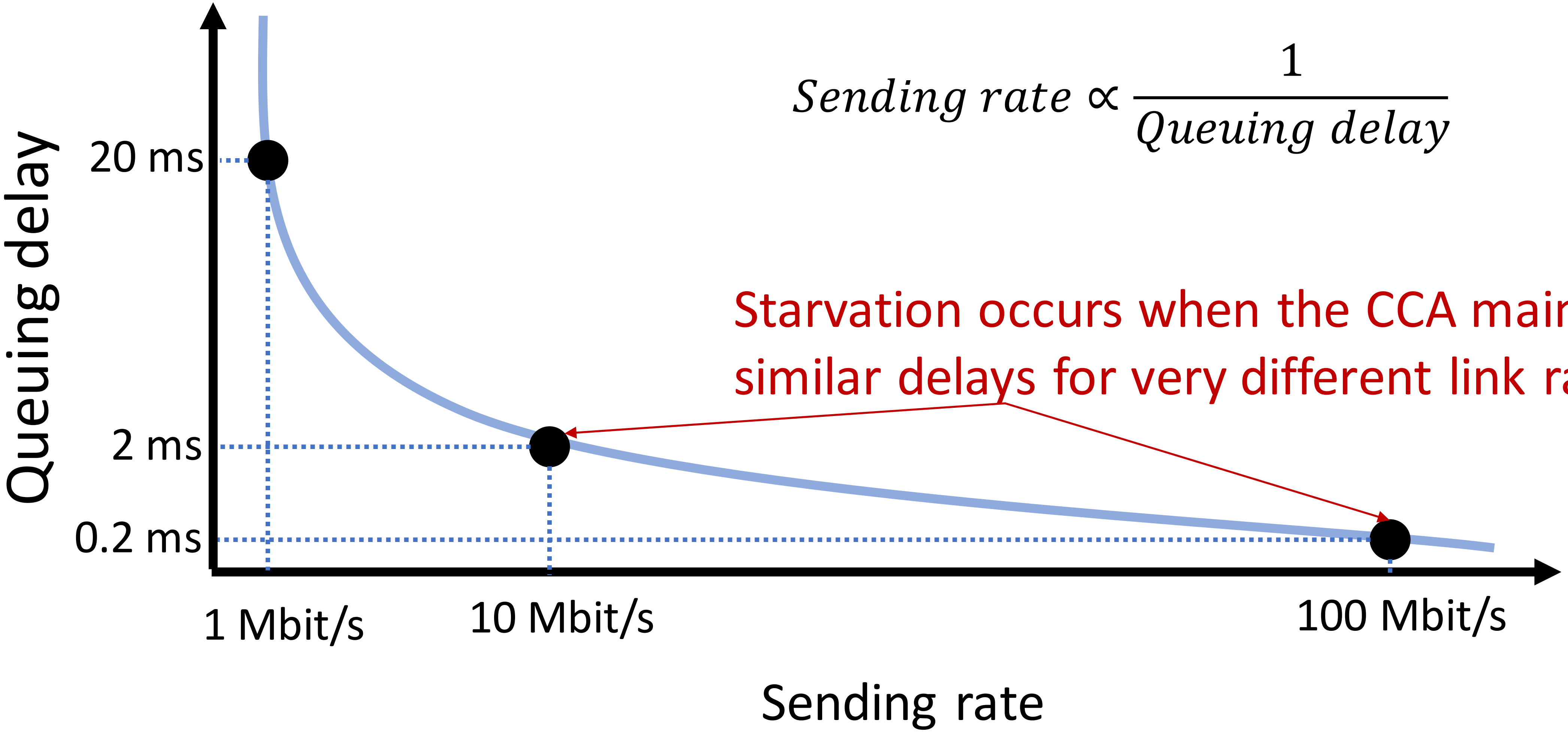
Average, max of average



Starvation (definition):

