



---

# Simulation Results for QCN-FbHat and other variants of QCN (Update)

Guenter Roeck, Teak Technologies

August 2007



- ECM
  - As specified
- QCN
  - As specified
- QCN-H
  - QCN using Fb-Hat, as specified
- QCN-P
  - QCN with CP-directed probes
- QCN-HP
  - QCN-H with CP-directed probes
- QCN-SP, QCN-PP
  - Sub-path probing (QCN-SP), Path probing (QCN-PP)



## Simulation Parameters

### Traffic

- Bernoulli
- Uniform destination distribution
- 1500 byte frames

### System

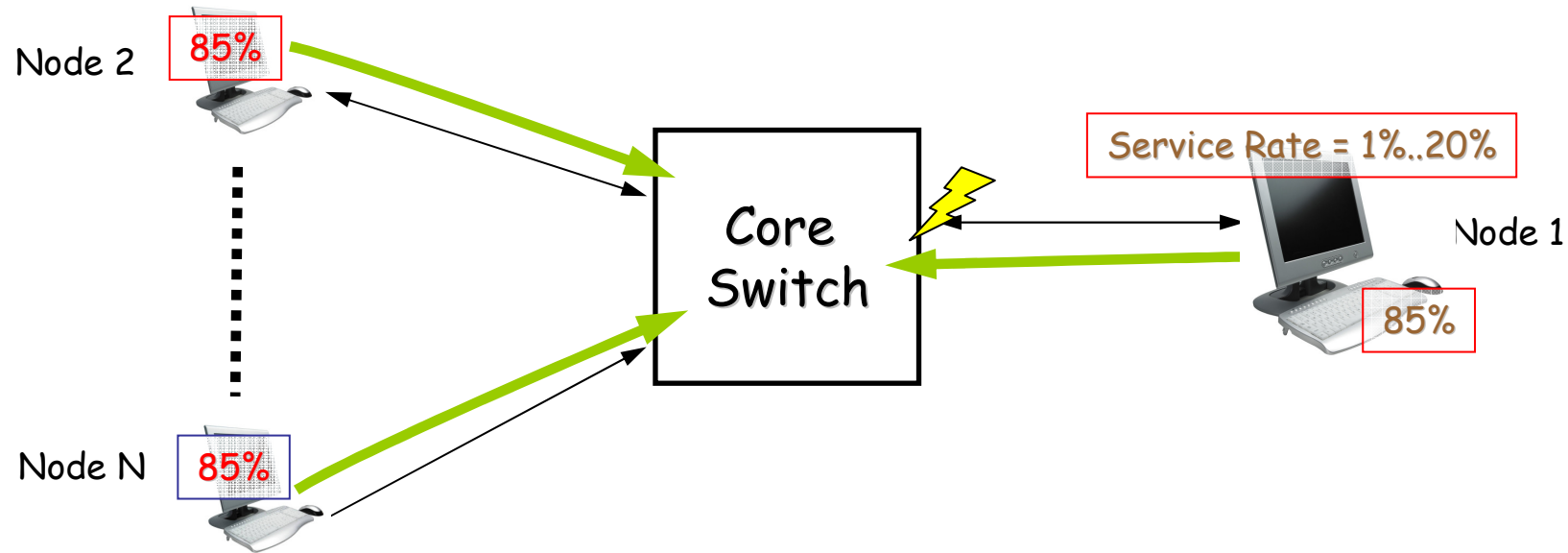
- Switch latency (processing time) = 1us
- Link latency = 500ns
- Switch frame capacity = 200kB, 250 packets
- PAUSE generated by switch
- RP egress buffer size 100 packets

### QCN-xx

- Drift factor = 1.005
- Timer period = 500 uS
- Extra fast recovery enabled
- EFR MAX disabled
- A = 12 Mbit (QCN-H: 24 Mbit)
- Fast Recovery Threshold = 5
- Gd = 1/128
- TO\_THRESH = 150 kBytes
- Qeq = 24kB
- QCN packet processing latency = 5uS
- Hyperactive Increase enabled/disabled
- Psample = 1% .. 10%



## Test 1: Output Generated Single Hotspot, 10 nodes

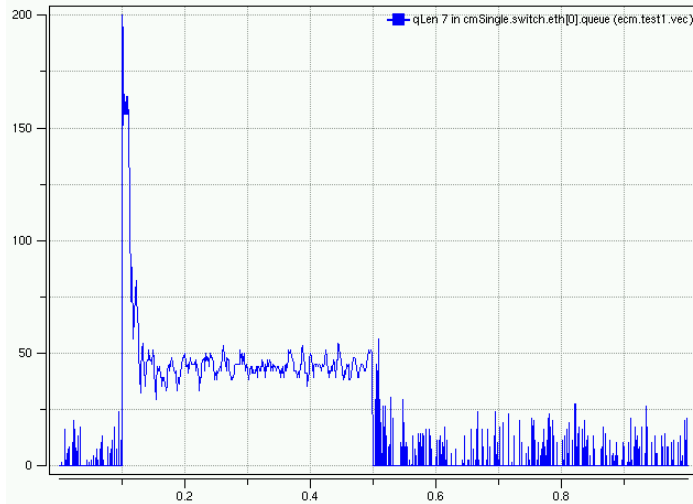


- All nodes (10): Bernoulli distribution, load: 8.5 Gb/s
  - From  $t=0$  to 1s
- Node 1 (hotspot) service rate: 0.1Gbps, 0.5Gbps, 1Gb/s, 2Gbps
  - Duration: 400mS from  $t_i=100$ ms to 500 ms

