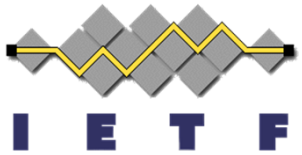
On Queuing, Marking, and Dropping

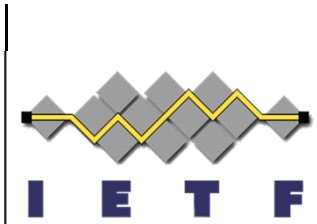
draft-baker-aqm-sfq-implementation

Fred Baker

AQM at IETF 90

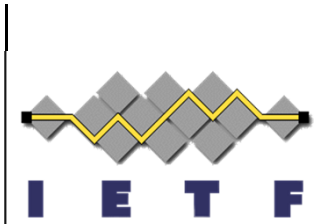


What am I trying to achieve in this draft?



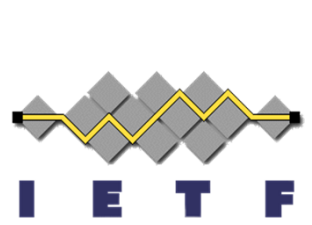
- I am making a simple observation:
  - Queuing algorithms and mark/drop algorithms differ in objective and effect, and should not be confused
- This is not to say that one or the other is bad
  - I personally greatly favor WFQ/WRR as a policy enforcement mechanism
  - I personally greatly favor AQM, and especially ECN, and delay/jitter-based TCP Congestion Control algorithms as latency control