1. The ratio of throughputs they get is arbitrarily large
2. It remains that way forever

Starvation in Vegas/FAST/Copa

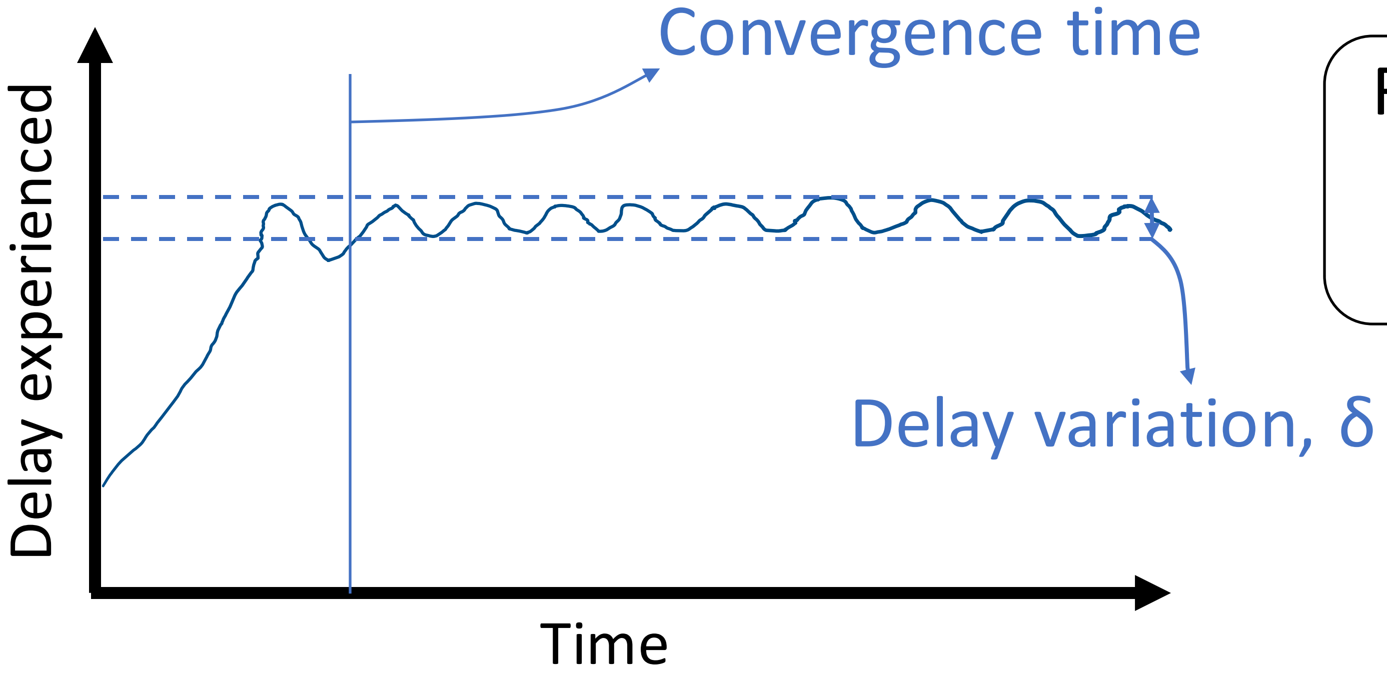


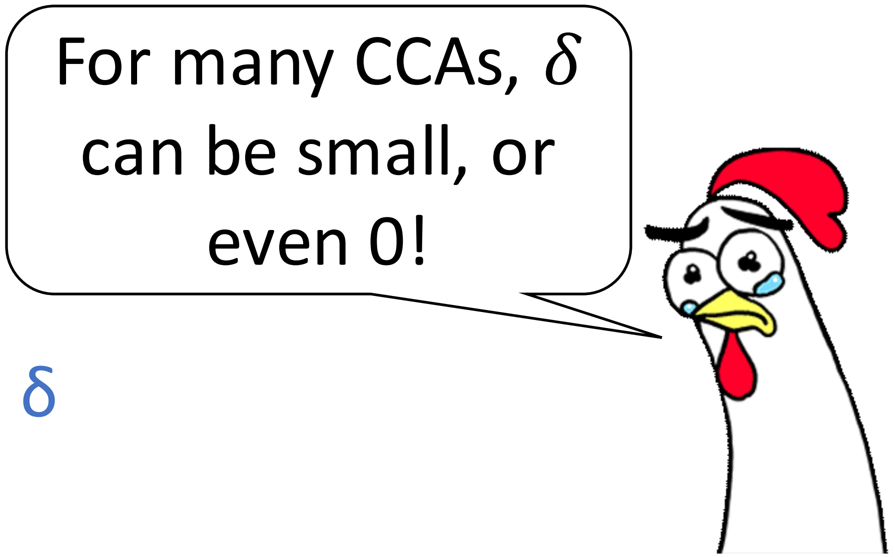
$$Sending\ rate \propto \frac{1}{Queuing\ delay}$$

Starvation occurs when the CCA maintains similar delays for very different link rates

Ok, the Vegas family maps different link rates to similar delays. What about the rest?

