Congestion Management Protocol Characteristics in Complex Simulation Scenarios

Guenter Roeck, Teak Technologies

IEEE 802.1Qau
Stockholm Interim Meeting, September 2007

- Determine protocol characteristics in corner cases and in complex scenarios
- Get a better understanding of protocol limitations

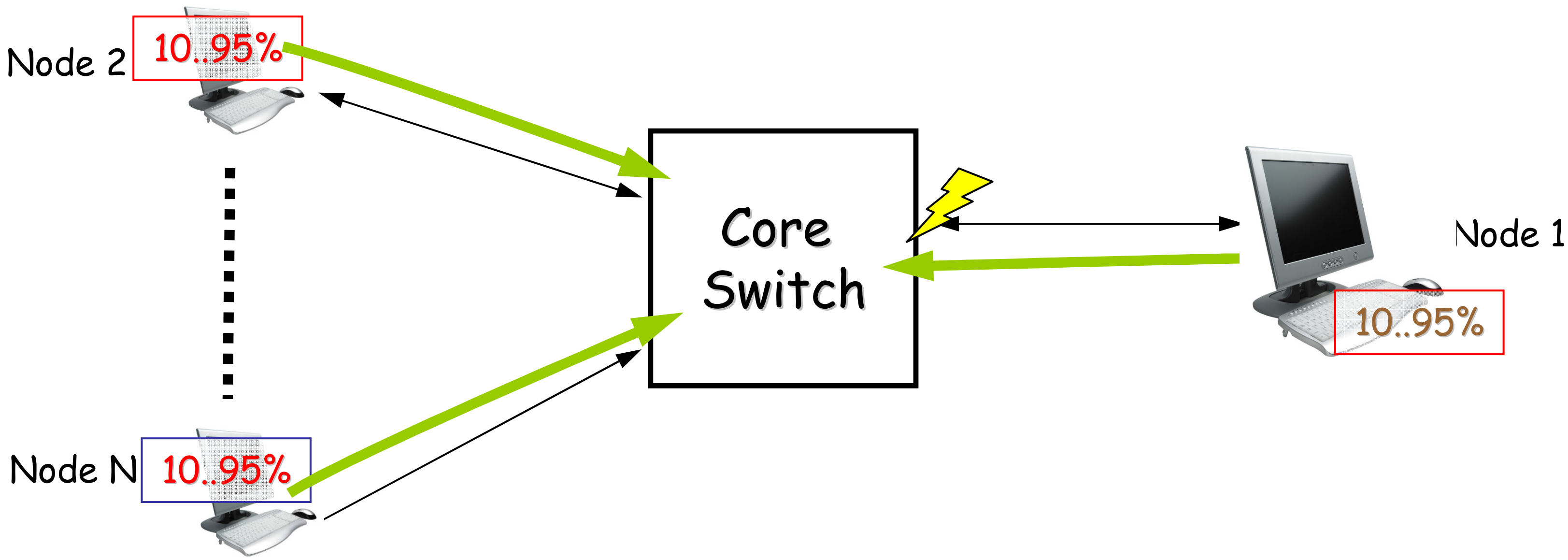
- Test Scenarios
- Simulated Protocols
- Simulation Results
- Summary and Conclusions

Test Scenarios

- No hotspots, variable load
  - Simulate normal network operation
  - Look for
    - Number and frequency of protocol messages
    - Number of created rate limiters
- OG hotspot with oscillating service rate
  - Simulate transient congestion in higher priority CoS
  - Look for overall throughput
- Baseline scenario with large forward latency
  - Simulate network with large BW * latency product
  - Look for stability (throughput, queue length)
- Large number of hotspots with dynamic load
  - Simulate complex network with high load and many CPs
  - Look for overall protocol performance (throughput)
  - Look for effects of CPID Thrashing

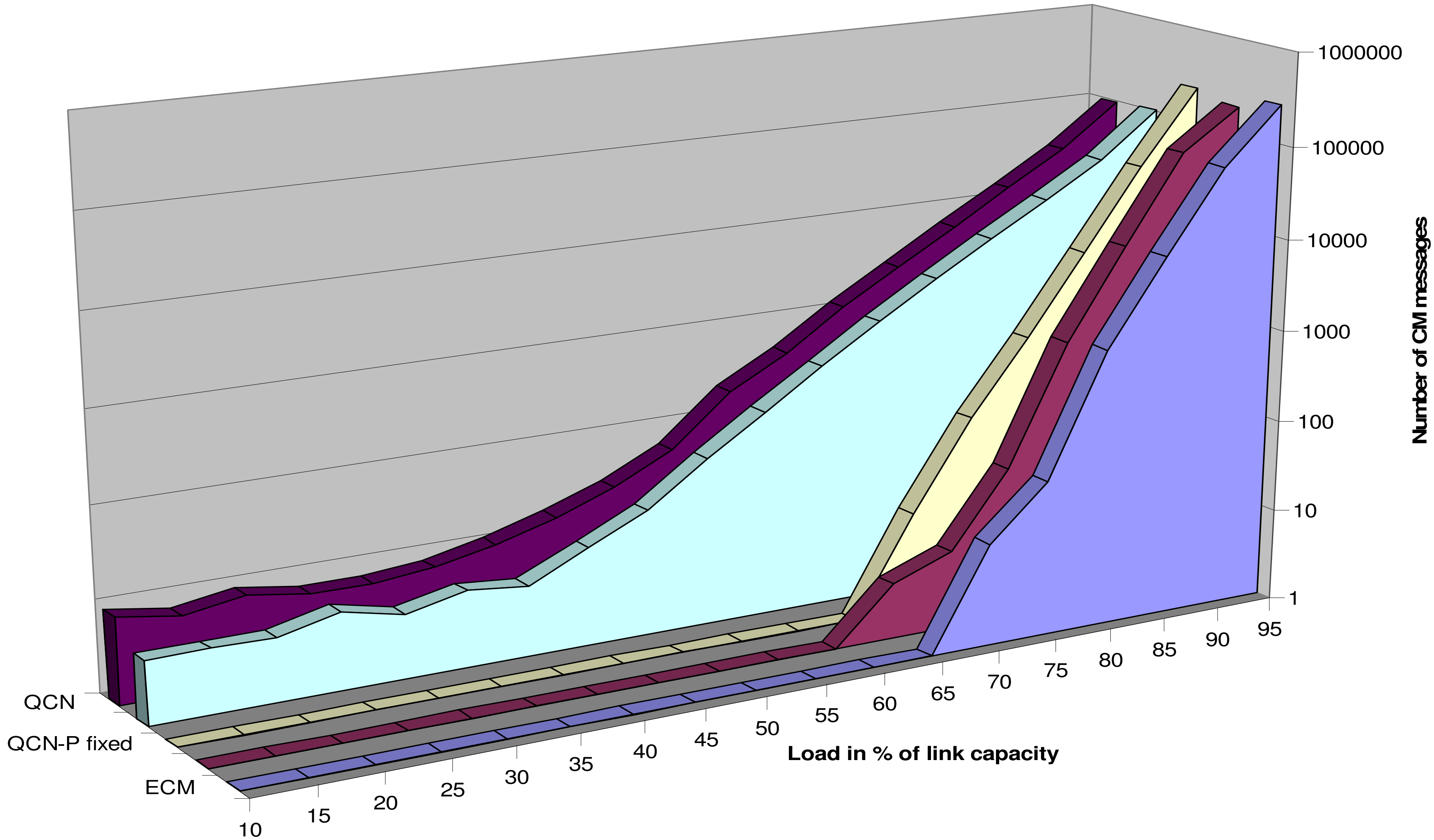
- ECM
  - As specified
- QCN, QCN-FbHat
  - As specified
- ECM-P, ECM-SP
  - ECM with CP-directed probes (-P) and Sub-Path probes (-SP)
- QCN-P
  - QCN with CP-directed probes
- QCN-HP
  - QCN-FbHat with CP-directed probes
- QCN-SP, QCN-PP
  - Sub-path probes (QCN-SP), Path probes (QCN-PP)

No Hotspot, 20 nodes

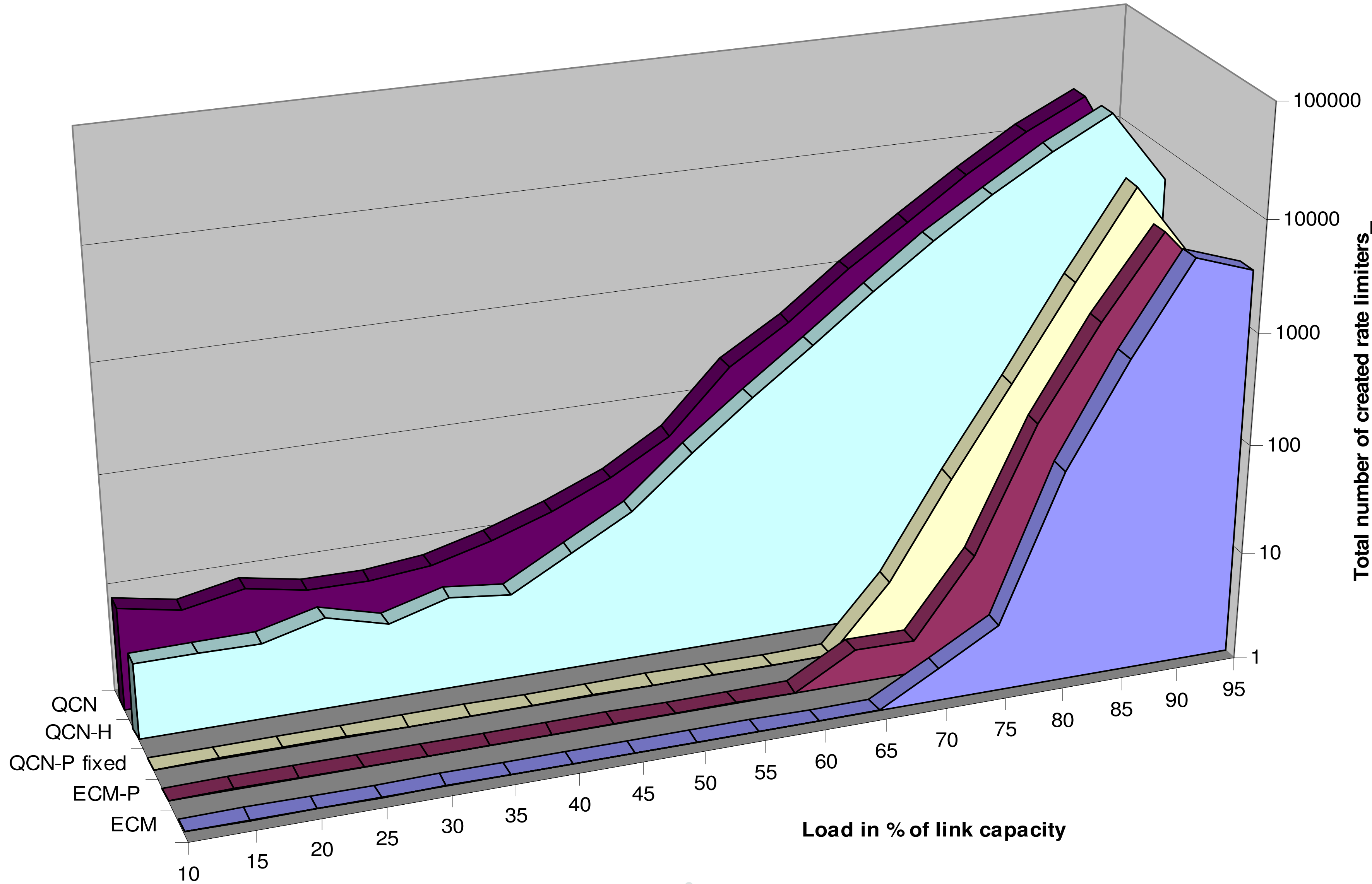


- All nodes (20): Bernoulli distribution, load: 1Gbps .. 9.5 Gb/s
  - From t=0 to 1s
- No hotspot
- Measure number of CM messages and number of created Rate Limiters

