The Origin, Evolution and Current Status of QCN

Berk Atikoglu, Abdul Kabbani, Rong Pan,
Balaji Prabhakar, Mick Seaman

Overview

- A quick summary of work-to-date on QCN
- This presentation is brief
  - More next week
  - Rong will present the remaining benchmark tests at Los Gatos

QCN: Evolution Summary

- Develop a simple, universally stable congestion management scheme
  - ***Universally stable*** means there are no tunable parameters; all parameters fixed regardless of number of sources (N) or round trip time (RTT)

- We began with BCN
  - First, just quantized it and removed the RLT
  - Later, rediscovered BIC and hence improved the self-increase feature
  - This is pretty much what we know as 2-pt QCN

- Response time
  - Since this is important, tried various things
    - 3-pt QCN, Fb-hat, SONAR, Fb99
  - 3-pt QCN impeded by multipath; others either had poor response time (Fb-hat) or were hard to make universally stable

- Finally: used a timer at the source in conjunction with the byte-counter, and put HAI in series with AI to get universal stability + good response time

Current status

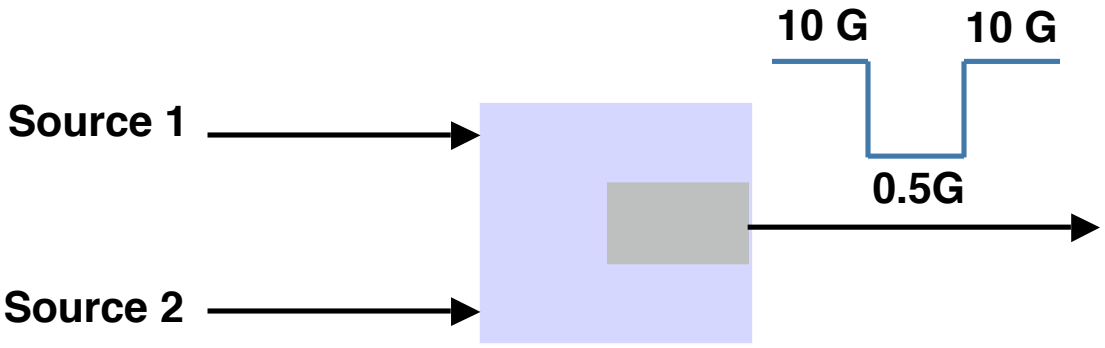
- Have an algorithm which is
  - Universally stable
  - Works in a shared RL (multipath) environment
  - Has a good response time
  - Benchmarks complete; will be presented at Los Gatos

Universal stability

- Worth understanding universal stability some more…
- AIMD schemes like TCP don't possess it; feedback compensation needed
  - Choice of parameters which stabilize scheme for long RTT make it sluggish
  - As we shall see, this is also true for BCN (which is AIMD)
- However, BIC and QCN have the property of universal stability

Simulations

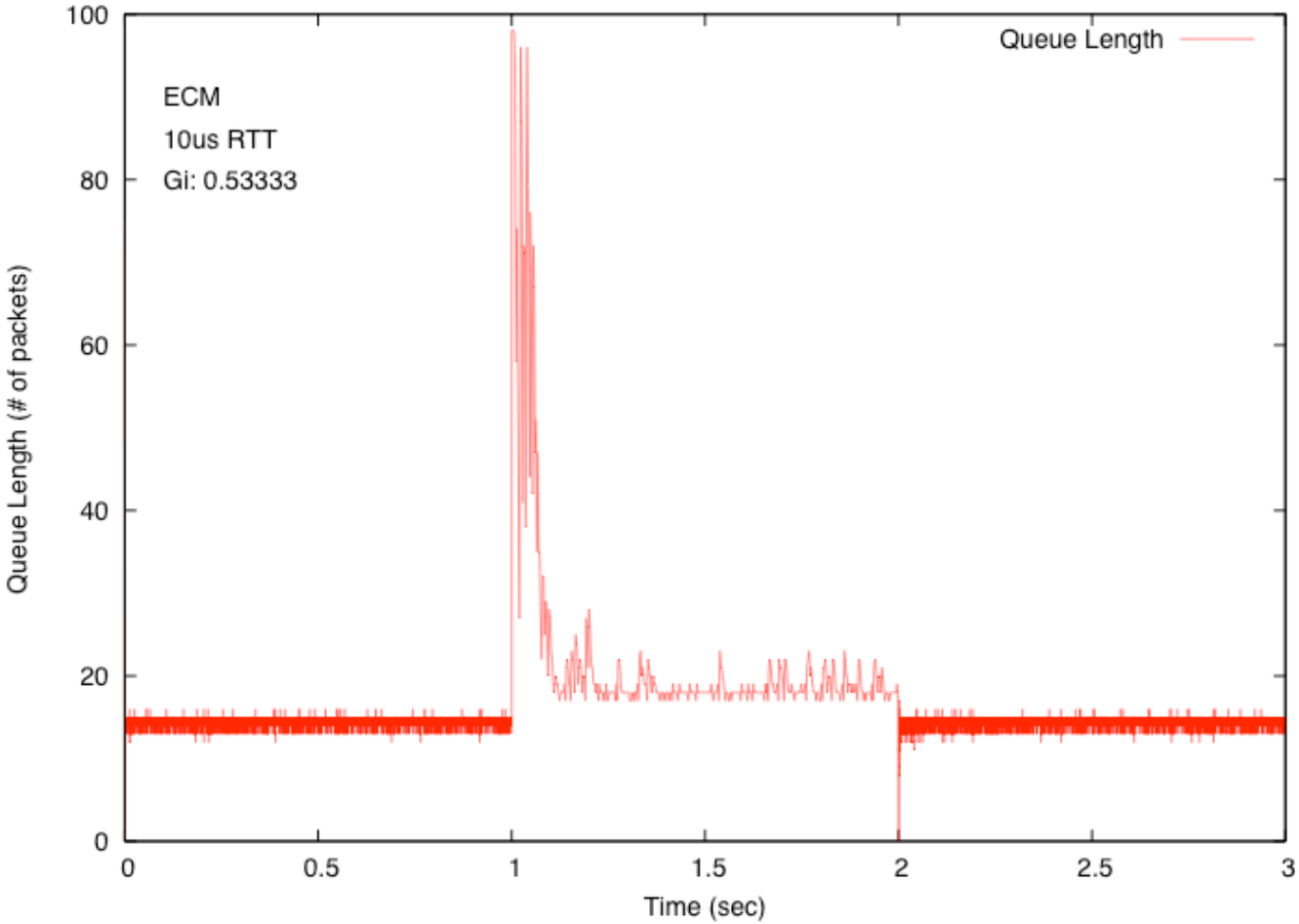
- Consider the Baseline scenario
  - Single link, 2 sources
  - OG hotspot; hotspot severity: 0.5G; hotspot duration 1s
  - Vary RTT: 10 us, 200 us
  - Study: behavior of QCN and BCN: stability and response time