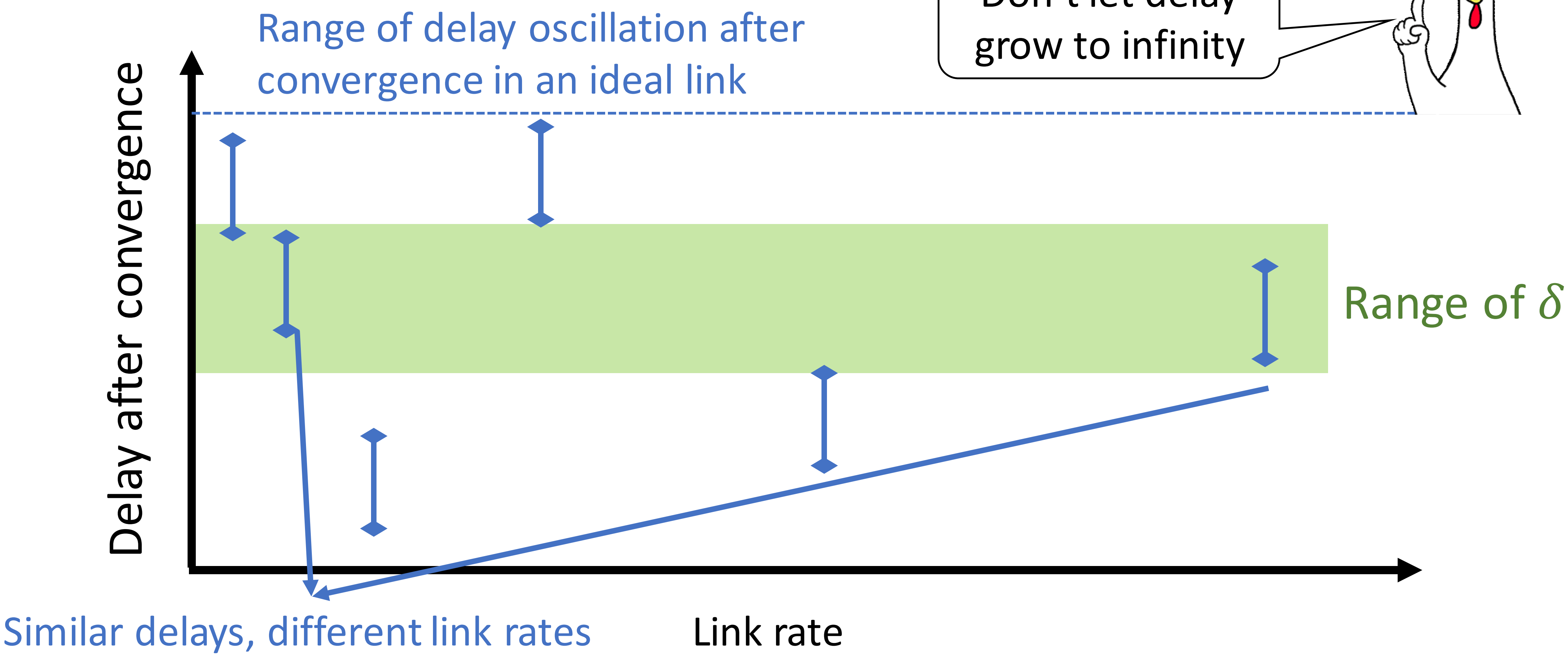
Key Result: All delay-convergent CCAs starve