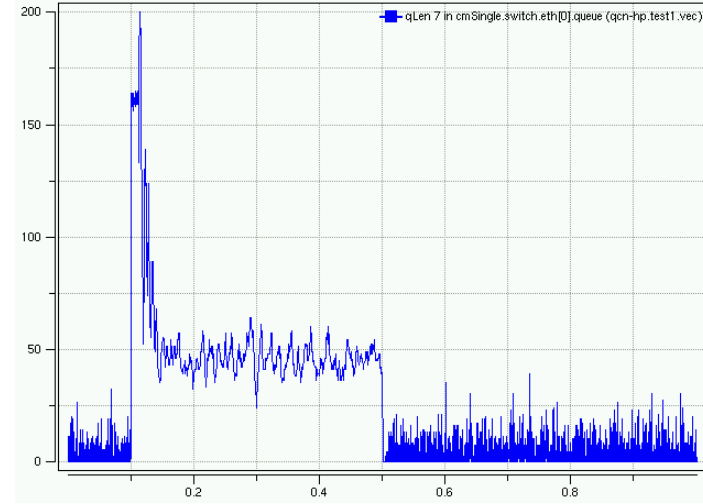


# Test 1: Queue Length (10% service rate)

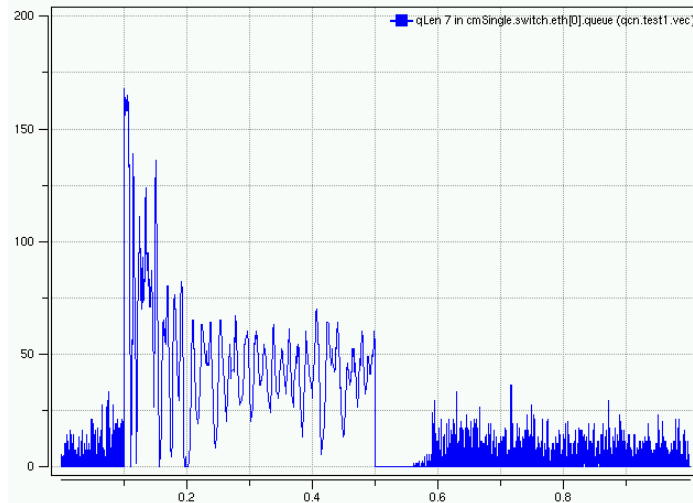
ECM



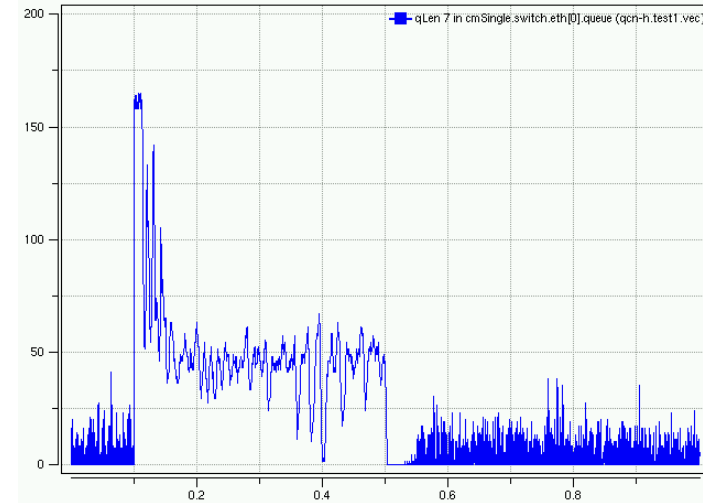
QCN-HP



QCN w/ Hyperactive Increase



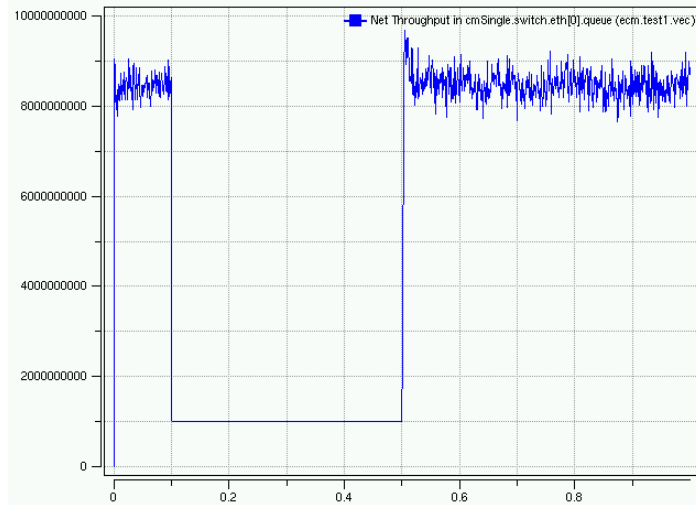