Simulation Parameters

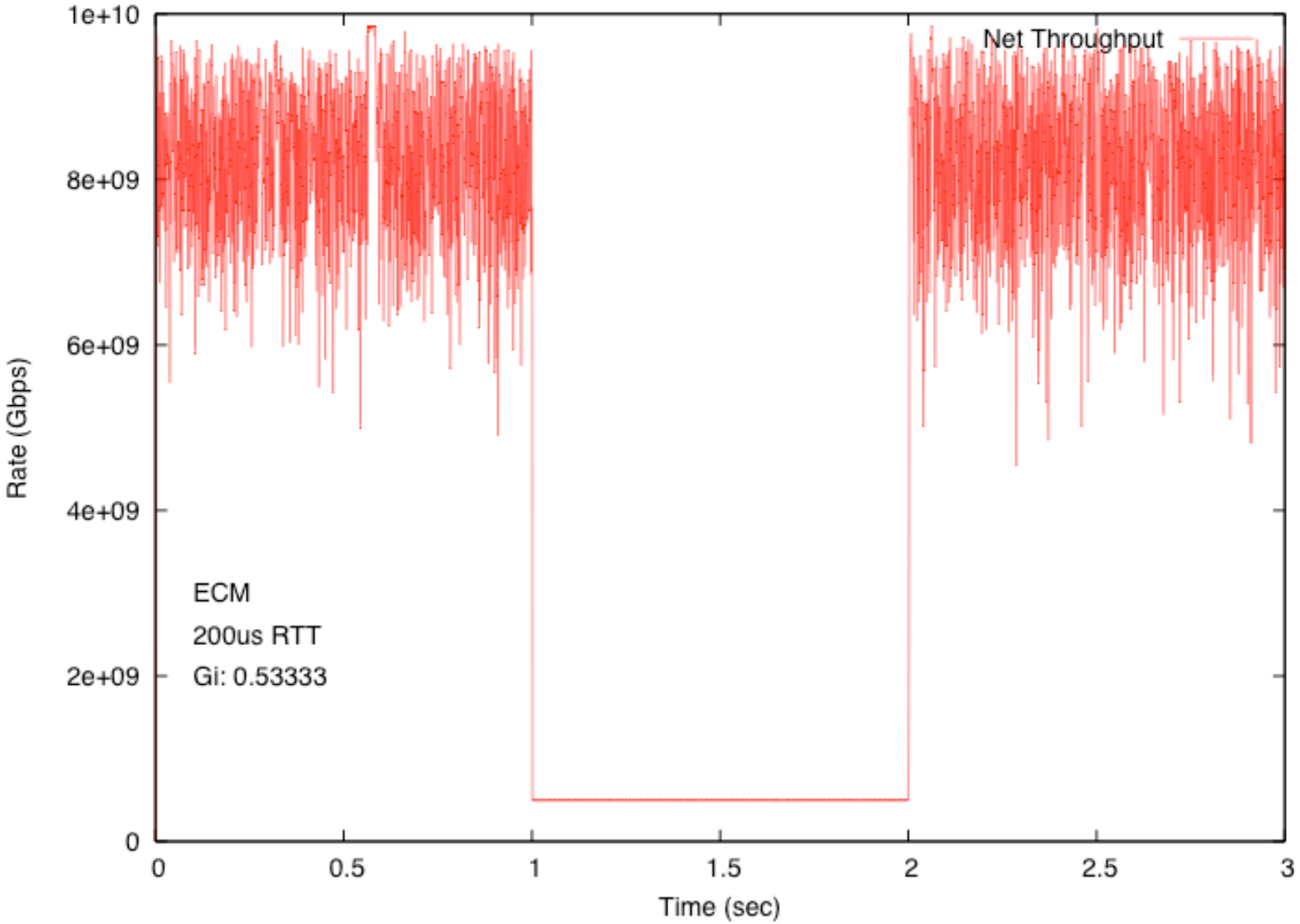
- QCN
  - W = 2.0
  - Q_EQ = 33 KB
  - GD = 0.0078125
  - Base marking: once every 150 KB
  - Margin of randomness: 30%
  - R_{unit} = 1 Mb/s
  - MIN_RATE = 10 Mb/s
  - BC_LIMIT = 150 KB
  - TIMER_PERIOD = 15 ms
  - R_AI = 5 Mbps
  - R_HAI = 50Mbps
  - FAST_RECOVERY_TH = 5
  - Quantized_Fb: 6 bits
- ECM
  - Qeq = 375
  - Qsc = 1600
  - Qmc = 2400
  - Qsat disabled
  - Ecm00 disabled
  - Gi = 0.53333 (varies with RTT)
  - W=2
  - Gd = 0.00026667
  - Ru = 1,000,000
  - Rd = 1,000,000
  - Td = 1ms
  - Rmin = 1,000,000

