Explicit Congestion Notification (ECN)

Jinjing Jiang and Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@wustl.edu

IEEE 802.1 Congestion Group Meeting, Dallas, TX
Nov 14, 2006

- Explicit Congestion Notification (ECN): What and Why?
- Source/Switch/Queue Control Algorithms
- Simulation Results
- Other Variations of ECN
- Comparison of ECN and BCN

Requirements for a Good Scheme

1. Fast convergence to stability
2. Fast convergence to fairness
3. Good for bursty traffic
4. Predictable performance: No local minima
5. Stable rates $\Rightarrow$ TCP Friendly (IETF feedback)
6. Easy to deploy:
    1. Small number of parameters
    2. Easy to set parameters
    3. Parameters applicable to a wide range of configurations (number of sources), link speeds, traffic types.

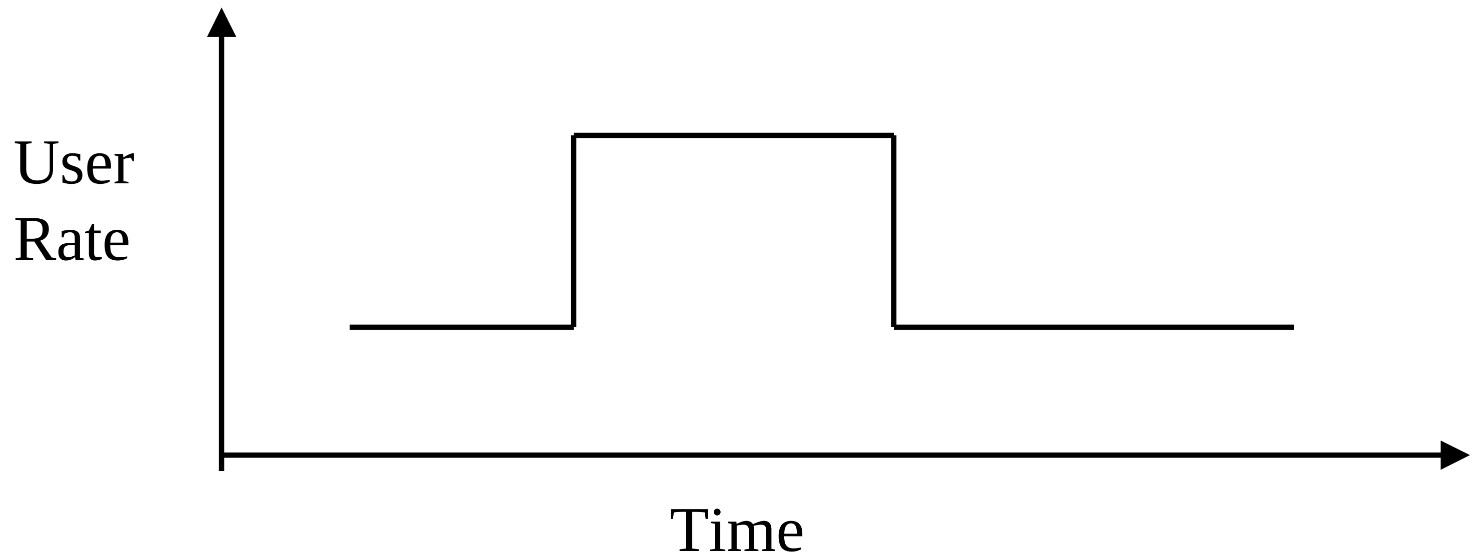
Convergence to Stability

- Convergence to the desired queue length in a few ms

=/=> Convergence of user rates.

User rates may still be off from the desired fair values.

May even have multiple stable states.



Large Oscillations and Metastability

- Symmetric Topology

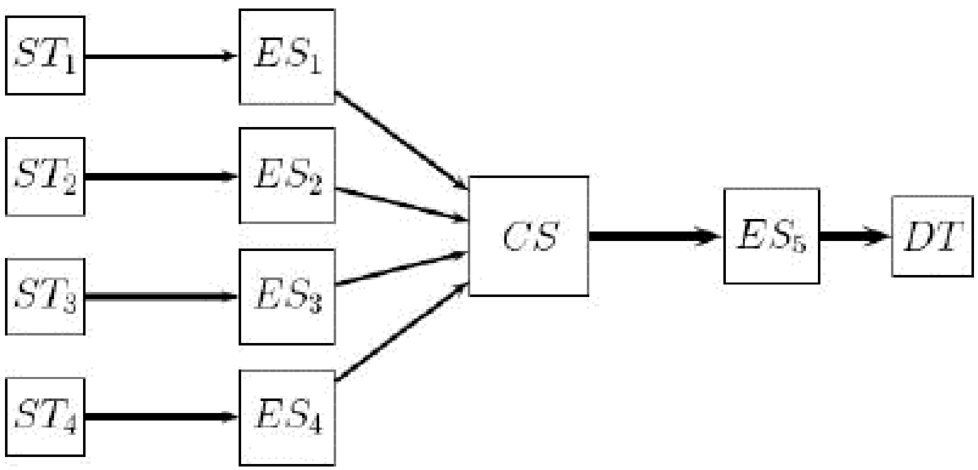
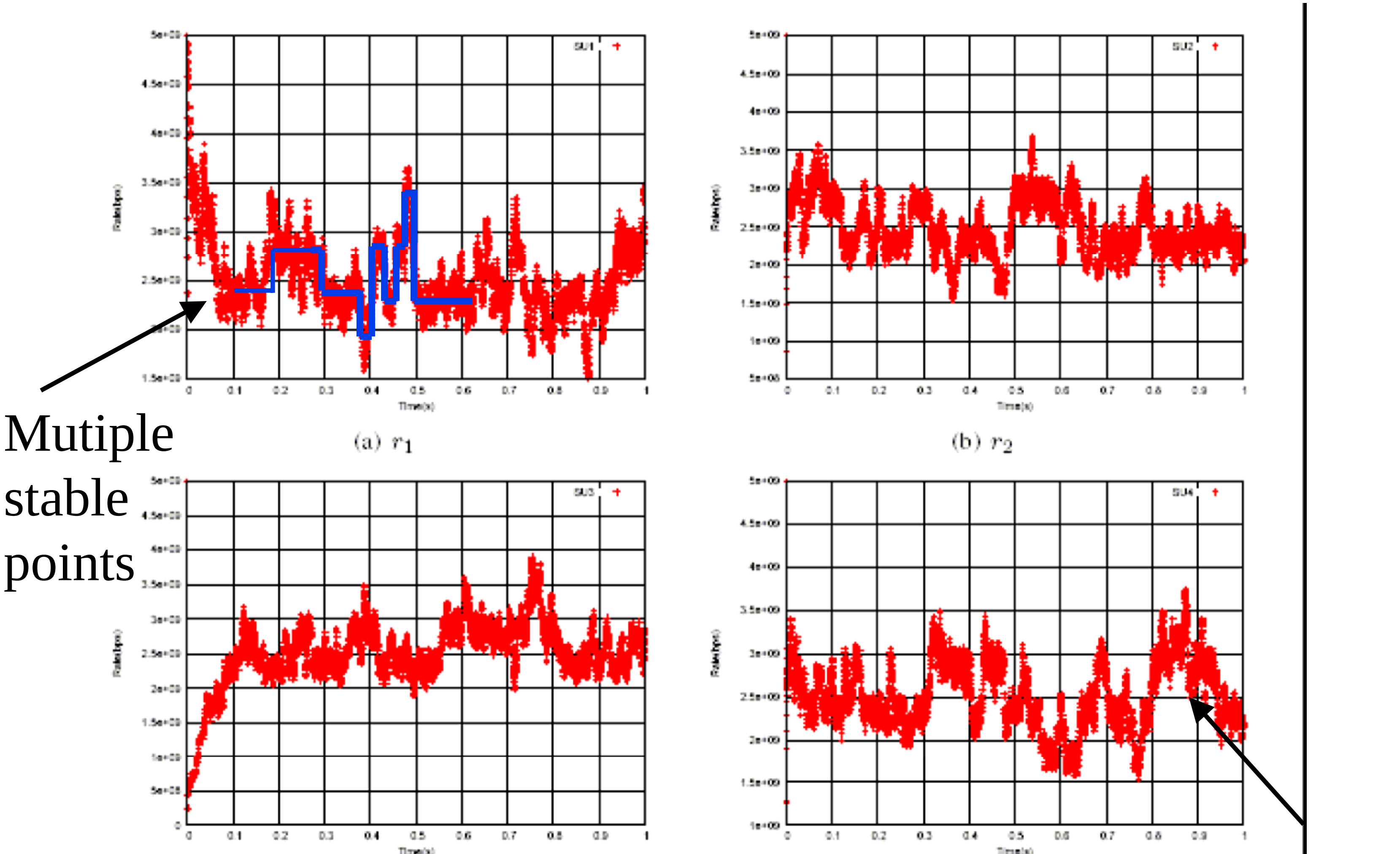


Fig. 1. A simple symmetric topology

Simulation Results: BCN



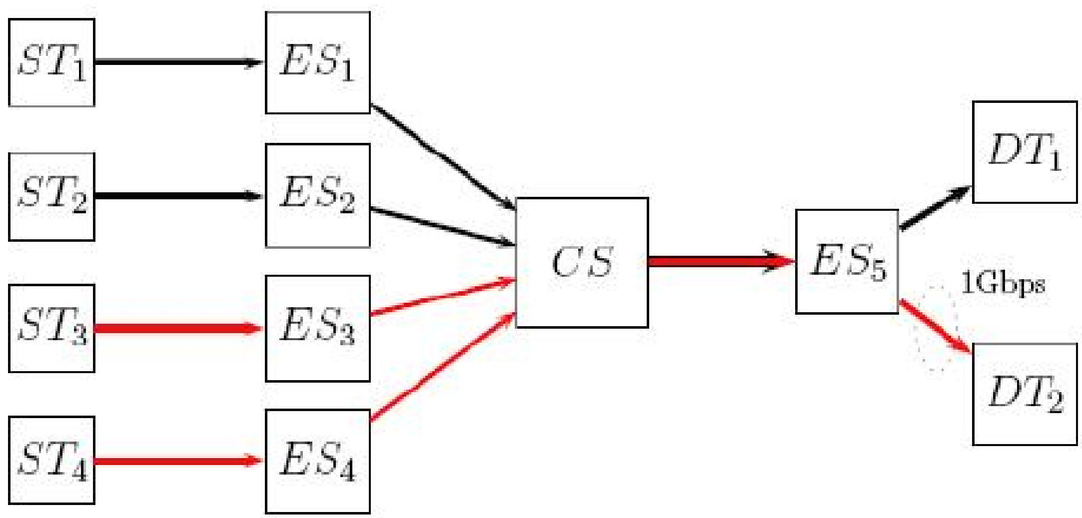
(a) r_1 (b) r_2 (c) r_3 (d) r_4

Time to Convergence

- Time to stability depends on the sampling size (in kB). Sampling interval is much much larger than round trip delays $\Rightarrow$ Convergence times of 100's of ms.
- The system can be unstable with incorrect sampling size
- The rate increase parameter Ru, sampling size, and link capacity are related
- When there are multiple congestion points, BCN's rate oscillations are high