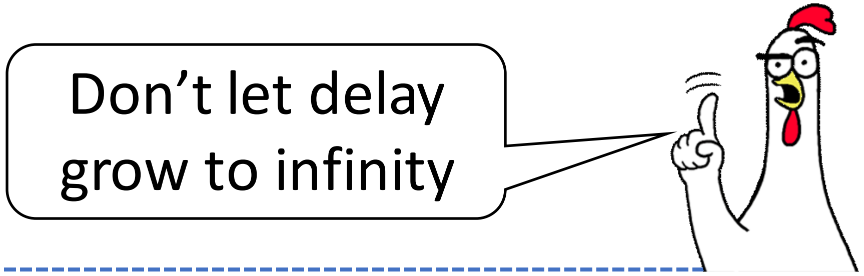




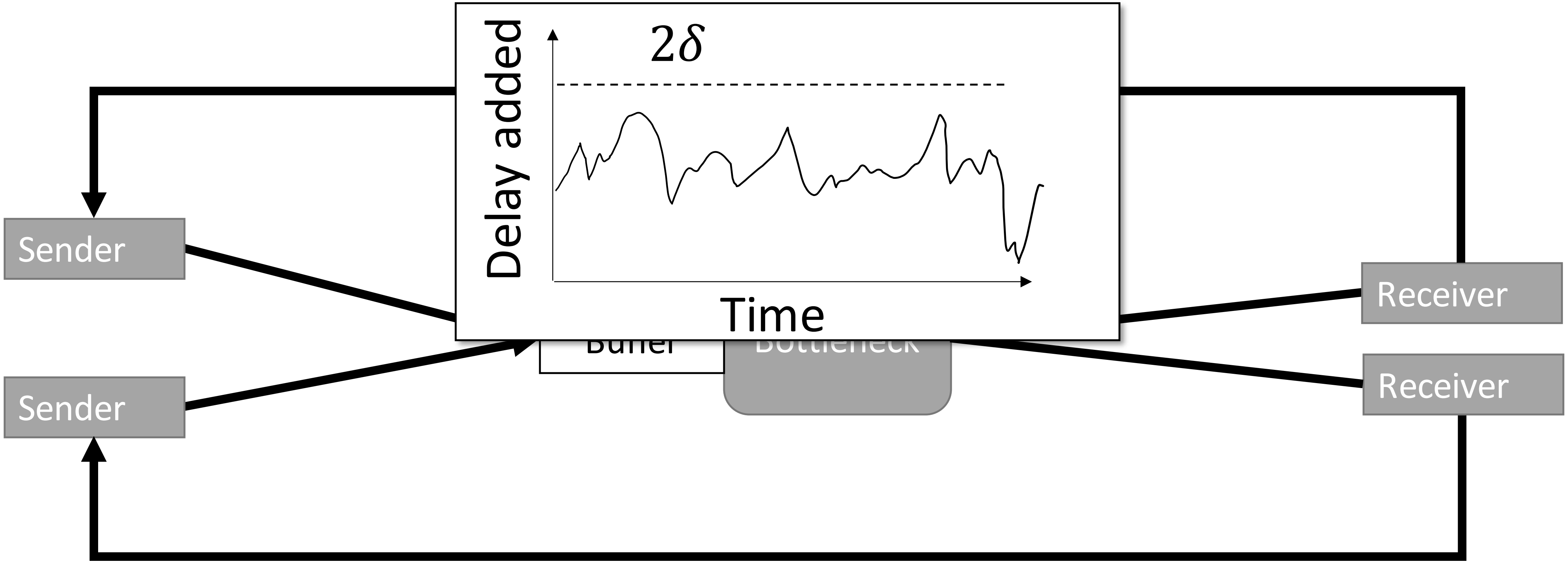
Theorem: We can always construct non-congestive delay smaller than D such that starvation occurs (for any $\mathrm{D} > 2\delta$)