Number of CM Messages



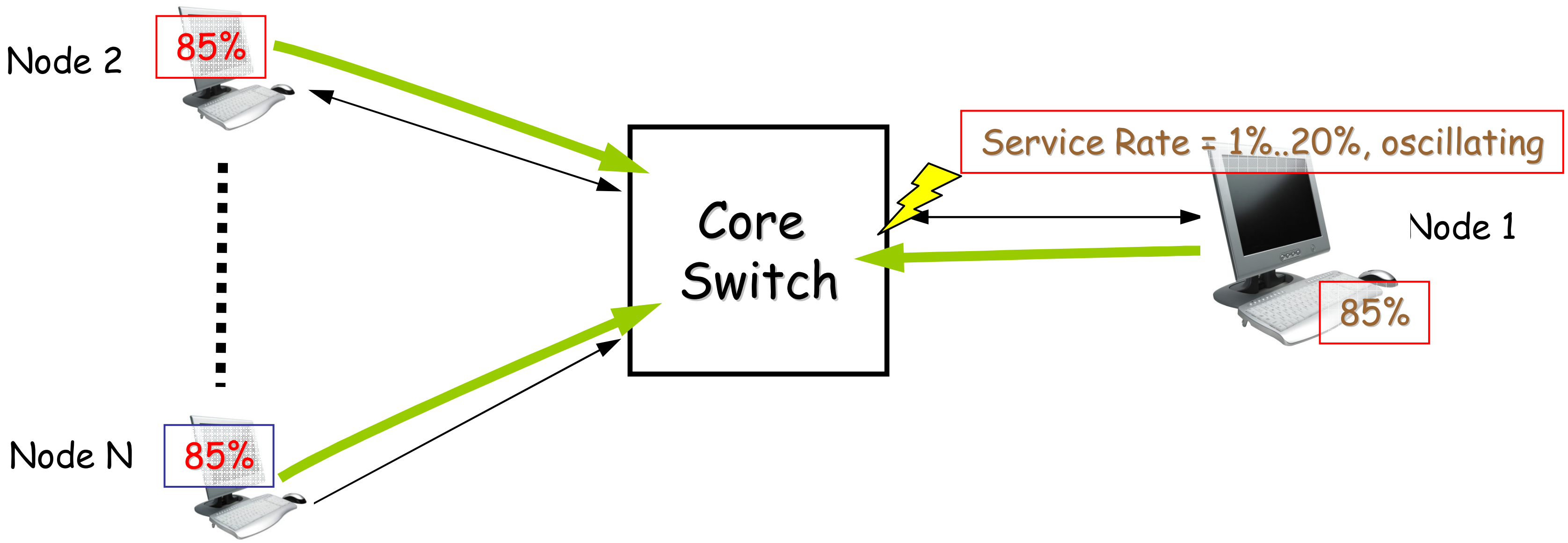
Number of created Rate Limiters



Observations

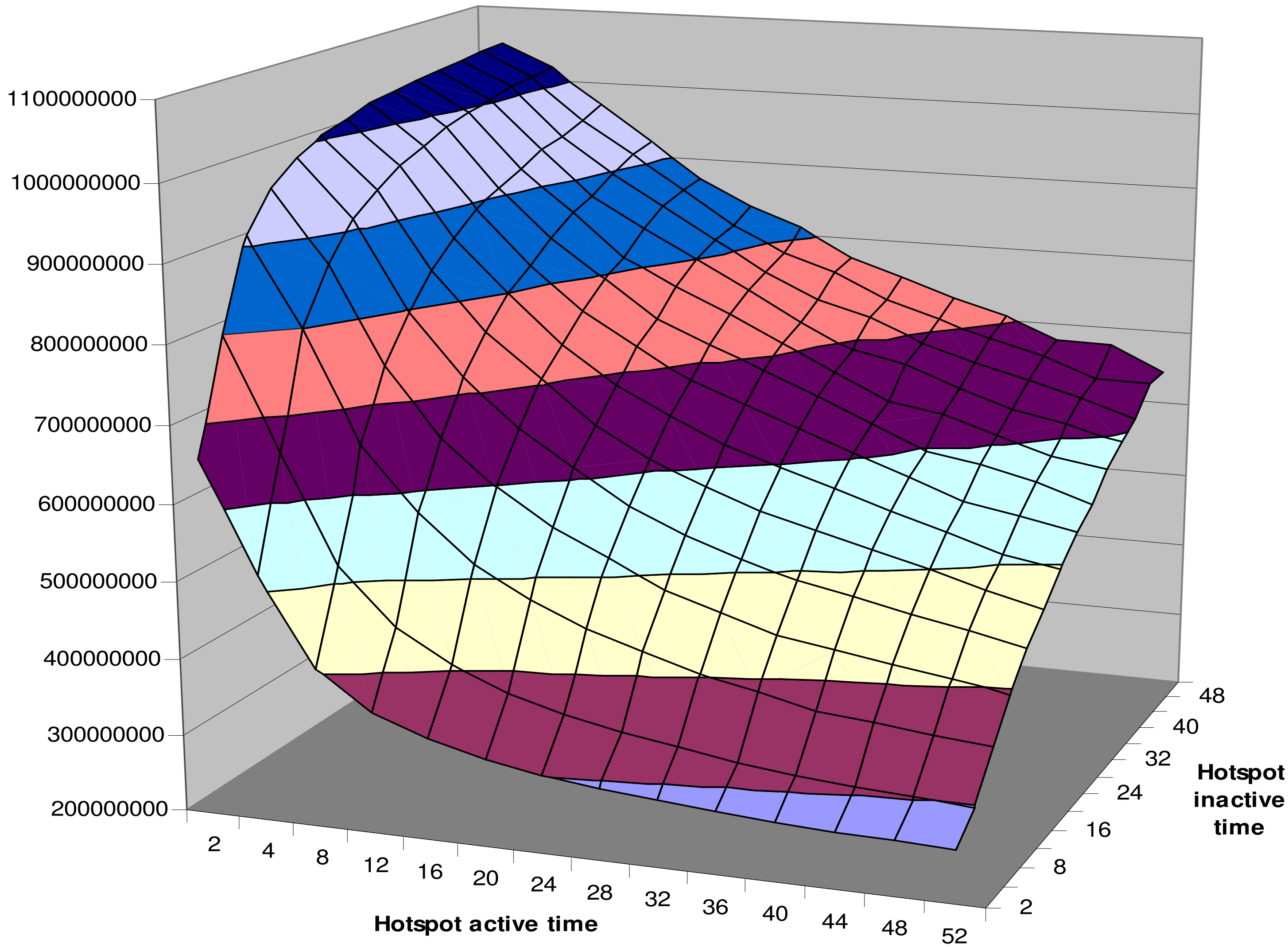
- QCN provides negative feedback at any given load
- Number of spurious rate limiters significantly higher than with protocols using Qoff to determine if to send negative feedback
- Caused by using Fb to calculate if negative feedback should be sent
  - Can send negative Fb with Qlen = Qeq/3 (if W=2)
  - Can occur after single Jumbo frame was received and queued
    - Qlen = 9k, QlenOld = 0: Fb = (24k-9k) – 2*9k = 15k – 18k = -3k
  - Can not fix by using Qoff, since CM messages with negative Fb are needed after RL was created
  - Must send Qoff and Qdelta instead of Fb to limit creation of spurious Rate Limiters

Single Oscillating Hotspot, 20 nodes



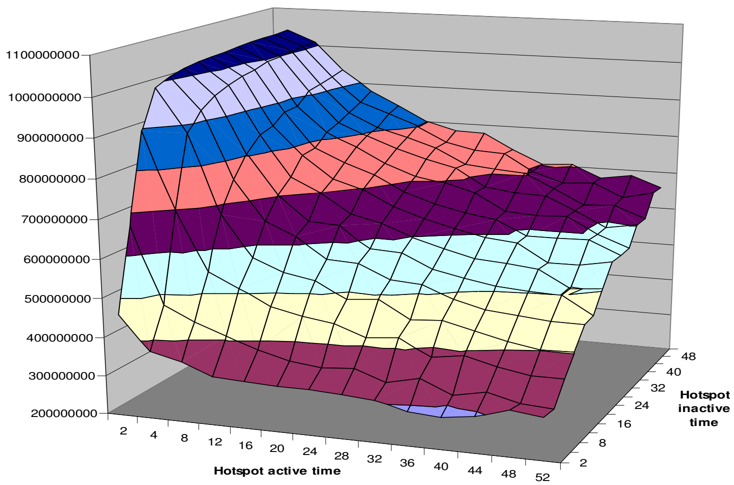
- All nodes (20): Bernoulli distribution, load: 8.5 Gb/s
  - From t=0 to 1s
- Node 1 (hotspot) service rate: 1Gb/s
  - Duration: 800mS from ti=100ms to 900 ms
  - Frequency: tOn=2..50ms, tOff=2..50ms
- Looking for Throughput distribution and bandwidth loss
- Real world scenario: Higher priority CoS with recurring transient congestion

Expected Throughput Distribution

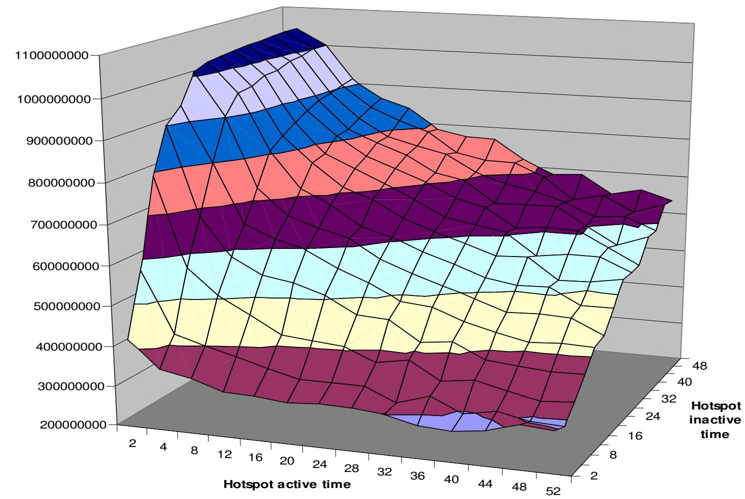


Oscillating Hotspot: Throughput Distribution

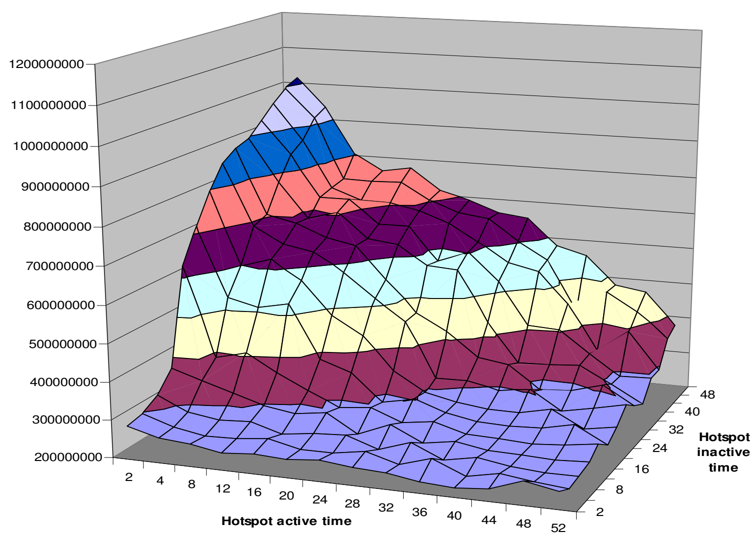
ECM



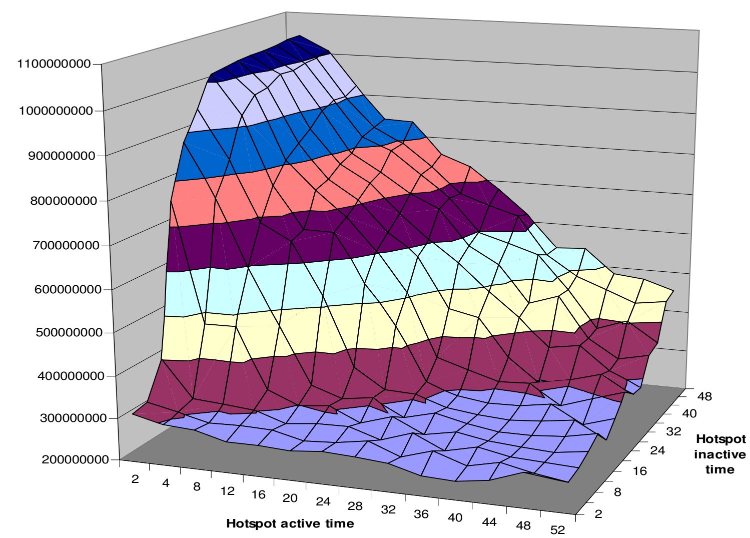
QCN-P



QCN

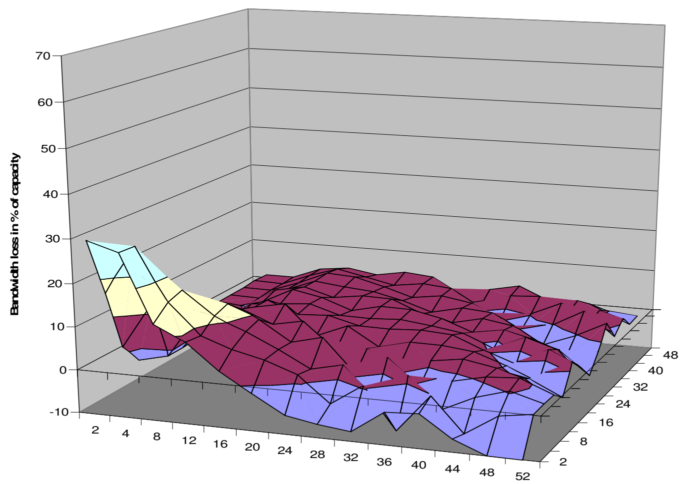


QCN-FbHat

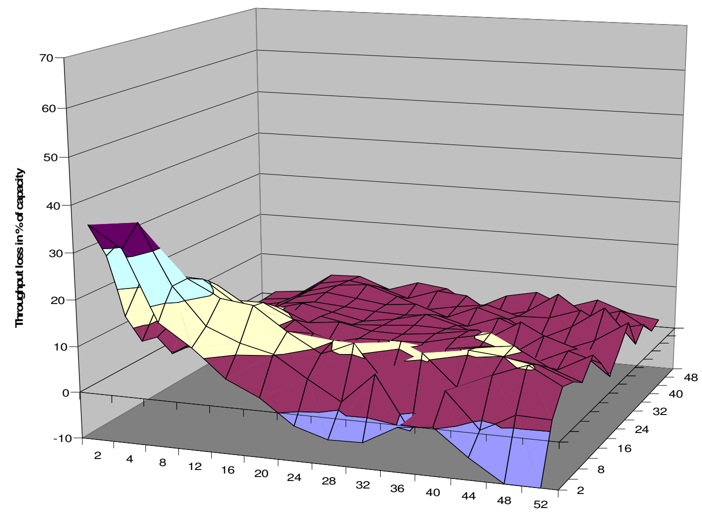


Oscillating Hotspot: Bandwidth Loss

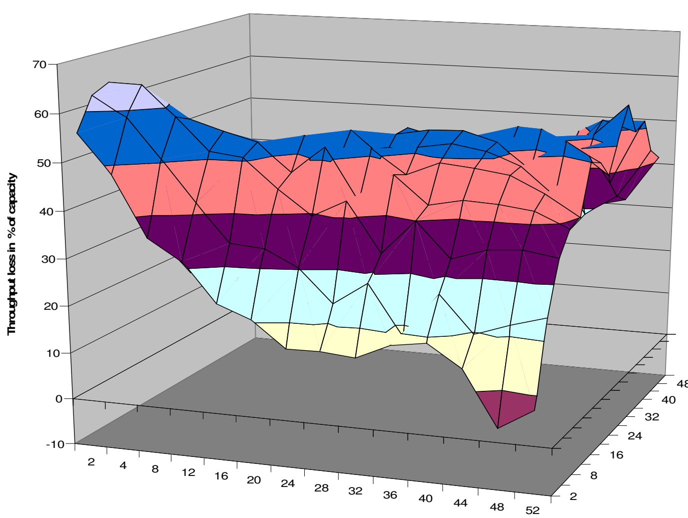
ECM



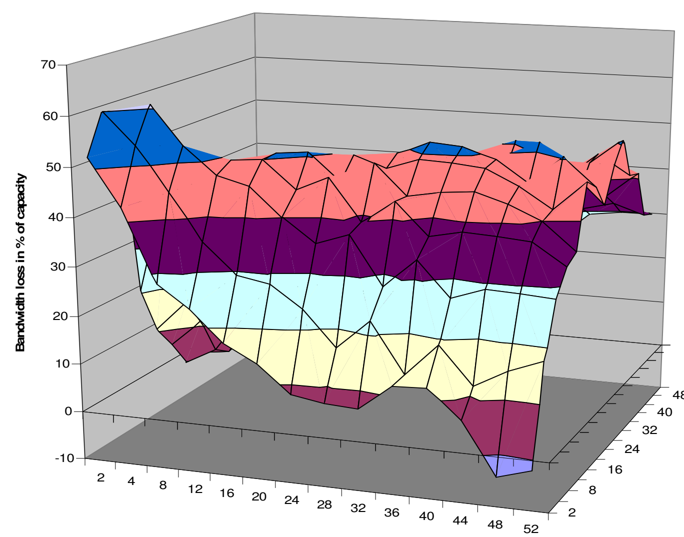
QCN-P



QCN



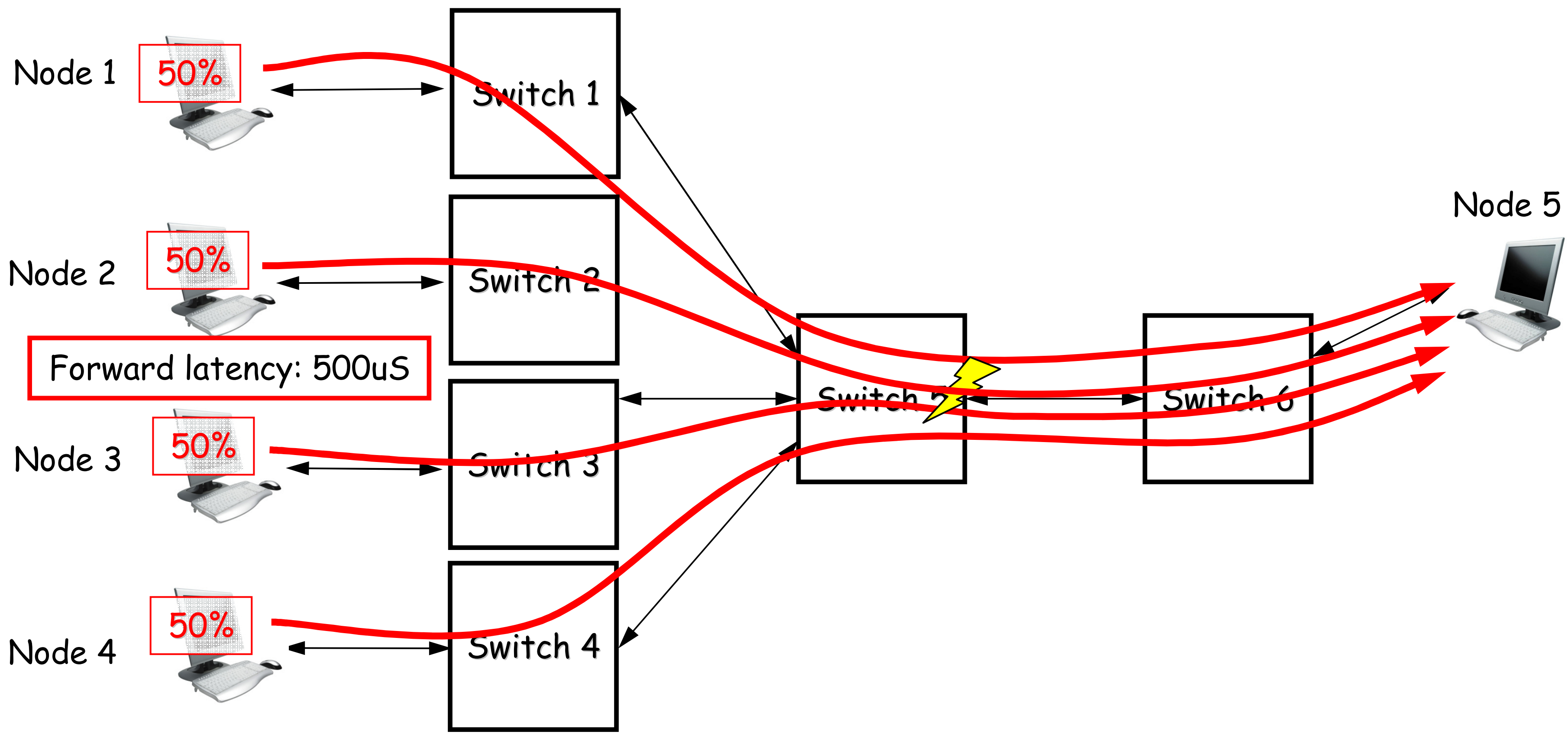
QCN-FbHat



Observations

- QCN does not perform well with recurring OG hotspots
- ECM has best performance
  - Due to tagging, positive feedback is almost immediate
- Results for QCN-P and ECM-P not as good as ECM, but acceptable

