QCN: Algorithm for P-code

Abdul Kabbani, Rong Pan,
Balaji Prabhakar and Mick Seaman

Outline

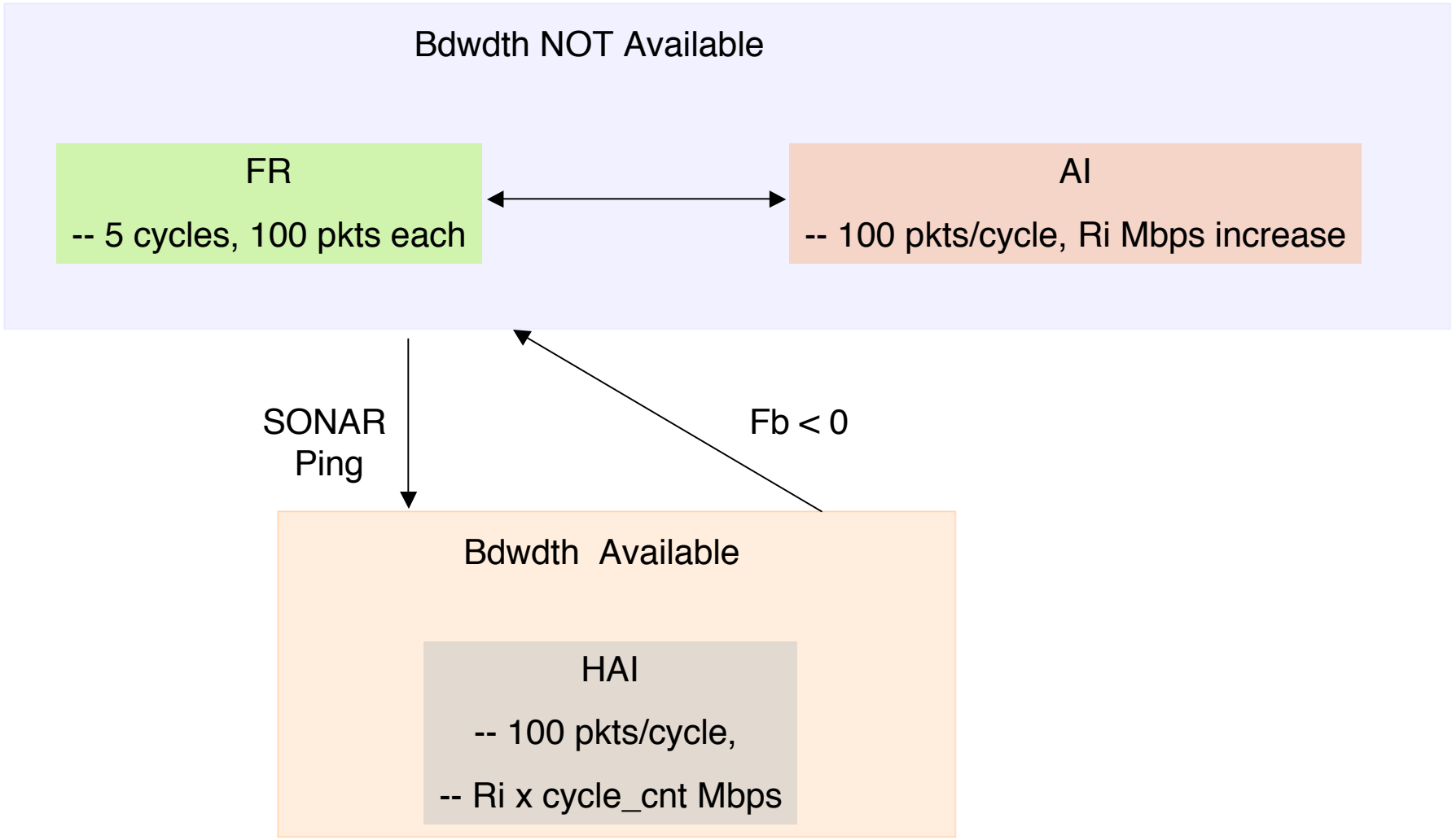
- Two items to be discussed

1. SONAR/Fb99 approaches
   - Sensitivity of probing for available bandwidth
   - Complexity of having timer at switch (issue raised at Atlanta)
2. Timer-supported basic QCN
   - Adapts the drift timer at the source
   - This gives low recovery time in addition to stability
   - P-code will be posted by Rong after call

Recall: SONAR and Fb99

- In SONAR and Fb99
  - A source (RL) performs an endless loop of FR and AI; meanwhile, it is also probing for “available bandwidth”
  - Switches detect “bandwidth availability” based on queue going down “for a while”; timer used at the switch
  - When bandwidth becomes available on all paths fed by an RL
    - RL goes into Hyper AI