Claim: Delay-convergent CCAs have similar delays for different link rates

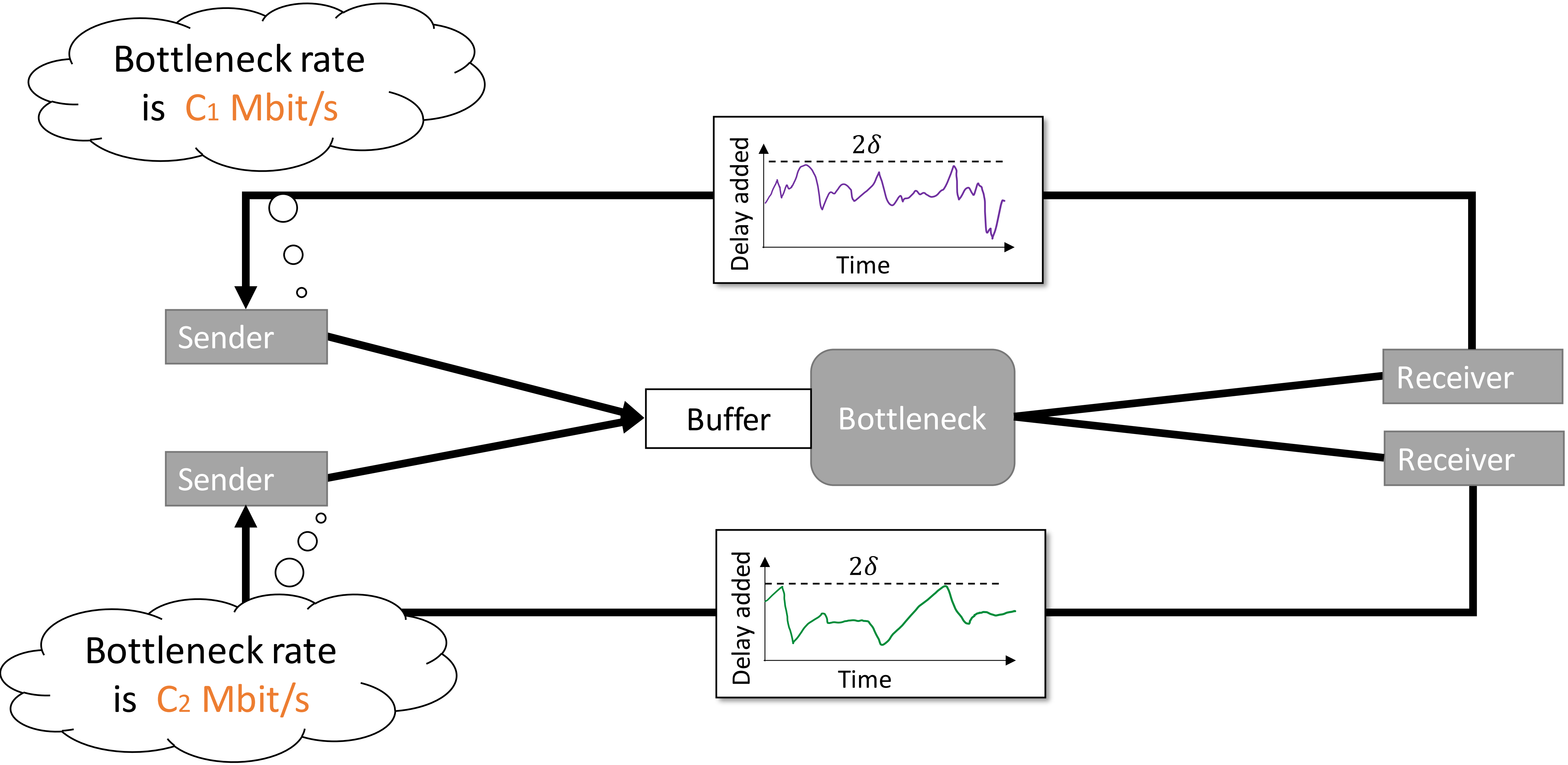




Proof: Constructing the non-congestive delay



Proof: Constructing the non-congestive delay



Bottleneck rate is C_1 Mbit/s

Sender

Sender

Buffer

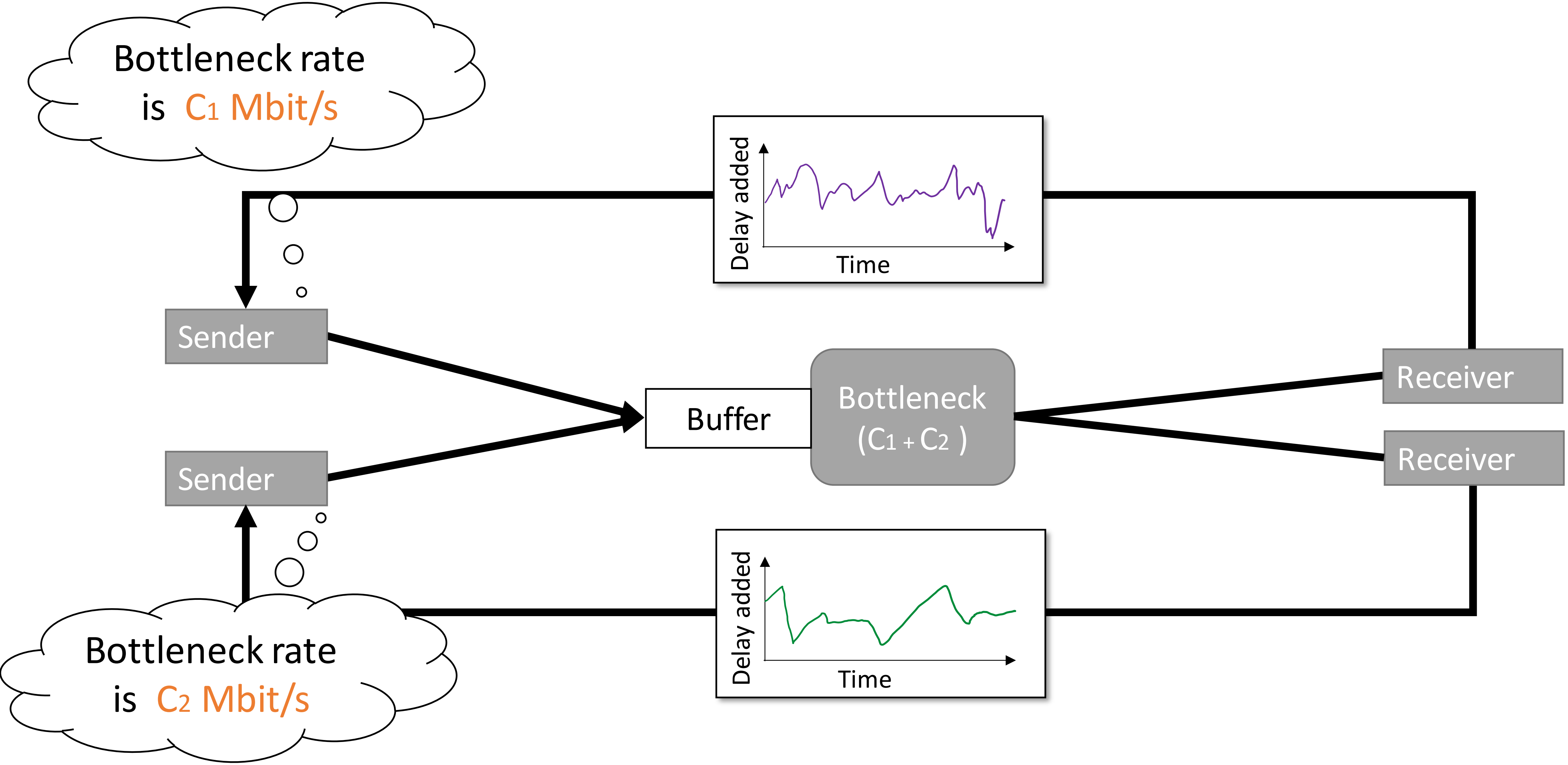