Asymmetric Topology and Multiple Congestion Points

Topology: Only one link is 1Gbps, others are all 10Gbps

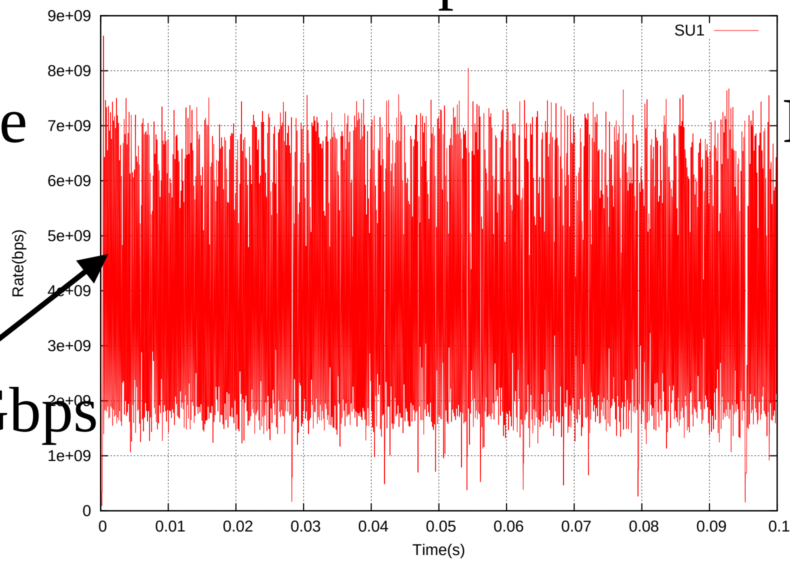


Multiple Congestion Points: BCN

- Large Oscillations

2-7 Gbps

Rate



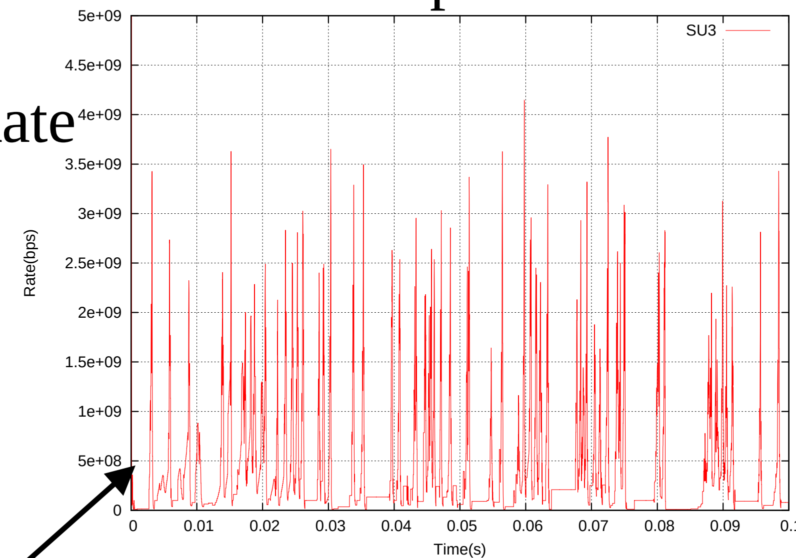
4.5Gbps

Time →

Optimal Rate
for ST1 and ST2

0-3 Gbps

Rate



Time →

0.5Gbps
Optimal rate for
ST3 and ST4

Slow Convergence to Fairness

- Analytical models have shown BCN to be fair
- However, the time to achieve fairness is too long
- The time to fairness depends upon the feedback delay which is dominated by the sampling interval
- In baseline simulations scenarios, the time to fairness can be several hundred ms.
  $\Rightarrow$ The system operates mostly unfairly if the traffic changes every few ms.
  $\Rightarrow$ Not good performance for bursty traffic.

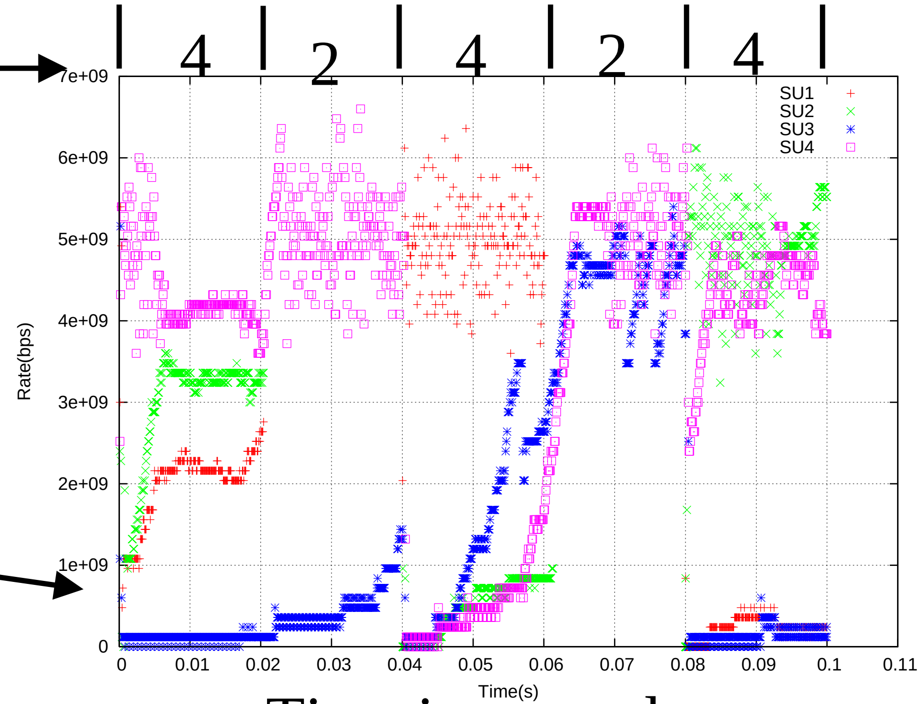
On/Off Sources

- Four source configuration shown earlier. Two sources (green+red) turn off at 20 ms and back on after 20 ms.

# of active sources

Allocated Rate in bps

Blue source received BCN (0,0)



Time in seconds

Overall Throughput: Pink ≠ Blue, Green ≠ Red => Unfair

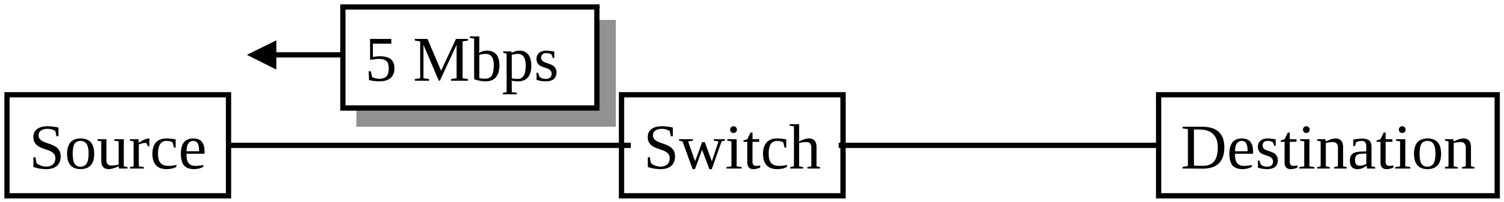
Fundamental Issues

- Sampling: RLT tags are sampled
  => Rate increase is matter of chance
- Overload = Qdelta/Sampling time
  Packet based sampling
  => sampling time depends upon the packet sizes
  => Byte based sampling
  => Sampling time depends upon the arrival rate
  => Qdelta is not a perfect indicator of overload
- Queue feedback: Meaningful only with capacity

Disclaimer

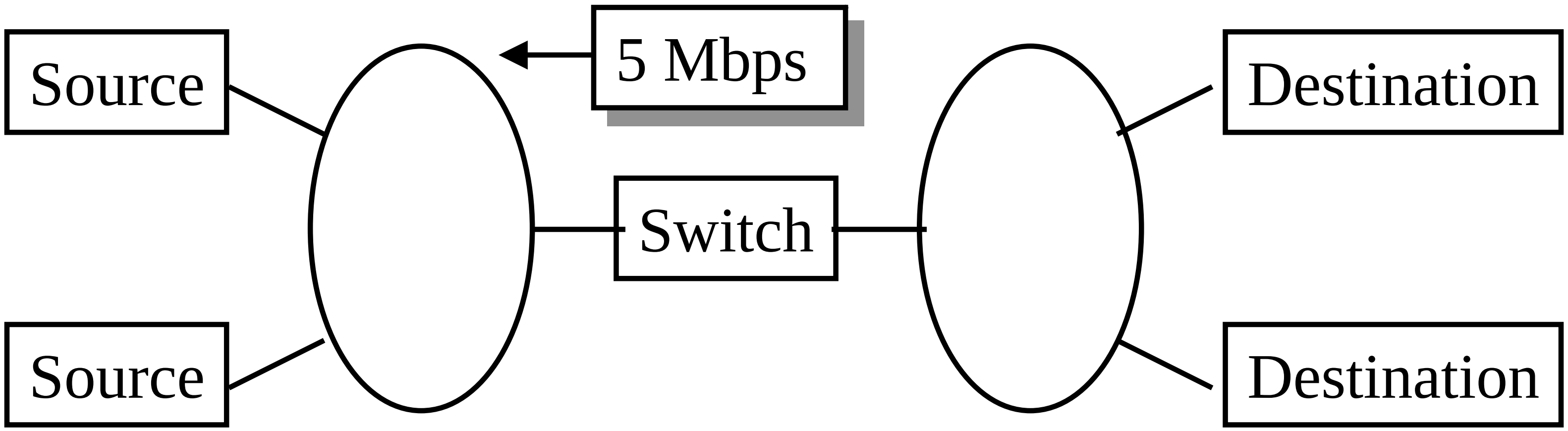
- This is a new scheme. Just developed.
- Some work will need to be done
- May not be able to answer all the questions
- Goal is to provide ideas for possible solutions to known problems
- There are many variations.
  Basic ECN will be described first.
  Variations later.

Explicit Congestion Notification (ECN)



- Switch sends a rate to the source. Source sets to that rate
- Only the feedback format has to be standardized
- No need to standardize switch algorithm.
- There are no source parameters
- Vendor differentiation: Different switch algorithms will “inter-operate” although some algorithms will be more efficient, more fair, and achieve efficiency/fairness faster than others.
- We present a sample switch algorithm and show that it achieves excellent performance.

Proposed Algorithm



- The switch sends its "**Advertised Rate**" to all sources
- All sources get the same feedback.
- The sources send at the rate received.

A Simple Switch Algorithm

0. Start with an Advertised Rate of r
1. Measure input rate every T interval
2. Compute overload factor z in the last T interval
3. Change the rate to r/z
4. Go back to step 1

Although this simple algorithm will work but:

- It will oscillate even if the rate is close to optimal.
- Queues will not be constant => ***Need a Q Control Fn***

A Sample Switch Algorithm

1. **Initialization**:

$$r_0 = \frac{C}{2}$$

Here C is the link capacity in bits/s. r_0 can be any other value too, e.g., C/4. It has no effect on convergence time.

2. **Measurement**: Let A_i be the measured arrival rate in bits/s then the load factor is A_i/C. We update this load factor based on the queue length so that the ***effective load factor*** is:

$$\rho_i = \frac{A_i}{f(q_i) \times C}$$

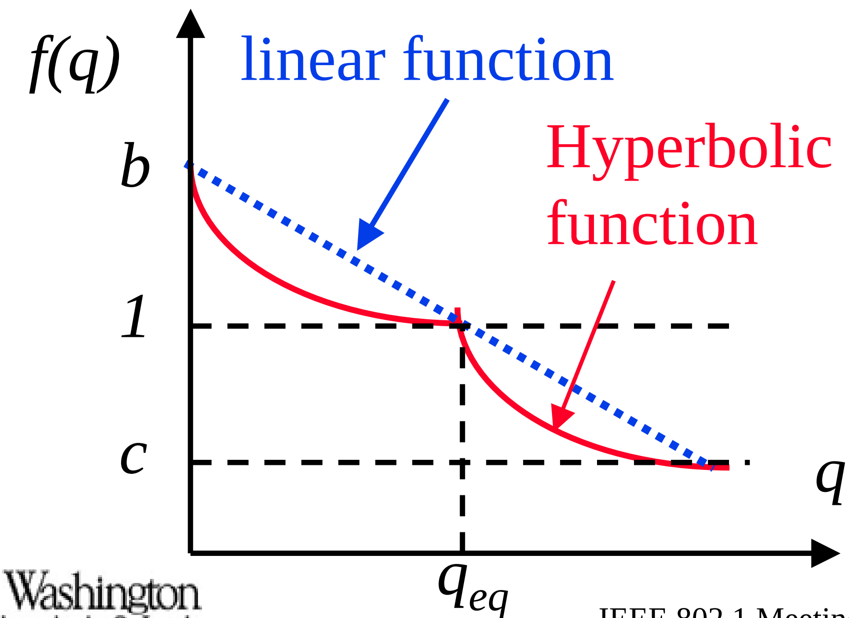
3. **Bandwidth Allocation**:

$$r_{i+1} = \frac{r_i}{\rho_i}$$

Queueing Control Function: *f(q)*

Idea: Give less rate if queue length is large and more if queue length is small compared to desired queue length of q_{eq} and $f(q_{eq})=1$

$$f(q) = \begin{cases} \geq 1 & q \leq q_{eq} \\ = 1 & q = q_{eq} \\ \leq 1 & q \geq q_{eq} \end{cases}$$