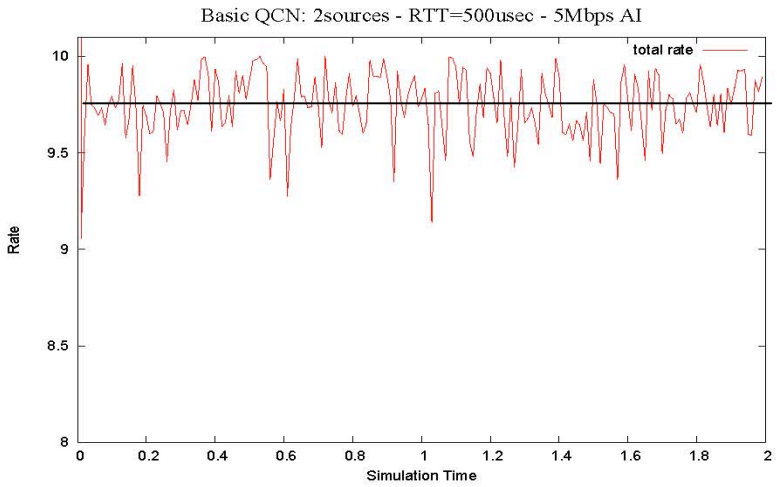
Recall: The SONAR Algorithm at RL

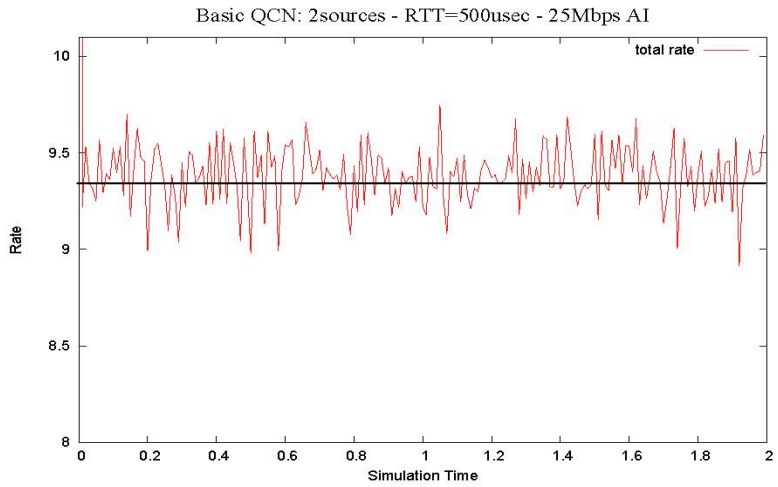


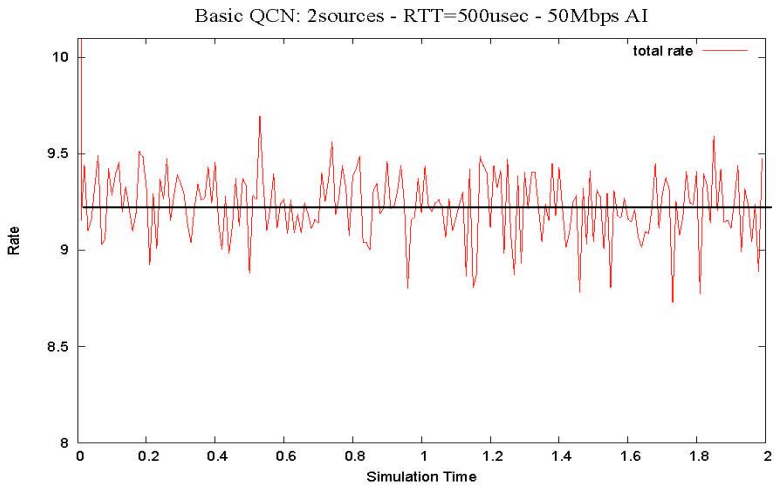
Remarks

- The RL can be in one of 3 states: FR, AI or HAI
- SONAR and Fb99 put HAI ***in parallel*** with FR and AI
  - That is, when bandwidth availability is detected, RL goes into HAI
  - It ***does not send any further packets*** in FR and AI modes
- The parallel structure achieved
  - Stability: Safe operation of RL during congestion: endless loop of FR--AI
  - Rapid response: Because of HAI triggered by a ***timer***
- But there is a lot of sensitivity to correctly detecting "bandwidth availability"
  - We will see that it is not an easy condition to detect, esp for 2 sources and large RTT
- Conclusion: Place HAI ***in series*** with FR and AI for safe operation
  - In order to achieve rapid response use a timer at RL in addition to byte-counter