RFC 2309 on scheduling

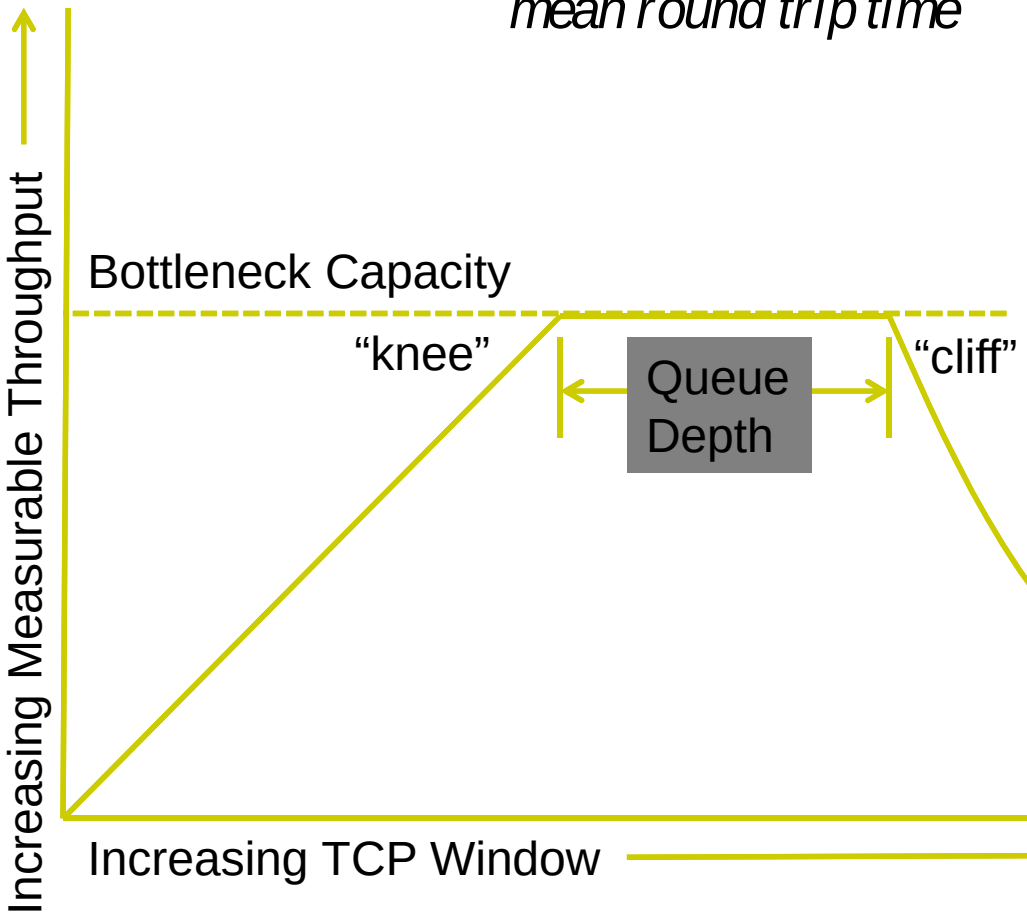


“It is useful to distinguish between two classes of router algorithms related to congestion control: "queue management" versus "scheduling" algorithms. To a rough approximation, queue management algorithms manage the length of packet queues by dropping packets when necessary or appropriate, while scheduling algorithms determine which packet to send next and are used primarily to manage the allocation of bandwidth among flows. While these two router mechanisms are closely related, they address rather different performance issues.”

Simple model of TCP throughput dynamics: What is AQM trying to do? Minimize Latency



$$mean\ throughput = \frac{effective\ window\ in\ bytes}{mean\ round\ trip\ time}$$

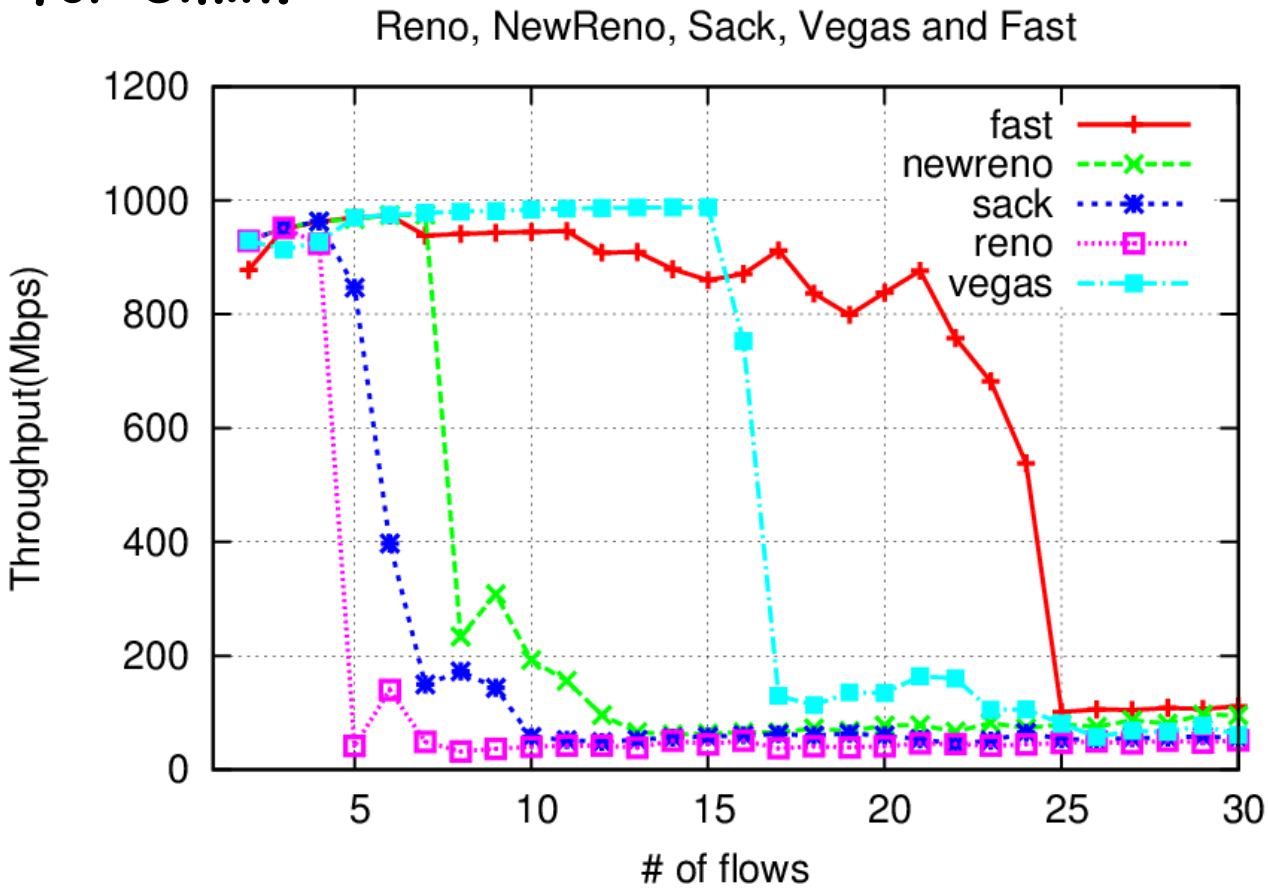


- **Effective Window:** the amount of data TCP sends each RTT
- **Knee:** the lowest window that makes throughput approximate capacity
- **Cliff:** the largest window that makes throughput approximate capacity
- Note that throughput is the same at knee and cliff. *Increasing the window merely increases RTT, by increasing queue depth*