Bottleneck

Receiver

Receiver

Bottleneck rate is C_2 Mbit/s

Proof: Constructing the non-congestive delay



Bottleneck rate is C_1 Mbit/s

Sender

Sender

Buffer

Bottleneck ($C_1 + C_2$)

Receiver

Receiver

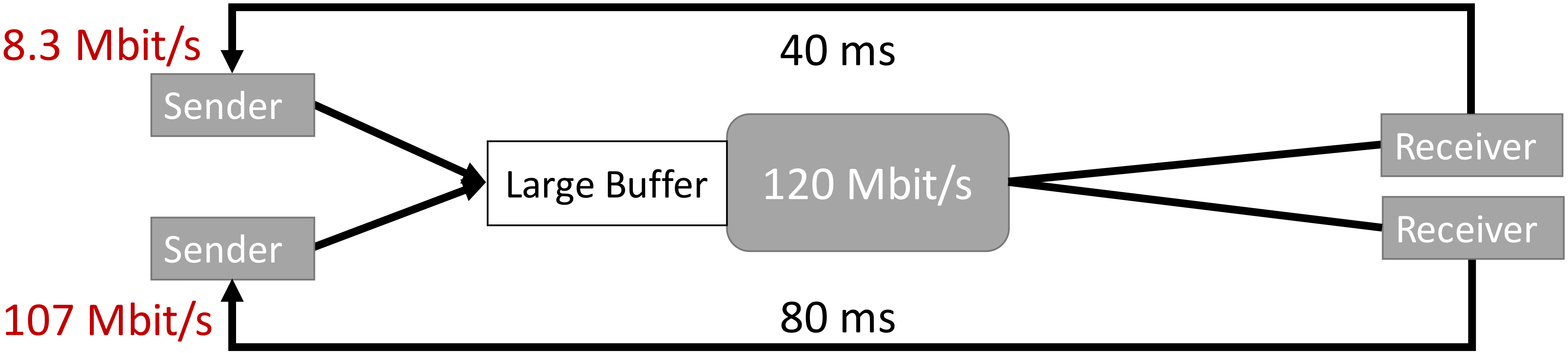
Bottleneck rate is C_2 Mbit/s

Is this construction realistic?