QCN-FbHat w/ Hyperactive Increase



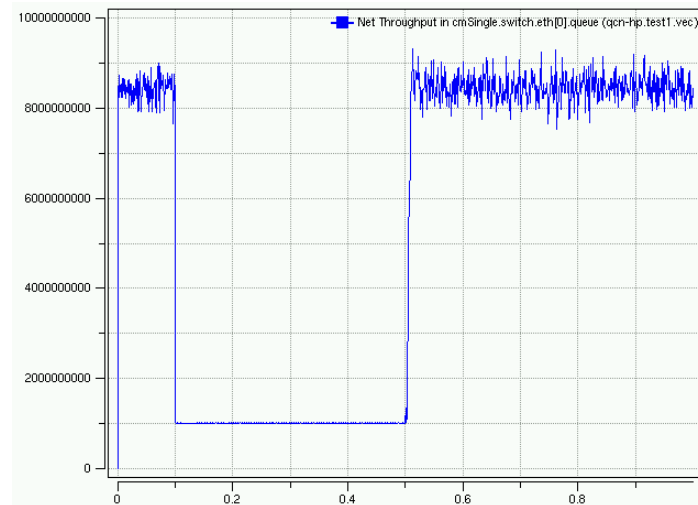


# Test 1: Throughput (10% Service Rate)

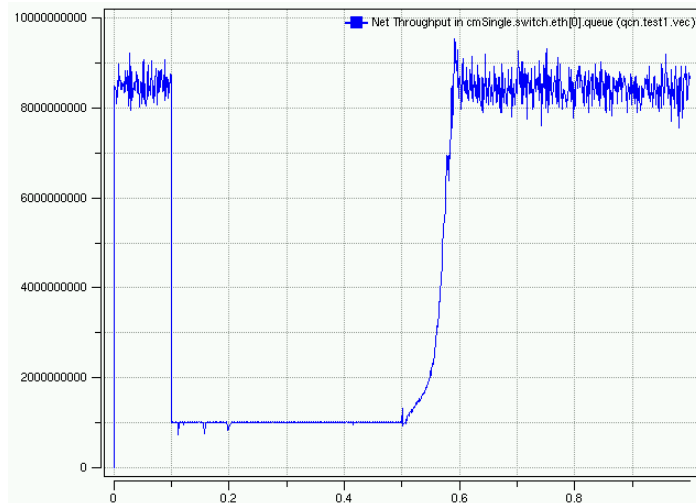
## ECM



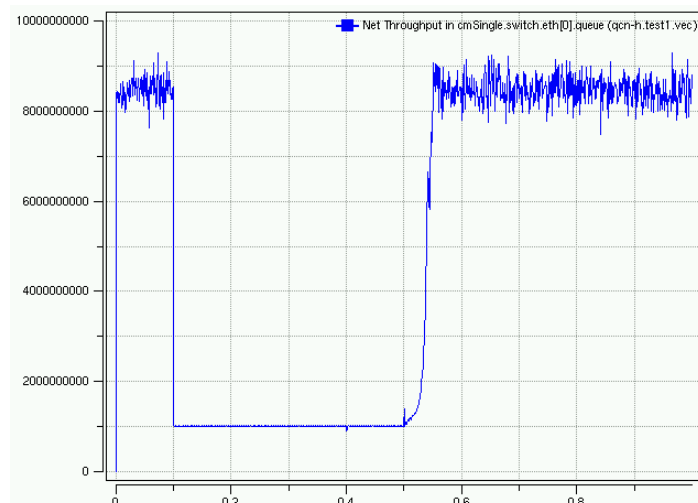
