Data Center Ethernet Congestion Management: Backward Congestion Notification

Cisco.com

Davide Bergamasco (davide@cisco.com)

Cisco Systems, Inc.

IEEE 802.1 Interim Meeting

Berlin, Germany

May 12, 2005

Contributors & Supporters

- **Valentina Alaria (Cisco)**
- **Andrea Baldini (Cisco)**
- **Flavio Bonomi (Cisco)**
- **Manoj K. Wadekar (Intel)**

Need for Layer 2 CM in DCE

- TCP ECN may not be sufficient
  - Not widely deployed or used
  - Performance issues
    - ECN does not perform well with small buffers
    - Timeout in case of severe congestion
  - Fairness issues in presence of other protocols
- In the DC there may be other protocols besides TCP
    - NFS over UDP
    - Tibco uses multicast
    - Oracle uses UDP
    - Veritas Cluster does not even use IP
    - Proprietary L3 protocols

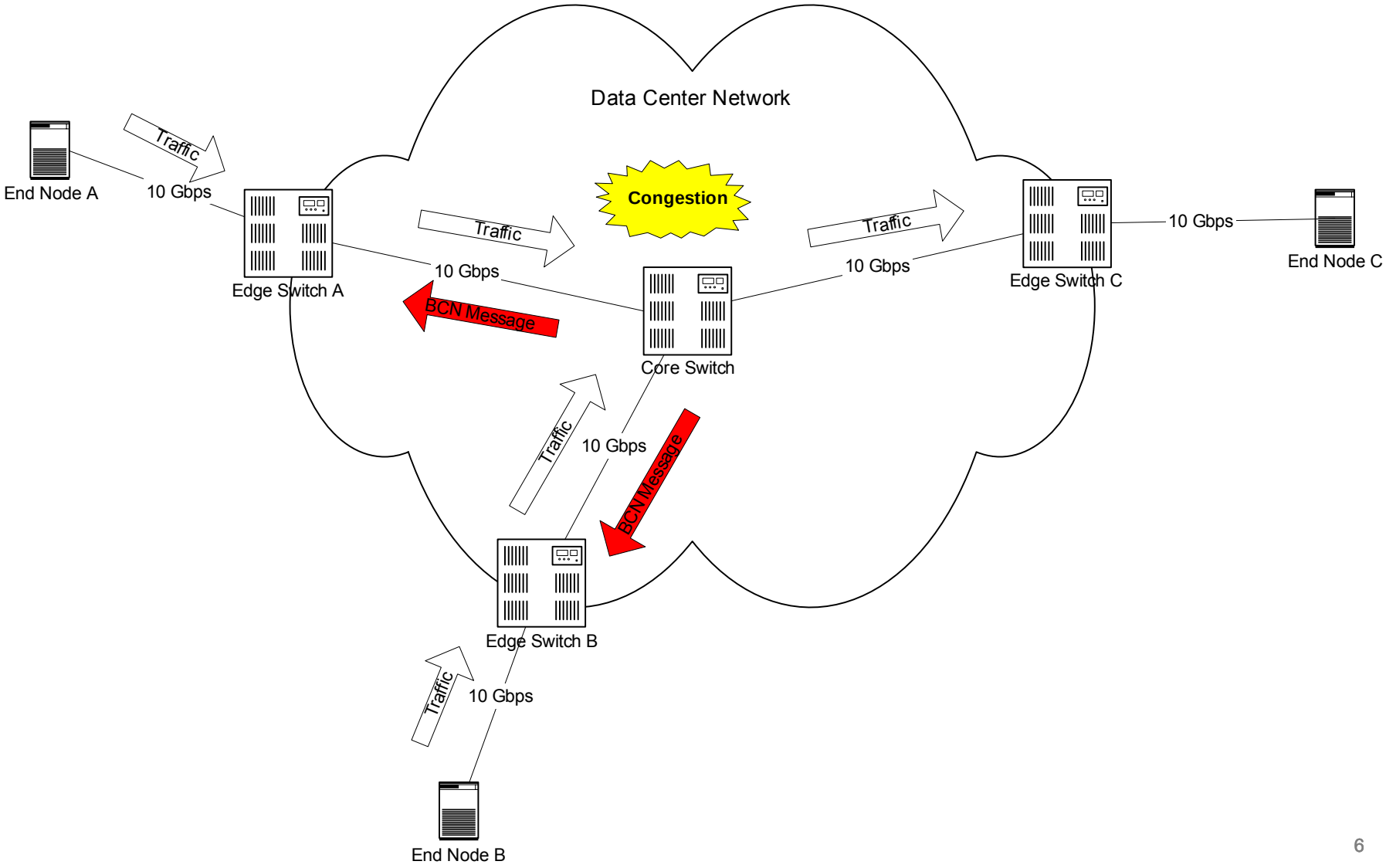
Backward Congestion Notification

- Principles
  - Push congestion from the core towards the edge of the network
  - Use rate-limiters at the edge to shape flows causing congestion
  - Tune rate-limiter parameters based on feedback coming from congestion points
- Inspired by TCP
  - AIMD rate control
    - TCP window increases linearly in absence of congestion
    - Decreases exponentially (gets halved) at every congestion indication (either implicit or explicit)
  - Self-Clocking Control loop (acknowledgements)