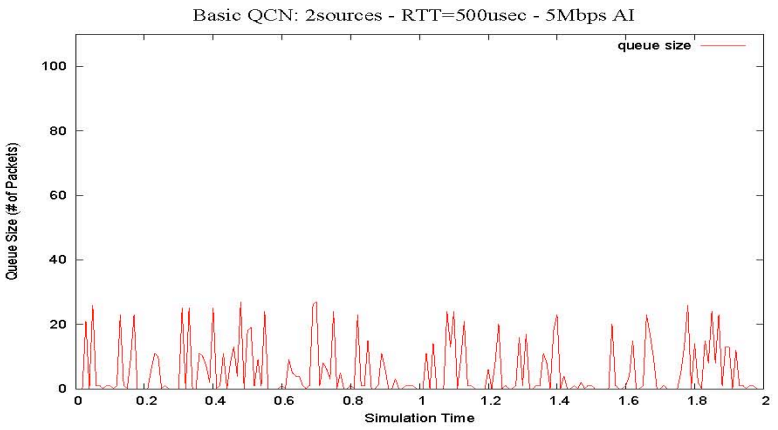
Basic QCN: FR--AI
Rates

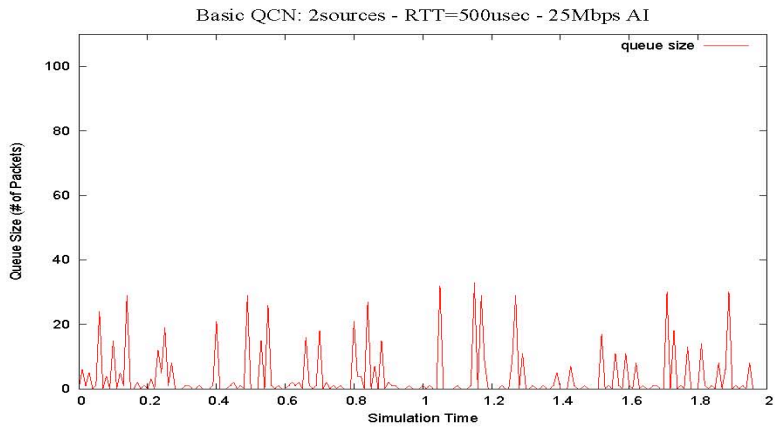


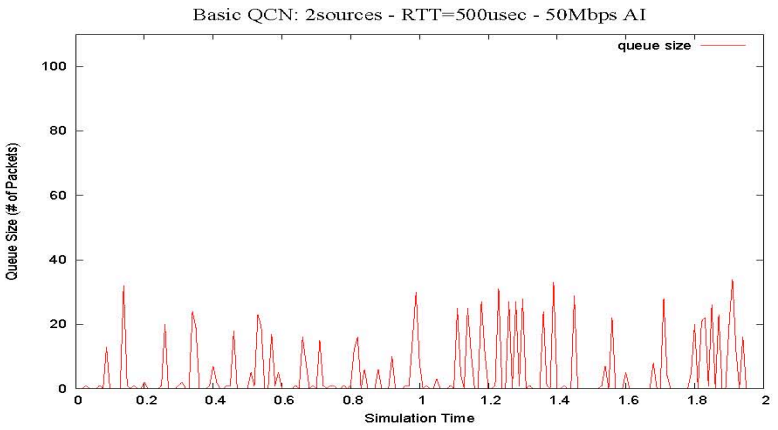




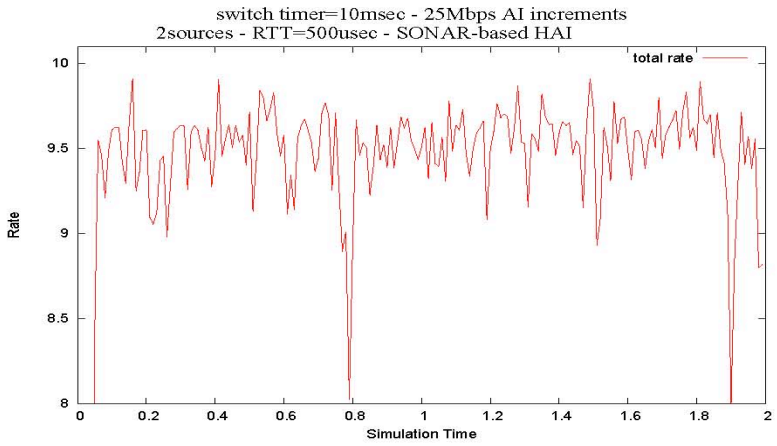
Basic QCN: FR--AI
Queue sizes

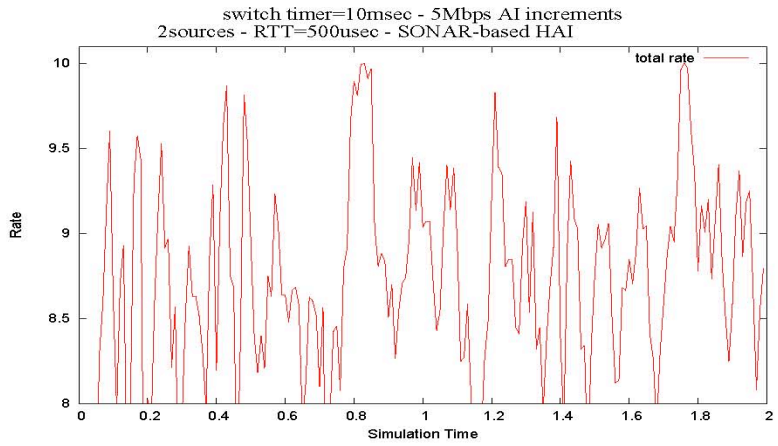


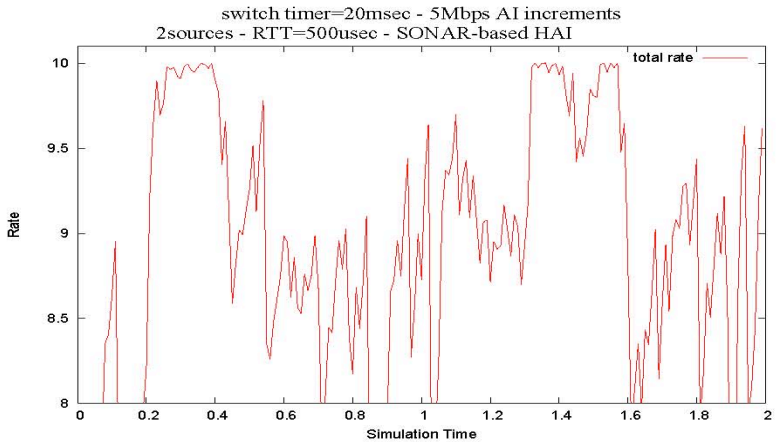




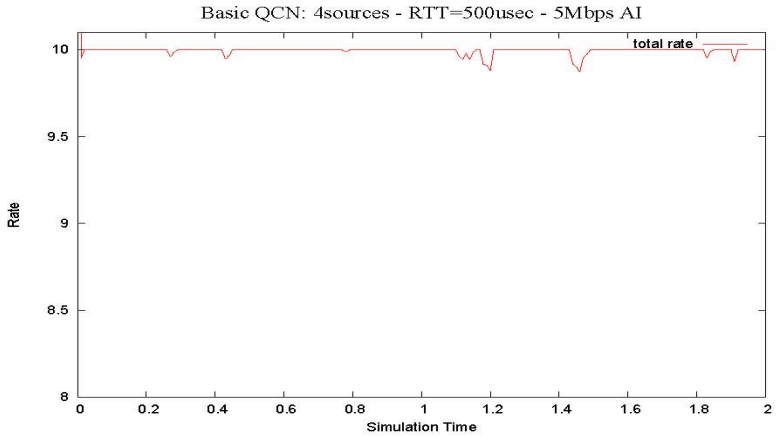
QCN with SONAR
HAI makes rate loss more severe

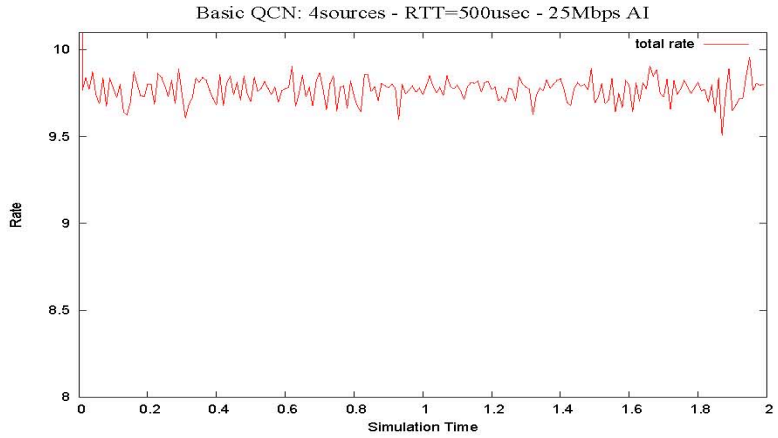


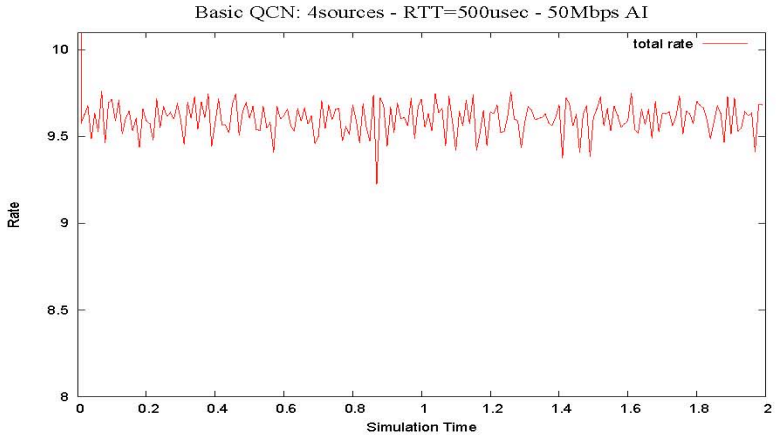




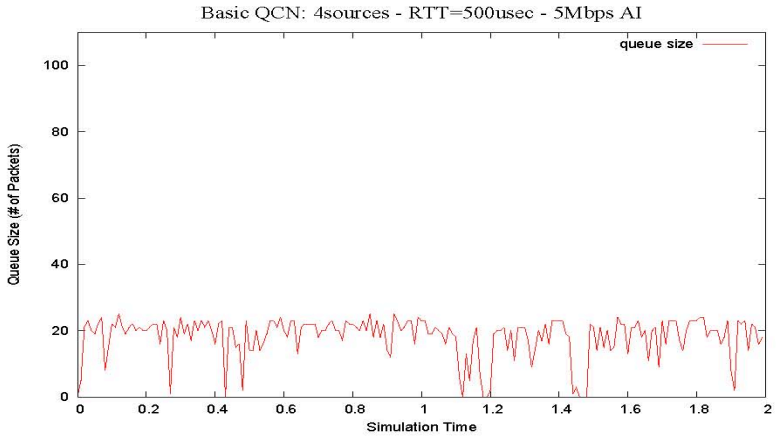
4srcs, rate

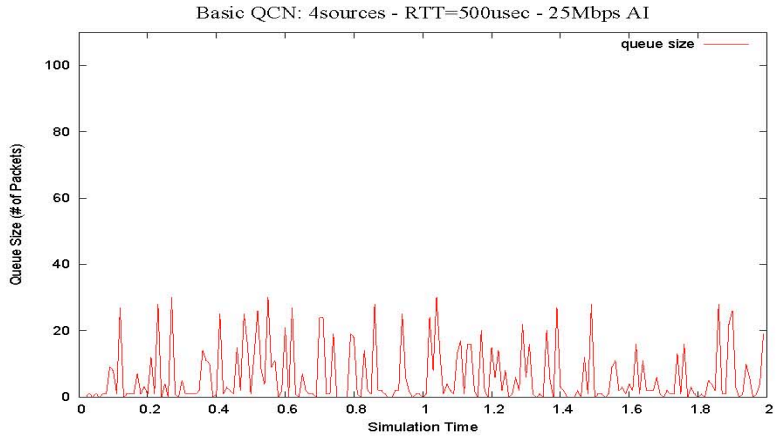

Basic QCN: 4sources - RTT=500usec - 5Mbps AI
total rate
Rate
Simulation Time
10
9.5
9
8.5
8
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

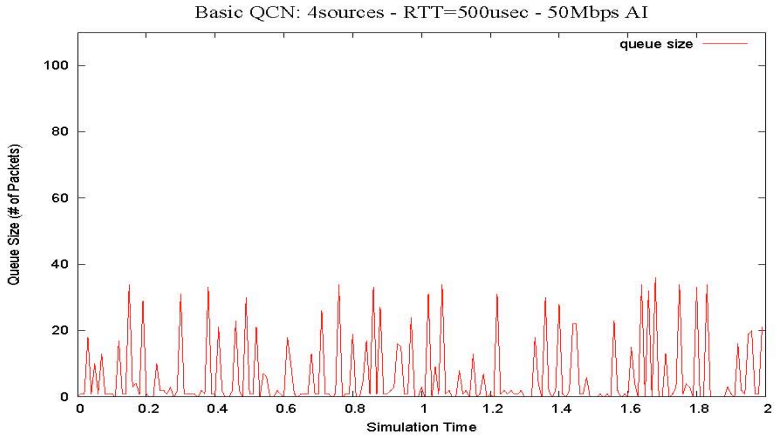

Basic QCN: 4sources - RTT=500usec - 25Mbps AI
total rate
Rate
Simulation Time
10
9.5
9
8.5
8
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