## QCN-HP



## QCN w/ Hyperactive Increase

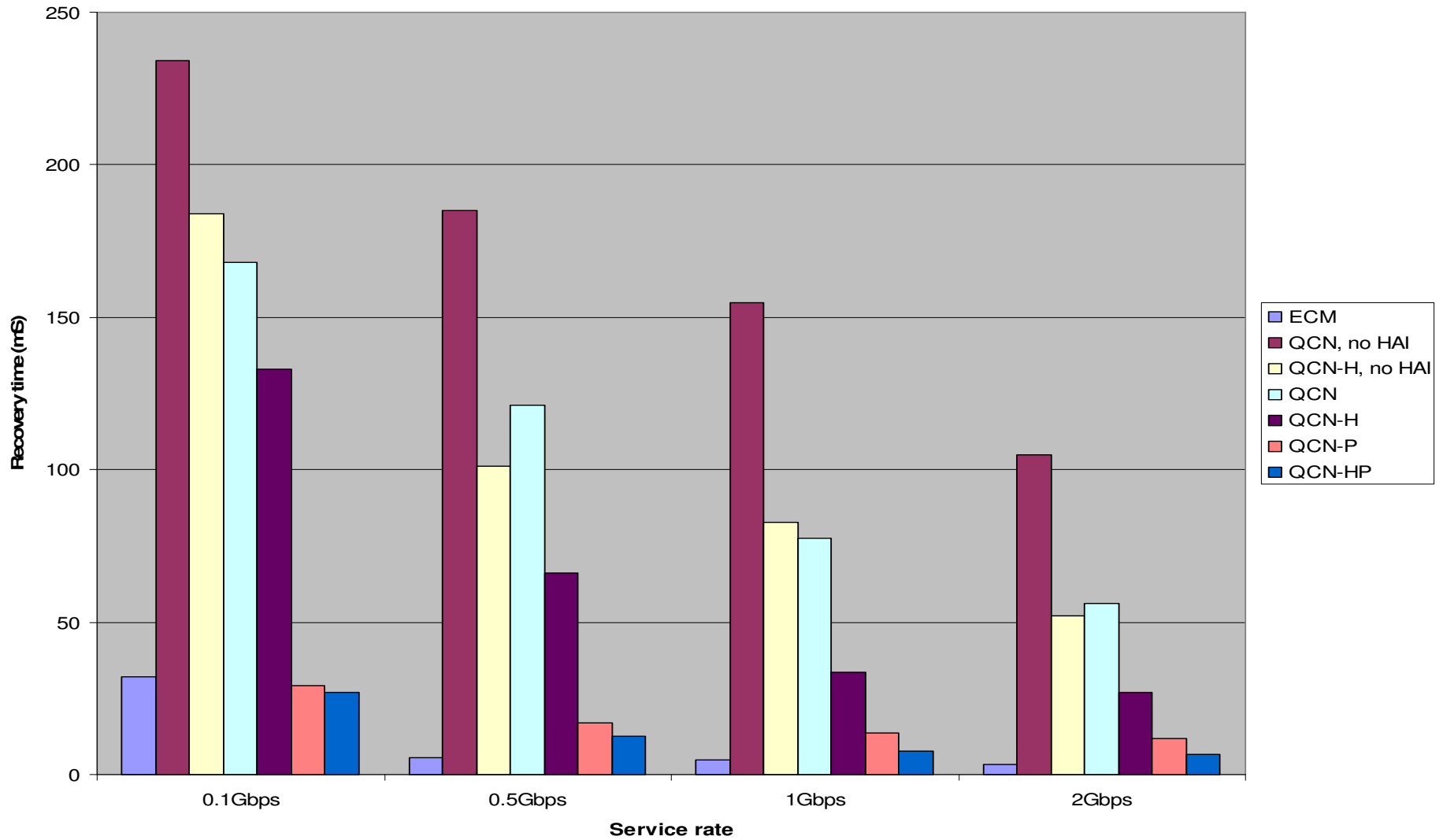


## QCN-FbHat w/ Hyperactive Increase



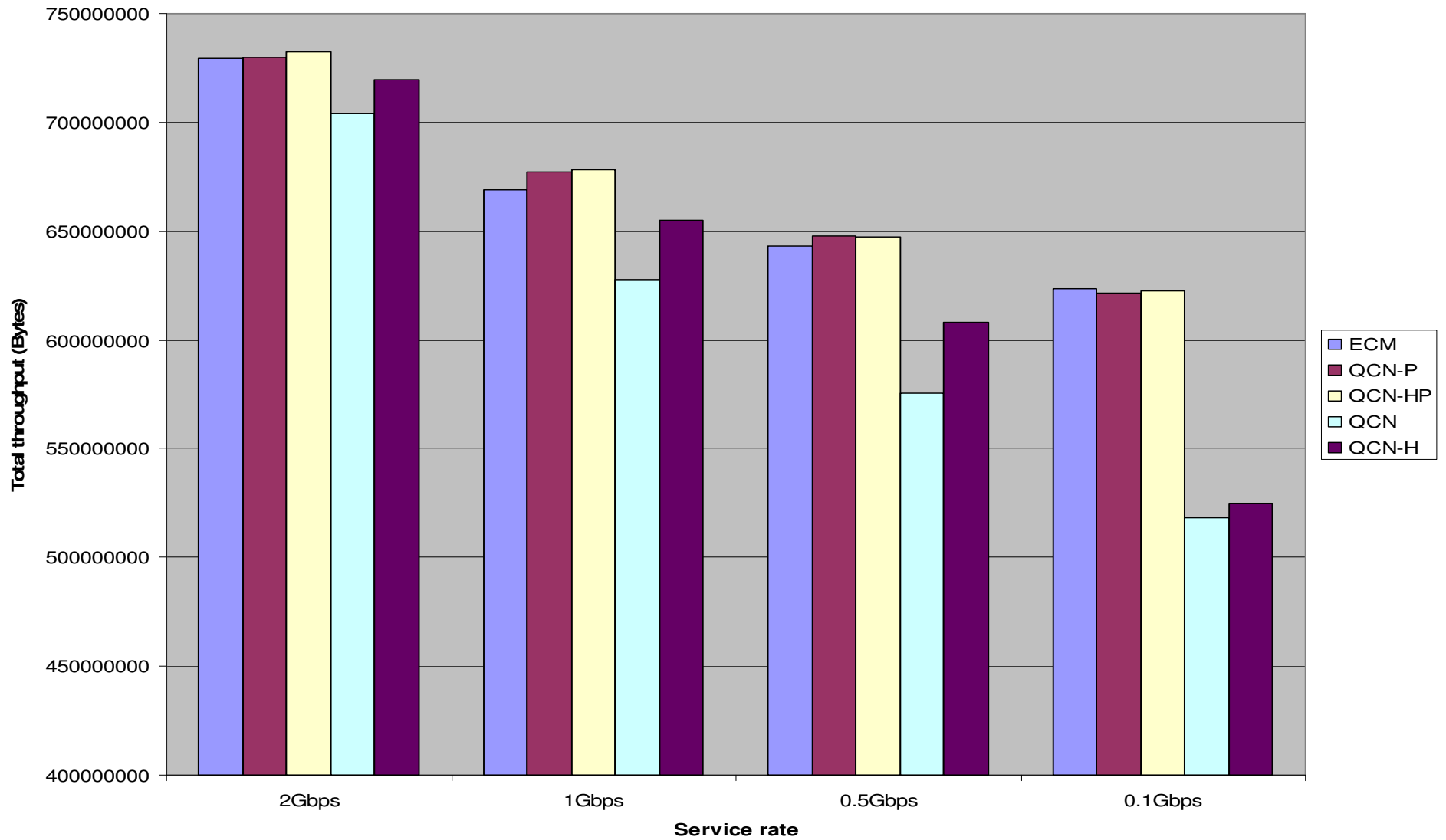


# Test 1: Recovery time





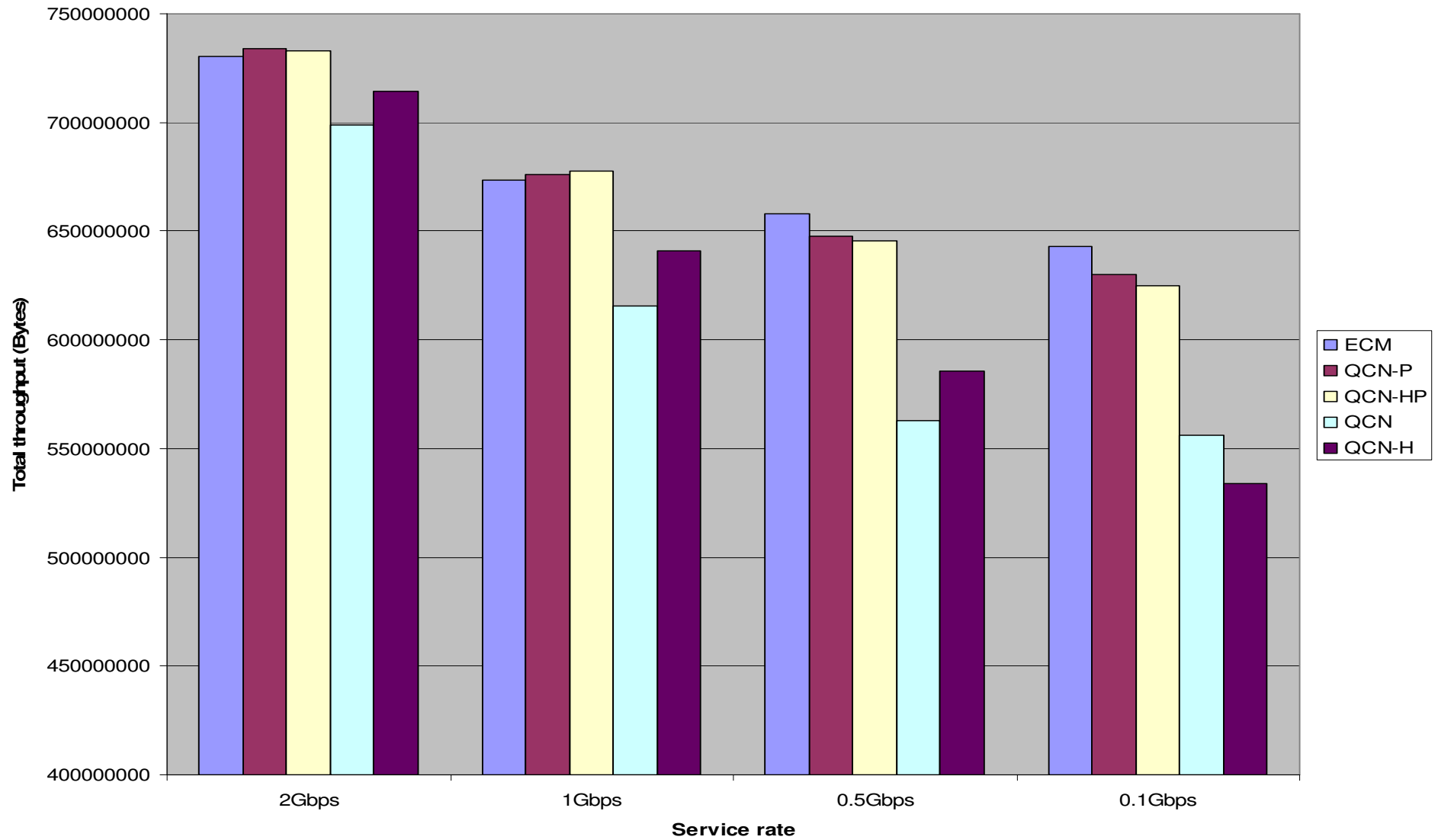
## Test 1: Total Throughput (N=10)