Yes, there is a more complex equation that takes into account loss. It estimates throughput above the cliff.

TCP Performance on short RTT timeframes

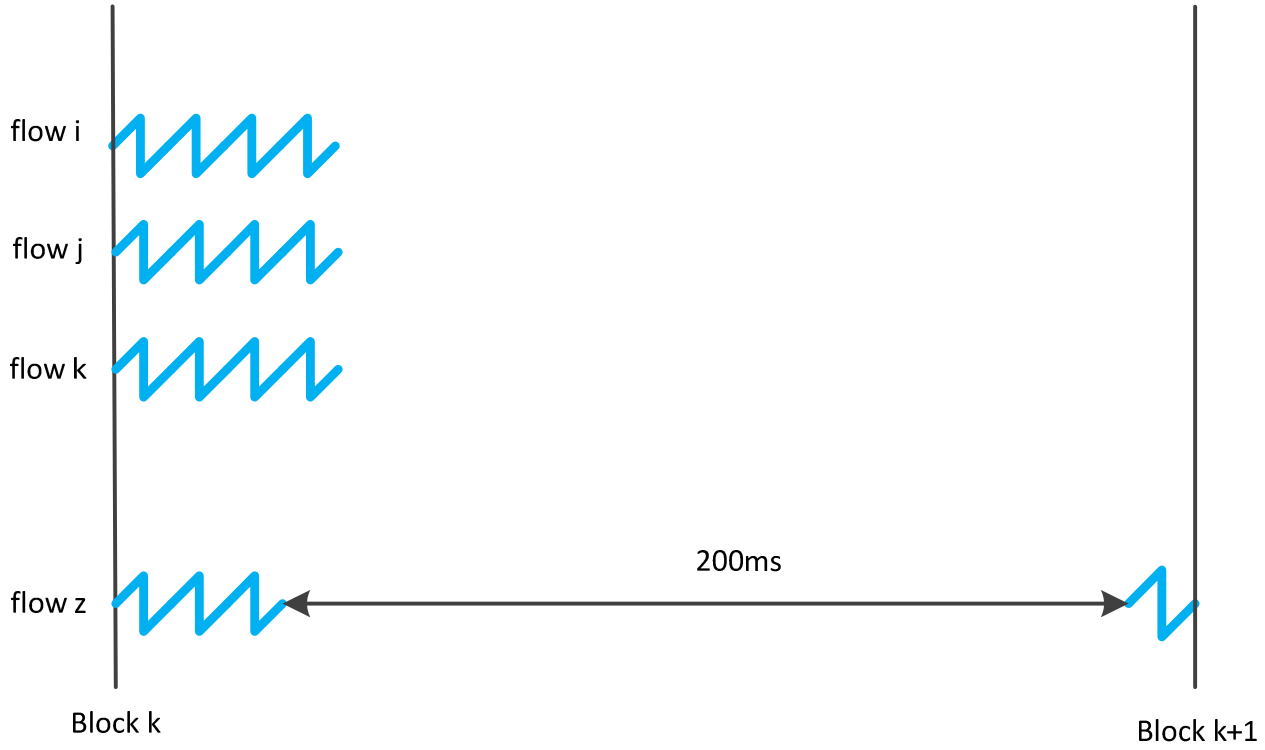
- Each flow responses 100KB data
  - Last for 5min.