ECM, RTT=10 usecs

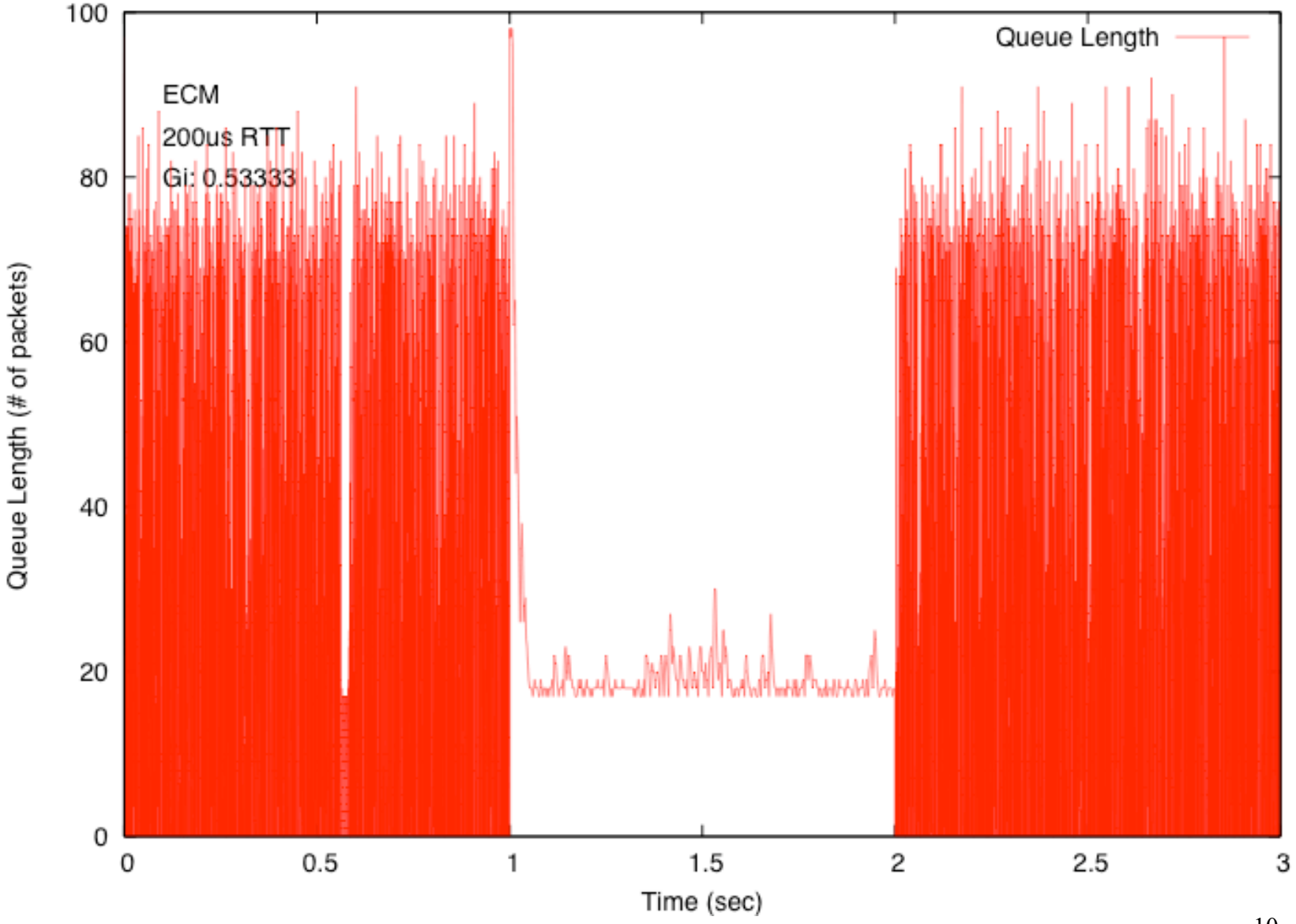


Recovery time = 3 msec