## Test 1: Total Throughput (N=20)

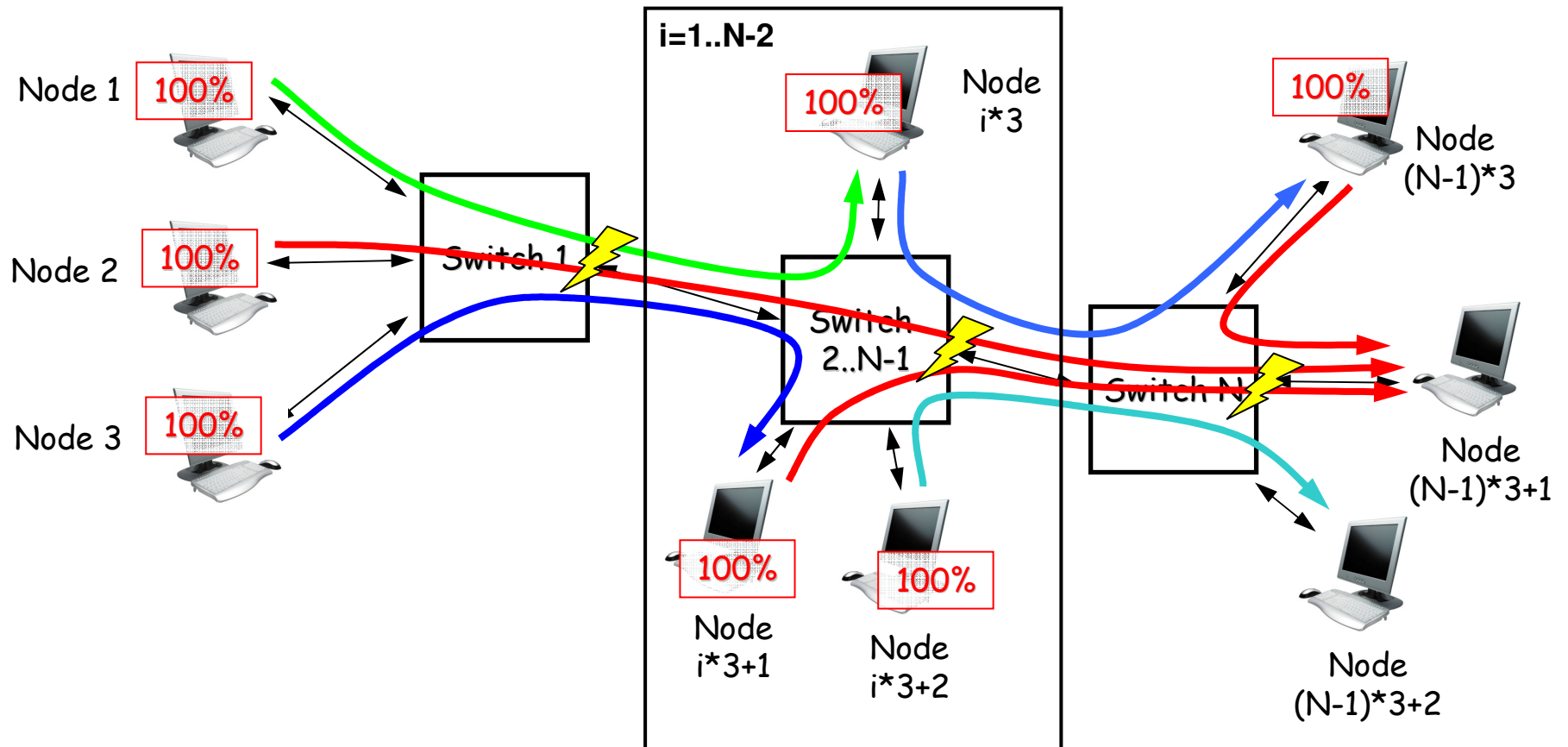




- QCN much improved with Fb-Hat
- QCN reacts (much) faster with Hyperactive Increase enabled
- QCN recovery time depends on hotspot severity
- QCN performance seems to get a little worse if more nodes are added
- Performance advantages with protocols using positive feedback
  - ECM has the fastest reaction time



## 20-stage Hotspot with dynamic load

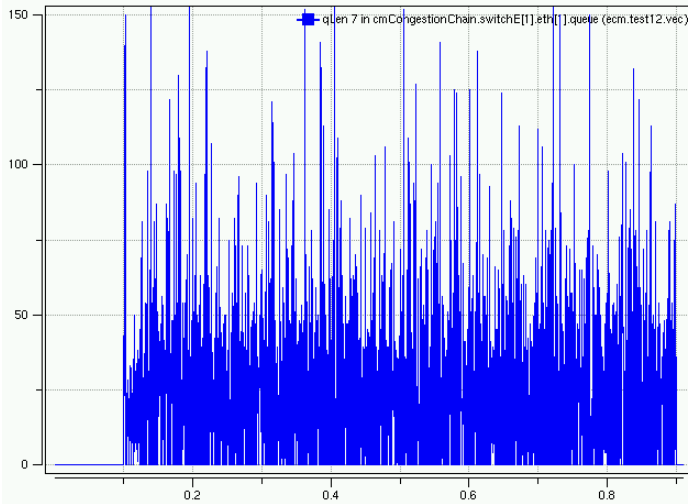


- $N=18$  switches; 3 hosts per switch
- Node  $\langle i \rangle$  sends to node  $\langle i+3 \rangle$ ; Node  $\langle i+1 \rangle$  sends to node  $(N-1)*3+1$ ; node  $\langle i+2 \rangle$  sends to node  $\langle i+4 \rangle$
- Node  $\langle i \rangle$  sending bursty traffic with interval  $1 + \langle i \rangle * 0.1$  ms to node  $\langle i+3 \rangle$
- 100% load from all nodes while sending traffic
- Node  $(N-1)*3+1$  receives traffic from  $\langle N \rangle$  sources
- $N$  hotspots