Courtesy Tsinghua University
Cisco/Tsinghua Joint Lab

Effects of TCP Timeout

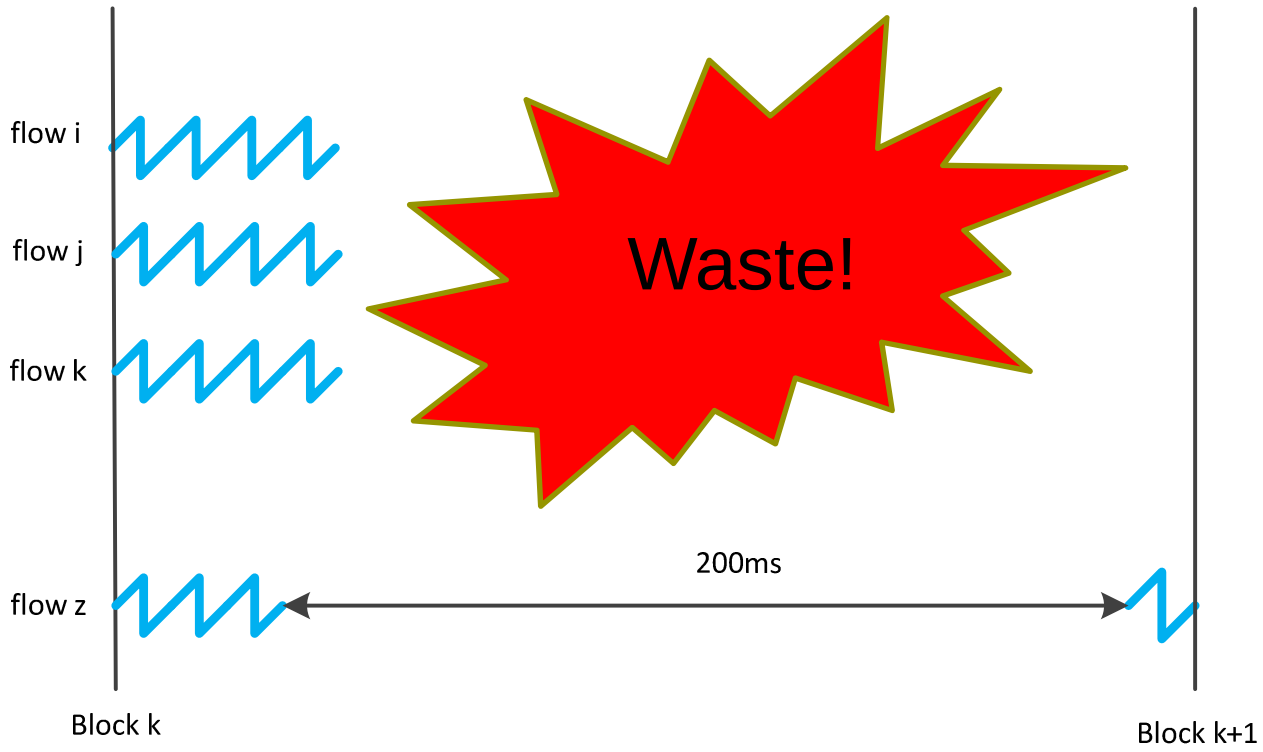
- The ultimate reason for throughput collapse in Incast is timeout.