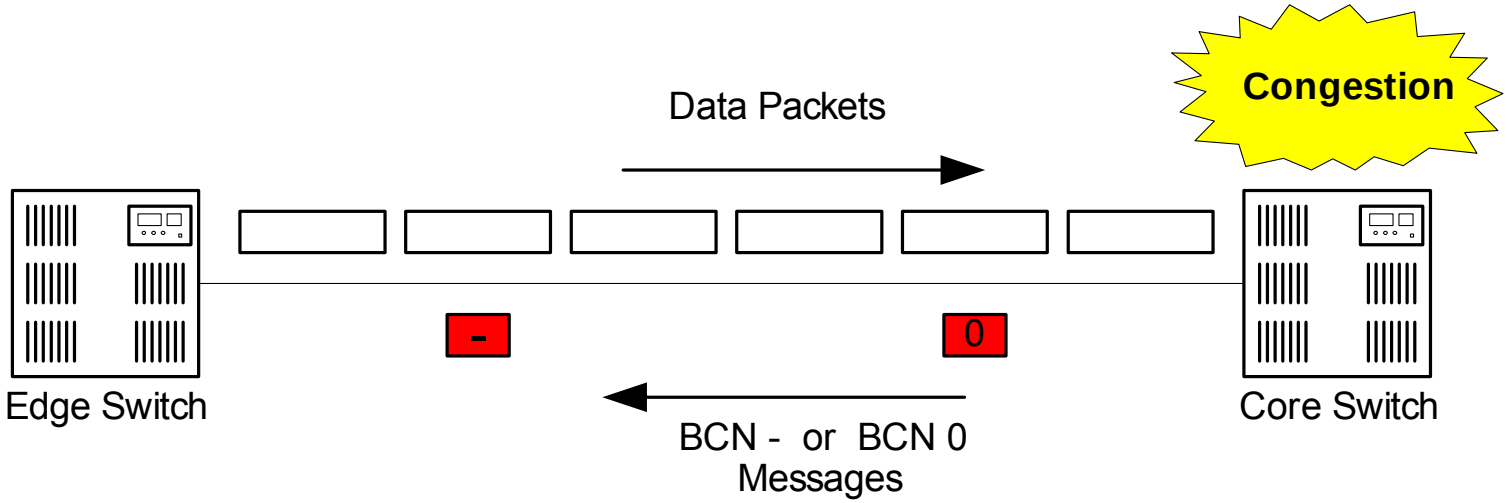
Backward Congestion Notification

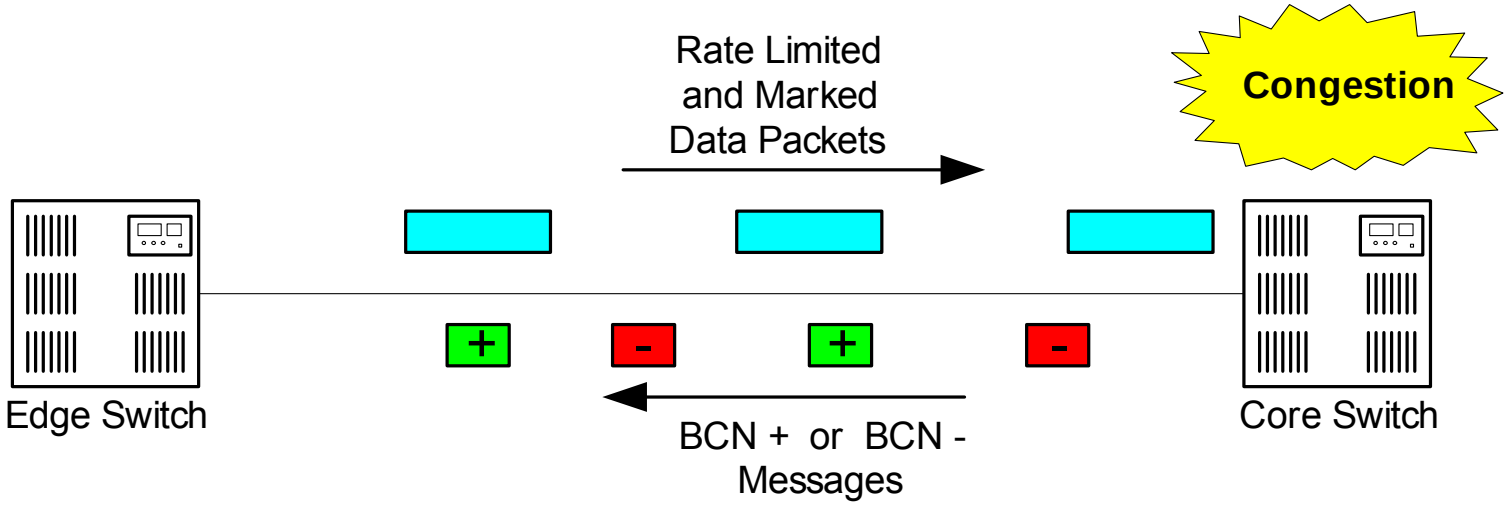
- **Advantages**
  - **Backward Congestion Notification (BCN) provides shorter control loop as compared with ECN forward marking**
  - **Supports non-TCP and non-IP protocols**
  - **L2 rate control triggered with backward congestion message may provide better “fairness” among DC flows**
- **Disadvantages**
  - **Overhead due to BCN messaging**