Basic QCN: 4sources - RTT=500usec - 50Mbps AI
total rate
Rate
Simulation Time
10
9.5
9
8.5
8
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

4srcs, qsize





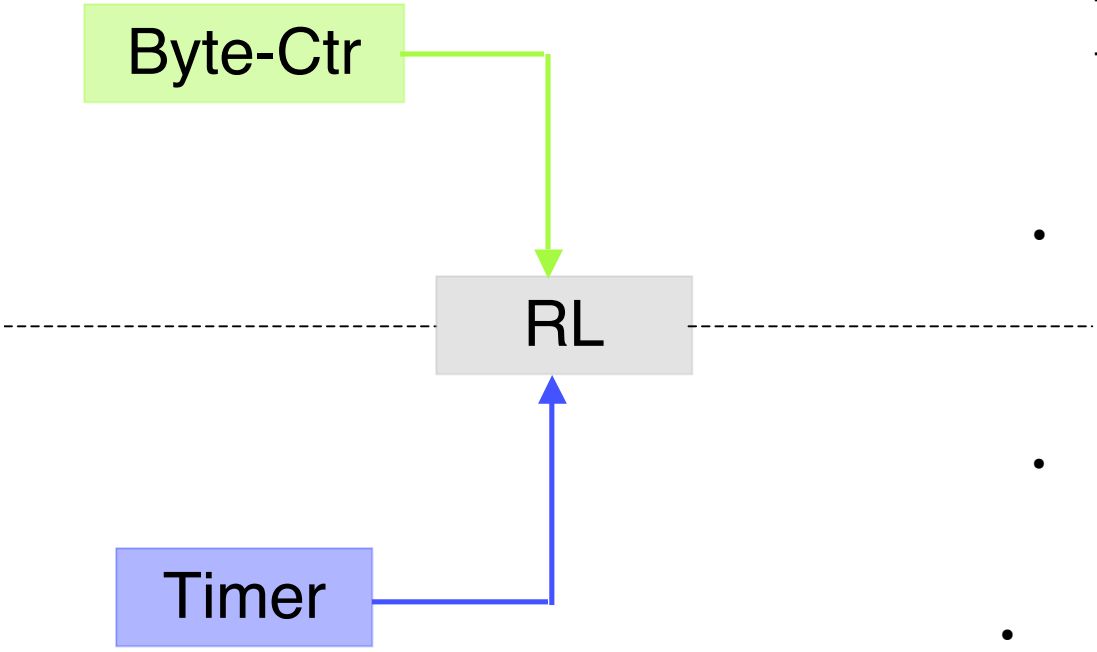


- NOTE
  - With 4 sources and long RTT of 500 us, the stability and utilization are pretty good
  - So, a very small number of sources AND a very large RTT cause problems for aggressive sources

Summary

- "Bandwidth availability" tricky to detect
  - Going into HAI based on incorrect detection degrades performance
- Need to put HAI ***in series*** with FR and AI
  - This ensures stability
- Need to use a timer at the source
  - This ensures quick recovery of bandwidth

Timer-supported QCN



- Byte-Counter
  - 5 cycles of FR (150KB per cycle)
  - AI cycles afterwards (75KB per cycle)
  - Fb < 0 sends timer to FR

- RL
  - In FR if ***both*** byte-ctr and timer in FR
  - In AI if ***only one of*** byte-ctr or timer in AI
  - In HAI if ***both*** byte-ctr and timer in AI
- Note: RL goes to HAI only after 500 pkts have been sent

- Timer
  - 5 cycles of FR (T msec per cycle)
  - AI cycles afterwards (T/2 msec/cycle)
  - Fb < 0 sends timer to FR

Rate Adjustments

When RL is in FR

- Upon completion of a byte-ctr or timer cycle: CR=(CT+TR)/2
- EFR and Target rate reduction enabled during first cycle of byte-ctr

When RL is in AI

- Upon completion of byte-ctr or timer cycle: TR = TR + R_{AI}; CR = (CR+TR)/2

(We've used R_{AI} = 5 Mbps)

When RL is in HAI

(This means at least 500 pkts have been transmitted since last ding)

- Events = completion of byte-ctr or timer cycles
- Events numbered i = 1, 2, …
- At the end of event number i:

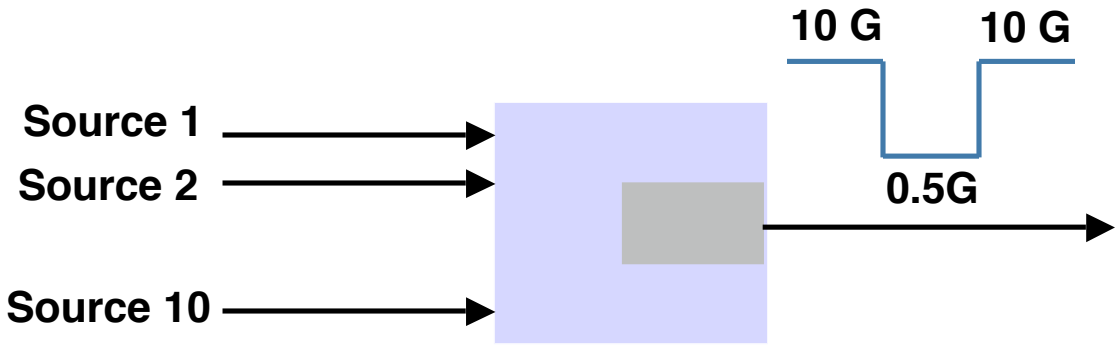
TR = TR + (i*R_{HAI}); CR = (CT+TR)/2;

(We've used R_{HAI} = 50 Mbps)