Courtesy Tsinghua University
Cisco/Tsinghua Joint Lab

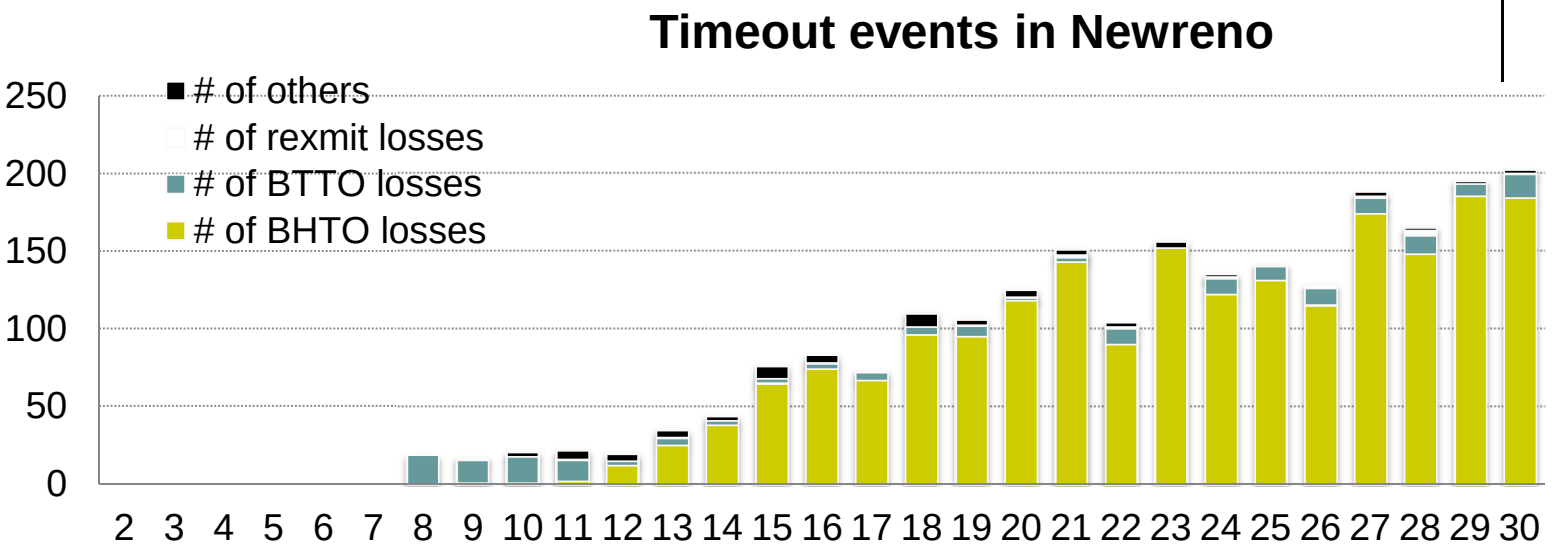
Effects of TCP Timeout

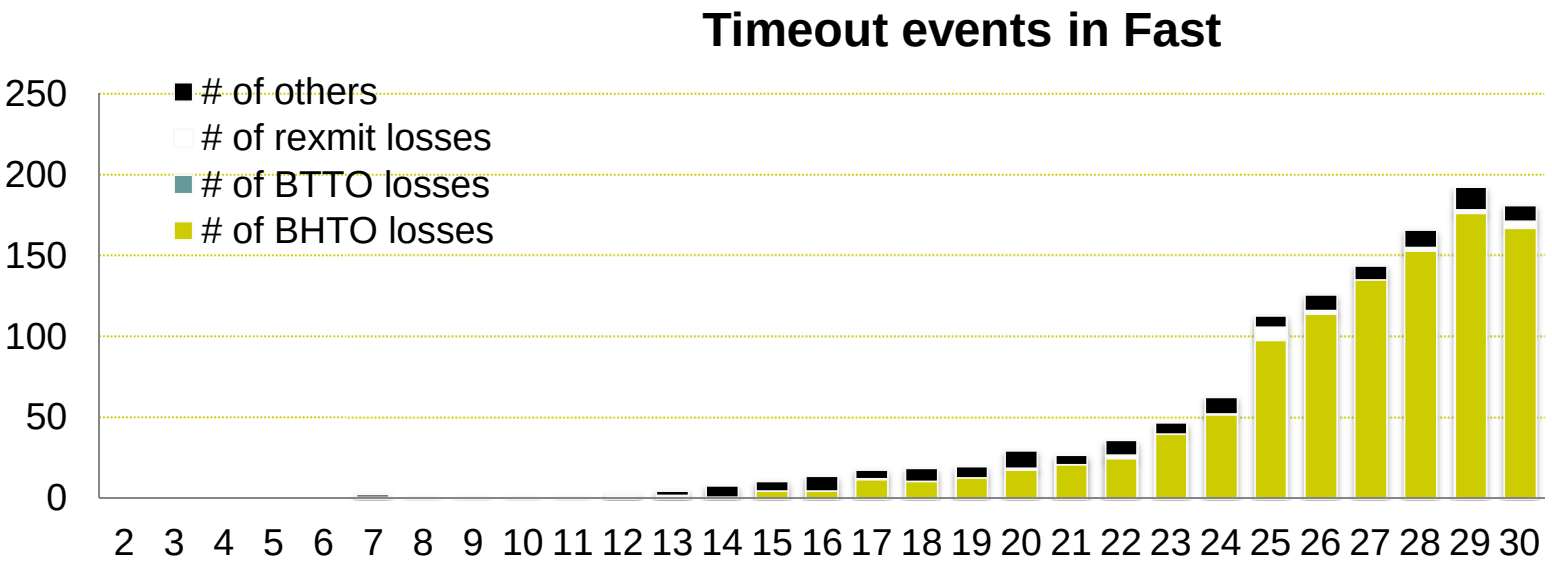
- The ultimate reason for throughput collapse in Incast is timeout.