Backward Congestion Notification: General Concepts

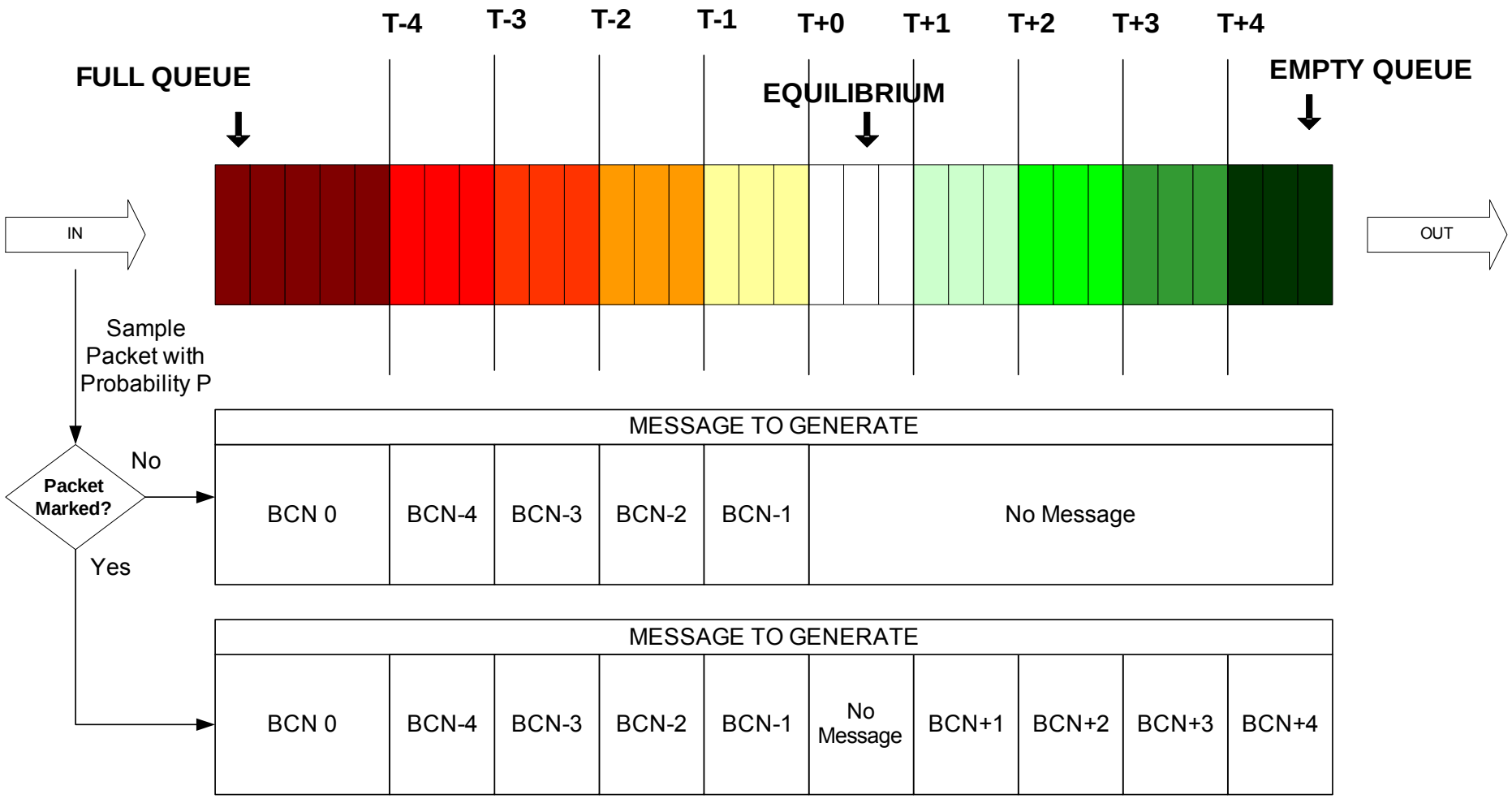


Backward Congestion Notification: Messaging