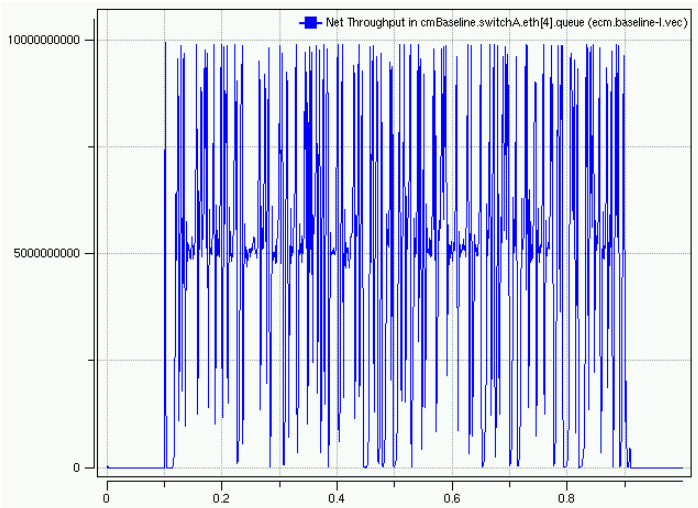
Symmetric Topology, Single HS, Large Forward Latency



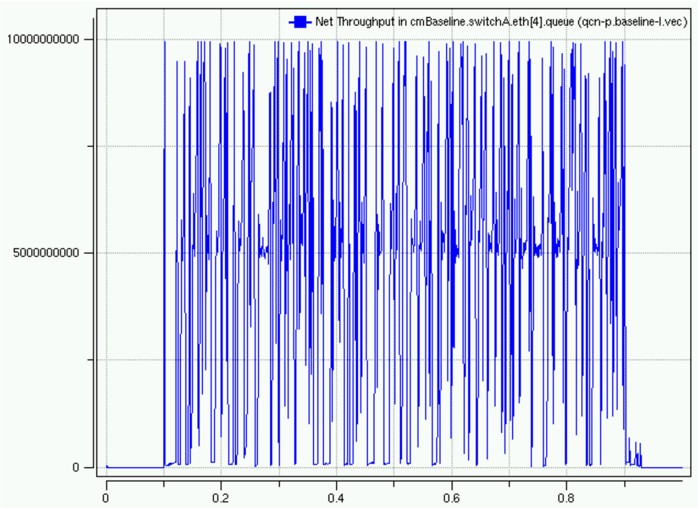
- Node 1 to 4 sending at 50% load to node 5
- Forward latency from Node 1..4 to switch: 500uS
- Simulation runtime 1s, with load from 0.1s to 0.9s
- Real world scenario: large number of hops and/or switches with large buffers in path to CP

500uS Latency: Throughput at Hotspot

ECM



QCN-P

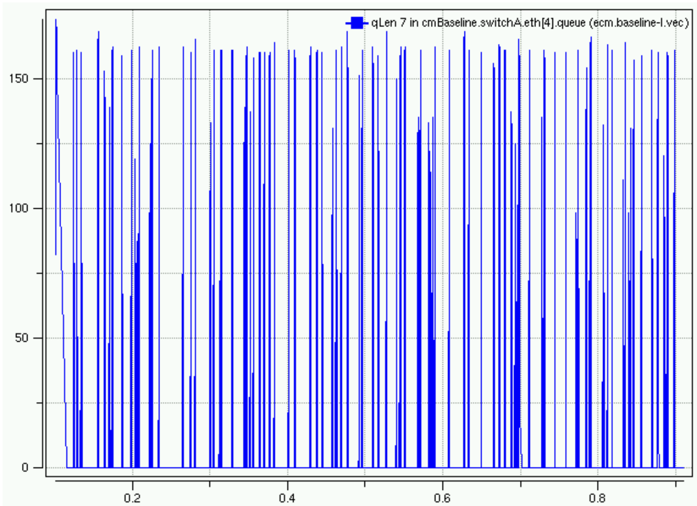


QCN

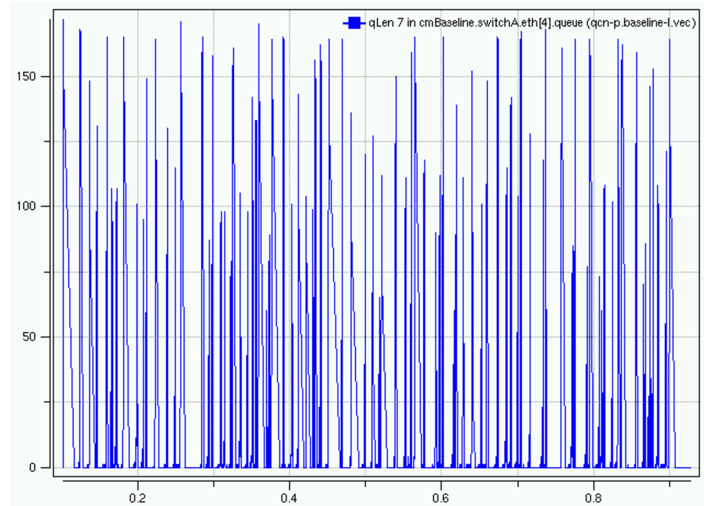
QCN-H

500uS Latency: Queue Length at Hotspot

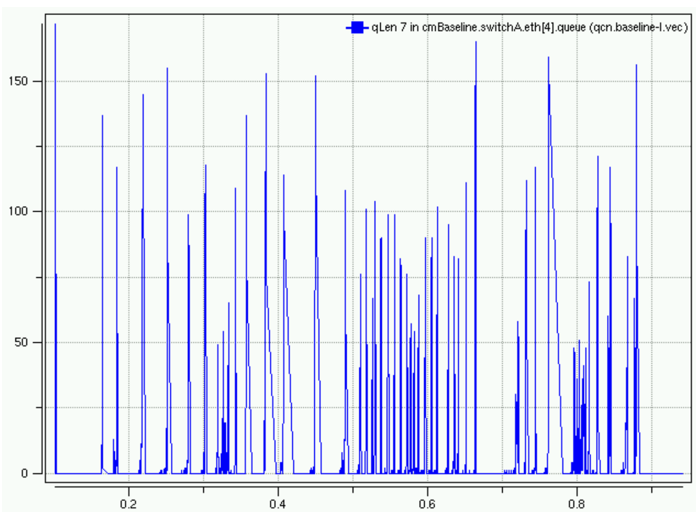
ECM



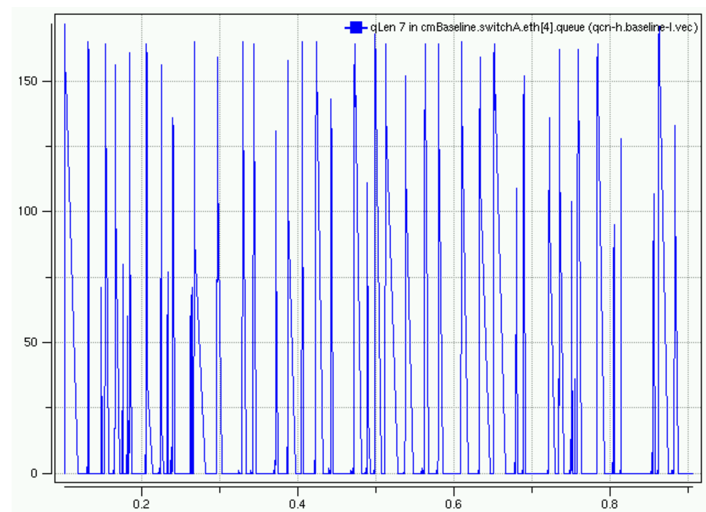
QCN-P



QCN



QCN-H

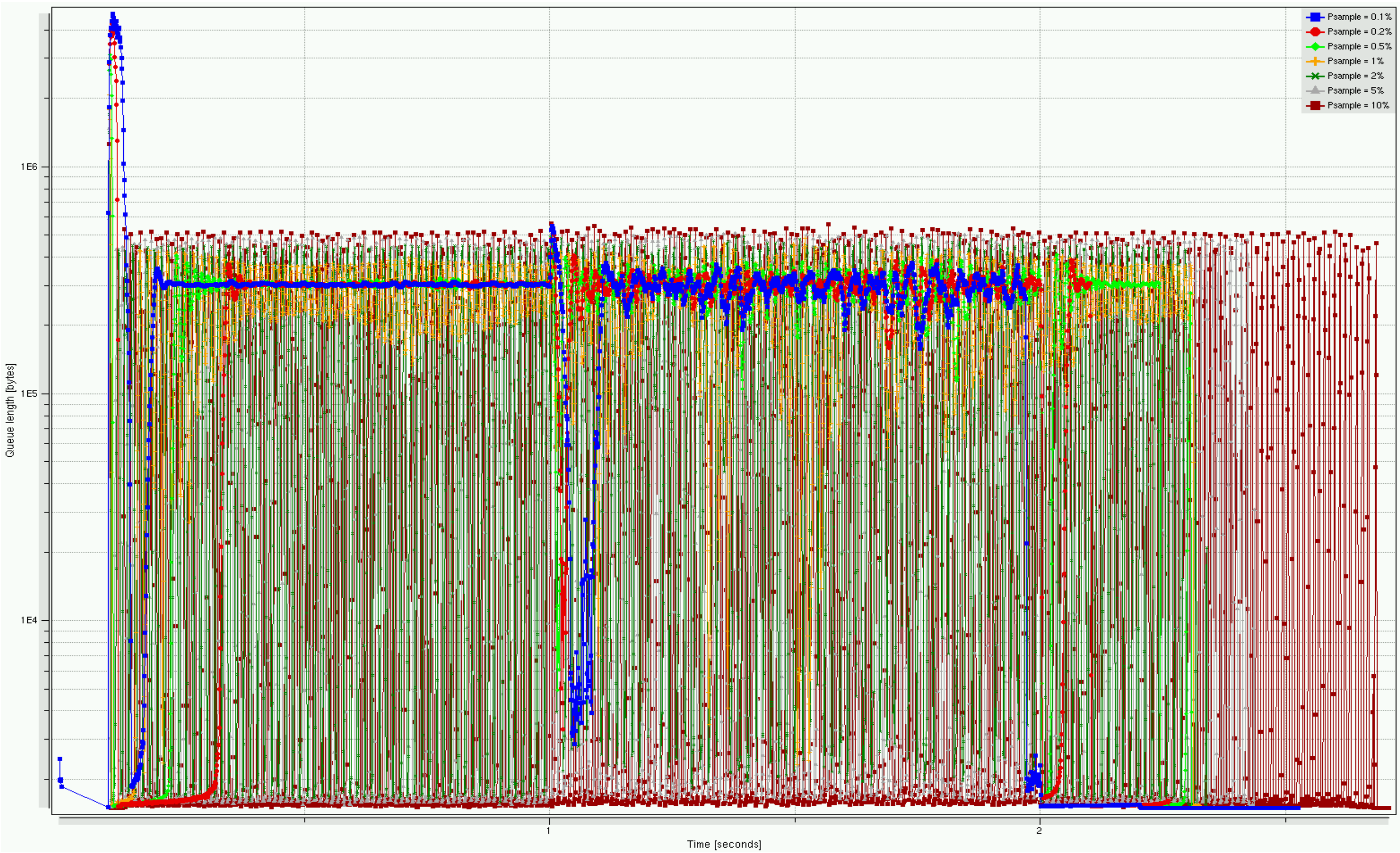


Observations

- Results unacceptable for all protocols
- Hypothesis
  - Protocols may fail if CP sends multiple Congestion Notifications before the impact of the first one is noticed, causing oscillations
  - With sampling probability p=1% (~150 kBytes), this would be around 120uS
  - To confirm, Cyriel ran simulations with p=<0.1%..10%>
    - Baseline scenario
    - Oversampling disabled
    - RTT=200uS
    - Switch buffer size 1.2 Mbytes
    - Qeq=300 kBytes

200 uS Latency, Queue Length, ECM

200uS Latency, Queue Length, QCN

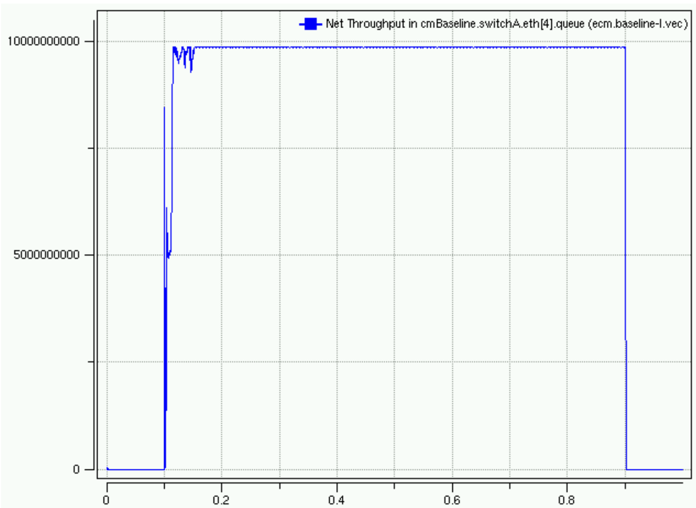


Conclusions

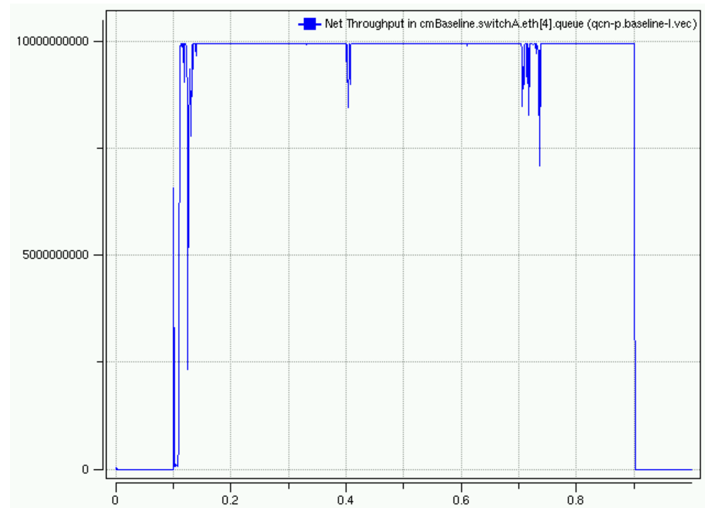
- Queue length unstable with psample > 0.5%
- Matches expected failure point of 250 kBytes
  - RTT = 200uS → 200uS * 10gBit/s = 2 Mbit = 250 kBytes
- Re-tested with 100uS latency, default parameters
  - RTT = 100uS → 100uS * 10gBit/s = 1 Mbit = 125 kBytes
- Re-tested with QCN and QCN-FbHat
  - Increase W with larger RTT and disable Hyperactive Increase (per Balaji's suggestion)
- Re-tested with ECM
  1) Optimize parameters for RTT
  2) Drop RL packets at RP if receive interval < RTT
     - In other words, accept only one RL packet per RTT from the same CPID