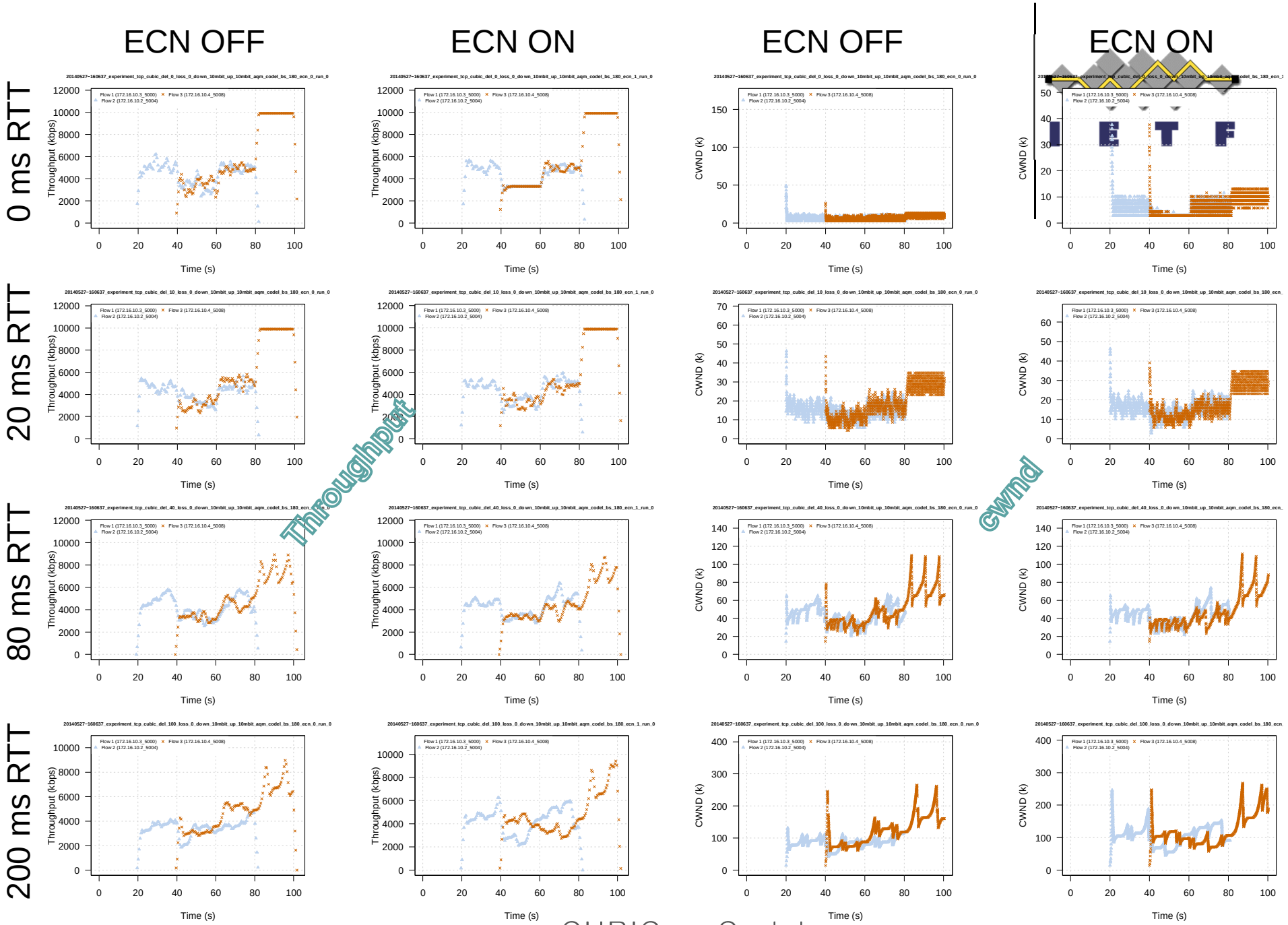
Courtesy Tsinghua University
Cisco/Tsinghua Joint Lab