We analyzed many different functions and recommend the hyperbolic function because it gives smaller oscillations. [See reference]

Queue Control Function: *f(q)*

- **Linear Function**: k is some constant

$$f(q) = 1 - k\frac{q - q_{eq}}{q_{eq}}$$

- **Hyperbolic function**: a, b, c are constants

$$f(q) = \begin{cases} \dfrac{bq_{eq}}{(b-1)q + q_{eq}}, & \mathit{if}\ \ q \le q_{eq}; \\ \max\left(c, \dfrac{aq_{eq}}{(a-1)q + q_{eq}}\right), & \mathit{otherwise.} \end{cases}$$

Analytical Results

- Wash U switch algorithm achieves max-min fairness and converges to desired queue length q_{eq}.
  1. Fairness
  2. Stability
  3. Convergence Time

Fairness Proof

Let:

- N = number of flows. (Note that we do NOT need to know N)
- Ai = Total arrival rate

$$r_{i+1} = \frac{r_i}{\rho_i} = \frac{A_i / N}{A_i / [f(q_i)C]} = \frac{f(q_i)C}{N}$$

- When $q_i = q_{eq}$, f(q) =1 => all sources get the fair share C/N which implies MAX-MIN fairness

Convergence Proof

- Easy to show that $\{q_i\}$ is a monotonic sequence converging to q_{eq}

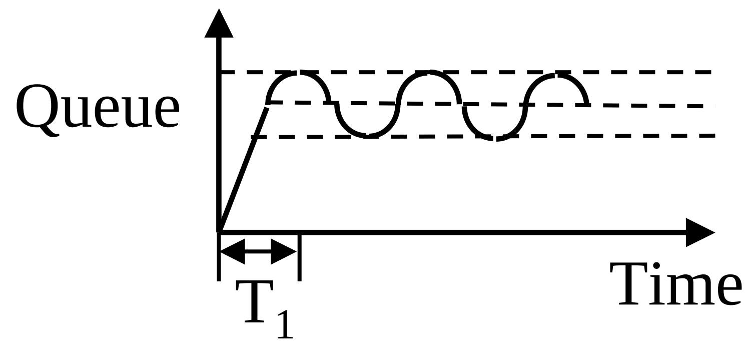
$$q_{i+1} = q_i + (Nr_{i+1} - C)T = q_i + CT(f(q_i) - 1)$$

- At $q=q_{eq}$, $f(q_{eq})=1$

Convergence Time

- Given any $\varepsilon > 0$, define the stable state (fair state) as

$$(1-\varepsilon)q_{eq} \le q_i \le (1+\varepsilon)q_{eq}$$



- For **linear** control function, the system will converge to stable state after n measurement intervals, where:

$$n \approx \frac{\log\left(\frac{\varepsilon}{|M-1|}\right)}{\log(1-\beta)}$$

where $\beta = \frac{CTk}{q_{eq}}, M = \frac{q_0}{q_{eq}}$, k is the slope of the linear queue fn, and q_0 is the initial queue length

Source Algorithm

- Source keeps two variables:
  - CP: Congestion Point ID (CPID) of the bottleneck switch in the last feedback received.
    Initially CP = -1 (No congestion point)
  - r : Current rate
- When the source gets a new BCN Message [r_i,*CPID*]

  IF $r_i < r$ *THEN* $r \leftarrow r_i$ *and CP* $\leftarrow$ *CPID*
  ELSE IF(CP=CPID or CP=-1) THEN $r \leftarrow r_i$

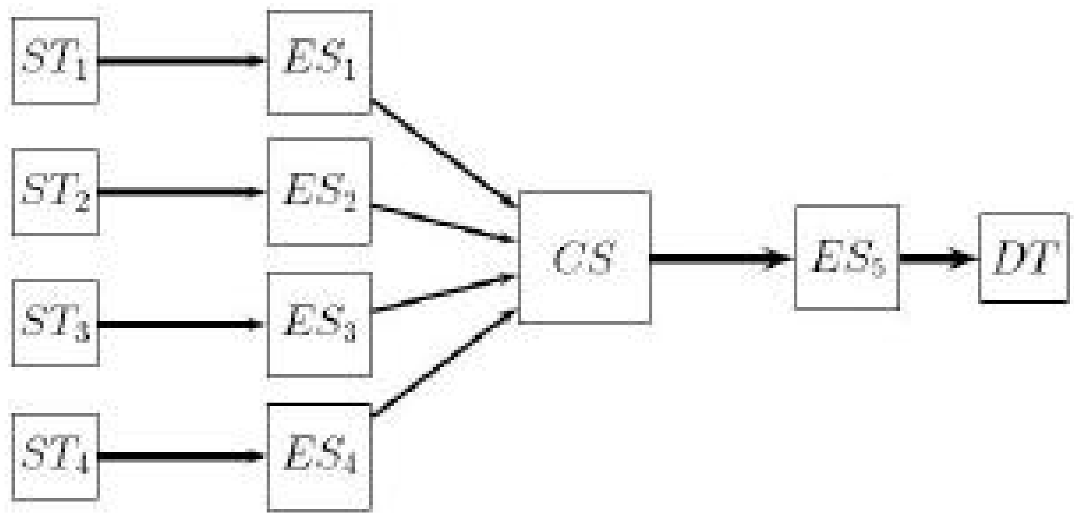
Simulation Parameters

- Measurement Interval: T = 0.03 ms
- Queue control function: Hyperbolic
  a = 1.05, b = 1.2, c = 0.5
- Packet size = 1500 B
- We compare performance with baseline BCN algorithm

Simulation Results

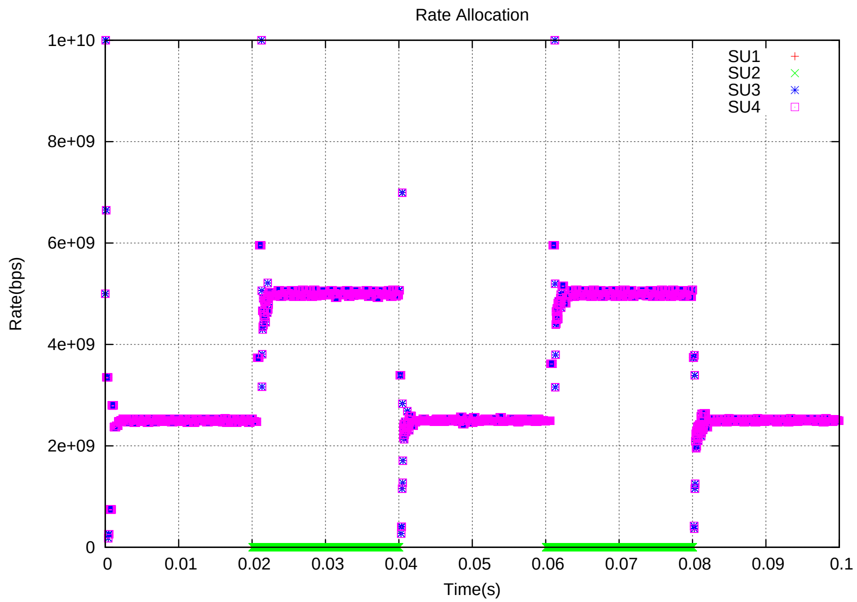
- Baseline Symmetric Topology
- Parking Lot Topology
- Asymmetric Topology
- Bursty Traffic

Symmetric Topology: Configuration



- UDP Bernoulli Traffic with 10 Gbps rate
- ST1 and ST2 are periodically turned off for around 20 ms, i.e., the exact time is not the same
- Simulation Time is *100* ms

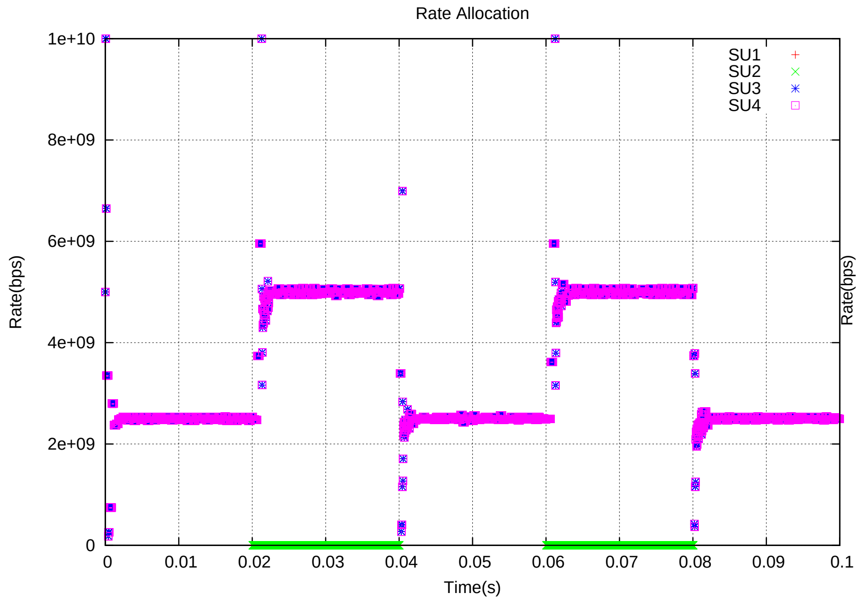
Symmetric Topology: Source Rates Rcvd

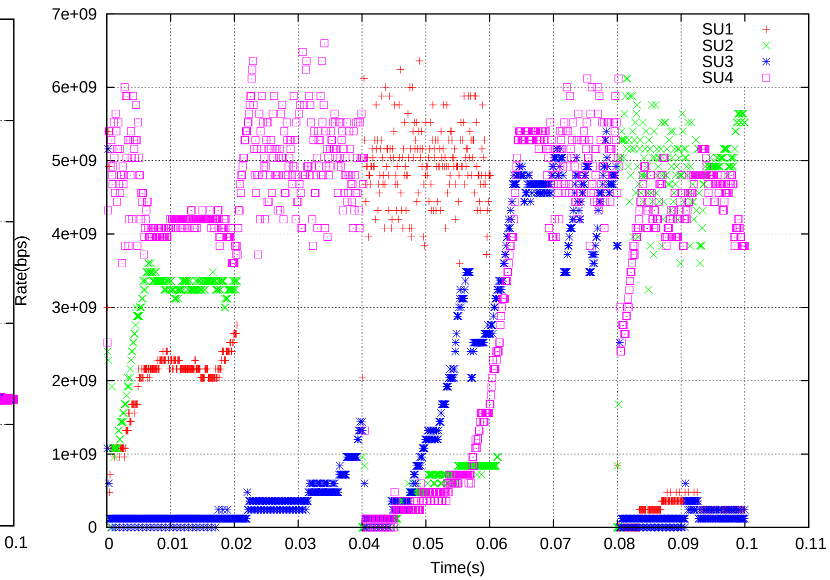


ECN

Conclusion: ECN converges very fast and remains stable. Note that ECN graphs have 4 curves. Perfect fairness results in only two visible curves.