100uS Latency: Throughput at Hotspot

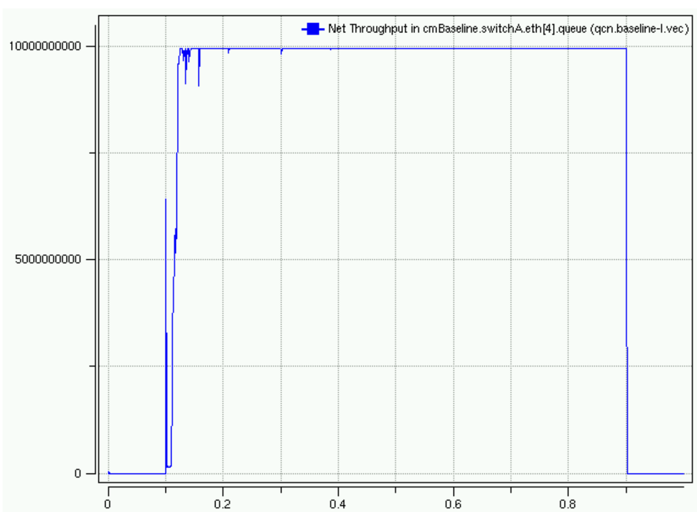
ECM



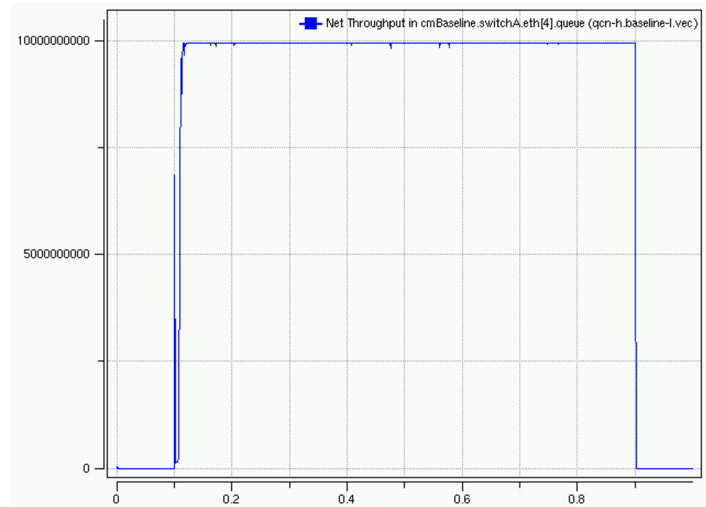
QCN-P



QCN

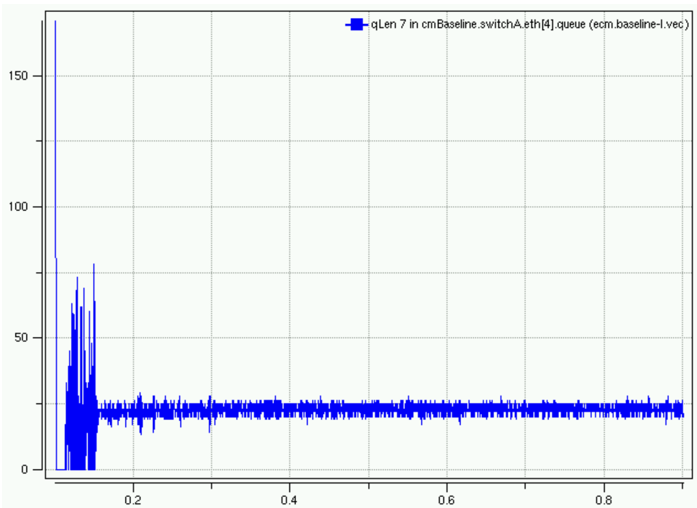


QCN-H

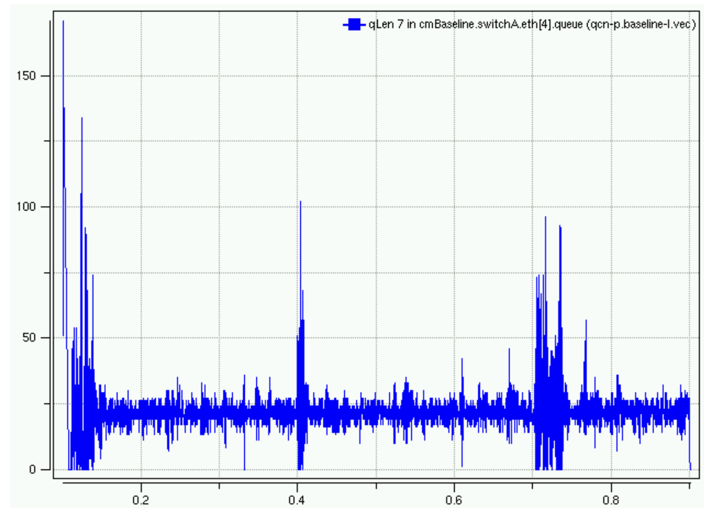


100uS Latency: Queue Length at Hotspot

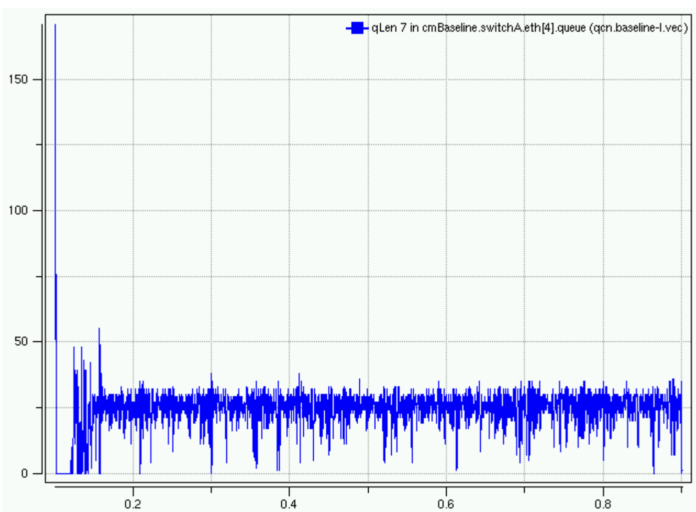
ECM



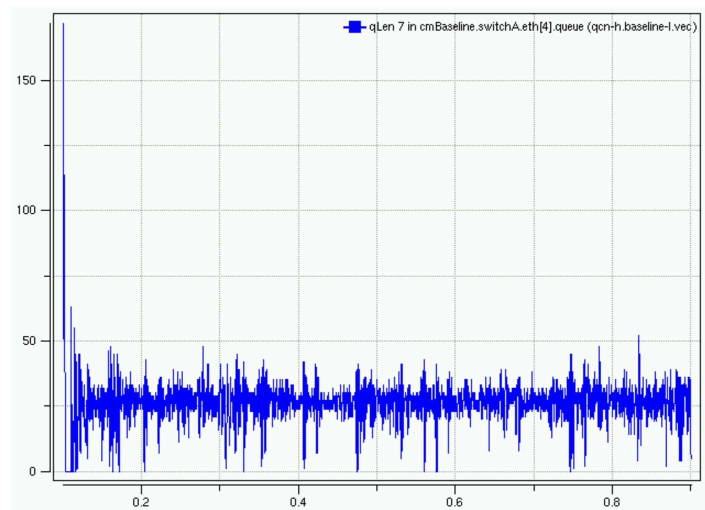
QCN-P



QCN

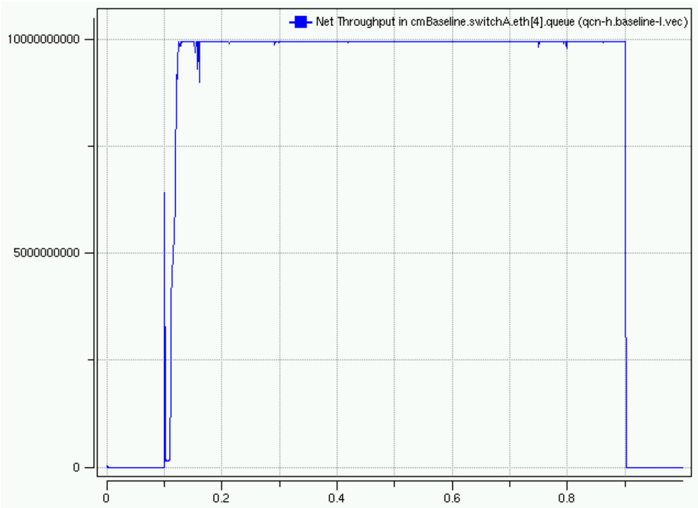


QCN-H

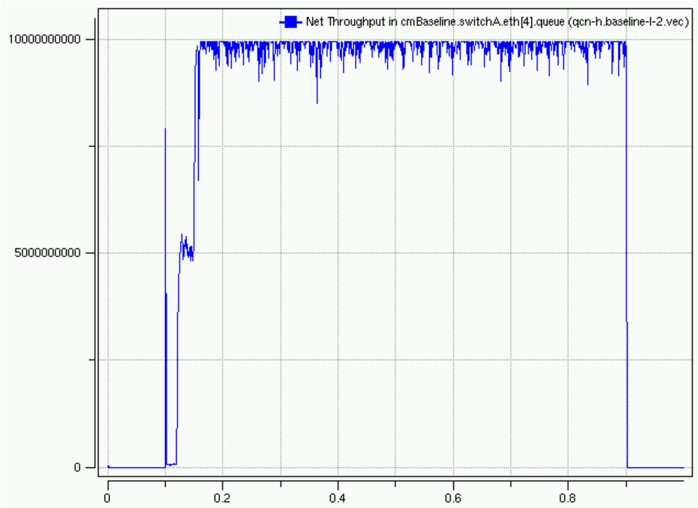


QCN-FbHat: Throughput at Hotspot, W=2.0, no HAI

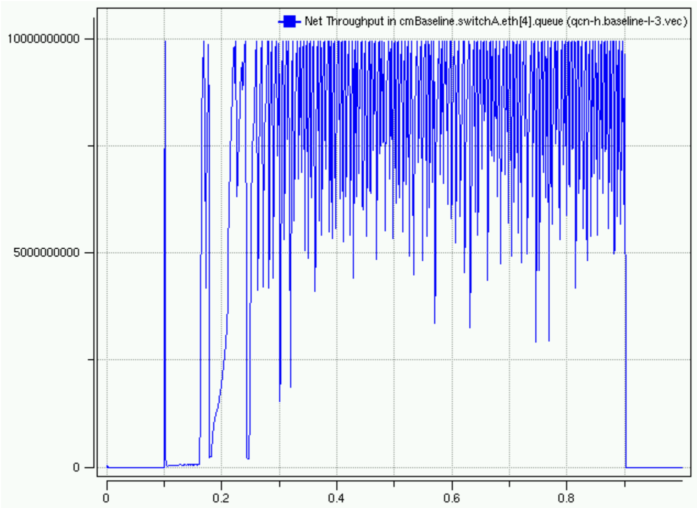
RTT=100uS



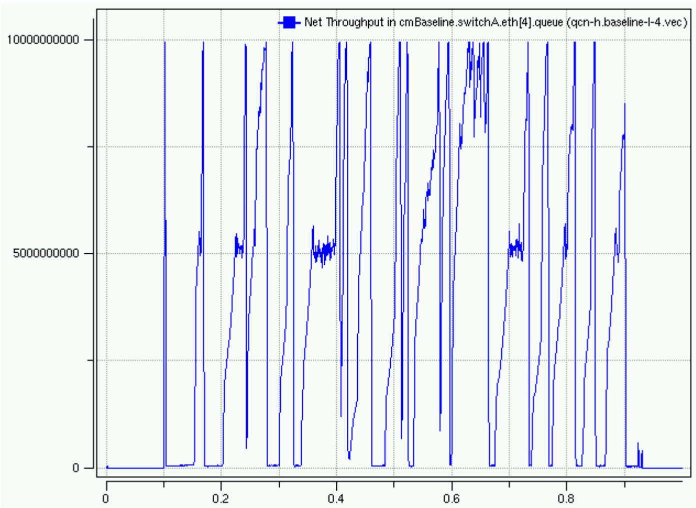
RTT=200uS



RTT=500uS

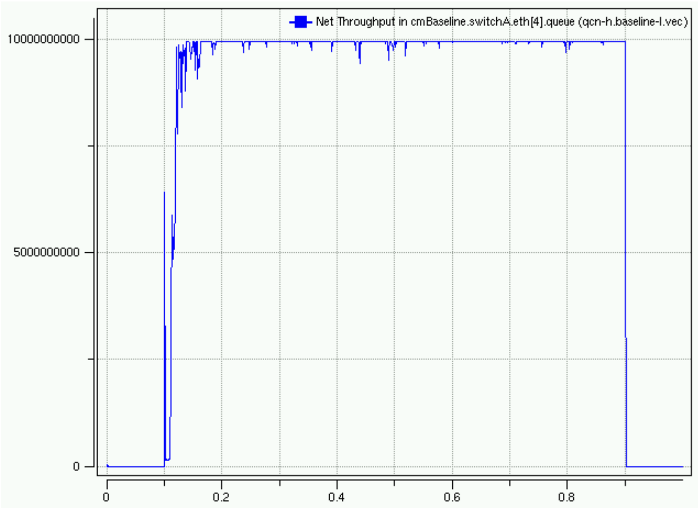


RTT=1mS

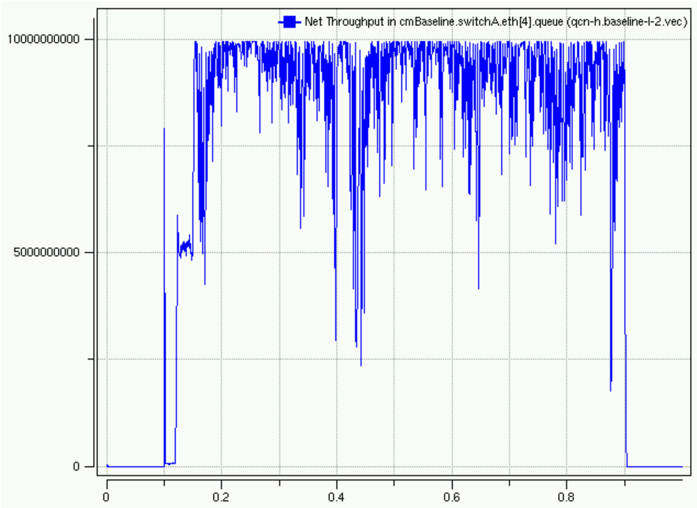


QCN-FbHat: Throughput at Hotspot, W=2.0, HAI

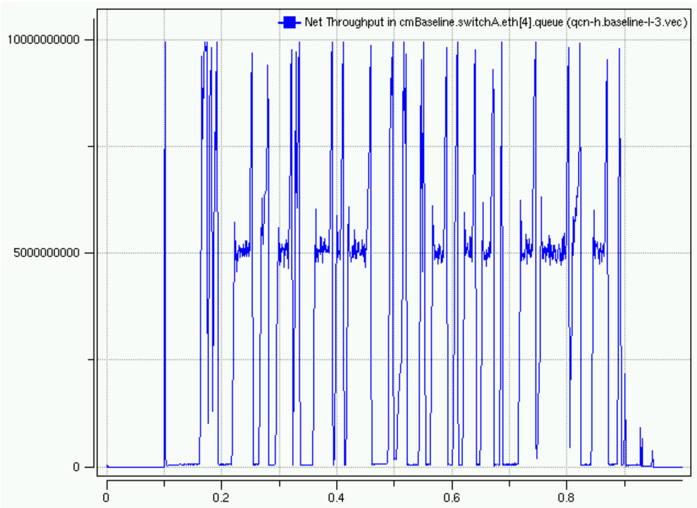
RTT=100uS



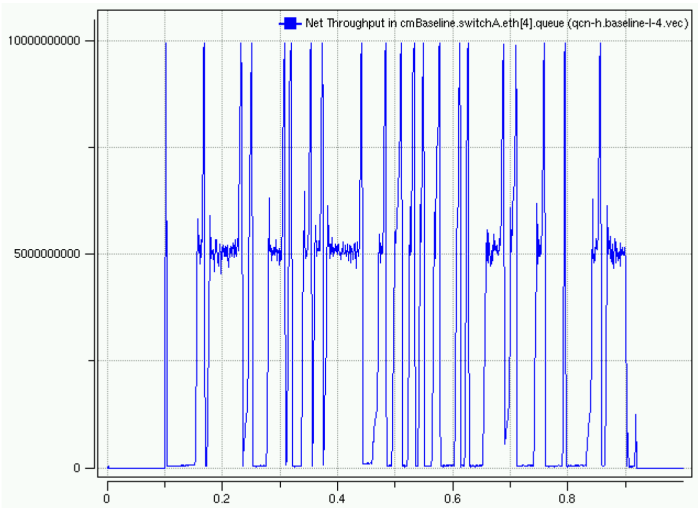
RTT=200uS



RTT=500uS

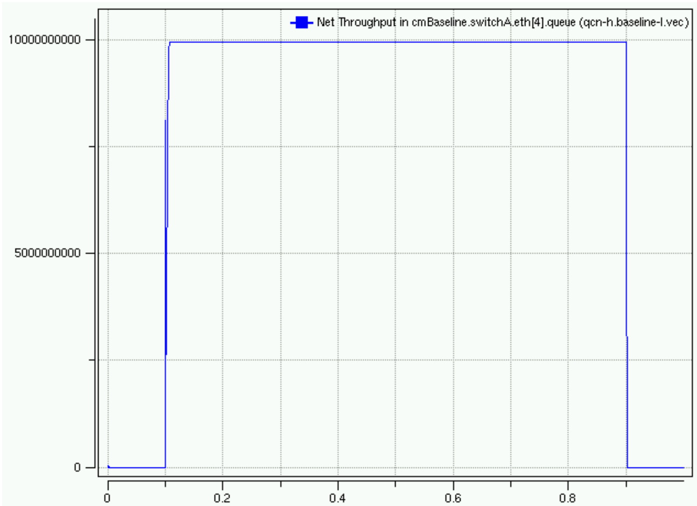


RTT=1mS

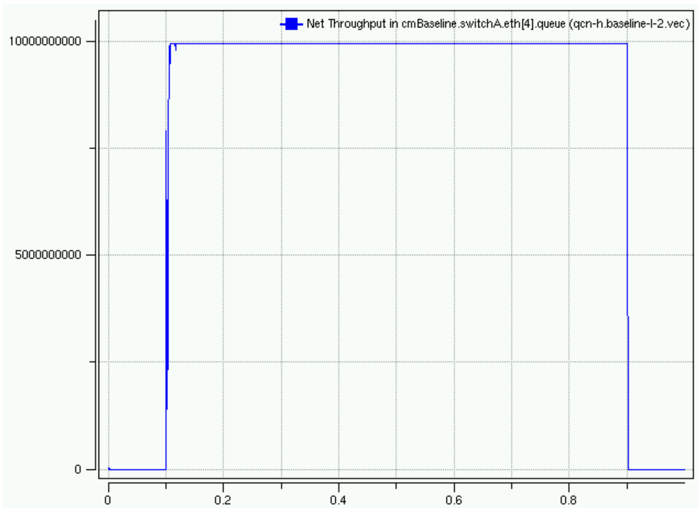


QCN-FbHat: Throughput at Hotspot, W=var, No HAI

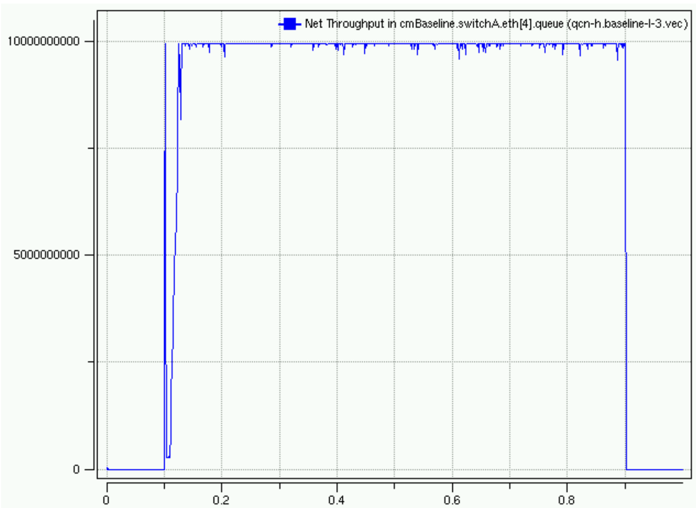
RTT=100uS, W=4.0



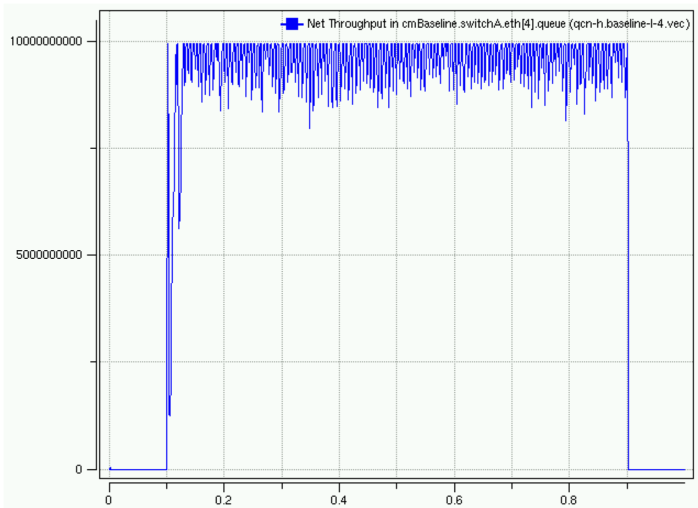
RTT=200uS, W=6.0



RTT=500uS, W=10.0

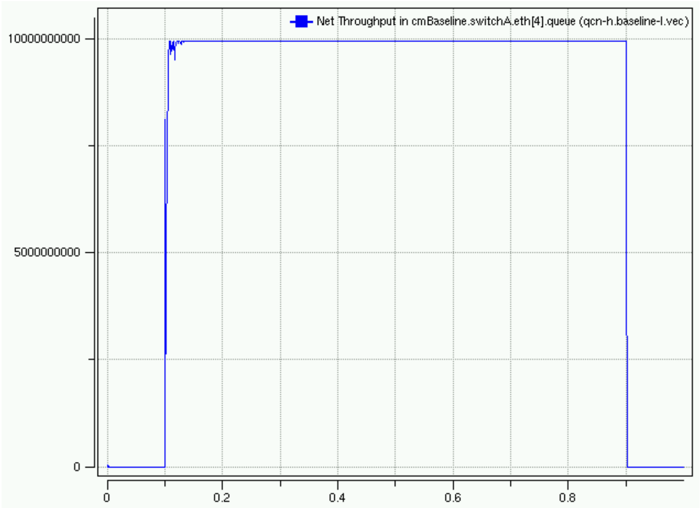


RTT=1mS, W=20.0

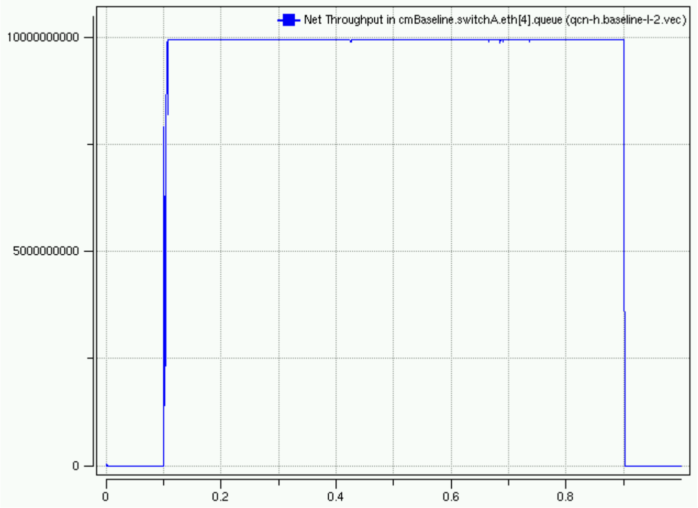


QCN-FbHat: Throughput at Hotspot, W=var, HAI