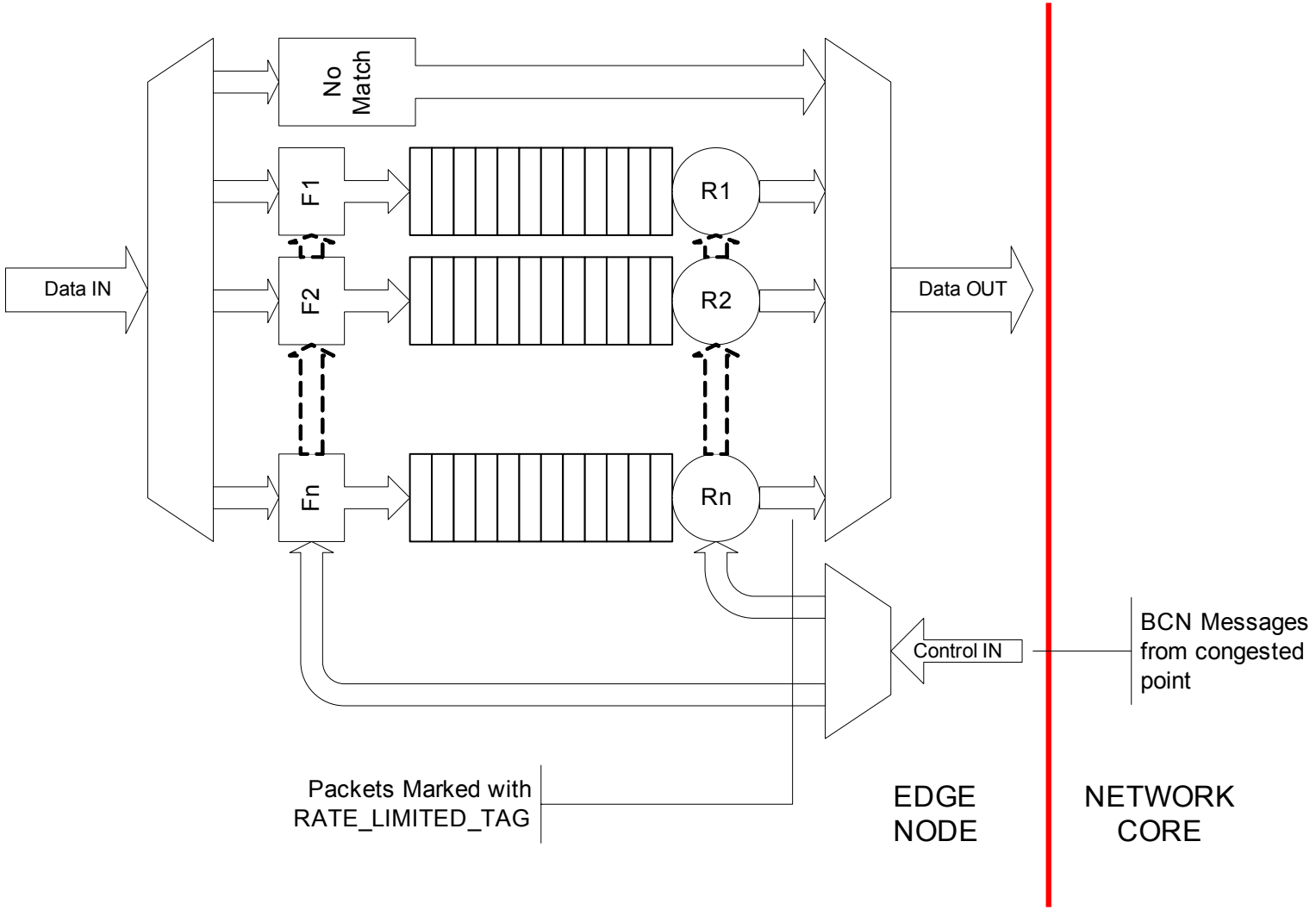




Backward Congestion Notification: Detection & Signaling