Symmetric Topology: Source Rates Rcvd



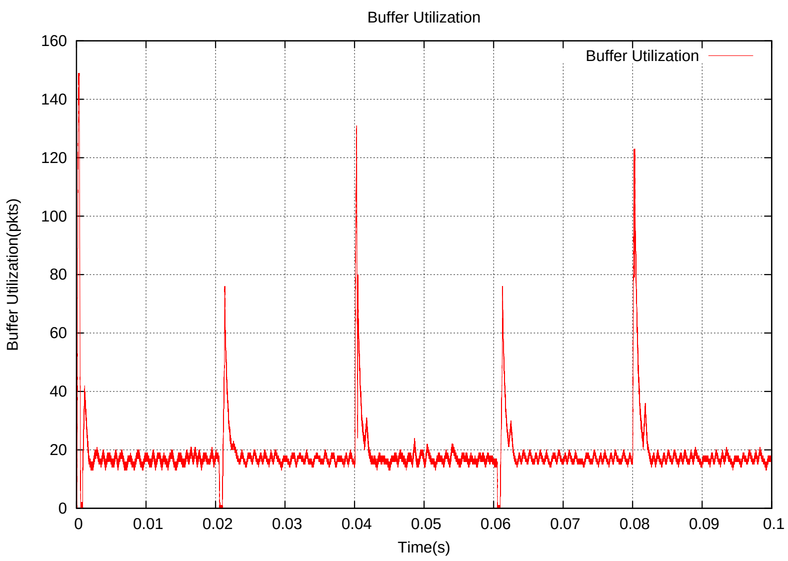


ECN BCN

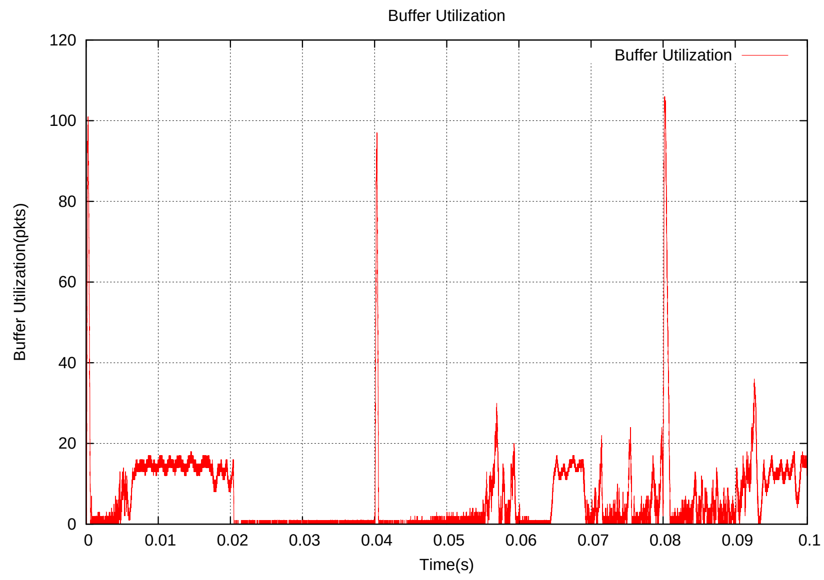
Conclusions:

1. ECN converges very fast and remains stable.
2. Perfect fairness results in only two visible curves. Note that ECN graphs have 4 curves.
3. Convergence time is a small multiple of measurement interval.

Symmetric Topology: Queue Length



ECN

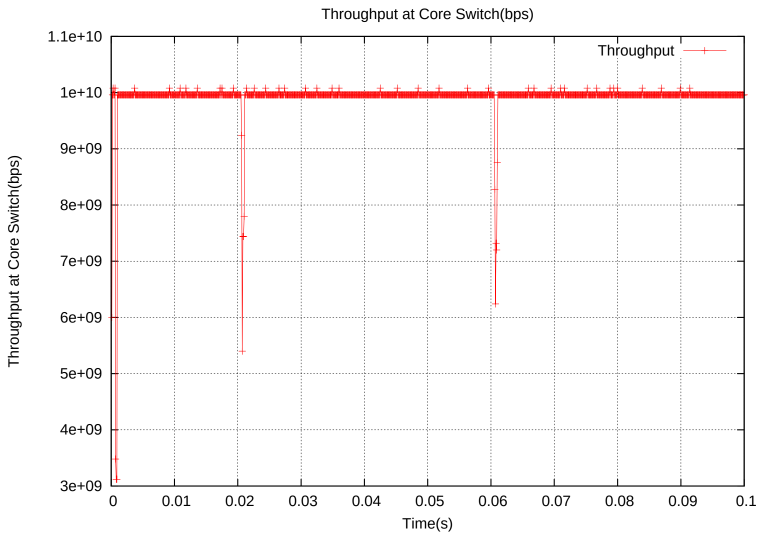


BCN

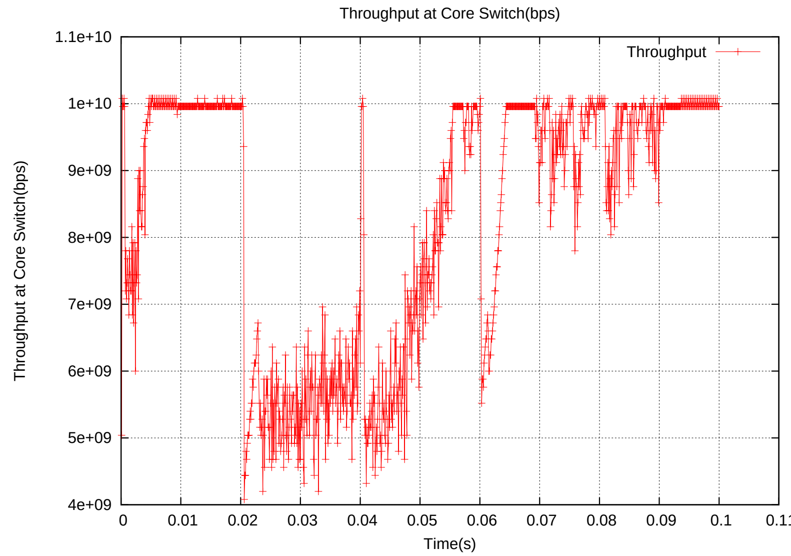
- **Conclusions**:

1. Queue approaches q_{eq} and stays there.
2. There is no under utilization (zero queue).

Symmetric Topology: Link Utilization



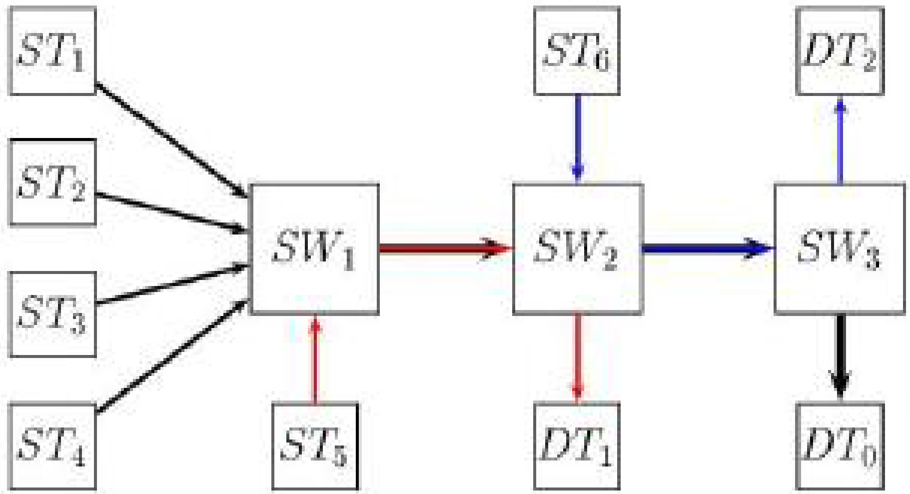
ECN



BCN

- **Conclusion**: ECN has much higher utilization

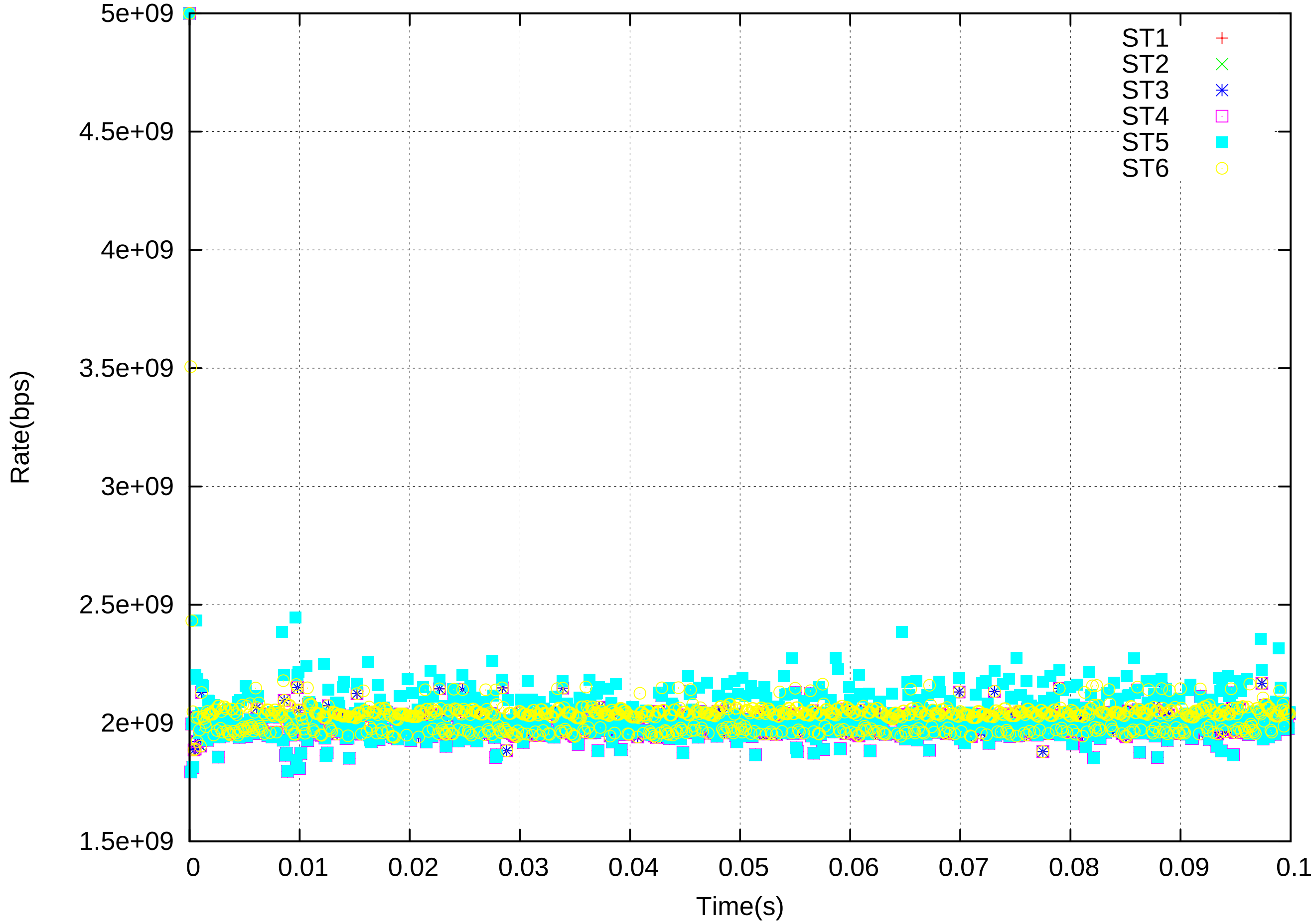
Parking Lot Topology



Goals:

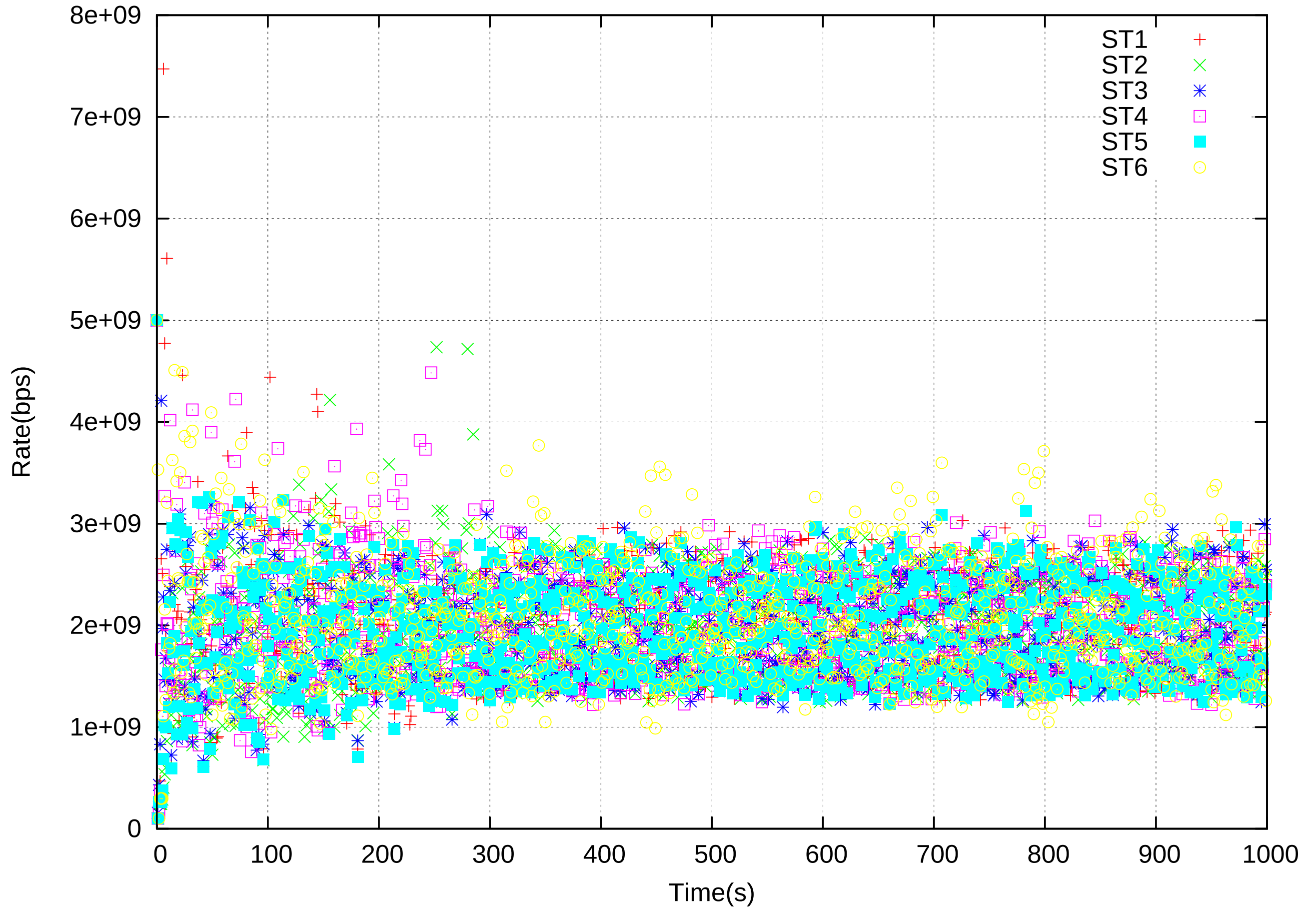
1. Check speed of convergence to fairness
2. Show that ECN gets Max-min (not proportional) fairness
   - Max-Min: All 6 sources get 1/5th of link rate
   - Proportional: ST1-ST4 get 1/6th and ST5-ST6 get 1/3rd

Parking Lot: Source Rates for ECN



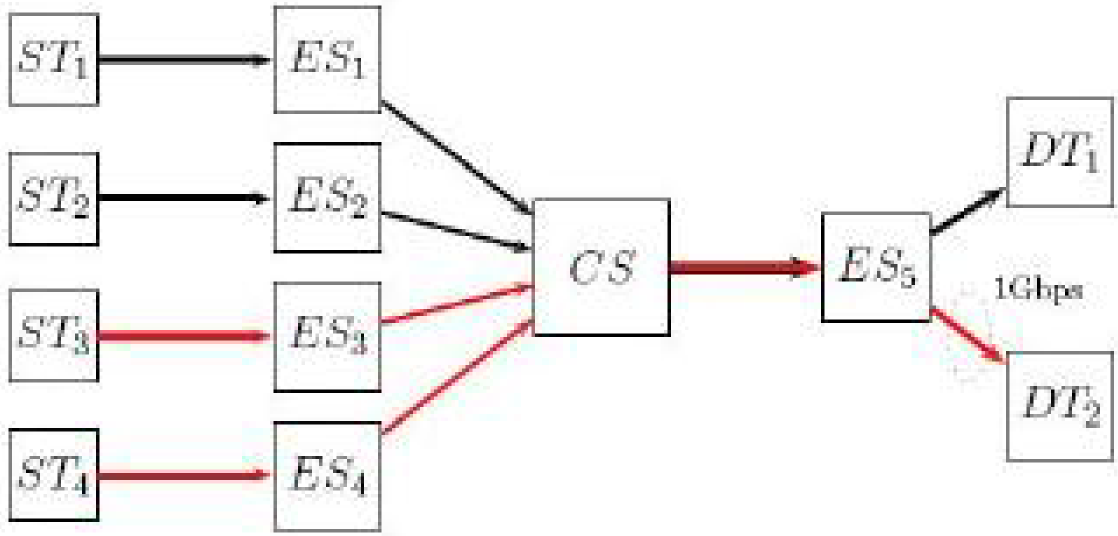
- Conclusion: All sources get 2 Gbps = C/5 => MAX-MIN Fairness

Parking Lot: Rates for BCN



- Large Oscillations

Simulation with Asymmetric Topology



Goal: Study multiple bottleneck case

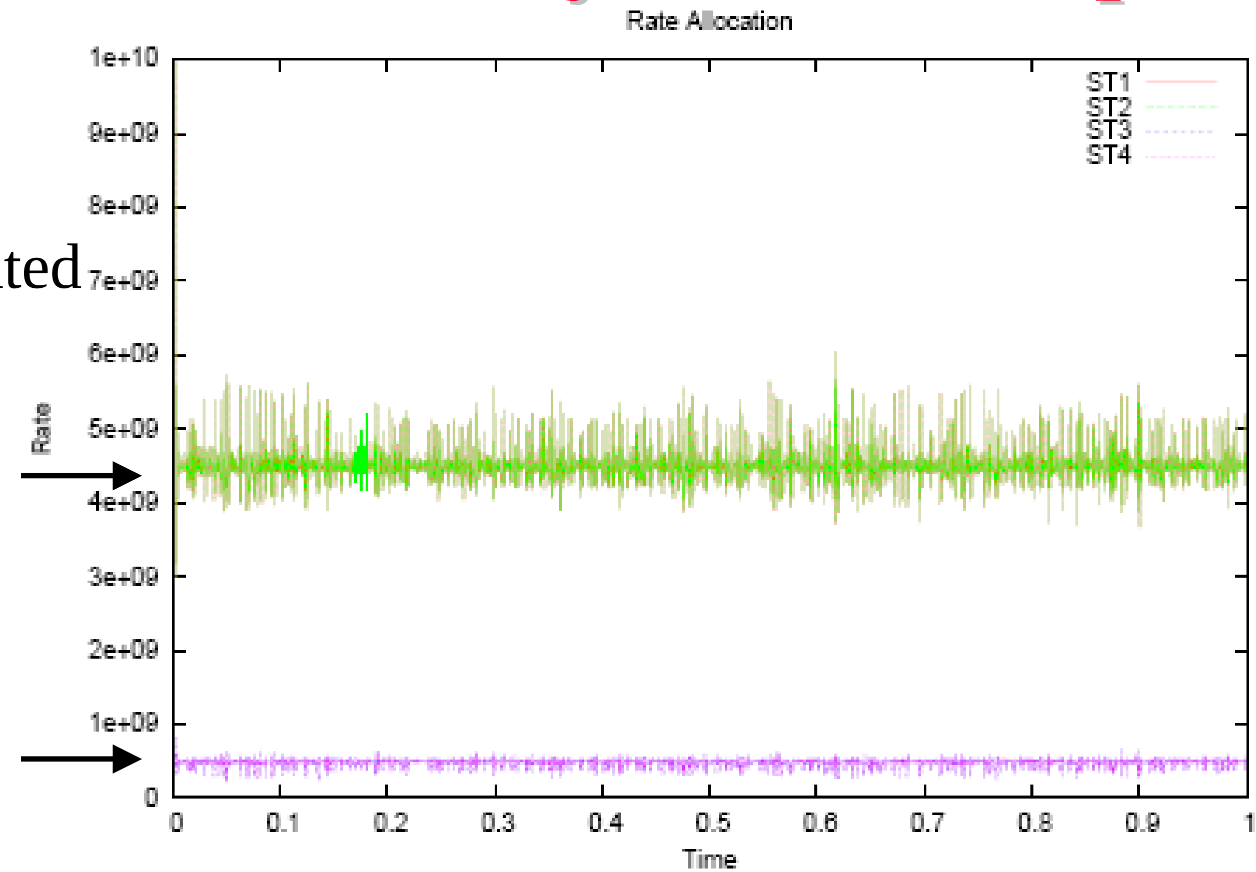
- Only one link is 1Gbps, others are all 10Gbps
- Two sources should converge to 5 Gbps and Two at 0.5 Gbps

Simulation with Asymmetric Topology

Rate Allocated

4.5G

0.5G

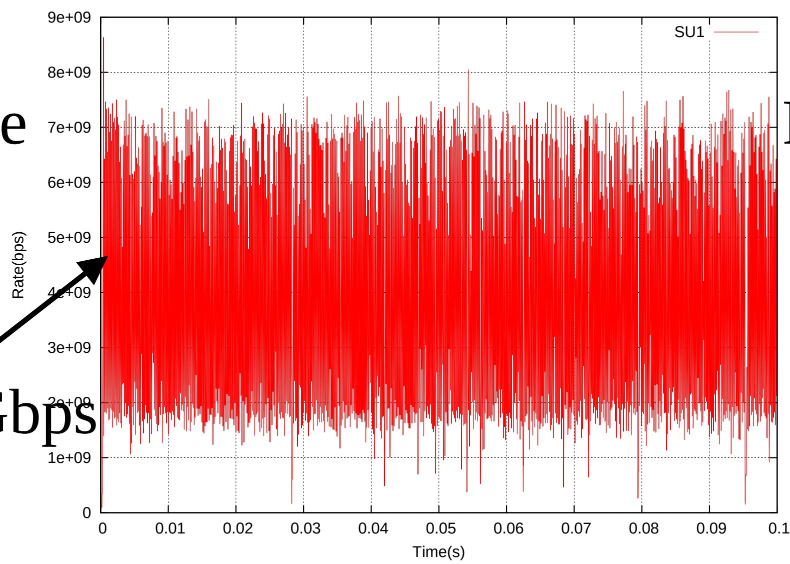


- **Conclusion**: ECN works perfectly with multiple bottlenecks. The rate variations are small.