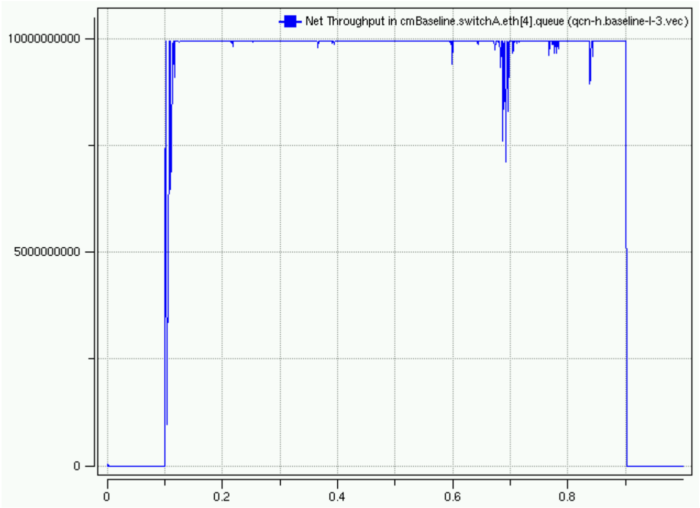
RTT=100uS, W=4.0



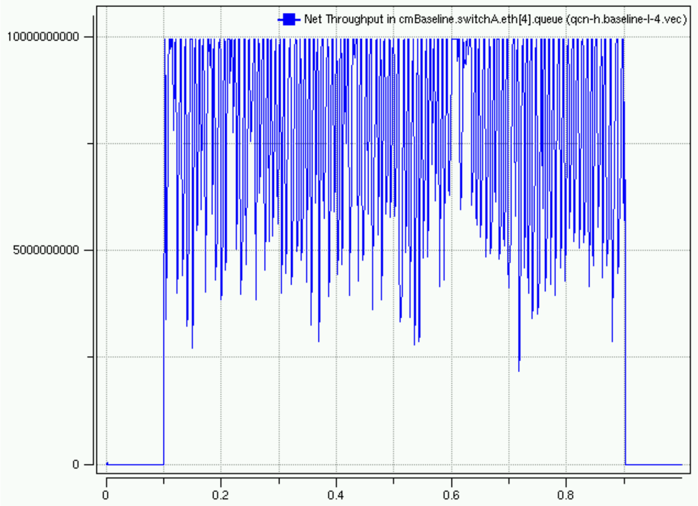
RTT=200uS, W=6.0



RTT=500uS, W=20.0

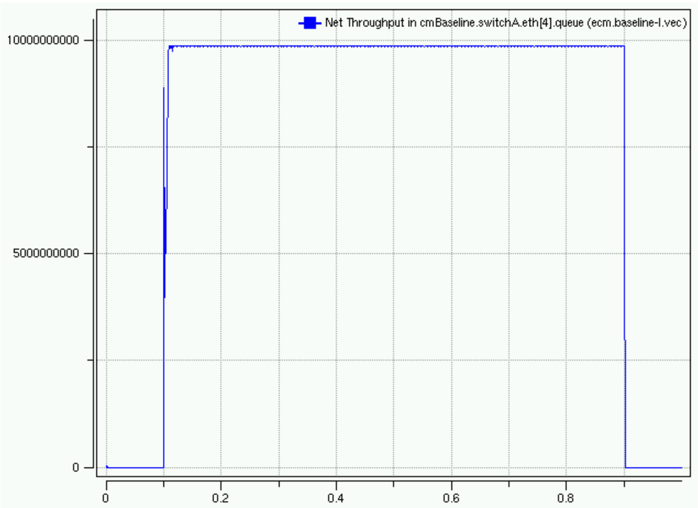


RTT=1mS, W=40.0

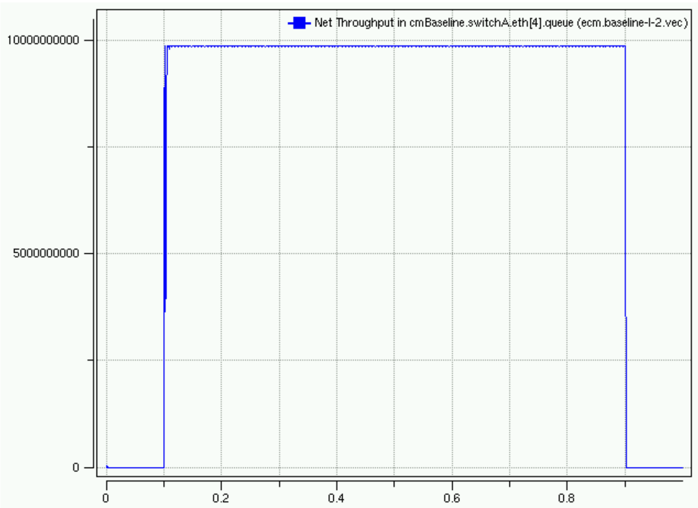


ECM: Throughput at Hotspot, No RTT triggered CM drops

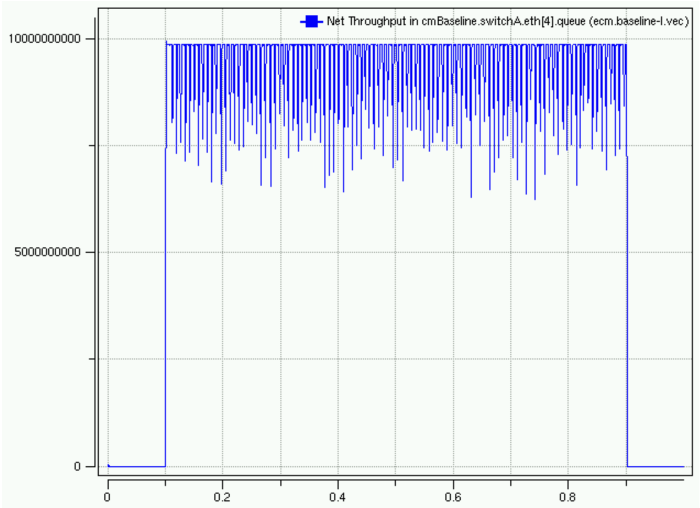
RTT=100uS: W=2.5, Gi=0.5, Gd=0.00010



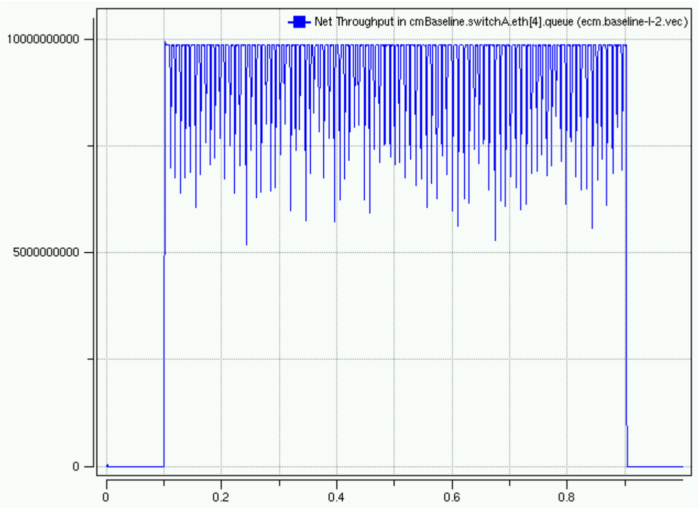
RTT=200uS: W=4.0, Gi=0.5, Gd=0.00003



RTT=500uS: W=0.5, Gi=0.5, Gd=0.00001

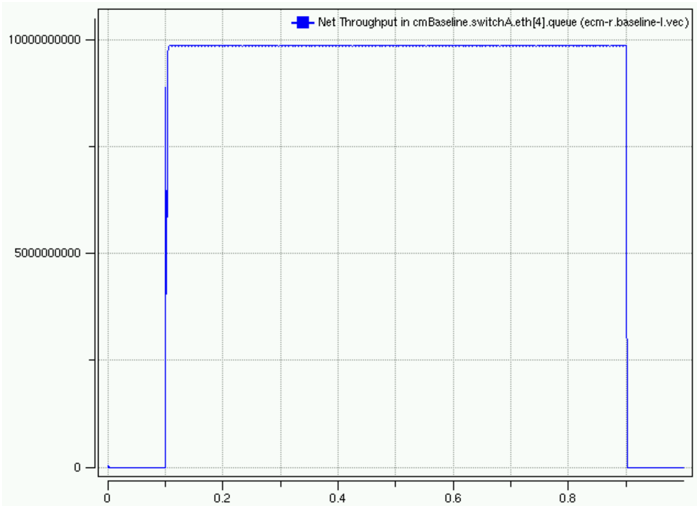


RTT=1mS: W=0.5, Gi=0.5, Gd=0.00001

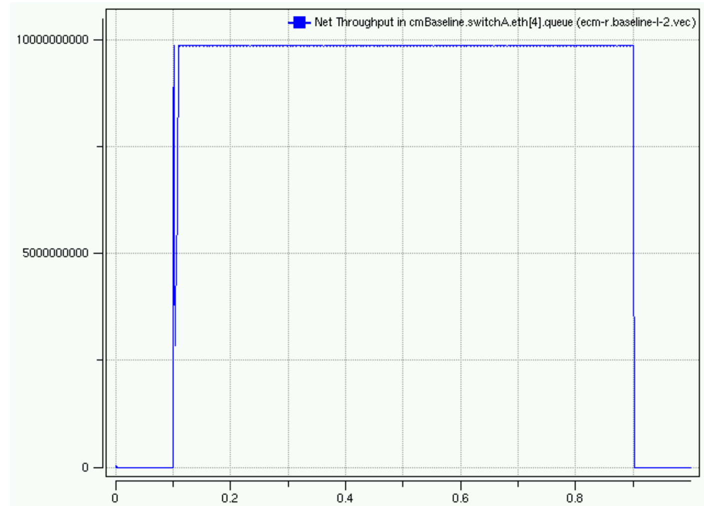


ECM: Throughput at Hotspot, RTT triggered CM drops

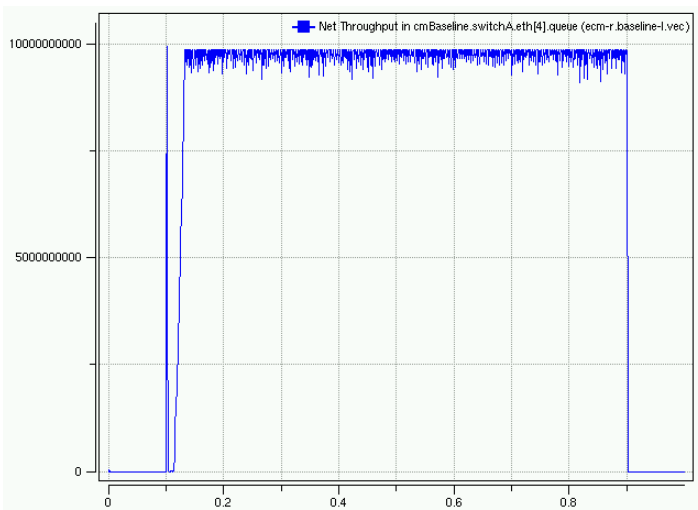
RTT=100uS: W=2.5, Gi=0.4, Gd=0.00015



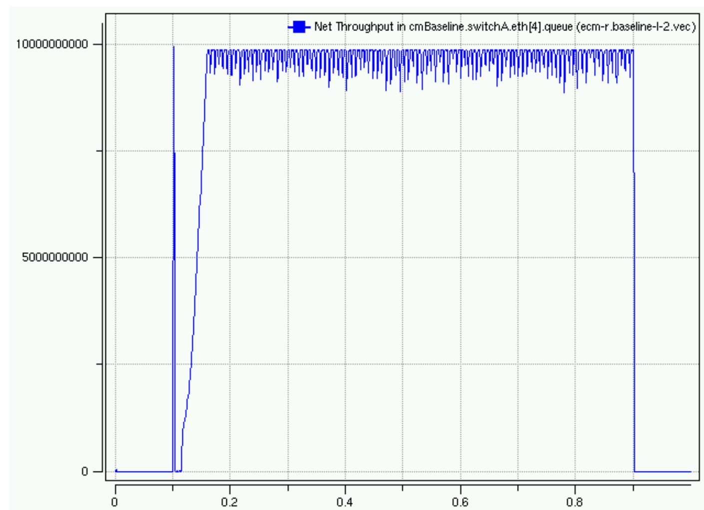
RTT=200uS: W=3.5, Gi=0.3, Gd=0.00004



RTT=500uS: W=3.5, Gi=0.3, Gd=0.00010



RTT=1mS: W=5.0, Gi=0.2, Gd=0.00006



ECM Stability Margins (+/-1% performance)

- No RTT based CM drops
  - RTT=200uS
    - Gi = <0.2 .. **0.5**>
    - Gd = <**0.00003** .. 0.00008>
    - W = <3.0 .. **4.0** .. 6.0>
  - RTT=500uS
    - Gi = <0.2 .. **0.5**>
    - Gd = 0.00001
    - W=0.5
  - RTT=1ms
    - Not calculated
- With RTT based CM drops
  - RTT=200uS
    - Gi = <0.2 .. **0.3** .. 0.45>
    - Gd = <0.00002 .. **0.00004** .. 0.00017>
    - W = <2.5 .. **3.5**>
  - RTT=500uS