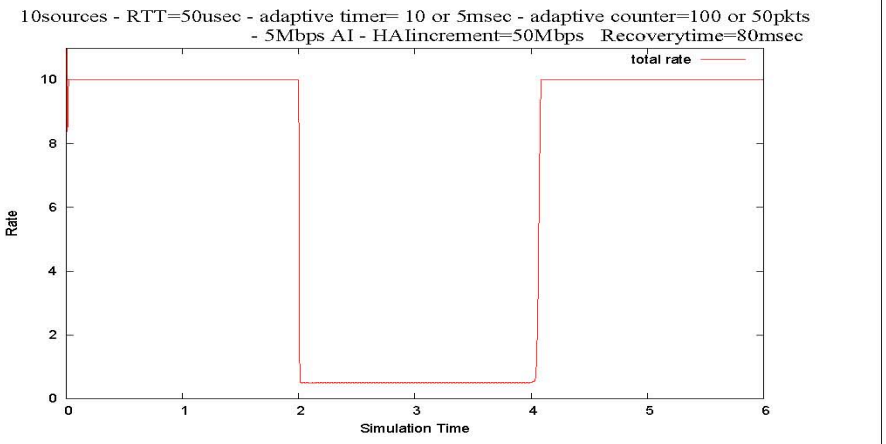
Simulations: OG Hotspot

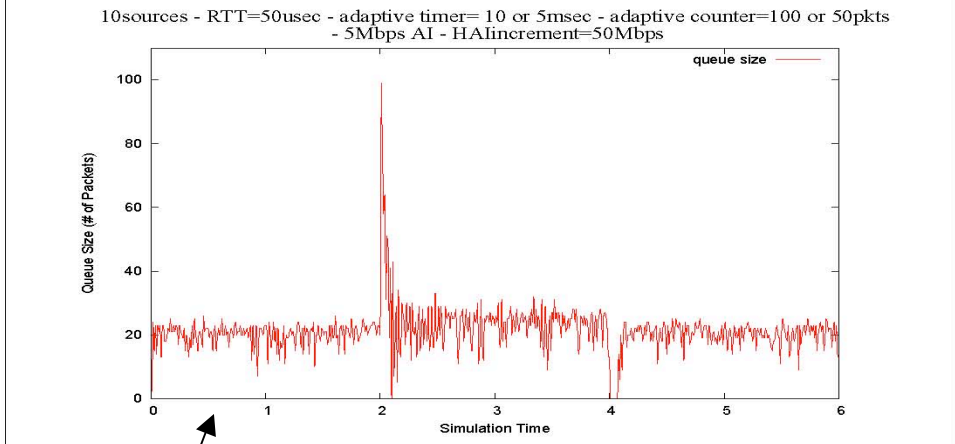
- Parameters
  - 10 sources share a 10 G link, whose capacity drops to 0.5G during 2-4 secs
  - Max offered rate per source: 1.05G
  - RTT = 50 usec
  - Buffer size = 100 pkts (150KB); Qeq = 22

  - T = 10 msecs
  - R_{AI} = 5 Mbps
  - R_{HAI} = 50 Mbps



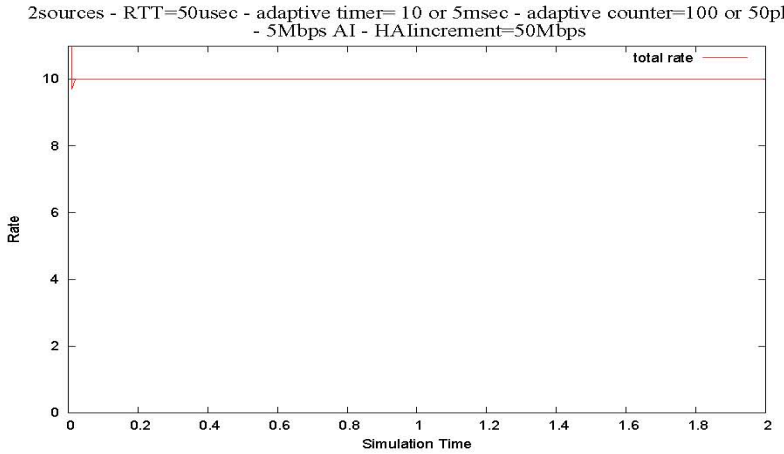
Recovery Time

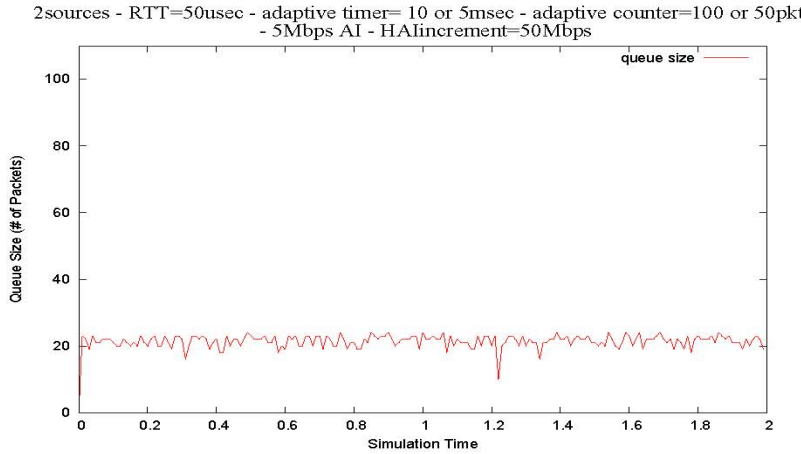


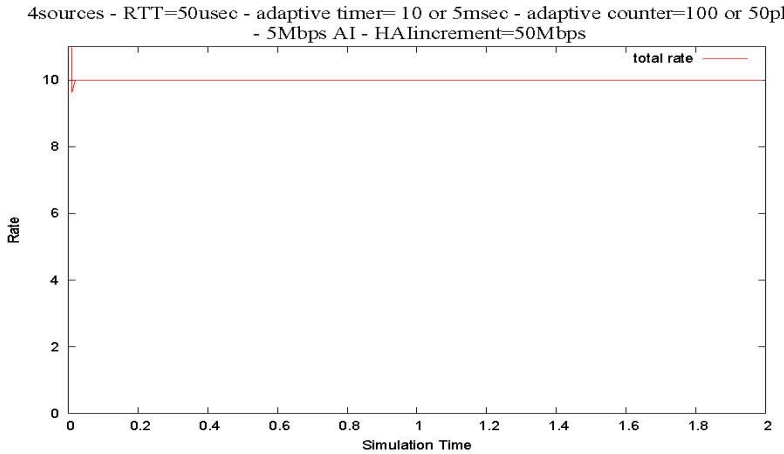


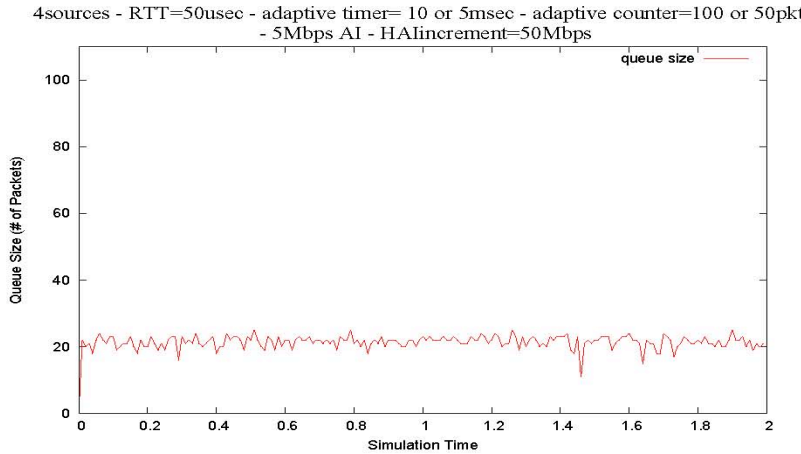
Stability not compromised
Recovery time = 80 msec

Stability (RTT = 50 usecs)