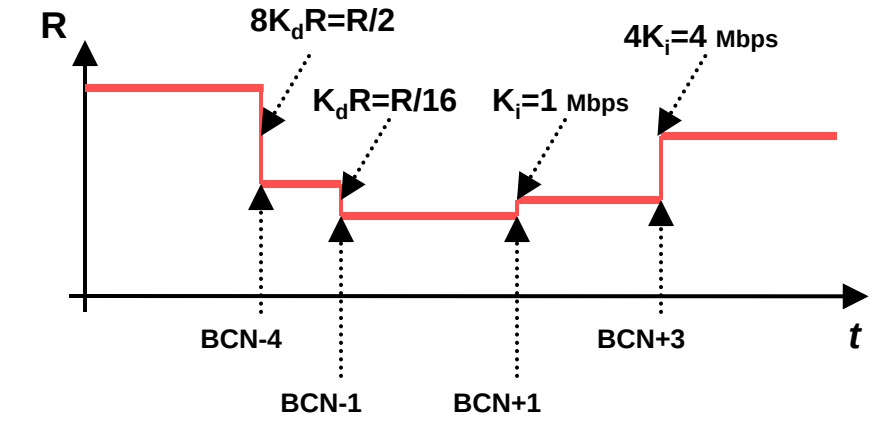


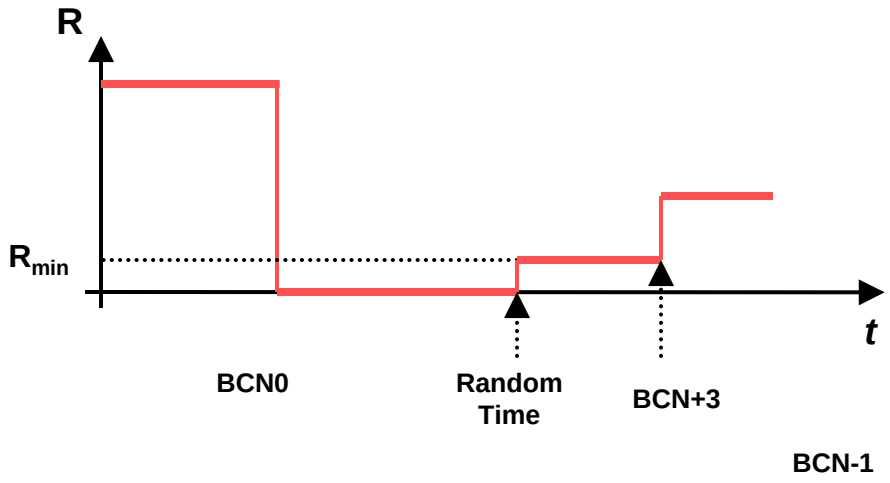
Backward Congestion Notification: Reaction (1)