Starvation in BBR

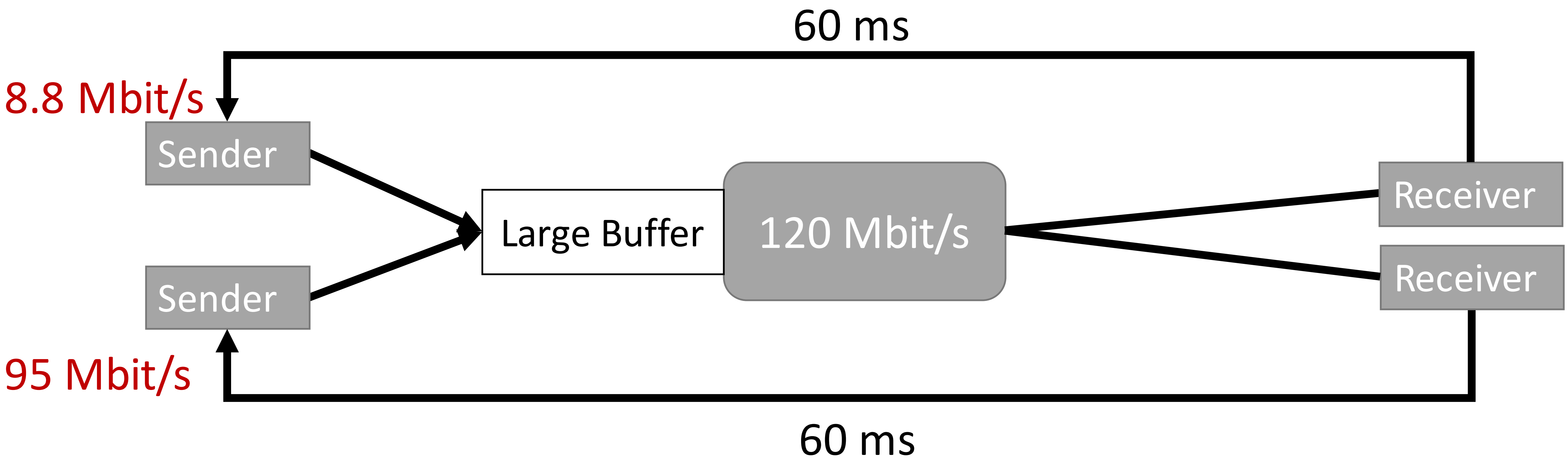
If the network has some jitter, BBR will maintain queuing delay equal to propagation delay

If propagation delay for two flows are different, the flow with the *smaller* propagation delay starves!