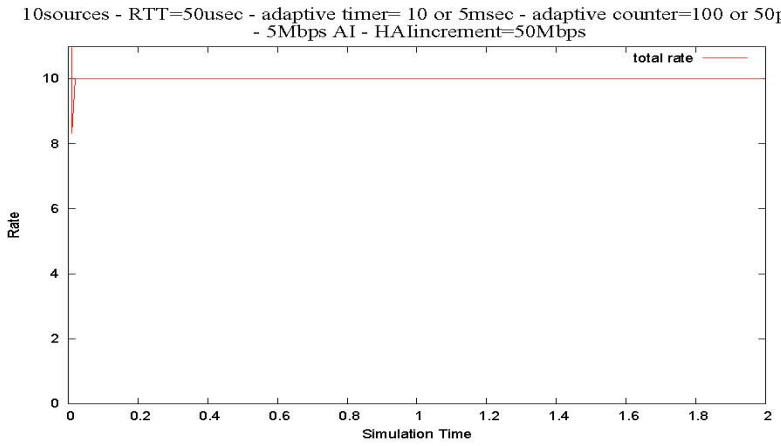


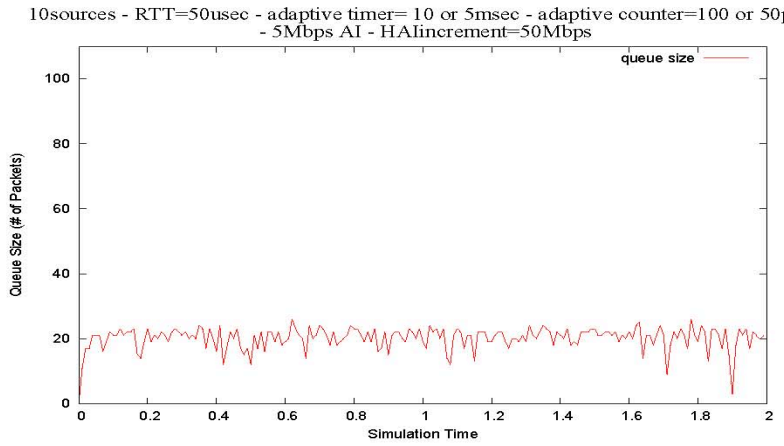


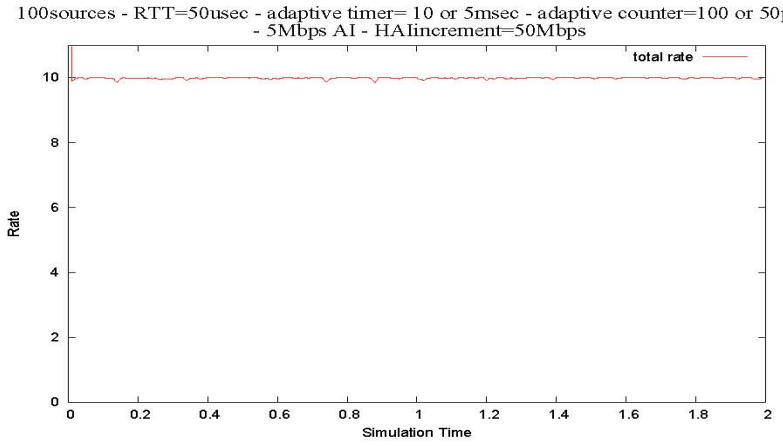


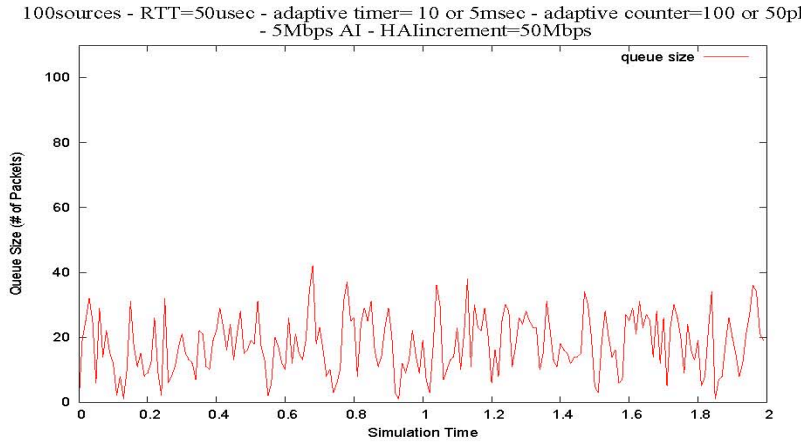


Stability (RTT = 50 usecs)

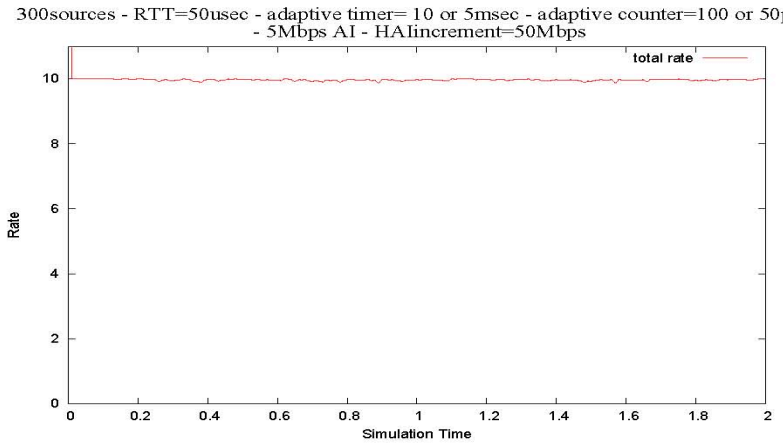


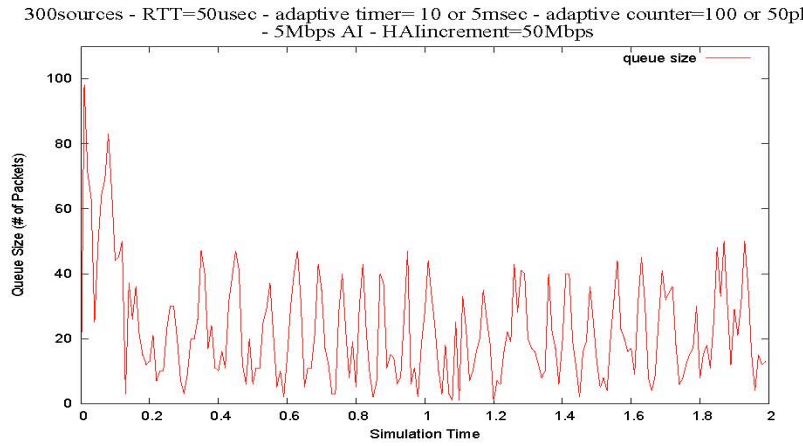


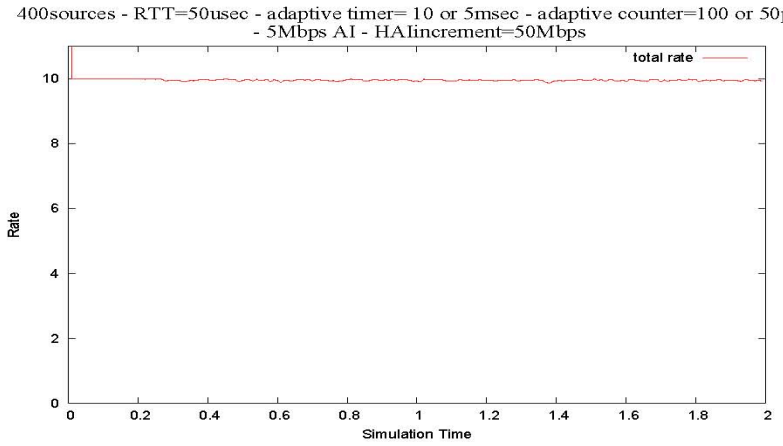


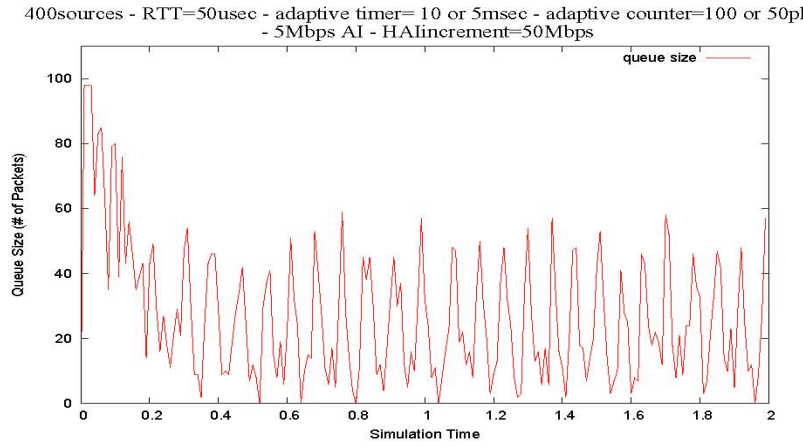


Stability (RTT = 50 usecs)