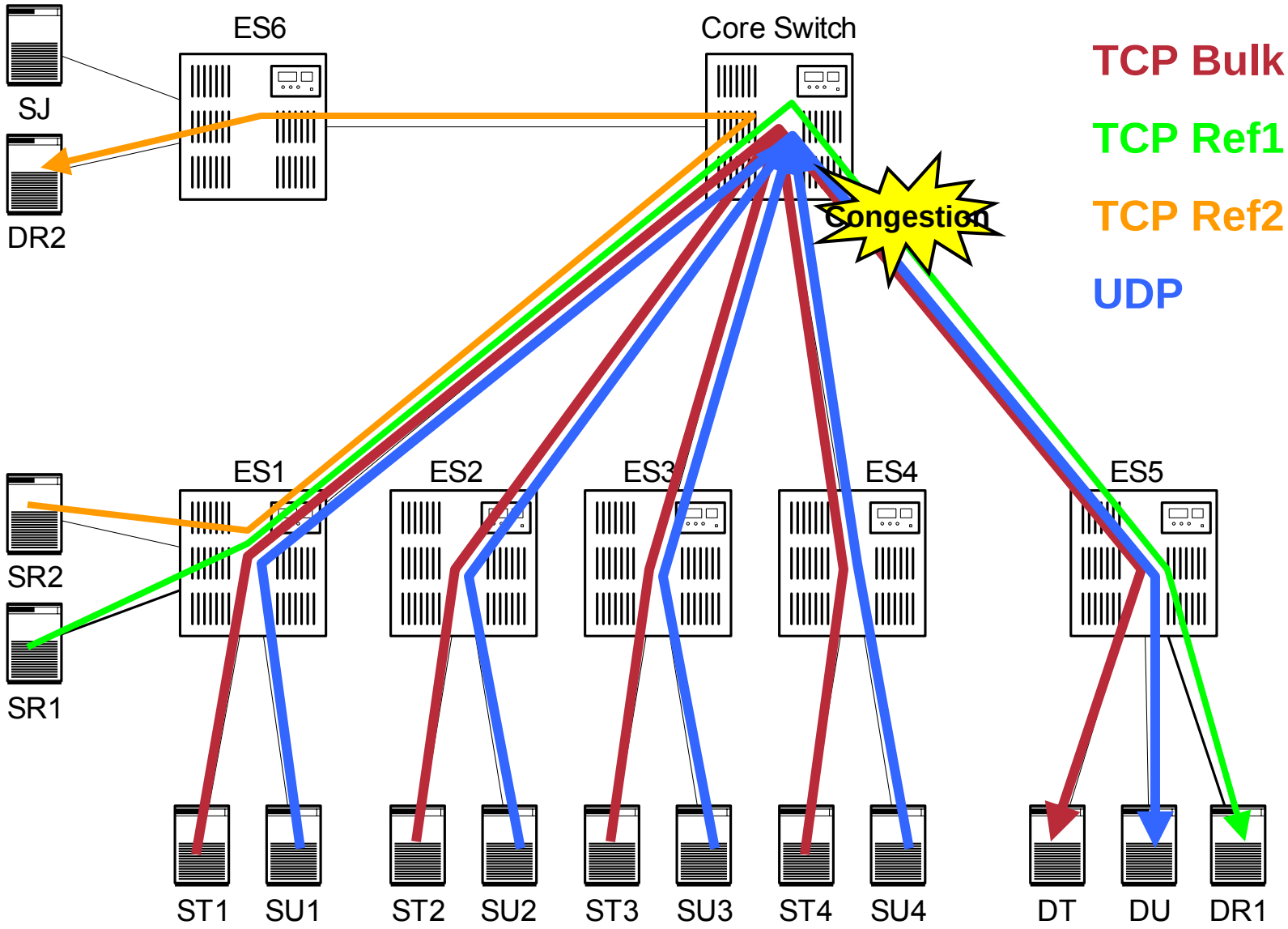
Backward Congestion Notification: Reaction (2)