Results for Baseline BCN

- Large Oscillations

Rate

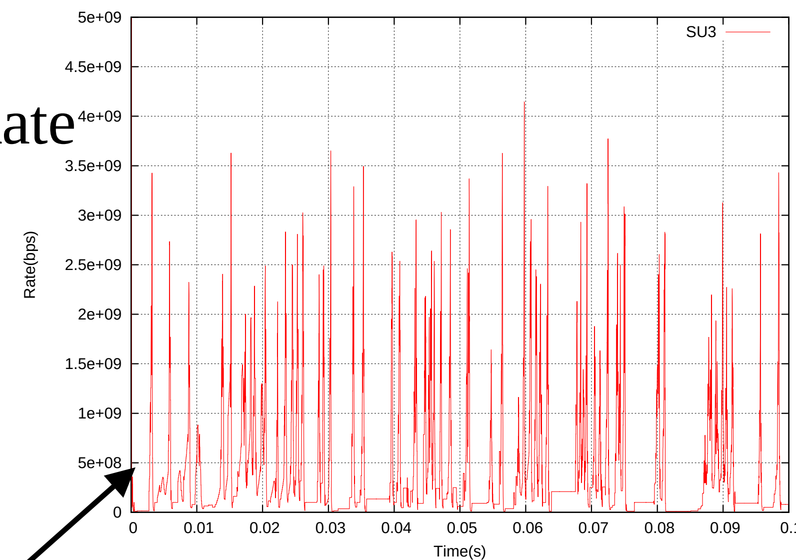


4.5Gbps

Time →

Optimal Rate
for ST1 and ST2

Rate

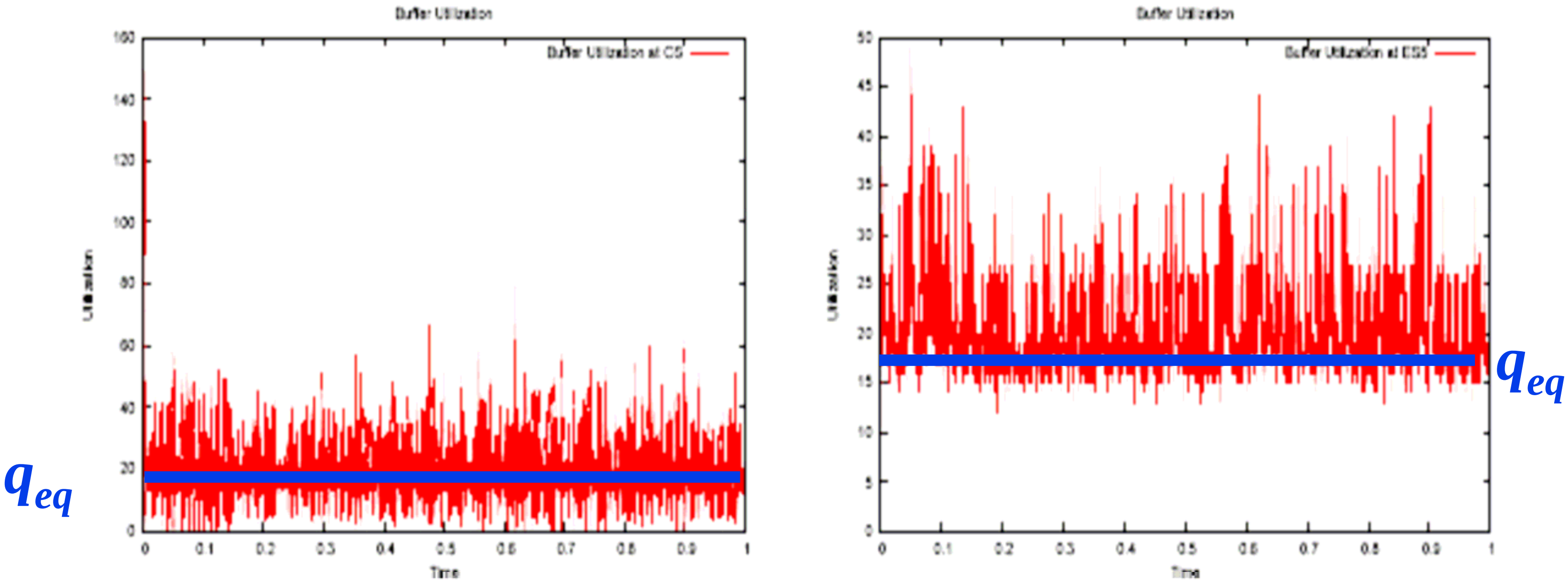


Time →

0.5Gbps
Optimal rate for
ST3 and ST4

Asymmetric Topology: Queue Lengths

- Buffer Utilization at two congestion points with ECN

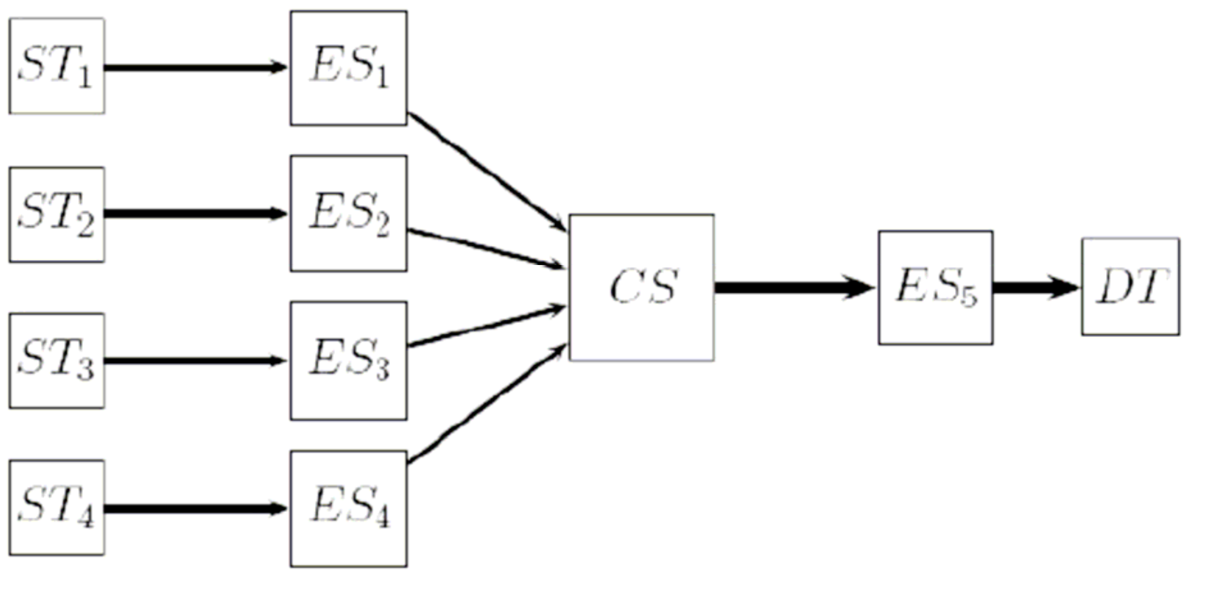


(a) Buffer Utilization at CS (b) Buffer Utilization at ES5

ECN BCN

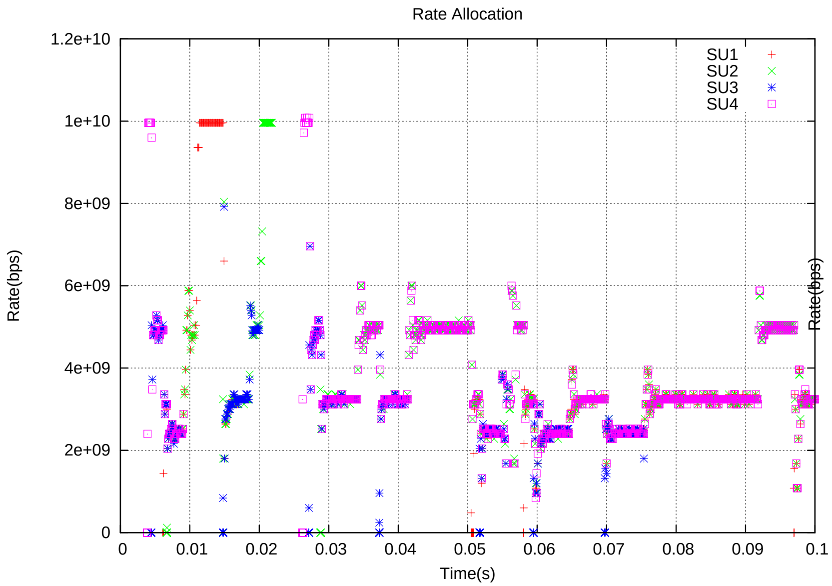
- **Conclusion**: Queues are stable in spite of different bottleneck rates.

Bursty Traffic: Configuration

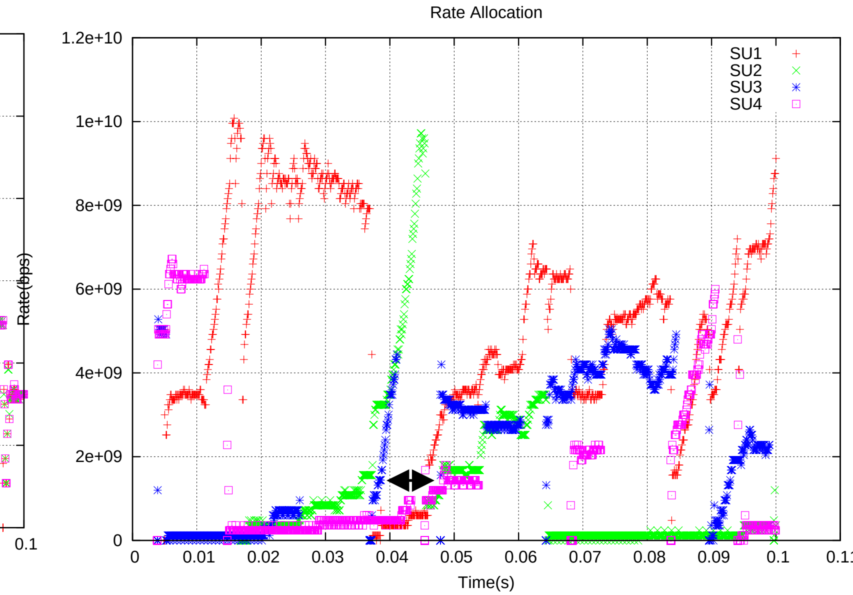


- On/Off UDP traffic with on/off periods taken from a pareto distribution
  - Average On/Off Time is *10* ms
  - Source rate at On Time is 10 Gbps

Bursty Traffic: Throughputs



ECN

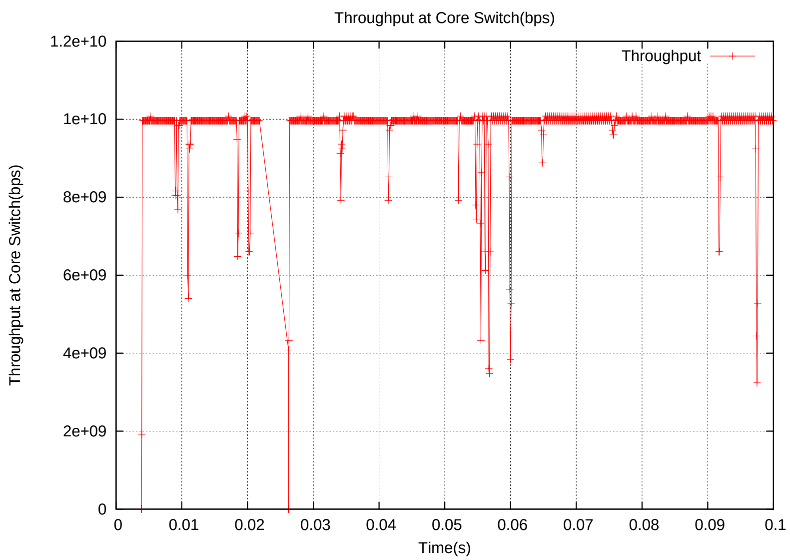


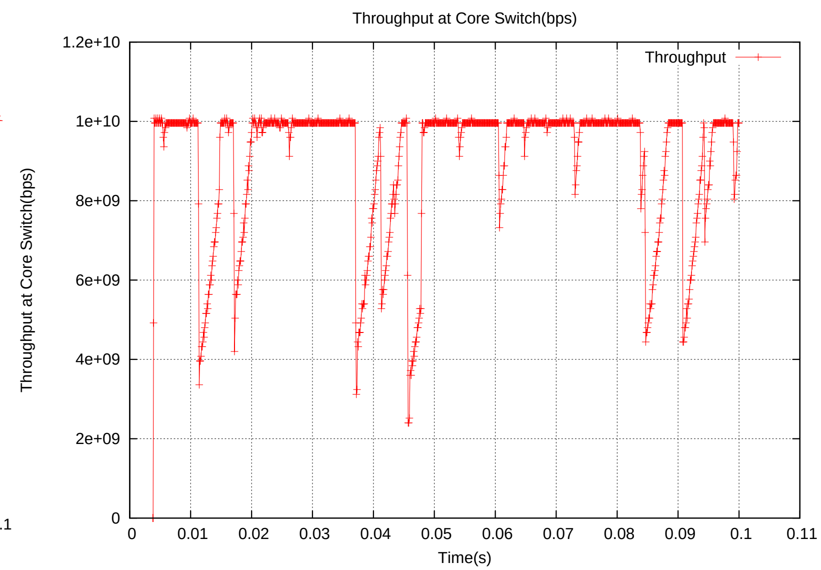
BCN

Conclusion:

- Four color curves are almost on the top of each other
  ⇒ ECN converges to fair state very fast