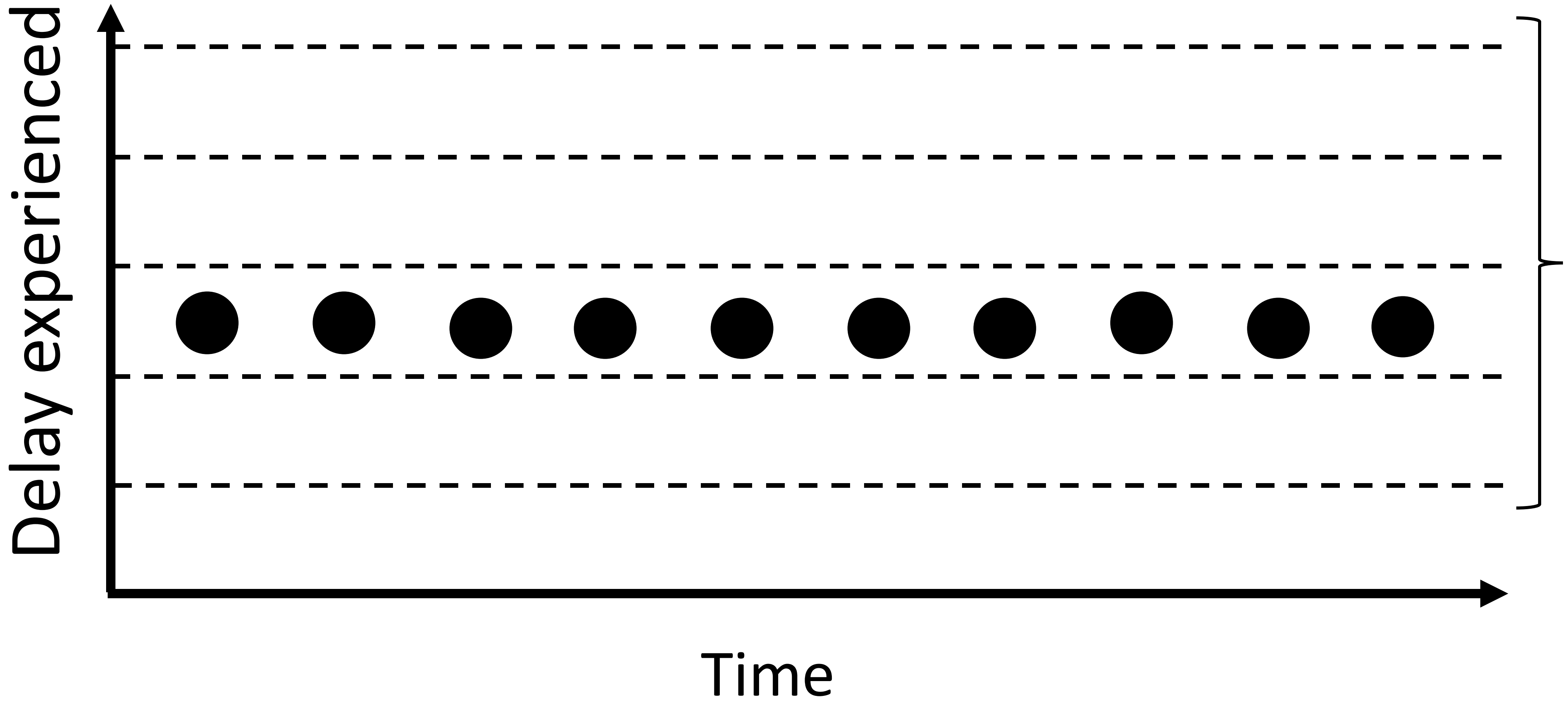


Starvation in Vegas/FAST/Copa

One packet gets acked in 59 ms

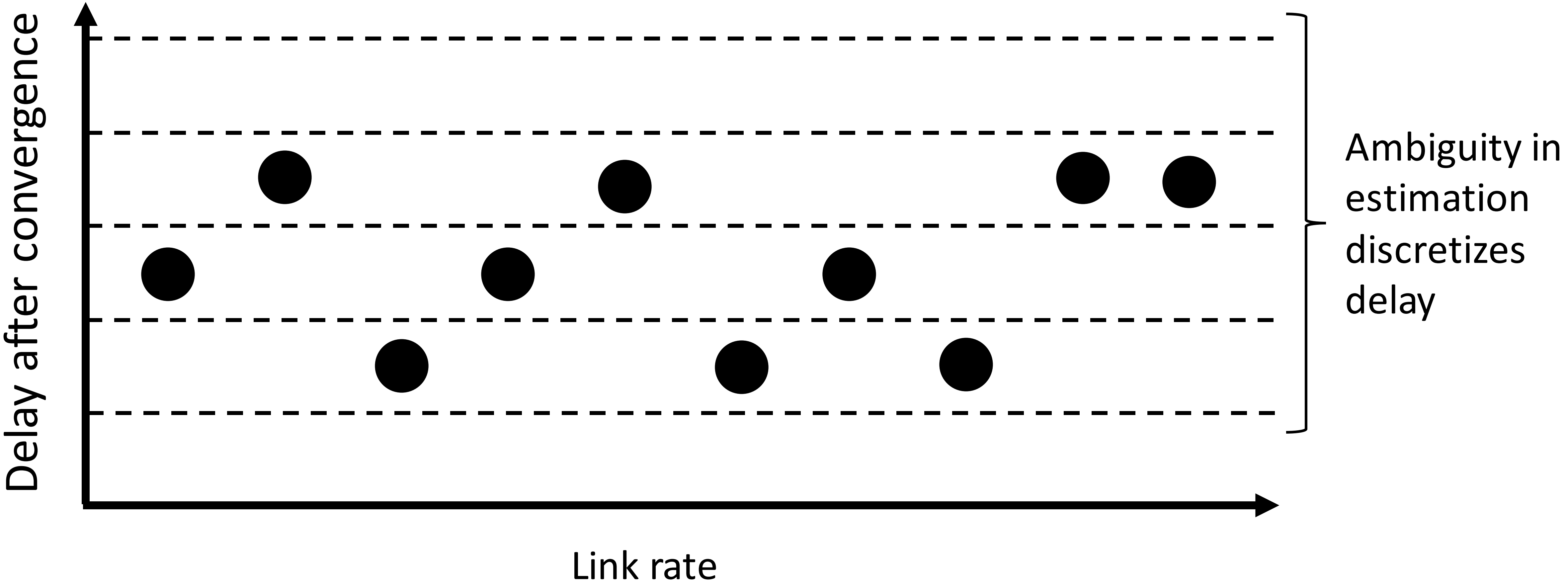


Could deliberately oscillating delay help?



Ambiguity in estimation discretizes delay

Why would deliberately oscillating delay help?



What next?

- Deliberately oscillate the delay
- Design for a finite link range [see paper for how]
- Use ECN, fair queuing, …