Signal	Reaction
BCN+4	$R = R_{old} + 8K_i$
BCN+3	$R = R_{old} + 4K_i$
BCN+2	$R = R_{old} + 2K_i$
BCN+1	$R = R_{old} + K_i$
BCN-1	$R = R_{old} * (1 - K_d)$
BCN-2	$R = R_{old} * (1 - 2K_d)$
BCN-3	$R = R_{old} * (1 - 4K_d)$
BCN-4	$R = R_{old} * (1 - 8K_d)$
BCN 0	R = 0





Simulation Environment (1)



Simulation Environment (2)

- Short Range, High Speed DCE Network
  - Link Capacity = 10 Gbps
  - Switch latency = 1 μs
  - Link Length = 100 m (.5 μ s propagation delay)
  - Control loop delay ~ 3 μs
- Workload
  1) TCP only
     - ST1-ST4: 10 parallel connections transferring 1 MB each
     - SR1: 1 connection transferring 10 KB (avg 16 μs wait)
     - SR2: 1 connection transferring 10 KB (1μs wait)
  2) 80% TCP + 20% UDP
     - ST1-ST4: same as above
     - SR1-SR2: same as above
     - SU1-SU4: variable length bursts with average offered load of 2 Gbps

Simulation Goals

- **Compare congestion management mechanisms in a DCE environment**
  - **None**
  - **RED**
  - **TCP ECN**
  - **BCN**
- **Metrics:**
  - **Throughput, Fairness and Latency @ Bottleneck Link**
  - **Throughput and Latency of Reference Flows**
  - **Buffer Utilization**