Bursty Traffic: Link Utilization



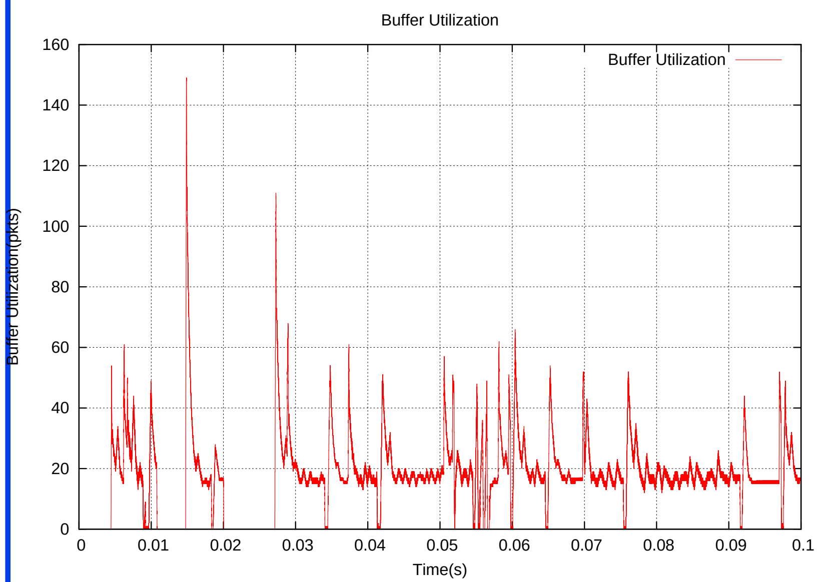


ECN

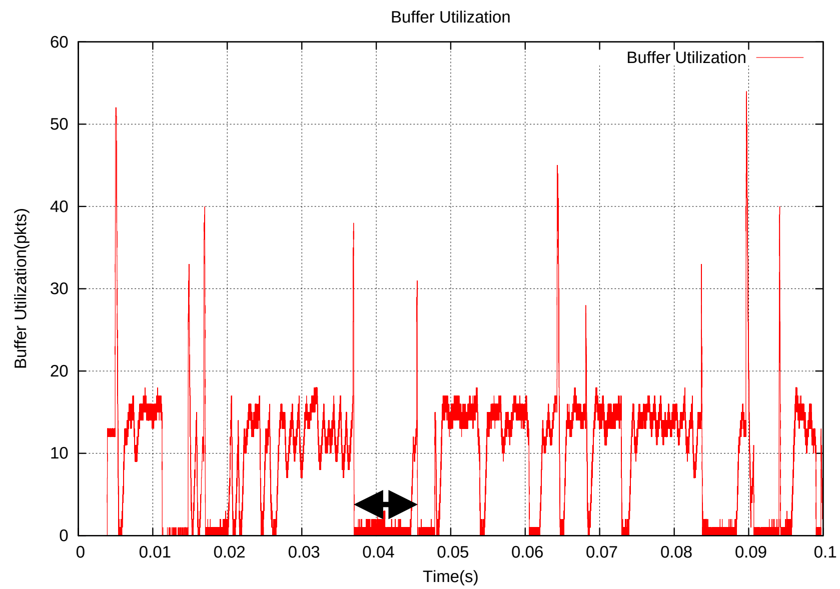
BCN

- **Conclusion**: ECN has higher link utilization

Bursty Traffic: Queue Lengths



ECN



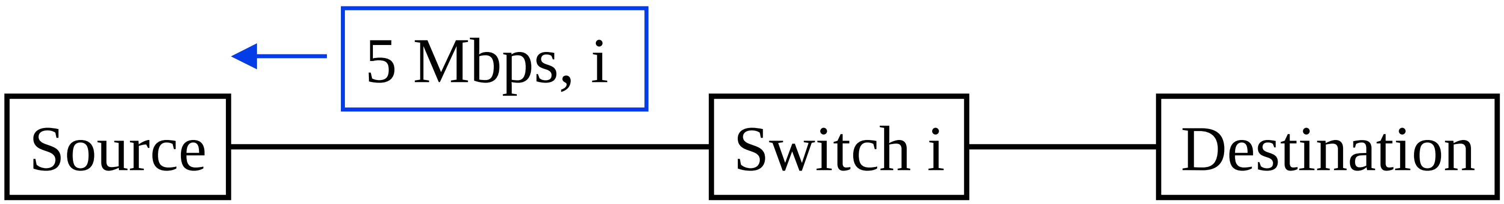
BCN

- **Conclusion**: ECN has less chances of zero queue => Higher link utilization

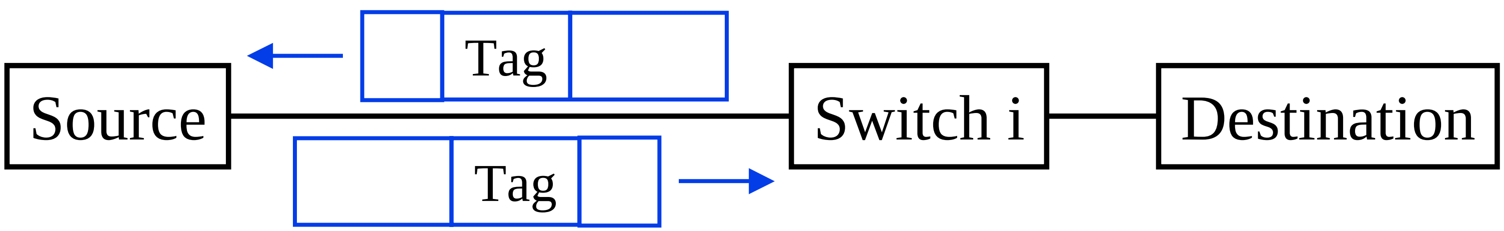
Variations of ECN

Choices for Congestion Notification

1. BCN messages: No tags required on the data packets. BCN contains rate and Congestion point ID (CPID).



2. Rate limiter tags: Works if you have ***bi-directional traffic***. The rates in the two directions do not have to be the same. No extra packets. Tags contains the rate (no CPID required).



3. Some combination of the two:
   e.g., tags, BCN messages periodically

RLT Tag Marking Algorithm

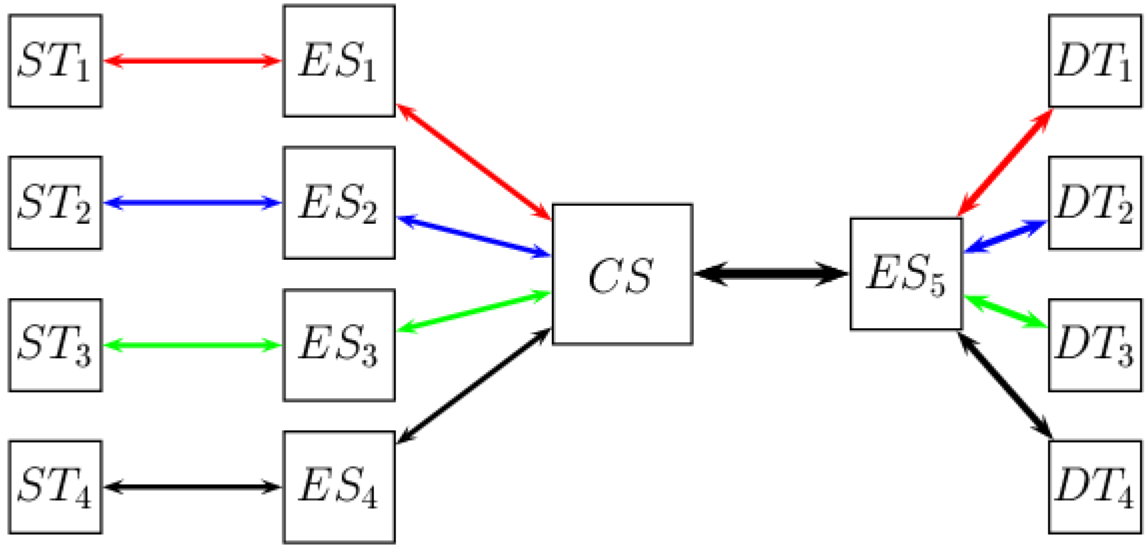
- Tags always start with r = -1 (infinite rate)
- Switch Marking Algorithm (Updates RLT Tags in all packets with the "advertised rate" of the reverse direction)

  IF r = -1 or $r > r_i$ THEN $r \leftarrow r_i$

  Note that tags do not need to contain CPID.
- Source Algorithm:

  $r \leftarrow r_i$

Symmetric Topology: 2-Way Traffic



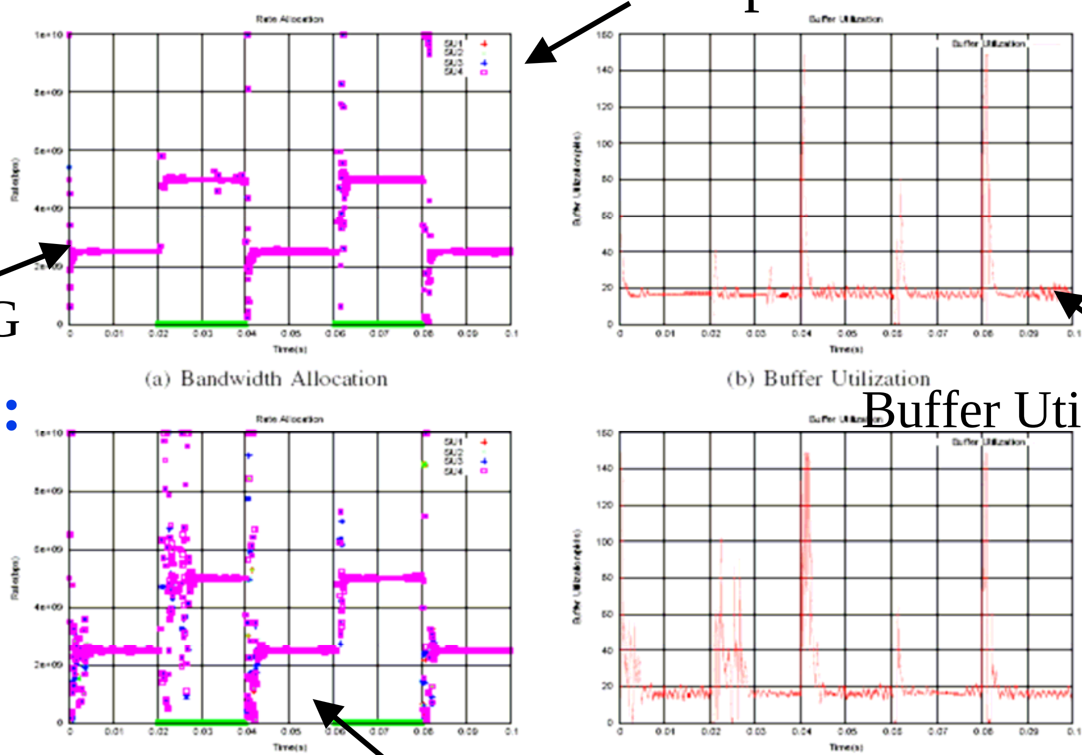
- Forward Traffic: UDP Bernoulli Traffic with 10 Gbps/source
- Reverse traffic: Case A: 2 Gbps/source
  Case B: 500 Mbps/source
- Forward ST1 and ST2 are periodically turned off for around 20 ms, i.e., the exact time is not the same
- Simulation Time is *100* ms

Symmetric Topology: RLT Tag Method

- **Case A:**

2G per source on reverse link

Rate=2.5G



(a) Bandwidth Allocation (b) Buffer Utilization

Buffer Utilization = 16pkts

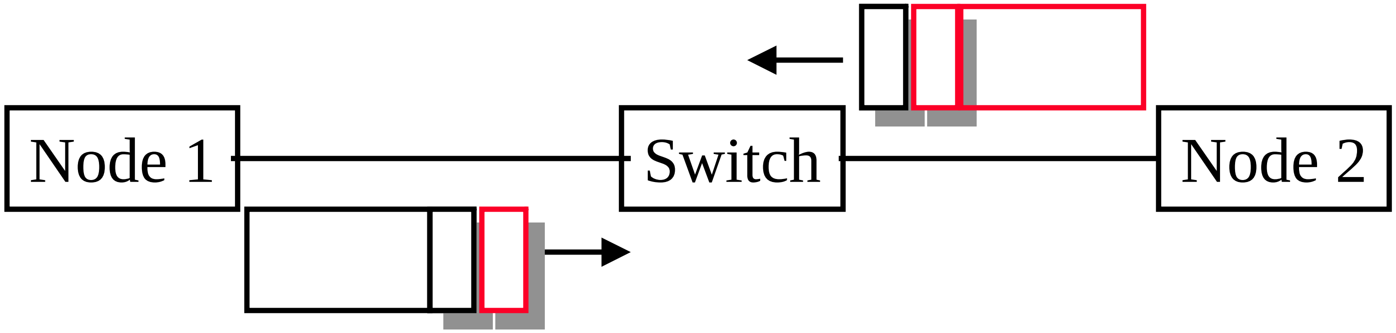
- **Case B:**

(c) Bandwidth Allocation (d) Buffer Utilization

0.5G per source on reverse link

- **Conclusion:** RLT tag performance similar to BCN messages.