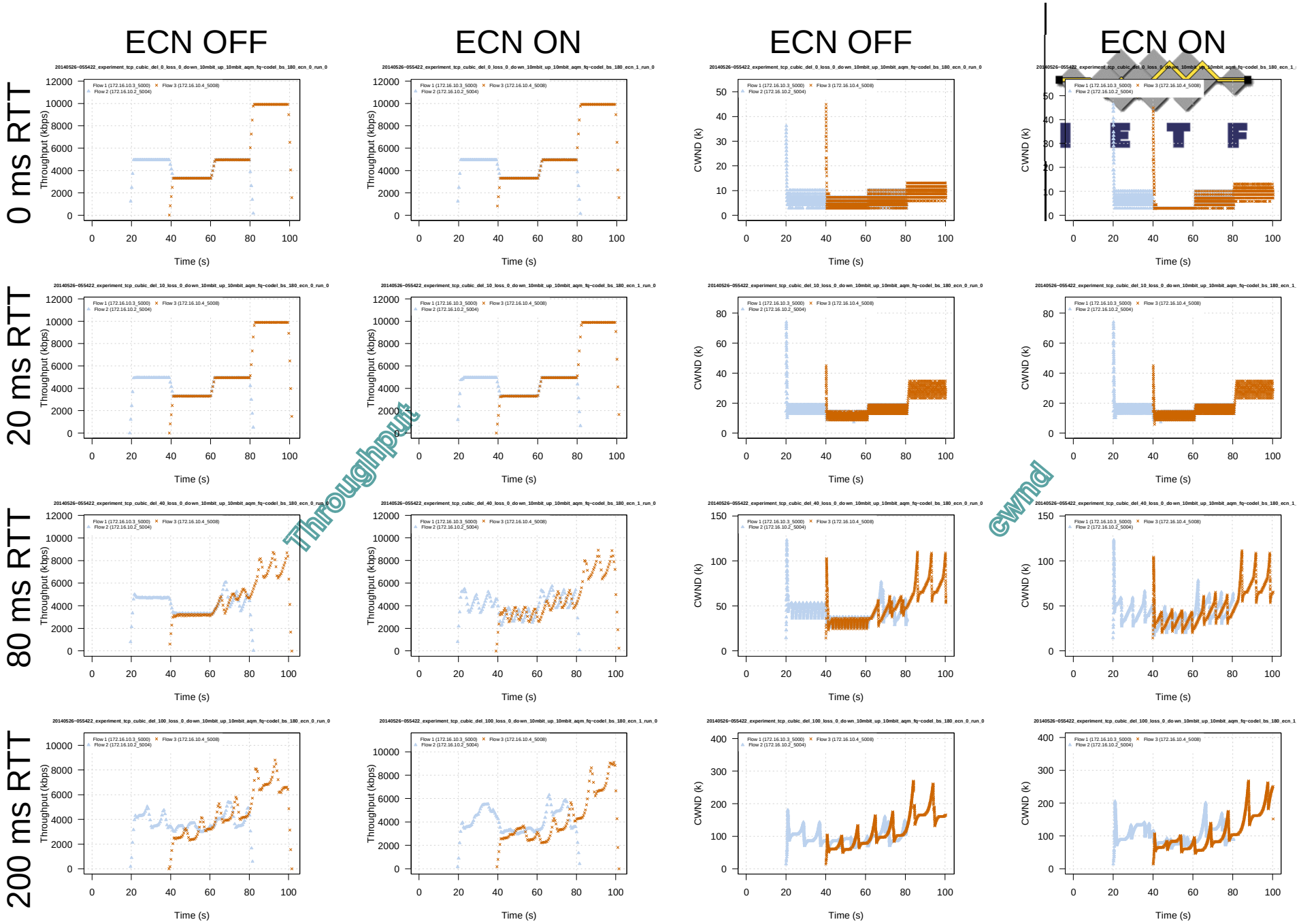
Prevalence of TCP Timeout





Courtesy Tsinghua University
Cisco/Tsinghua Joint Lab

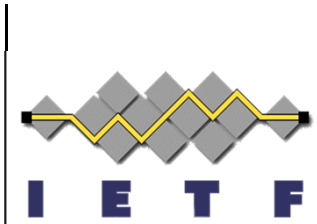

ECN OFF
ECN ON
ECN OFF
ECN ON
0 ms RTT
20 ms RTT
80 ms RTT
200 ms RTT
Throughput
cwnd
Throughput (kbps)
CWND (k)
Time (s)
CUBIC vs Codel
Courtesy Swinburne CAIA