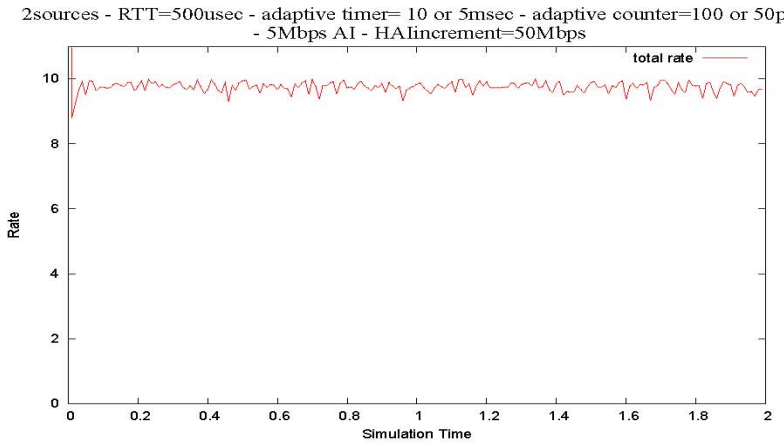


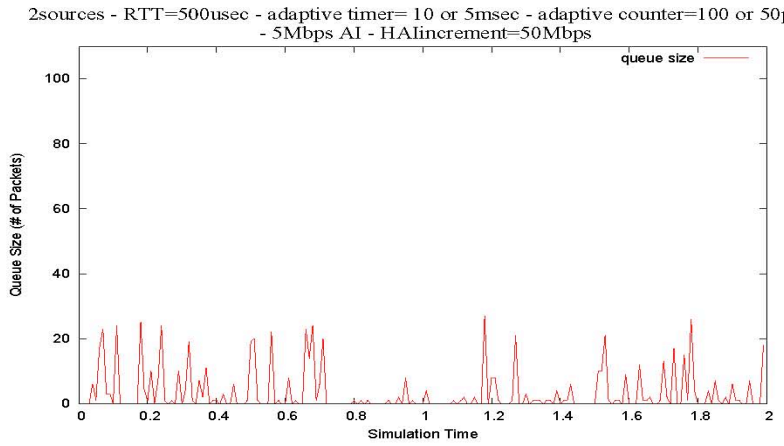


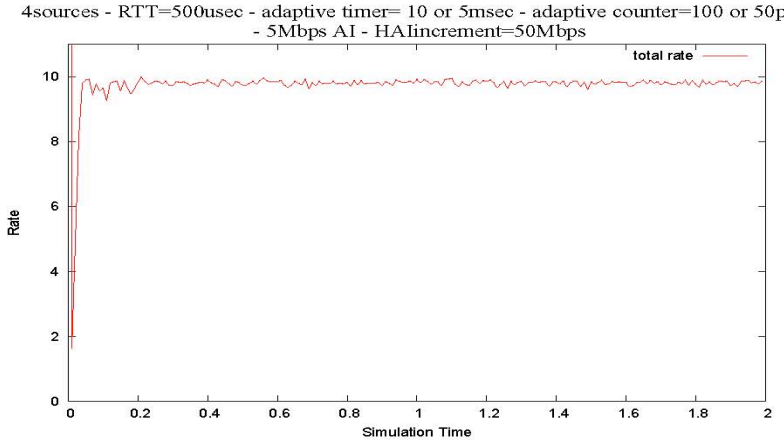


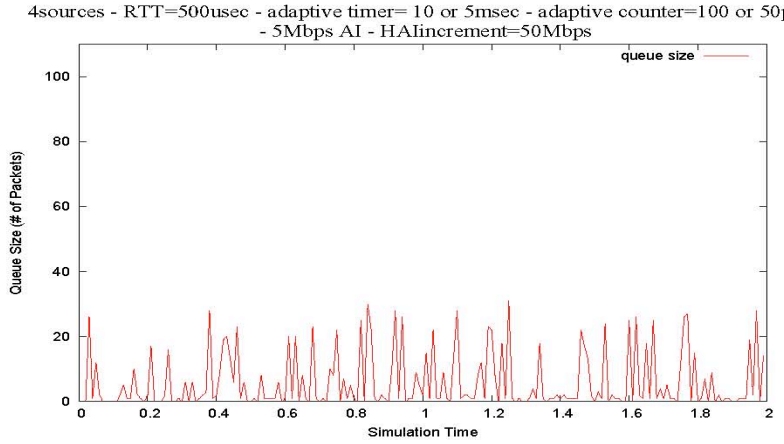


Stability (long RTT = 500 usecs)