FECN

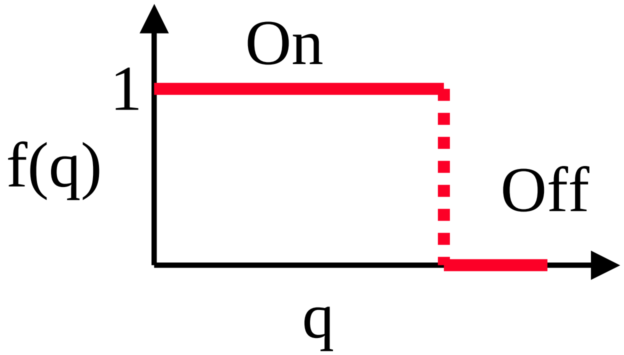


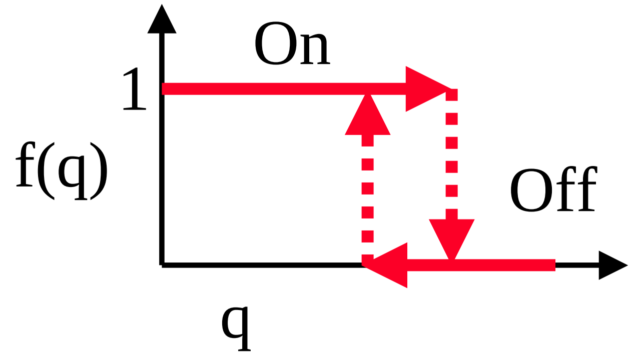
- Every n^{th} packet has two RLT tags (forward RLT tag and reverse RLT tag)
- The sender initializes the forward RLT tag
- The receiver copies the forward RLT tag in packets in the reverse direction on the same flow
- Source adjusts to the rate received
- The tags contain the rate and flow id

Advantages of ECN

- Flexibility:
  - Switches can base rates on resources other than one queue, e.g., sum of input and output queues, utilization of shared buffers, # of channels available on a wireless link, etc.
  - Switches can give different rate to a flow based on traffic type, class of service, types of sources, VLANs
- Works perfectly on variable link speeds, e.g., wireless links
- Vendor differentiation

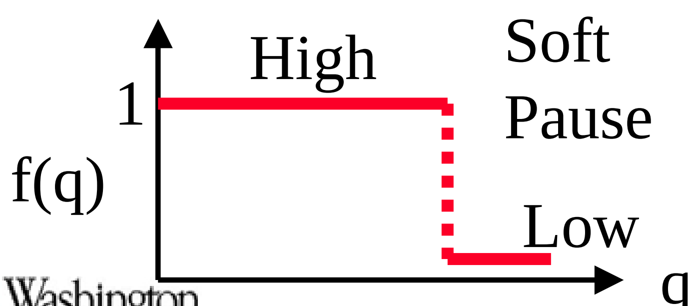
Pause and Queue Control Fn

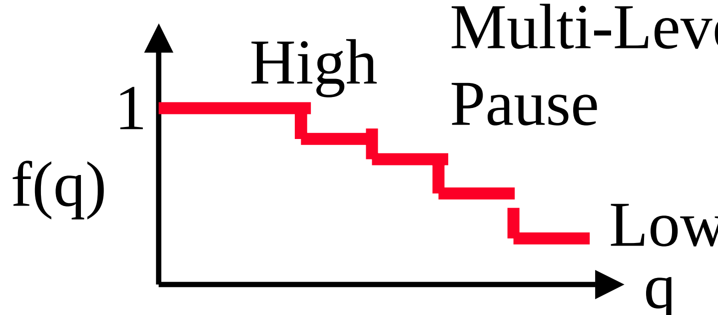




Pause with Hysterisis

- Pause is a special queue control function as shown above
- Using High/Low rather than on/off may avoid deadlocks
  ⇒ Multi-level Pause or soft-Pause
- Queue control idea is independent of ECN

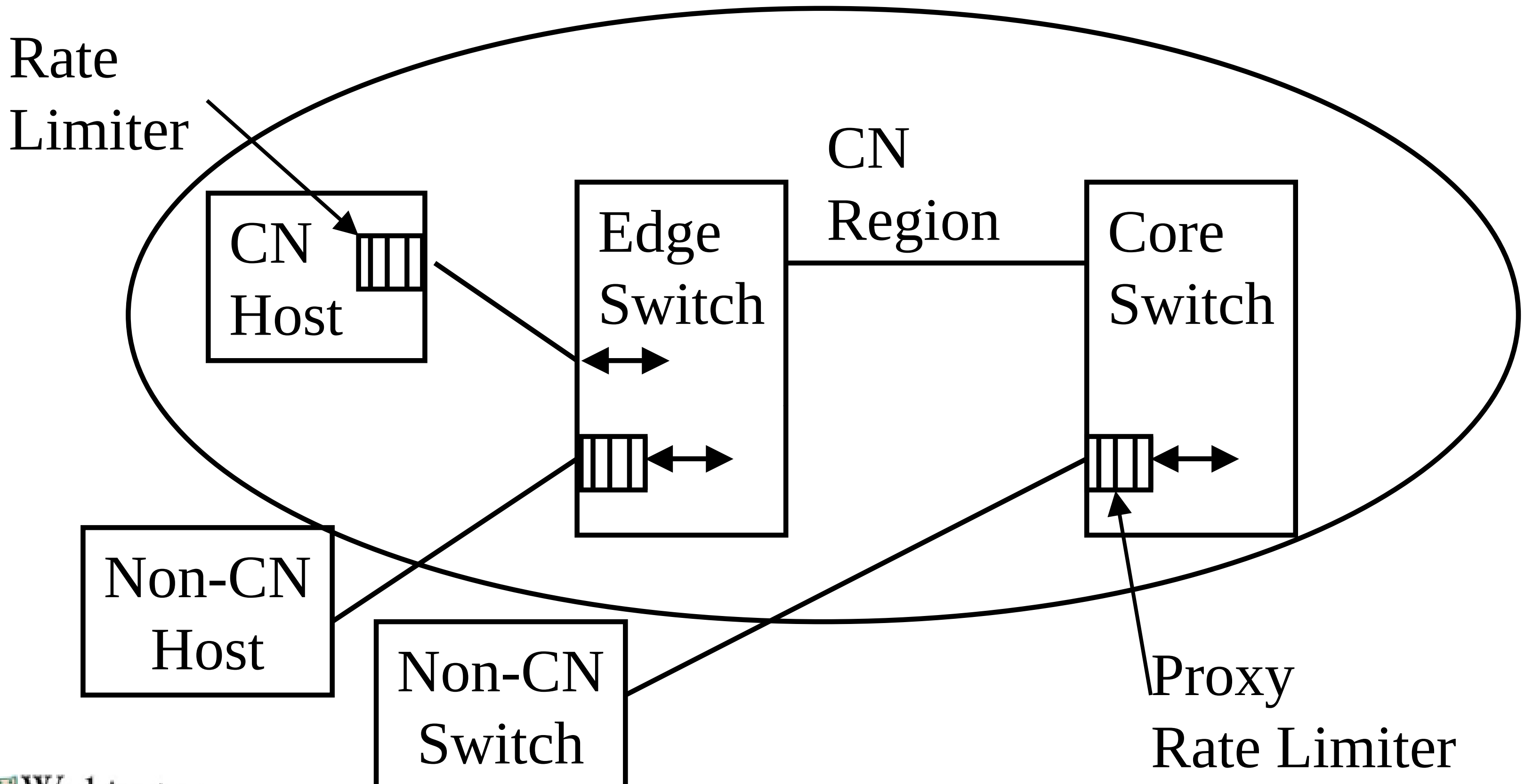




Multi-Level Pause

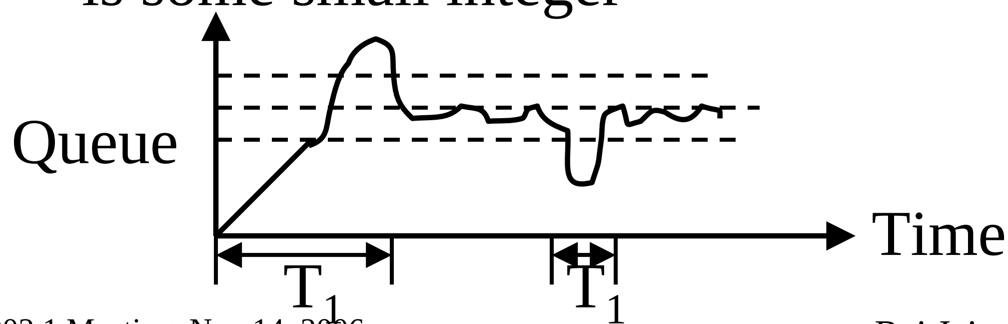
CN vs Non-CN Regions

- Edge switches could provide proxy Rate limiting queues



Adaptive Measurement Interval

- The load measurement interval T can be fixed or variable.
- Different switches can even use different T values or one switch can change its interval arbitrarily.
- Smaller T => Quick control but higher overhead if BCN messages (no effect on overhead if RLT tags)
- One possibility is to use large T when operating near the optimal and use small T when away from the optimal $T_1 < T_2$. Initially we set $T = T_1$.
  - If $|q_{eq}-q_i| < \delta$, $T = T_2$, δ is some small integer
  - *Otherwise*, $T = T_1$



Summary

#	Feature	BCN	FECN
1.	Convergence Time	Depends upon Sampling Interval	Depends upon measurement Interval
2.	Convergence Time	A probabilistic multiple of sampling interval =¿ Long	A small fixed multiple of measurement interval =¿ Short
3.	Convergence to Fairness	Slow	Fast
4.	Bursty Traffic	Fairness may not be achieved	Fairness is achieved quickly
5.	Bursty Traffic	Lower link utilization	Higher link utilization
6.	Control Overhead	BCN Messages and RLT Tags	RLT tags
7.	Source Parameters	Gi, Gd, W, Ru	Nothing

Summary (Cont)

#	Feature	BCN	FECN
8.	Source Algorithm	Complex (rate computation, drift, RLT tags)	Simple
9.	Switch Parameters	Qoffset, Qdelta, Sampling size, jitter, Qsc, Qeq	Measurement Interval, Qeq, Q-control fn
10.	State	Qdelta	Arrival rate, No state in switch. Destinations turn around tags.
11.	Sensitivity to parameters	Sensitive to sampling size	Not very sensitive
12.	Pause	Extra implementation	Part of Q control fn
13.	Vendor Differentiation	# of RL Queues in NIC	# of RL queues in NIC and Rate computation algorithm

References

- Bobby Vandalore, Raj Jain, Rohit Goyal, Sonia Fahmy, "Dynamic Queue Control Functions for ATM ABR Switch Schemes: Design and Analysis," Computer Networks, August 1999, Vol. 31, Issue 18, pp. 1935-1949. http://www.cse.wustl.edu/~jain/papers/cnis_qctrl.htm

Thank You!

Hvala

Grazie

धन्यवाद

Bedankt

Teşekkürler

Díky

Köszönettel

ขอบคุณ

شكراً

Ευχαριστώ

Merci

Gracias

Obrigado!

Vielen Dank

תודה