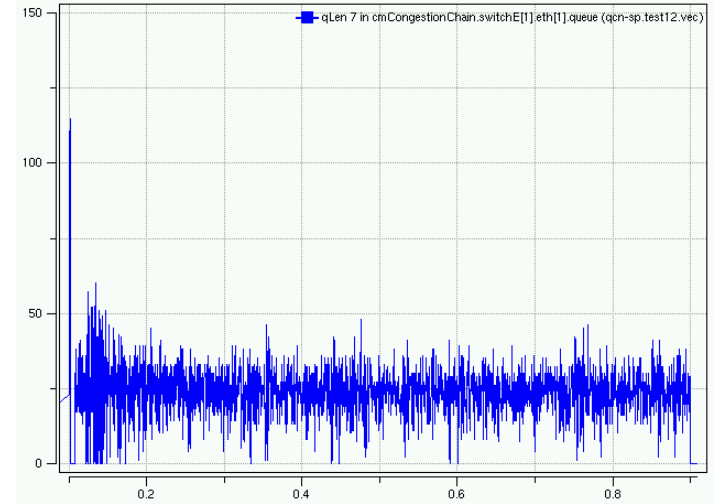


# 20-stage hotspot: Queue length at last switch

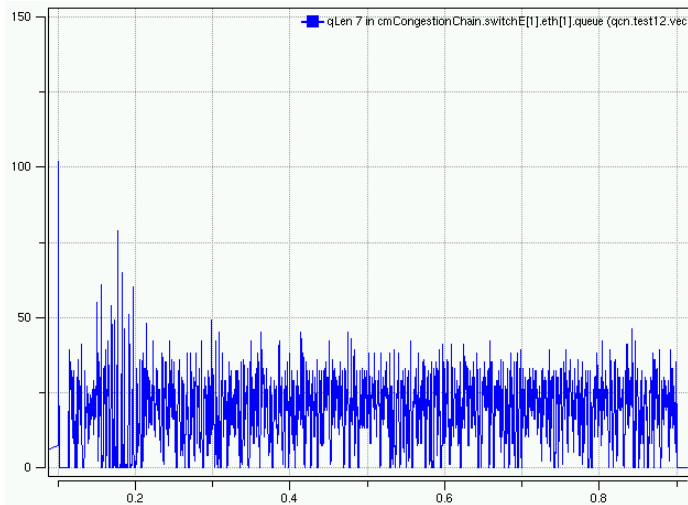
ECM



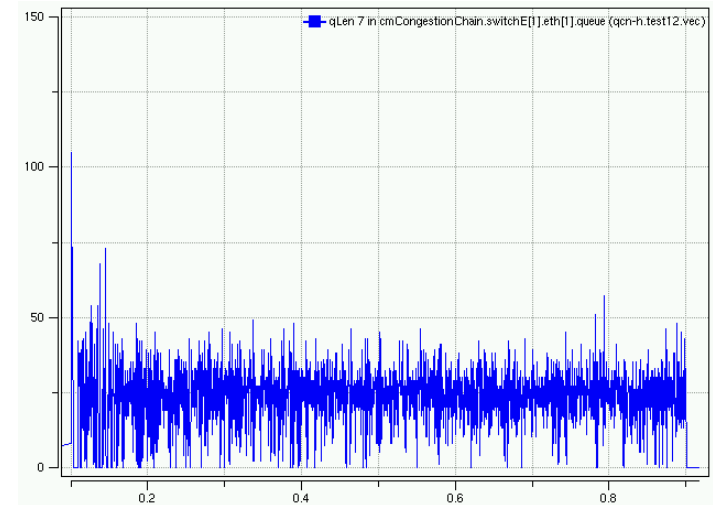
QCN-SP



QCN



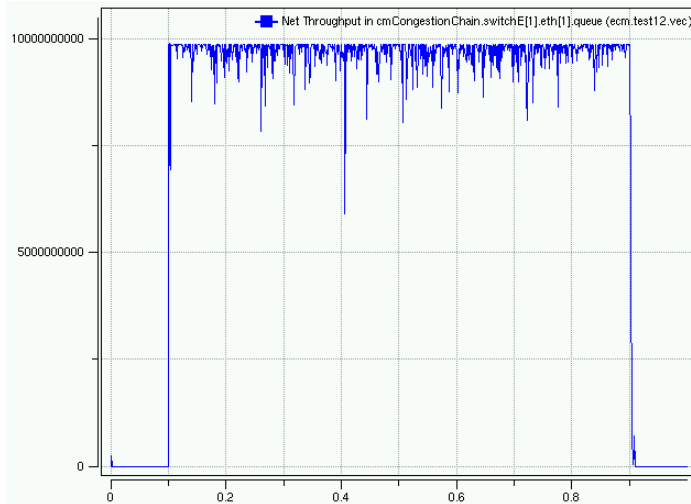
QCN-H



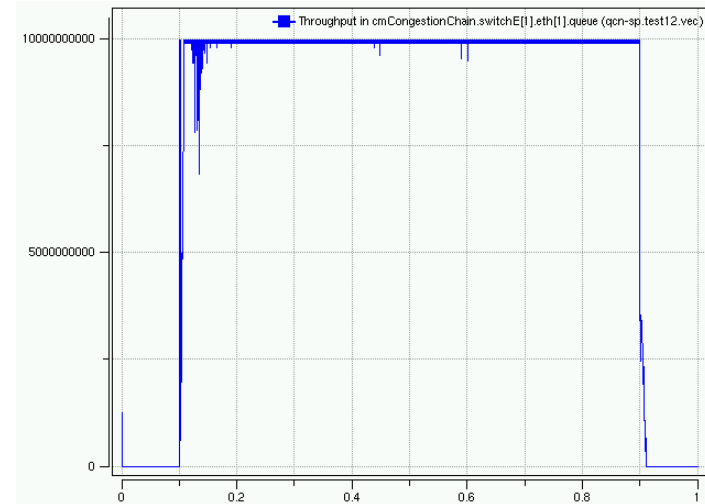


# 20-stage hotspot: Throughput at last hotspot

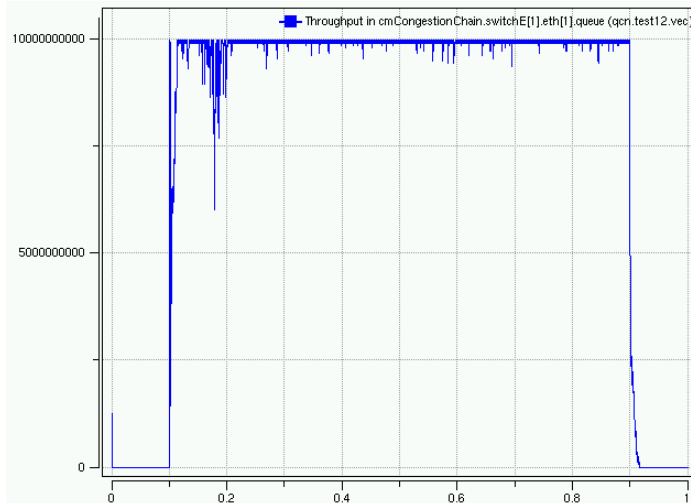
ECM



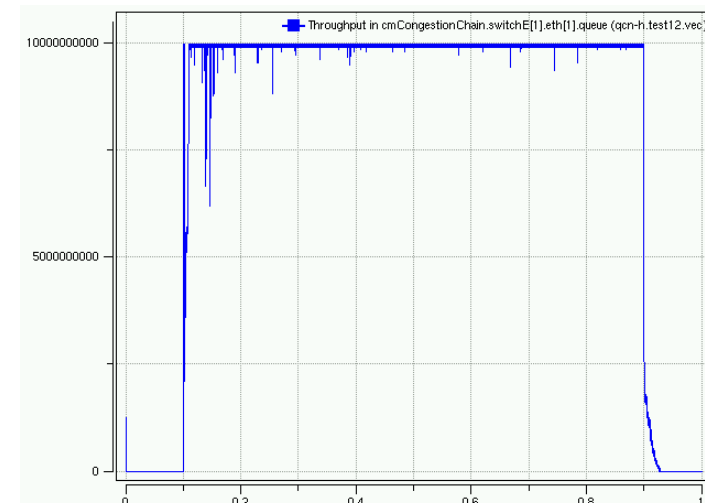