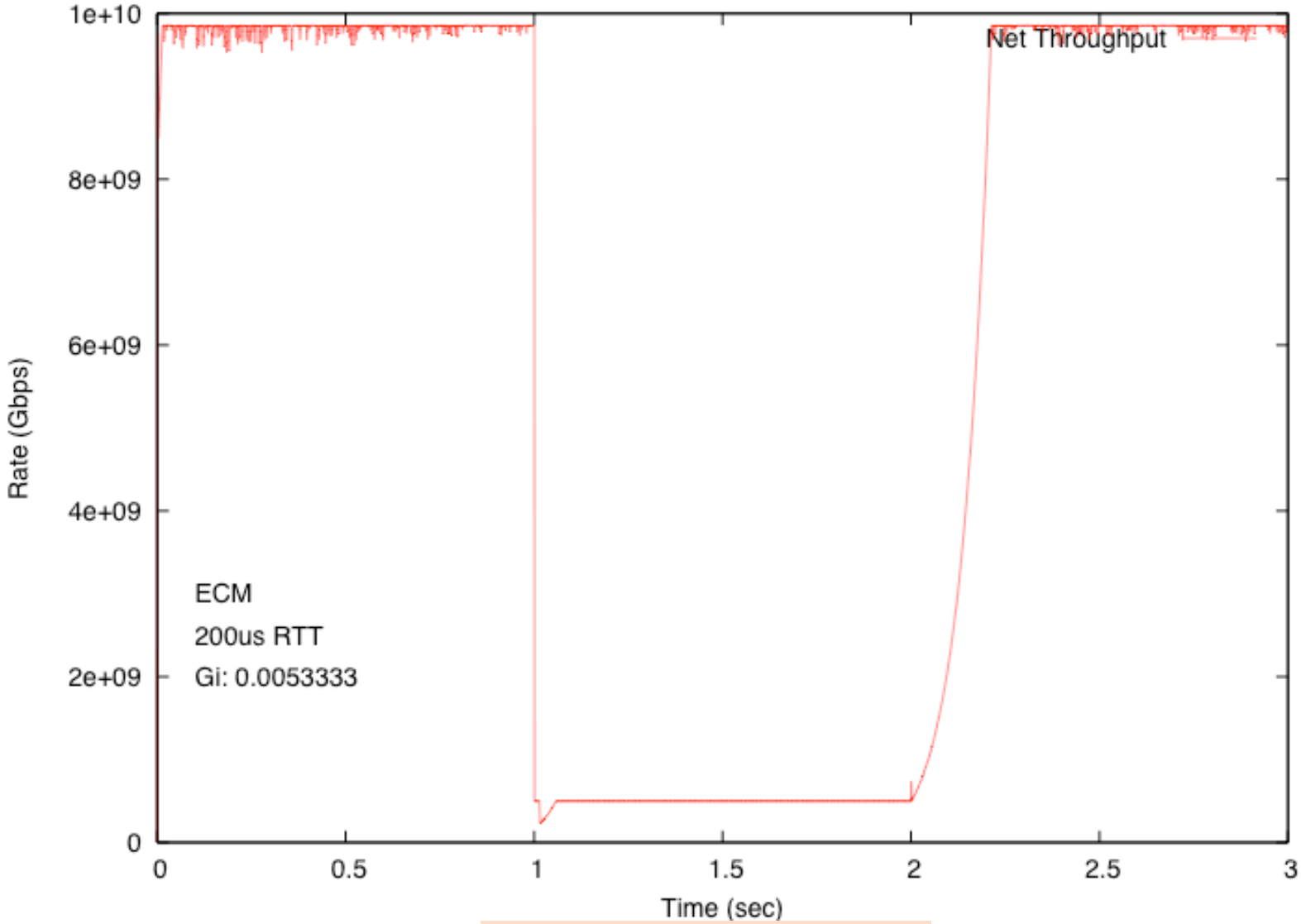
ECM, RTT=200 usecs, Throughput Gi = 0.53333