    - Gi = <0.2 .. **0.31** .. 0.4>
    - Gd = <0.00005 .. **0.00006** .. 0.00010>
    - W = <**2.5** .. 3.5>
  - RTT=1ms
    - Gi = <**0.2** .. 0.3>
    - Gd = <0.00006 .. **0.00008**>
    - W = <**5.0**>

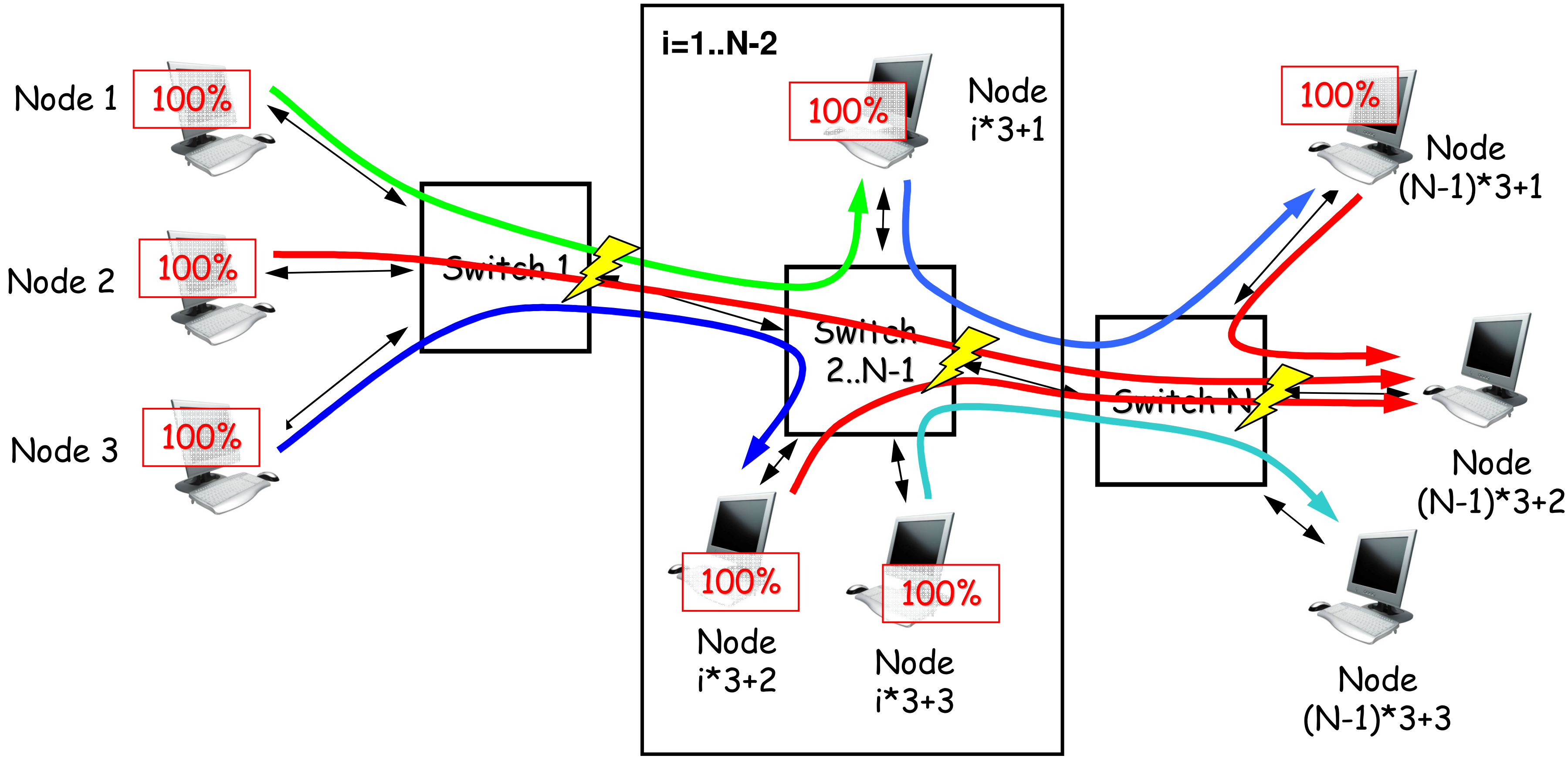
Observations and Conclusions

- QCN, QCN-FbHat
  - Increasing W improves its stability with large latencies
  - Less stable with Hyperactive Increase enabled
  - Optimal value of W depends on RTT
  - Increasing W affects reaction time in OG hotspot scenarios
    - Test 1: W: 2.0 -> 4.0 => Reaction time 7ms -> 60ms
- ECM
  - Larger latency requires CM message drop triggered by RTT to maintain stability
  - Margin for Gi, Gd reduced as RTT gets larger
  - Optimal value of W depends on RTT
    - But no strong relationship between RTT and W as with QCN

➢ Support for large latencies (> 100uS) requires RTT dependent operation and parameter optimizations

➢ Needs further study

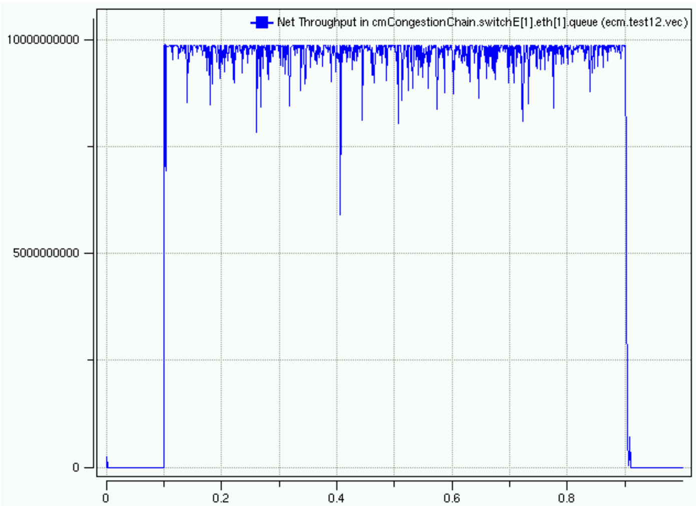
20-stage Hotspot with bursty load



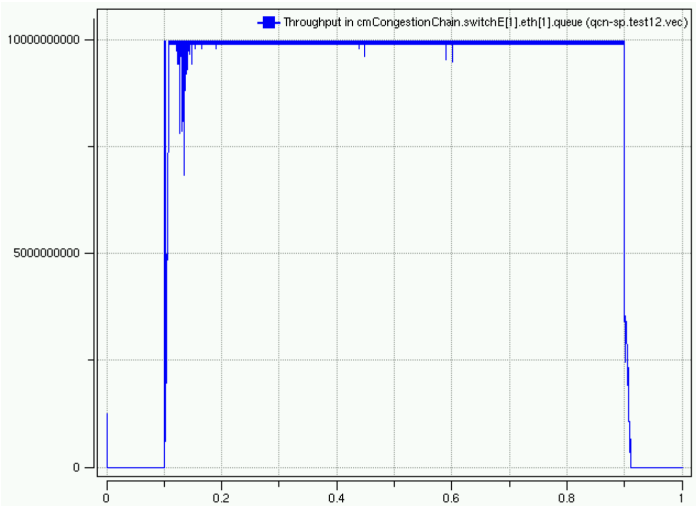
- N=18 switches; 3 hosts per switch
- Node <i> sends to node <i+3>; Node <i+1> sends to node (N-1)*3+2; node <i+2> sends to node <i+4>
- Node <1,4,7,...> sends bursty traffic with interval 1 + <i>*0.1 ms
- 100% load from all nodes
- Node (N-1)*3+2 receives traffic from <N> sources
- N hotspots