CUBIC vs fq_codel

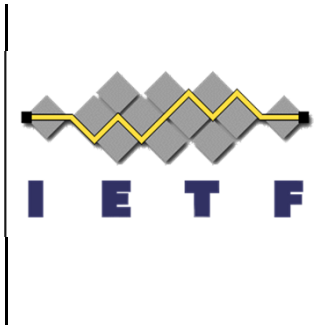
Courtesy Swinburne CAIA

Implementation discussion

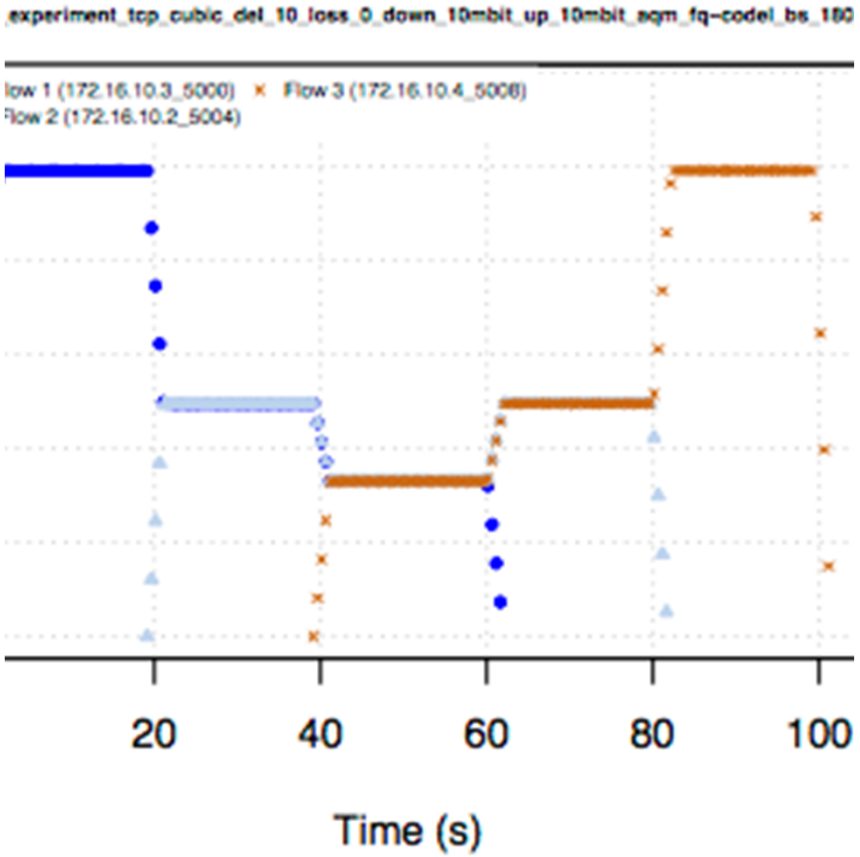


- In the draft, I spend quite a bit of time on WRR and WFQ (described in Zhang ‘90 and McKenney ‘91)
  - How they are commonly implemented
  - Trade-offs between them
  - I don’t note, but it was pointed out to me, that fq is a WRR variant that minimizes search time

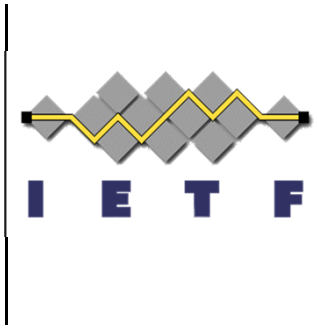
The sharp edges in the graph result from queuing, not AQM



- Having written and tested WFQ with tail drop…
  - It delivers essentially the same results as fq_codel
- Given a fair queue algorithm (WRR/WFQ)
  - How you mark or drop is almost irrelevant
  - The latency incurred is due to number of active queues, not queue depth



So – my point



- I am making a simple observation:
  - Queuing algorithms and mark/drop algorithms differ in objective and effect, and should not be confused

On Queuing, Marking, and Dropping

draft-baker-aqm-sfq-implementation

Fred Baker

AQM at IETF 90