ECM, RTT=200 usecs, Queue size Gi = 0.53333

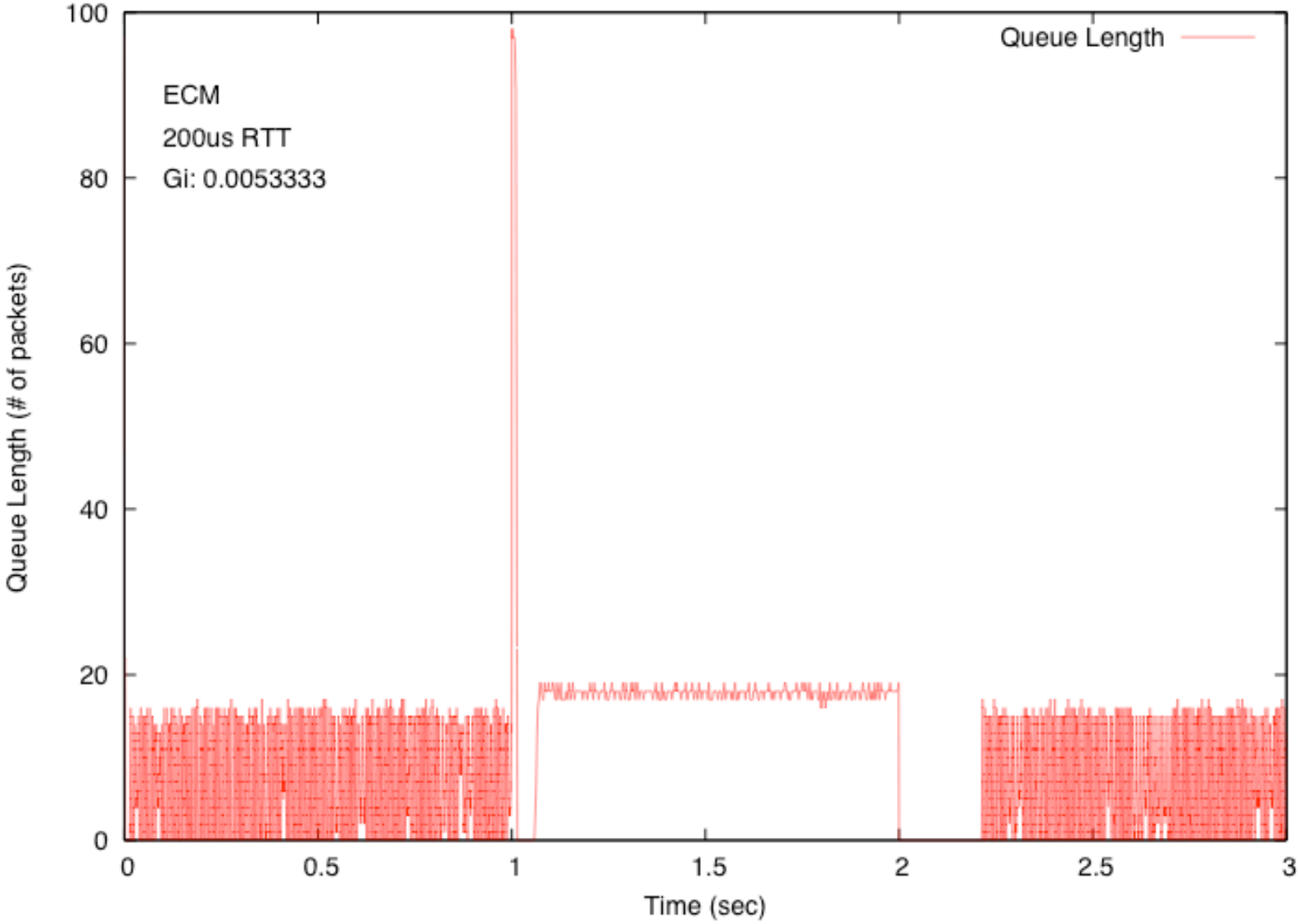


ECM, RTT=200 usecs, Throughput Gi = 0.0053333

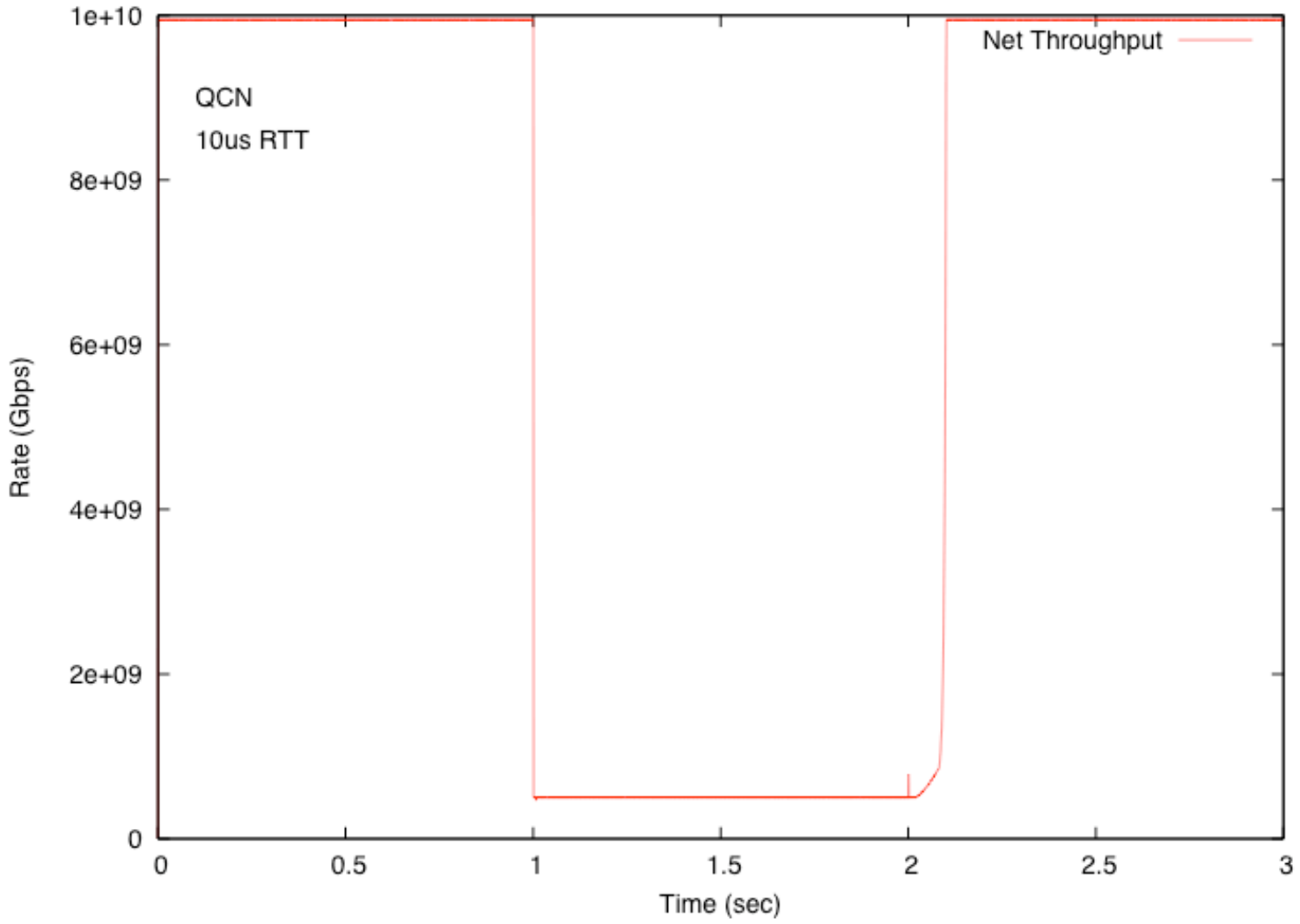


Recovery time = 214 msec

ECM, RTT=200 usecs, Queue size Gi = 0.0053333