QCN-SP



QCN



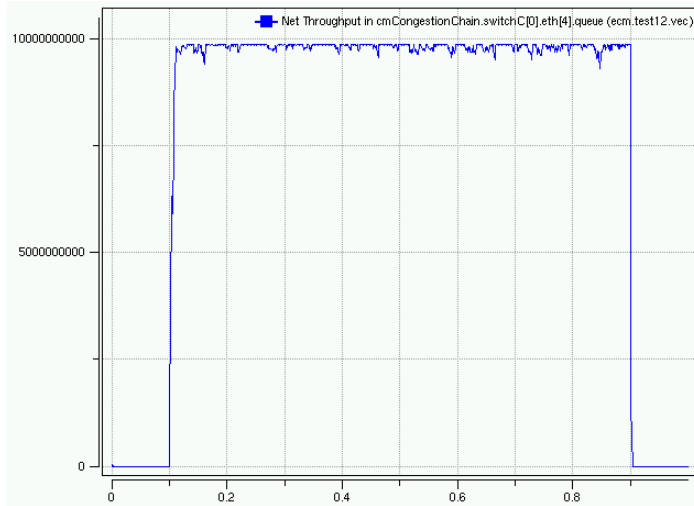
QCN-H



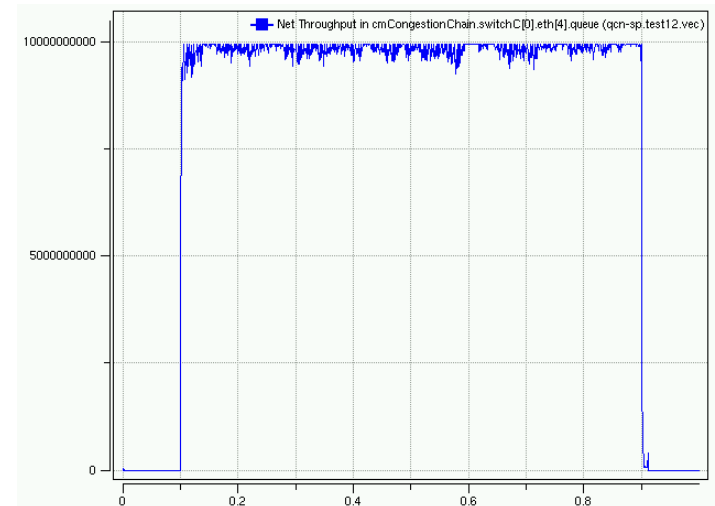


# 20-stage hotspot: Switch 2 Throughput

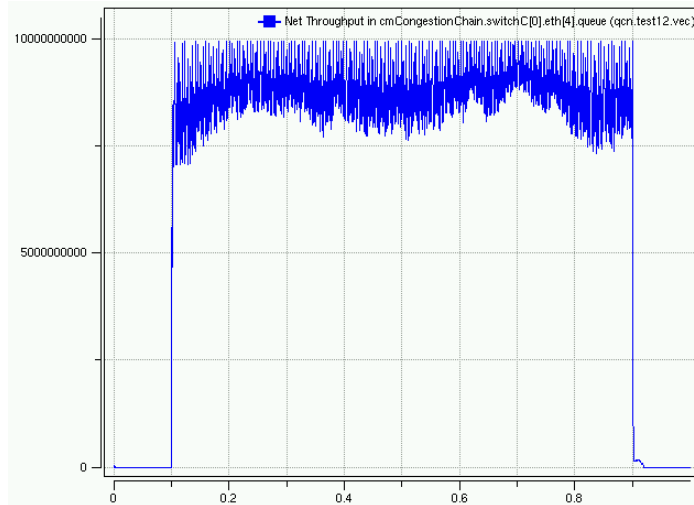
ECM



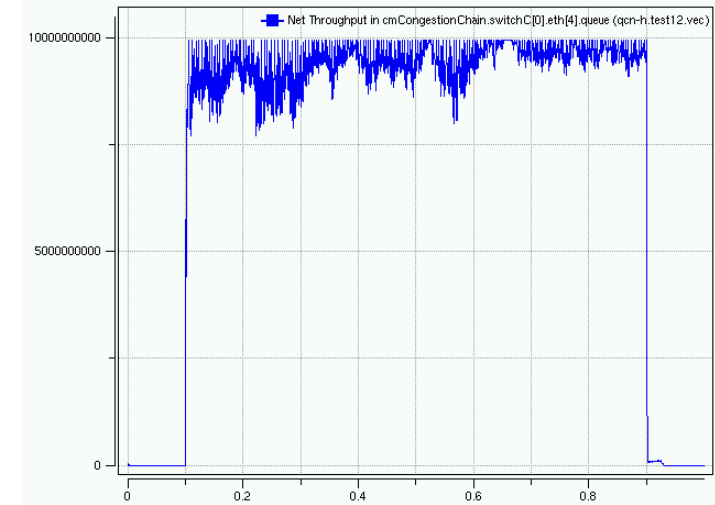
QCN-SP



QCN

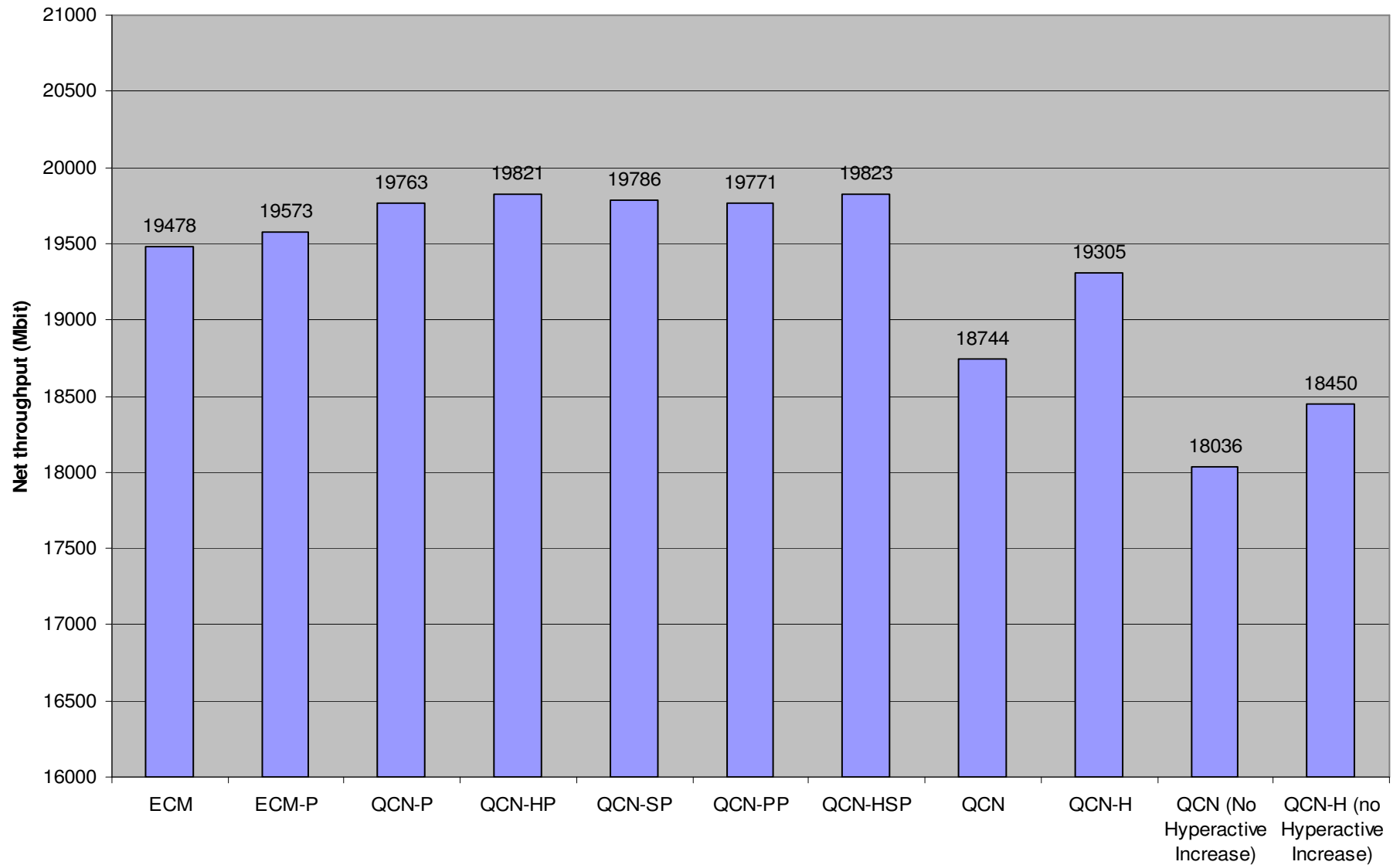


QCN-H





## 20-stage hotspot: Total Throughput through all hotspots





- QCN – same as before
- ECM performance suffers, though not much
  - Tagging
  - Large number of false positives discarded in RP
    - CPID thrashing (?)
  - Overall performance still better than QCN and QCN-H
- No CPID thrashing effects seen for QCN based protocols using probes (QCN-\*P)
- Best performance with Sub-path probing (RP<->CP)
  - QCN-SP, QCN-HSP
  - Even better than with full path probing
  - Only marginally better than direct CP probing