20-stage hotspot: Throughput at last hotspot

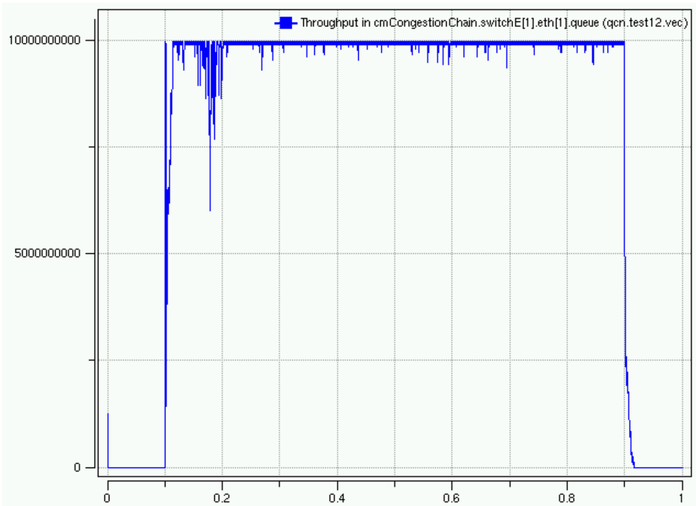
ECM



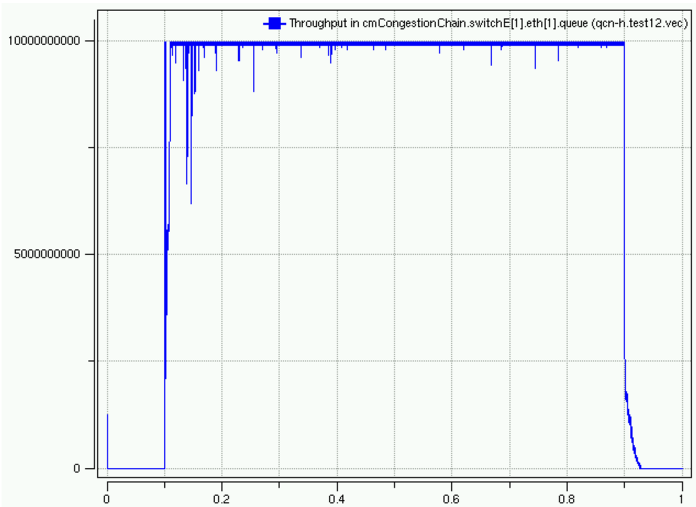
QCN-SP



QCN

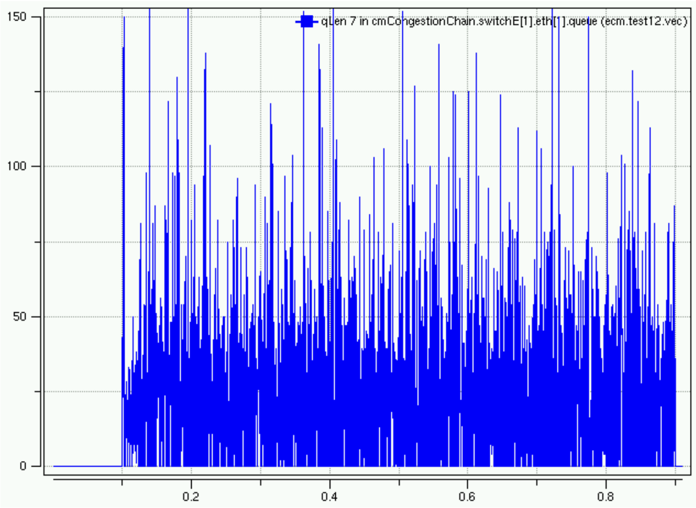


QCN-H

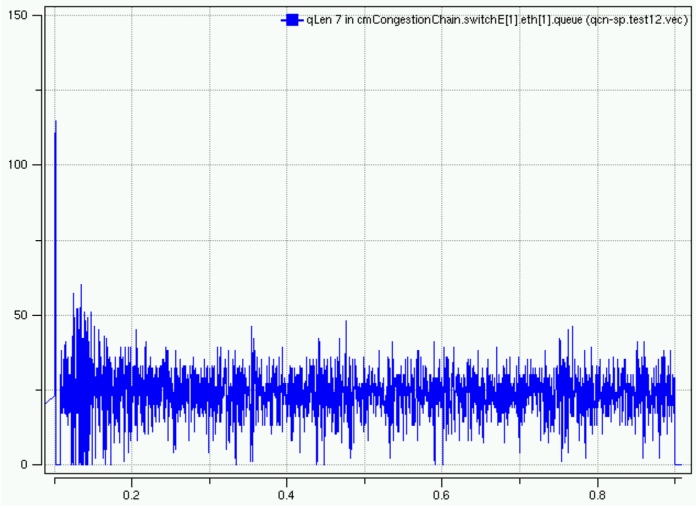


20-stage hotspot: Queue length at last hotspot

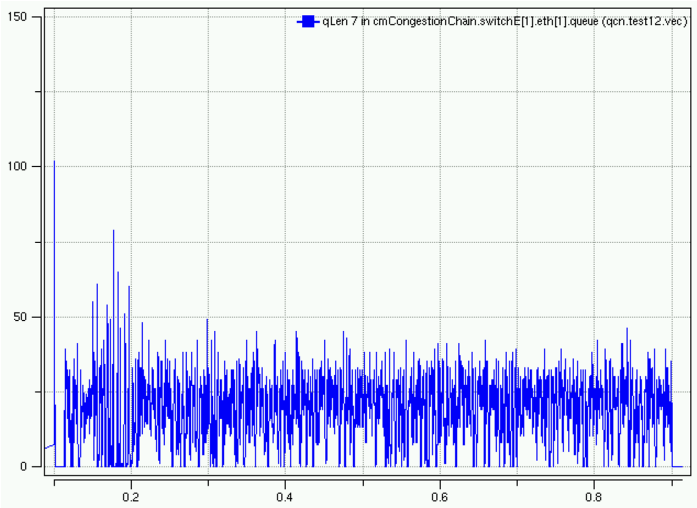
ECM



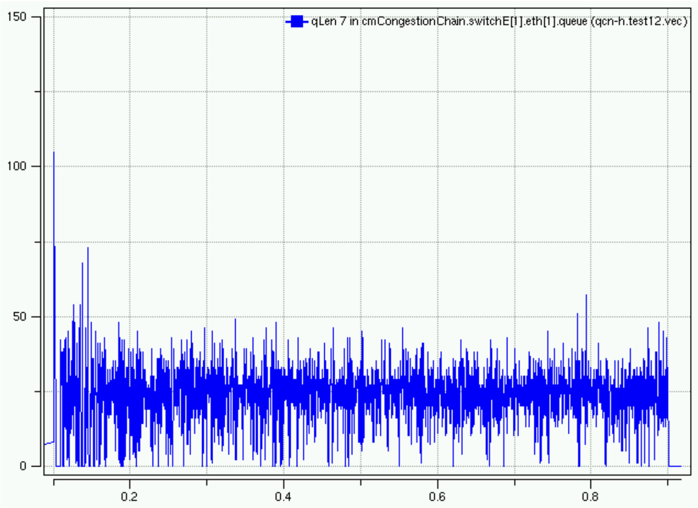
QCN-SP



QCN

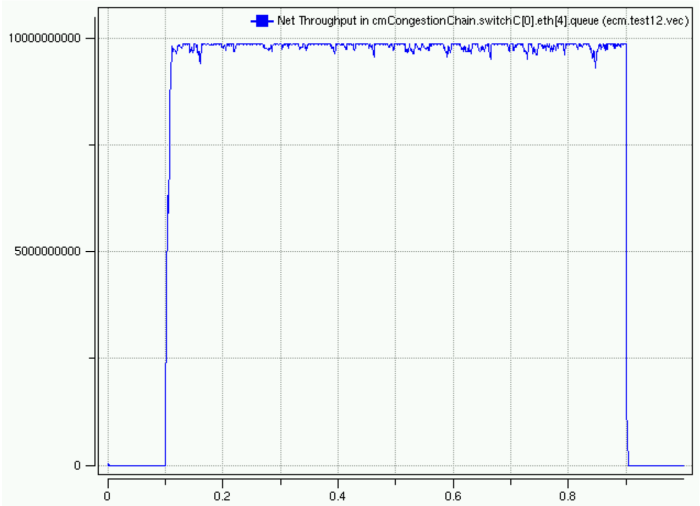


QCN-H

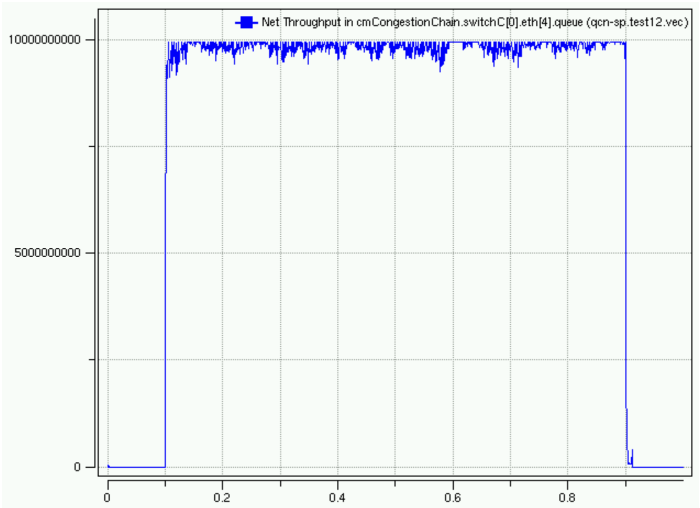


20-stage hotspot: Switch 2 Throughput

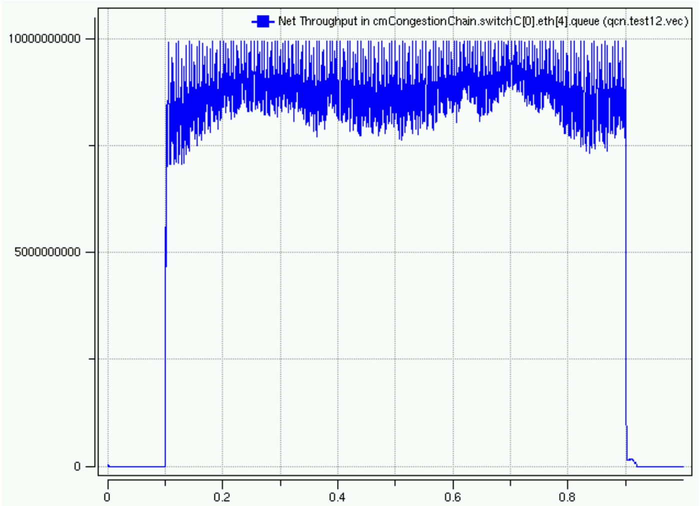
ECM



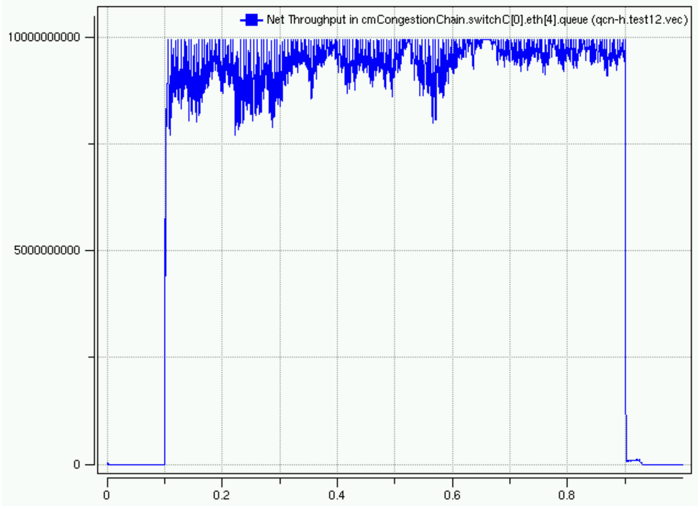
QCN-SP



QCN

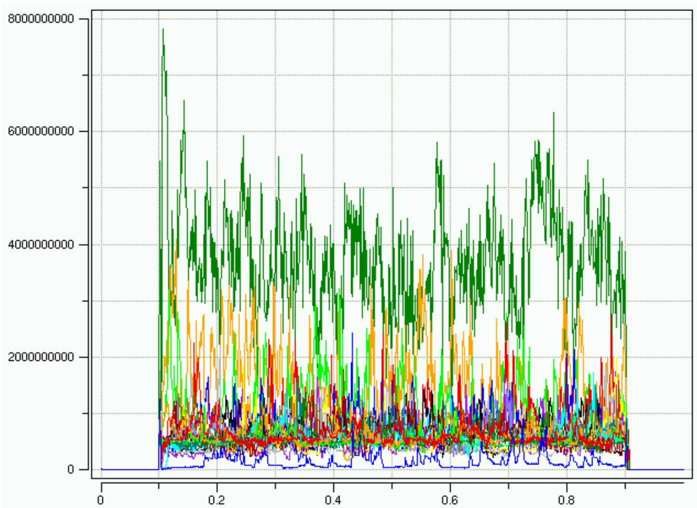


QCN-H

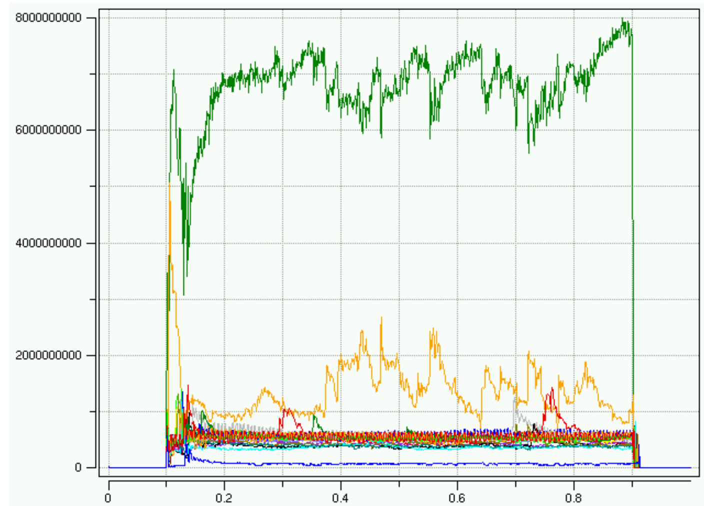


20-stage hotspot: Per-Flow Throughput

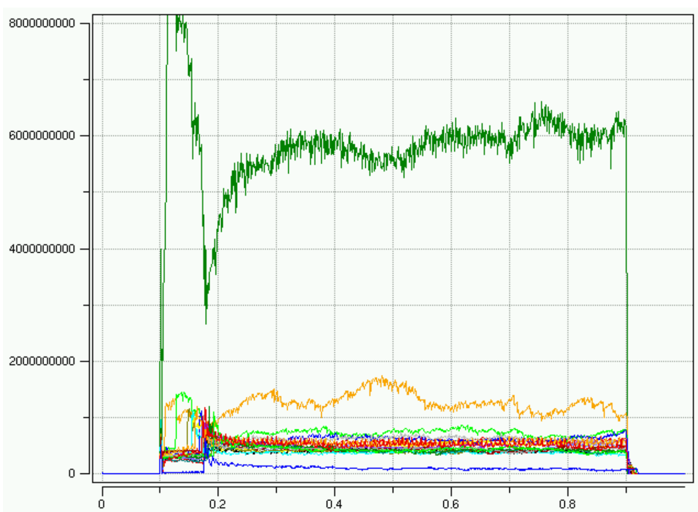
ECM



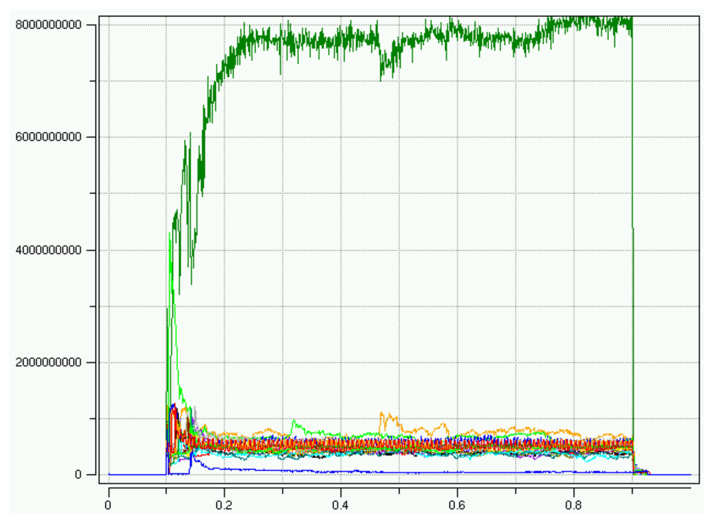
QCN-SP



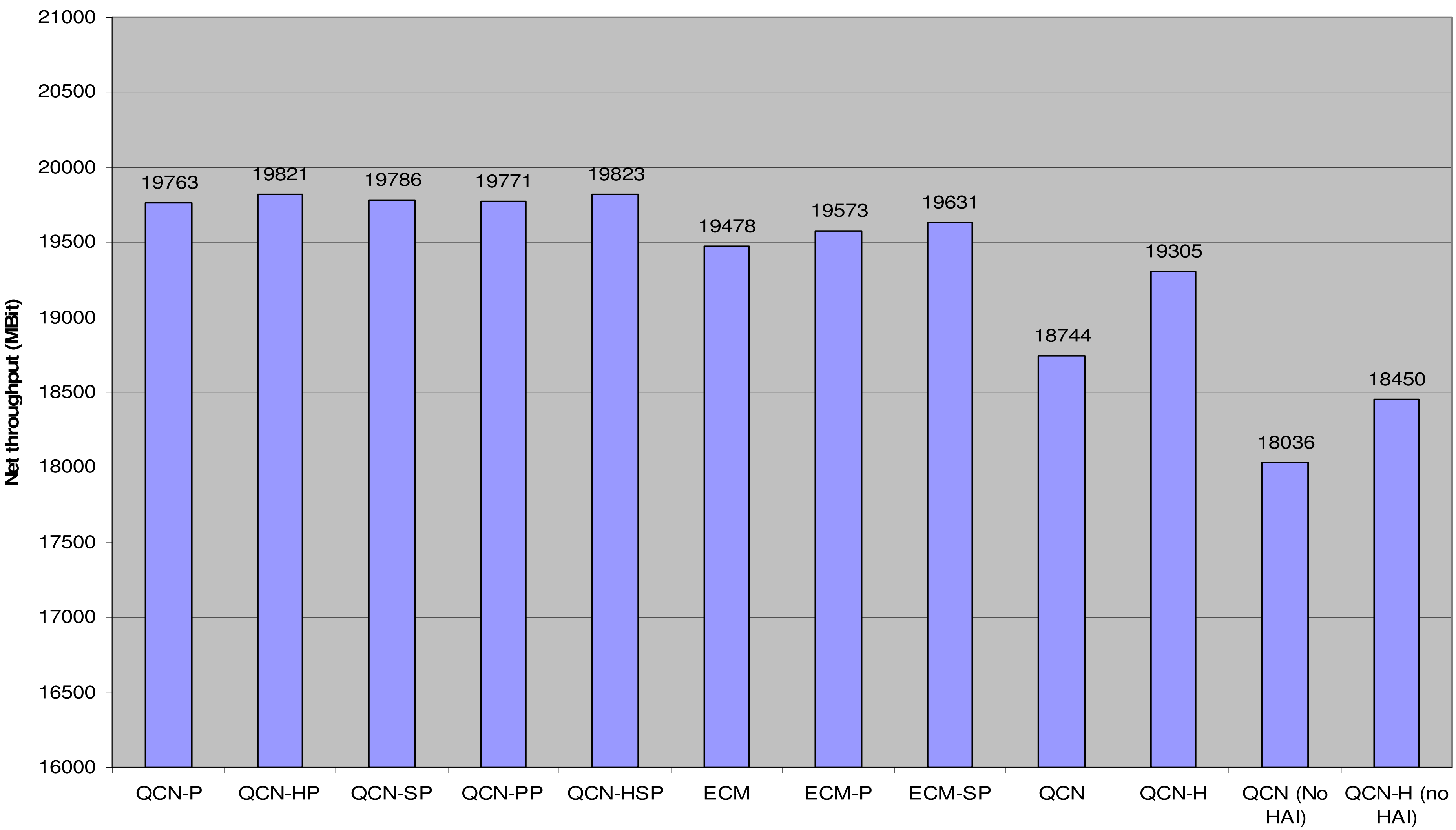
QCN



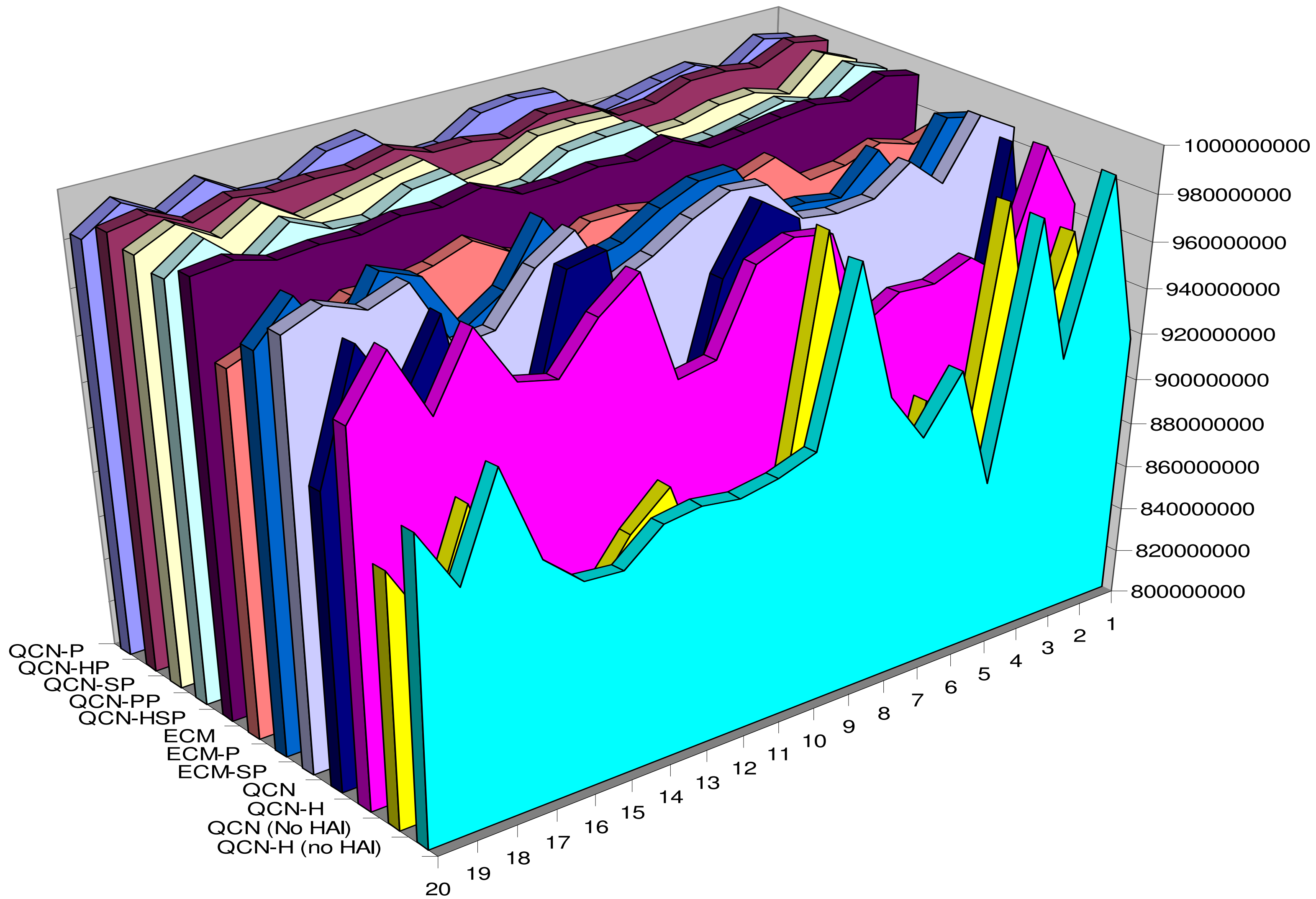
QCN-H



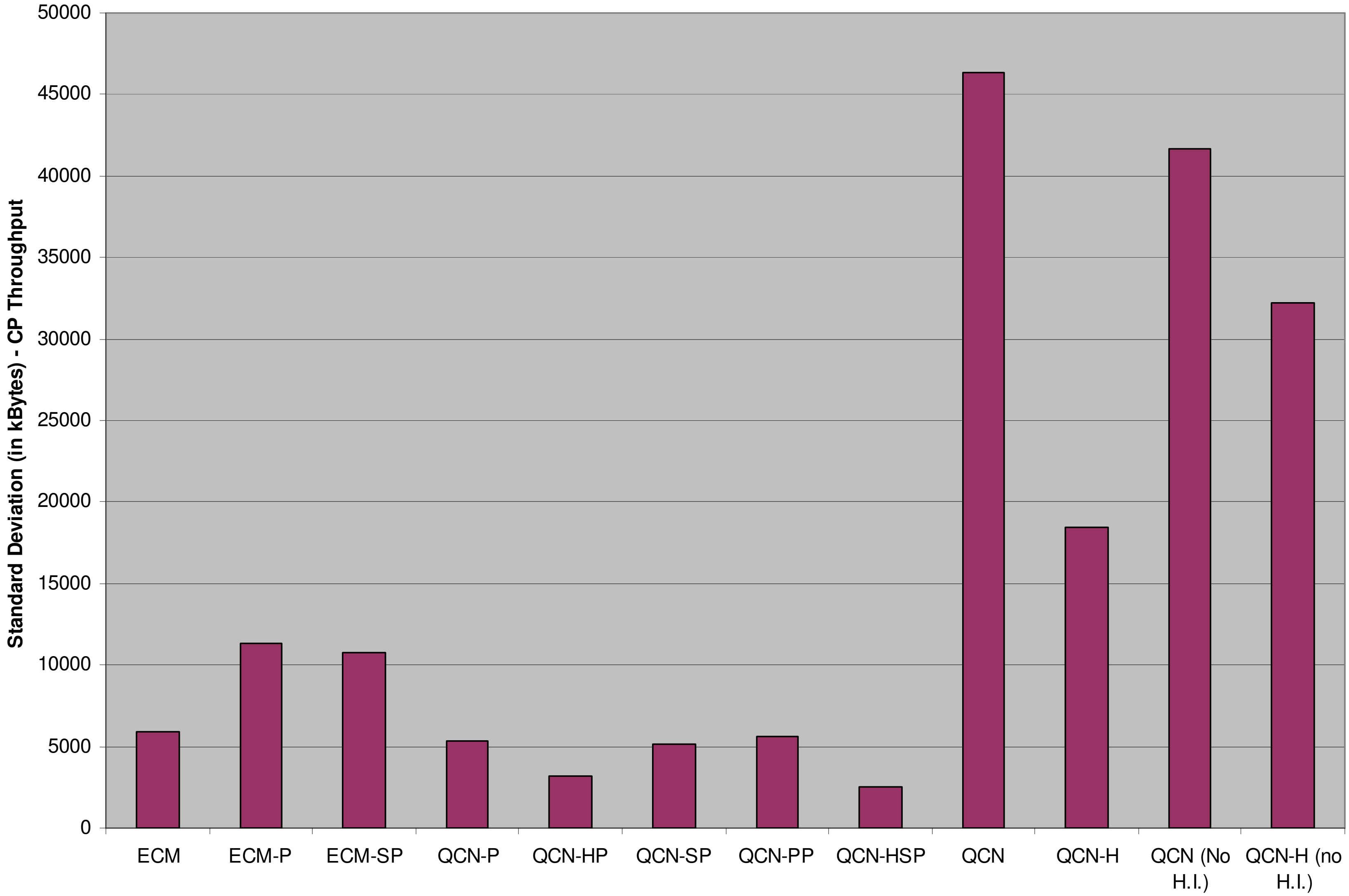
20-stage hotspot: Total Throughput through all hotspots



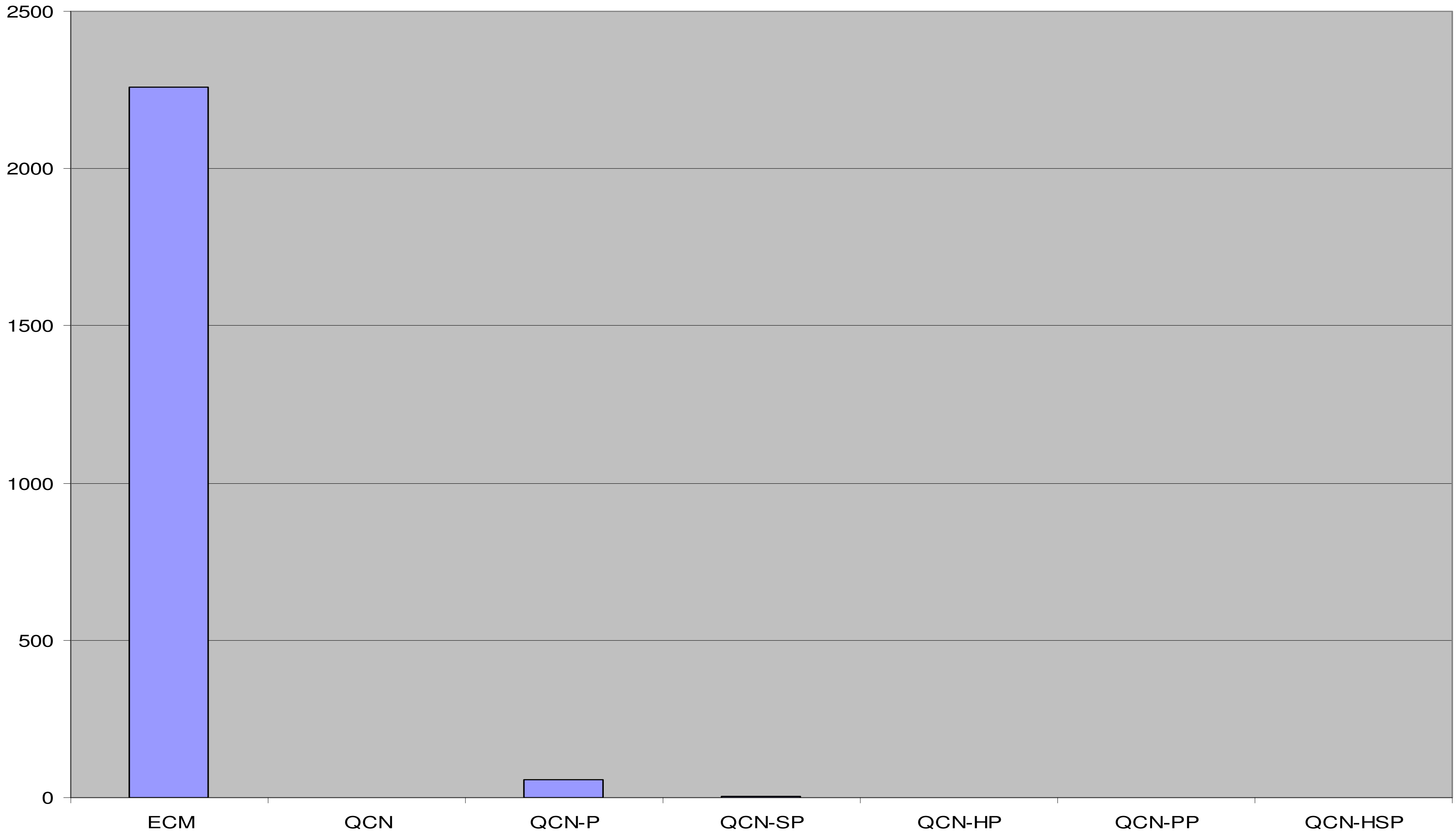
20-stage Hotspot: Throughput per switch



20-stage Hotspot: CP Throughput Standard Deviation



False Positives (Positive feedback from unexpected CP)



Observations

- QCN and derivatives
  - Use of FbHat shows improved performance
    - Still well below protocols utilizing positive feedback
  - High throughput Standard Deviation for QCN and QCN-FbHat
    - Throughput across switches somewhat unbalanced
- ECM
  - Tagging reduces net throughput
  - Large number of false positives
    - Feedback can be positive even if switch believes it to be negative