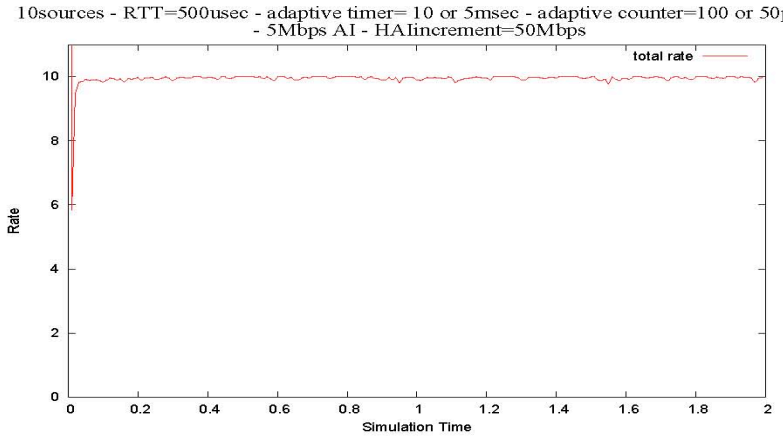


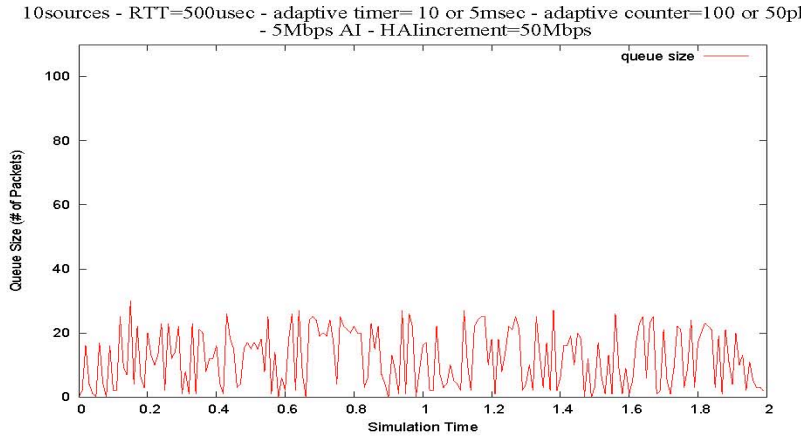


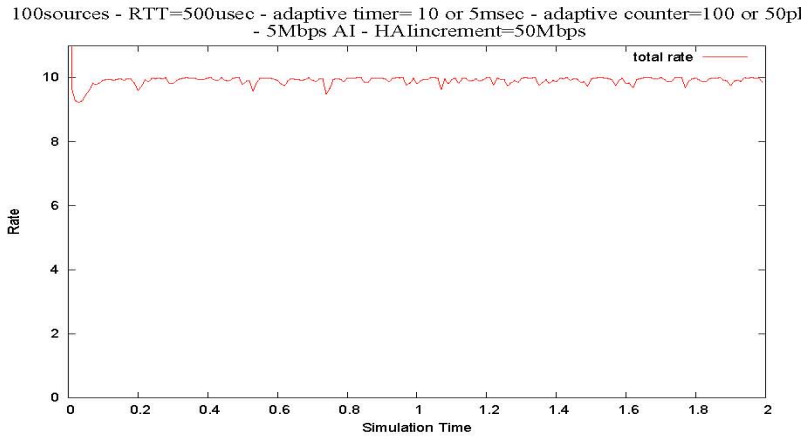


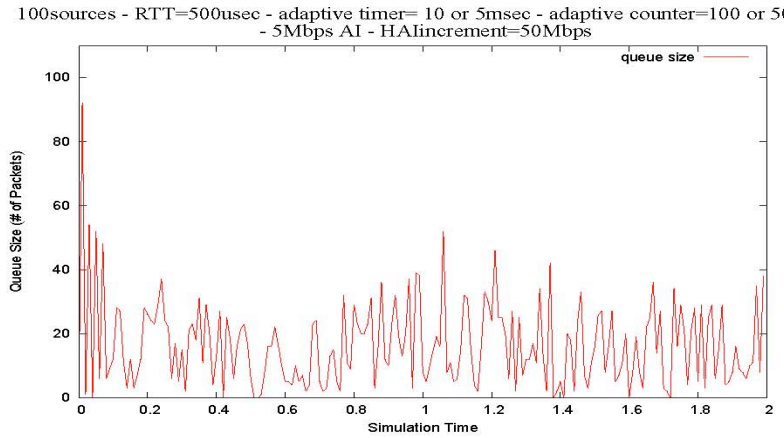


Stability (RTT = 500 usecs)

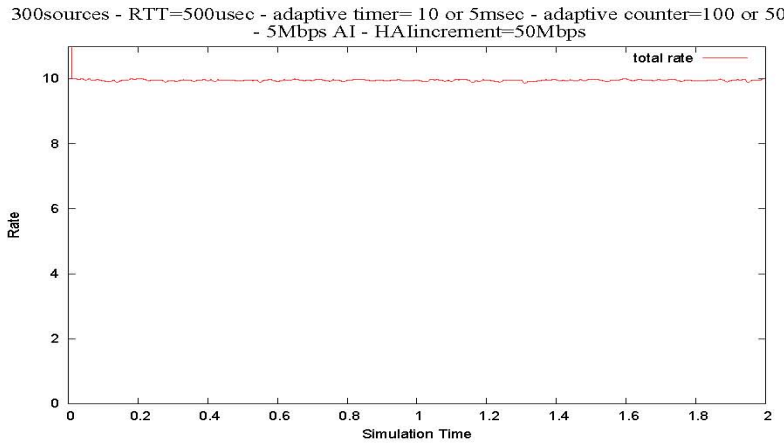


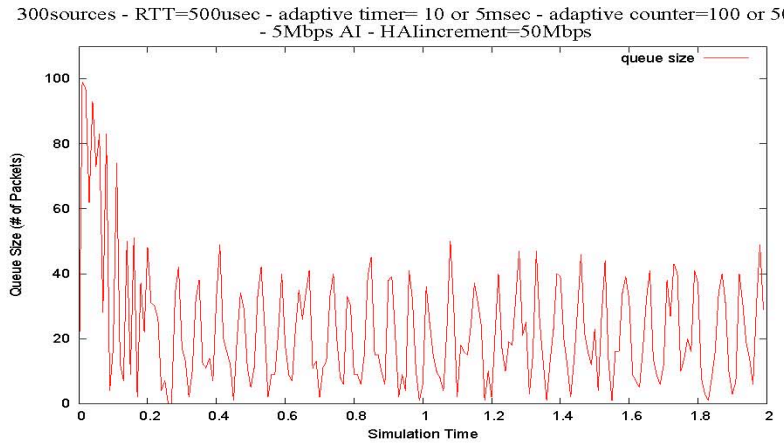


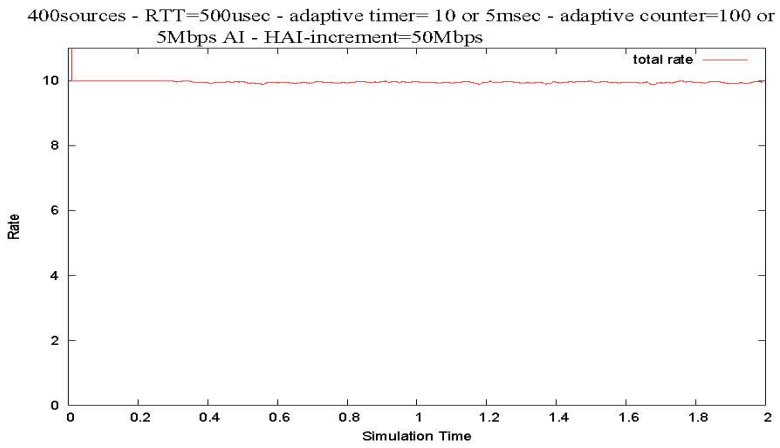


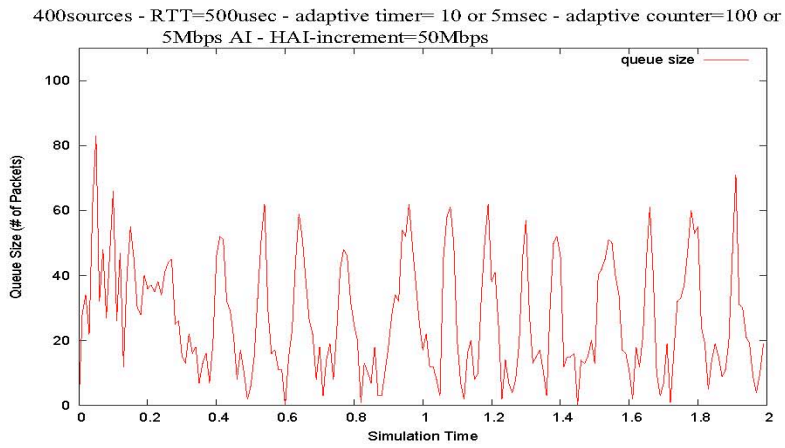


Stability (500 usecs)









Conclusion

- Discussed trickiness of detecting available bandwidth
- Need to put HAI ***in series*** with FR and AI
  - Gives good stability; timer gives good recovery time
- Timer-supported QCN
  - Key ideas and features
    - Transmit 500 pkts before going to HAI
    - Byte-ctr gives network sufficient opportunity to send feedback to RL
    - Timer hastens recovery for slow rate sources
    - No change to switch operation from basic QCN; esp no timer at switch
    - Parameter free (no dynamic parameter choice needed)
  - Very similar to basic QCN
  - Can be further enhanced, e.g. with Fb-hat or Fb99
    - Need to seriously consider benefits of enhancements first!