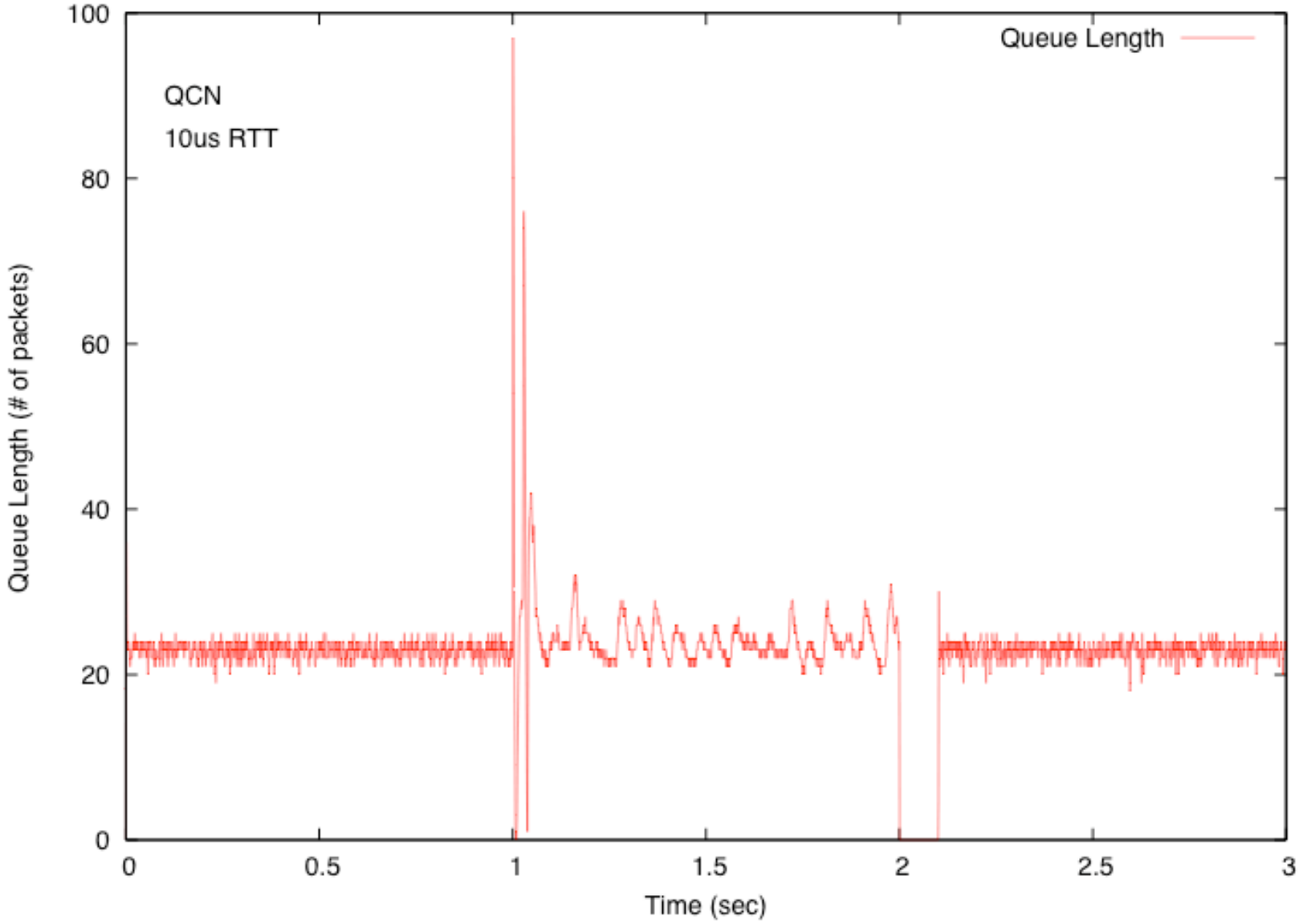


QCN, RTT = 10 us, Throughput

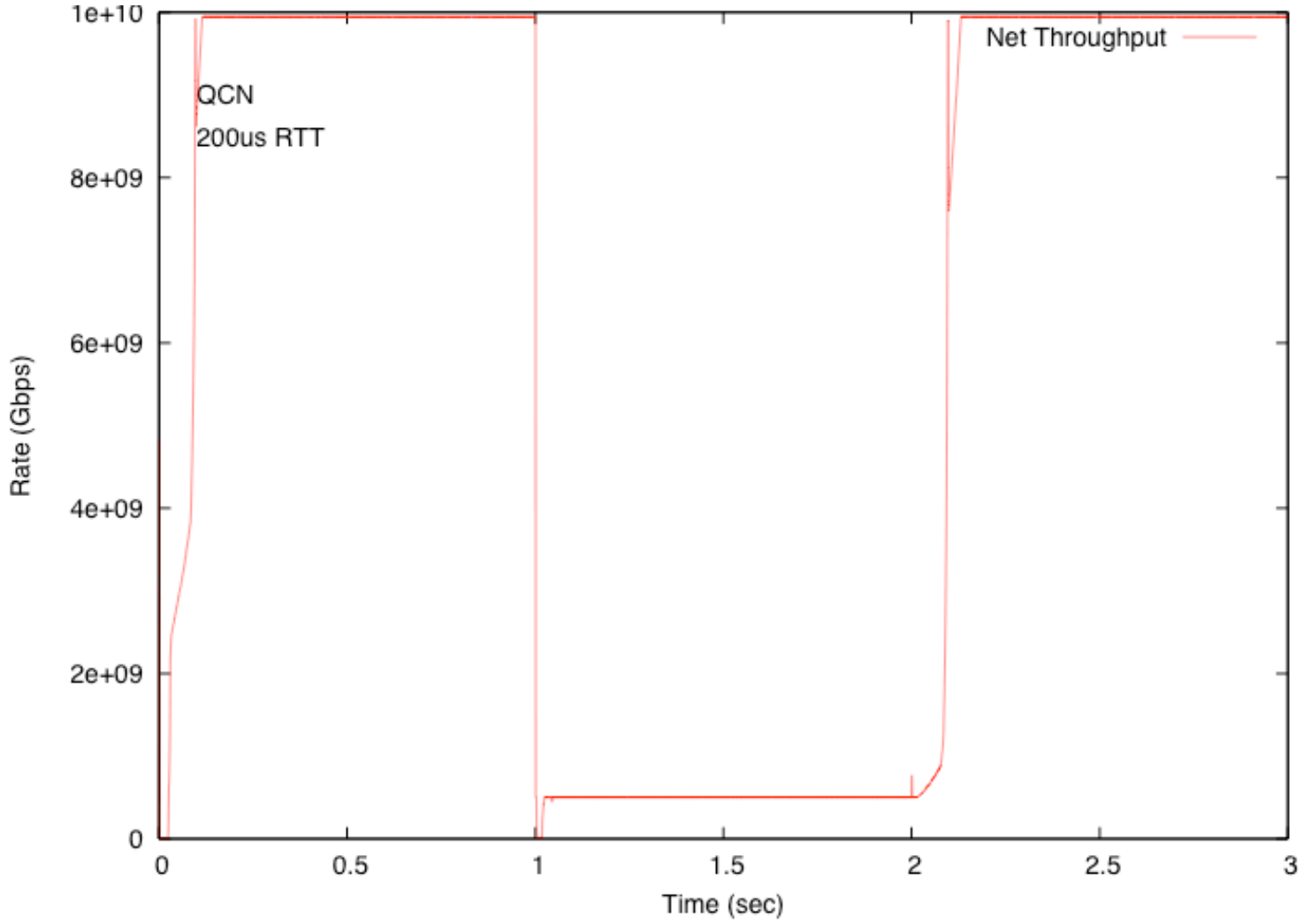


Recovery time = 94 msec

QCN, RTT=10 usecs, Queue size

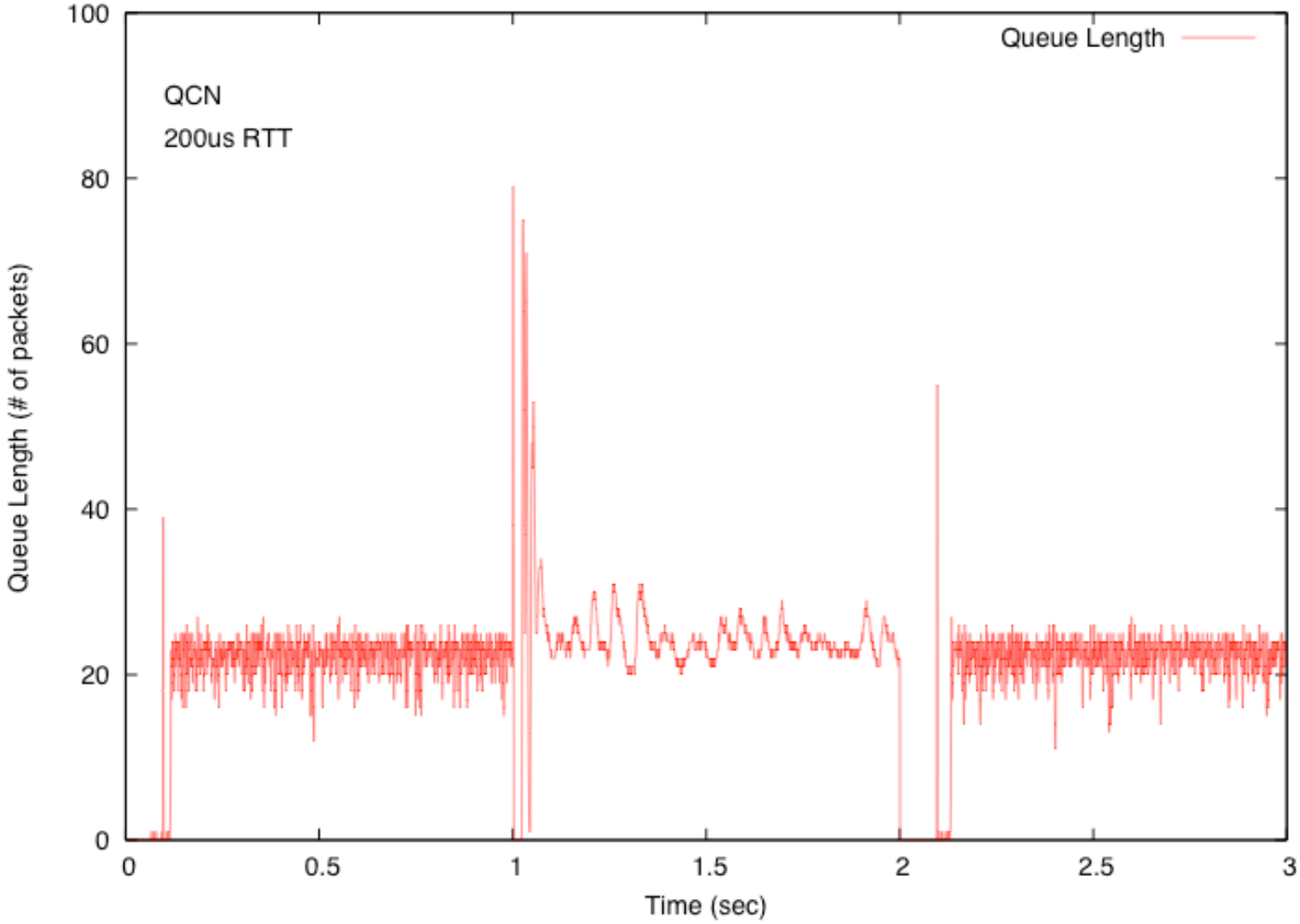


QCN, RTT = 200 us, Throughput



Recovery time = 108 msec

QCN, RTT=200 usecs, Queue size



Sumamry

- Universal stability is important property of QCN
  - BCN, like other AIMD schemes, doesn't have it
  - So, stability at large RTT makes system sluggish
  - In fact, this is precisely the improvement we sought for QCN over BCN

- More next week
  - Benchmark simulations
    - With different hotspot durations
    - Different RTTs
    - Different number of sources