    - CPID association changes after tagged packet was sent
  - Large queue length jitter in last switch
    - Maybe because RTT exceeds acceptable limit for some flows
  - Overall performance still better than with QCN and QCN-H

Observations - continued

- Did not observe effects of CPID Thrashing
- Best overall 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

Summary and Conclusions

- Closed-loop protocols perform better in all test cases
- Specific concerns
  - Excessive RL creation with QCN in non-congested conditions
    - QCN specific
  - Slow recovery of Open Loop protocols in OG hotspot scenarios
  - Protocol performance in large latency environments depends on RTT

➢ Closed-loop protocol required
  - To achieve acceptable performance in OG hotspot scenarios
    - Faster recovery due to positive feedback
  - To improve performance with large latencies
    - Enables RTT calculation and RTT based adjustments

Teak simulation code access

- OMNET++
  - Download from www.omnetpp.org
- INET framework
  - git access (linux):

```
git clone git://teaktechnologies.com/var/git/INET.git INET
cd INET
git checkout –b my_branch origin/teak
```

Thank You

Backup Slides

Simulation Parameters

- Traffic
  - Bernoulli
  - 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

Simulation Parameters - 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%

Simulation Parameters - ECM

- Qeq = 375
- Qsc = 1600
- Qmc = 2400
- Qsat disabled
- Gi = 0.53333
- Gd = 0.00026667
- Ru = 1000000
- Rd = 1000000
- Td = 1ms
- Rmin = 1000000
- W = 2.0
- samplingInterval = 150000