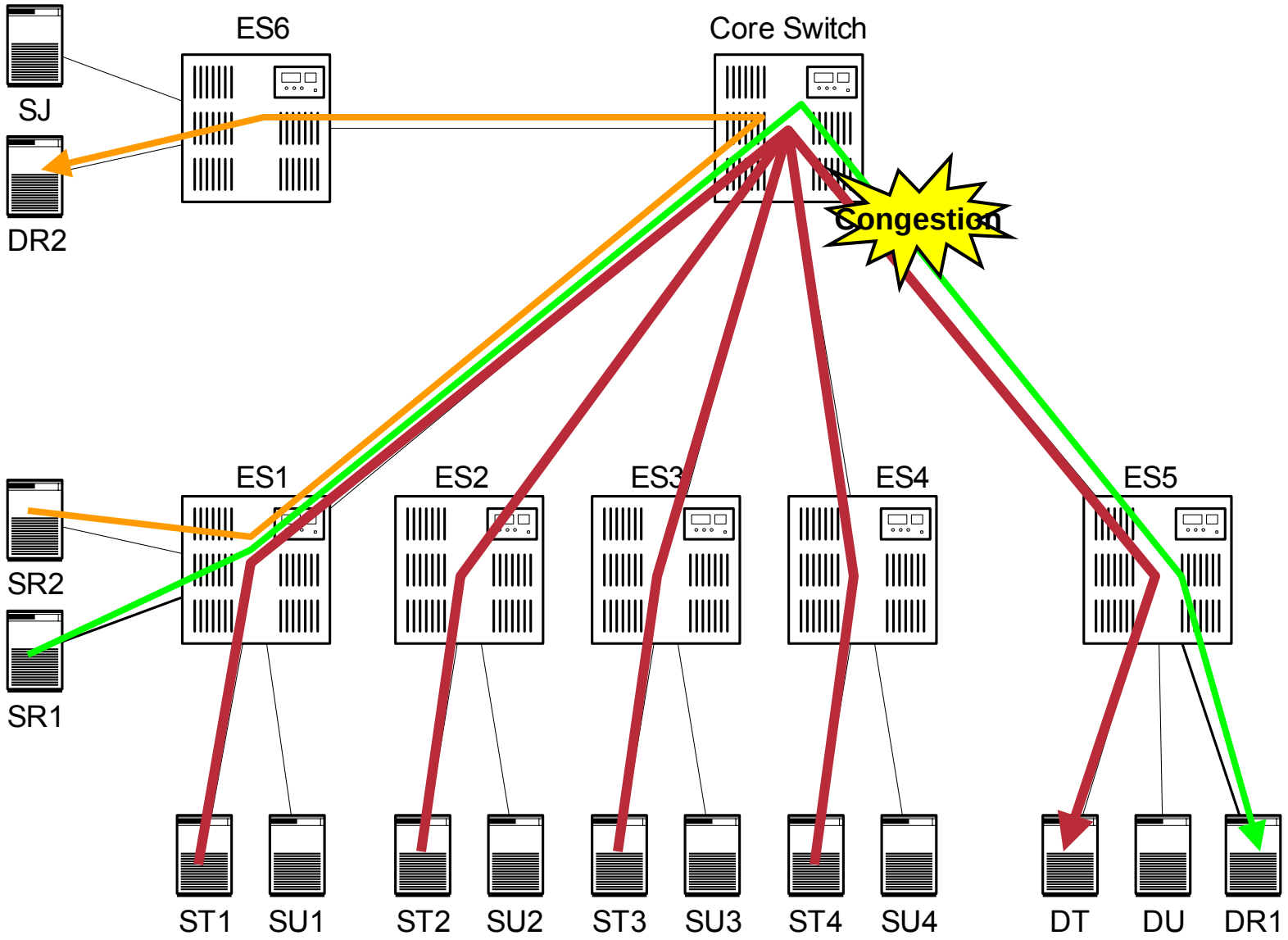
Application Throughput, Latency & Fairness @ Bottleneck Link (Workload 1: TCP Only)

Congestion Management Mechanism	Average Src Throughput (Tps)	Standard Deviation / Average (%)	Average Src Throughput (Gbps)	Standard Deviation / Average (%)	Throughput on Bottleneck link (Gbps)	Average Latency (μs)	Standard Deviation / Average (%)
None	30.00	0.00	2.486	0.73	10.000	32,570	0.00
RED	30.20	3.11	2.487	5.99	9.999	33,134	4.00
ECN	30.18	11.49	2.469	5.71	9.879	31,259	11.59
BCN	29.00	1.31	2.403	5.66	9.999	33,879	1.53

Best Worst N/A

- No clear winner here, however…

Simulation Environment



Application Throughput & Latency of Reference Flow (Workload 1: TCP Only)

	Reference Flow 1			Reference Flow 2		
Congestion Management Mechanism	Throughput (Tps)	Throughput (Gbps)	Latency (µs)	Throughput (Tps)	Throughput (Gbps)	Latency (µs)
None	609	0.05245	1,625	6,325	0.54476	157.100
RED	577	0.04984	1,665	32,050	2.76049	30.200
ECN	6	0.00052	3,596	32,261	2.77866	29,996
BCN	4,491	0.38680	206.394	31,515	2.71437	30.730

Best Worst N/A

- … BCN is the best at protecting fragile congested flow

Application Throughput, Fairness & Latency @ Bottleneck Link (Workload 2: TCP + UDP)

Congestion Management Mechanism	Average Src Throughput (Tps)	Standard Deviation / Average (%)	Average Src Throughput (Gbps)	Standard Deviation / Average (%)	Throughput on Bottleneck link (Gbps)	Average Latency (μs)	Standard Deviation / Average (%)
None	16.00	0.00	1.33774	1.51	10.000	60,524	0.18
RED	6.80	18.97	0.59337	116.75	9.900	141,850	1.04
ECN	0.93	9.46	0.12341	712.48	8.897	∞	N/A
BCN	17.25	1.73	1.43195	7.36	9.999	56,706	0.57

Best Worst N/A

- By properly dealing with UDP, BCN outperforms the rest

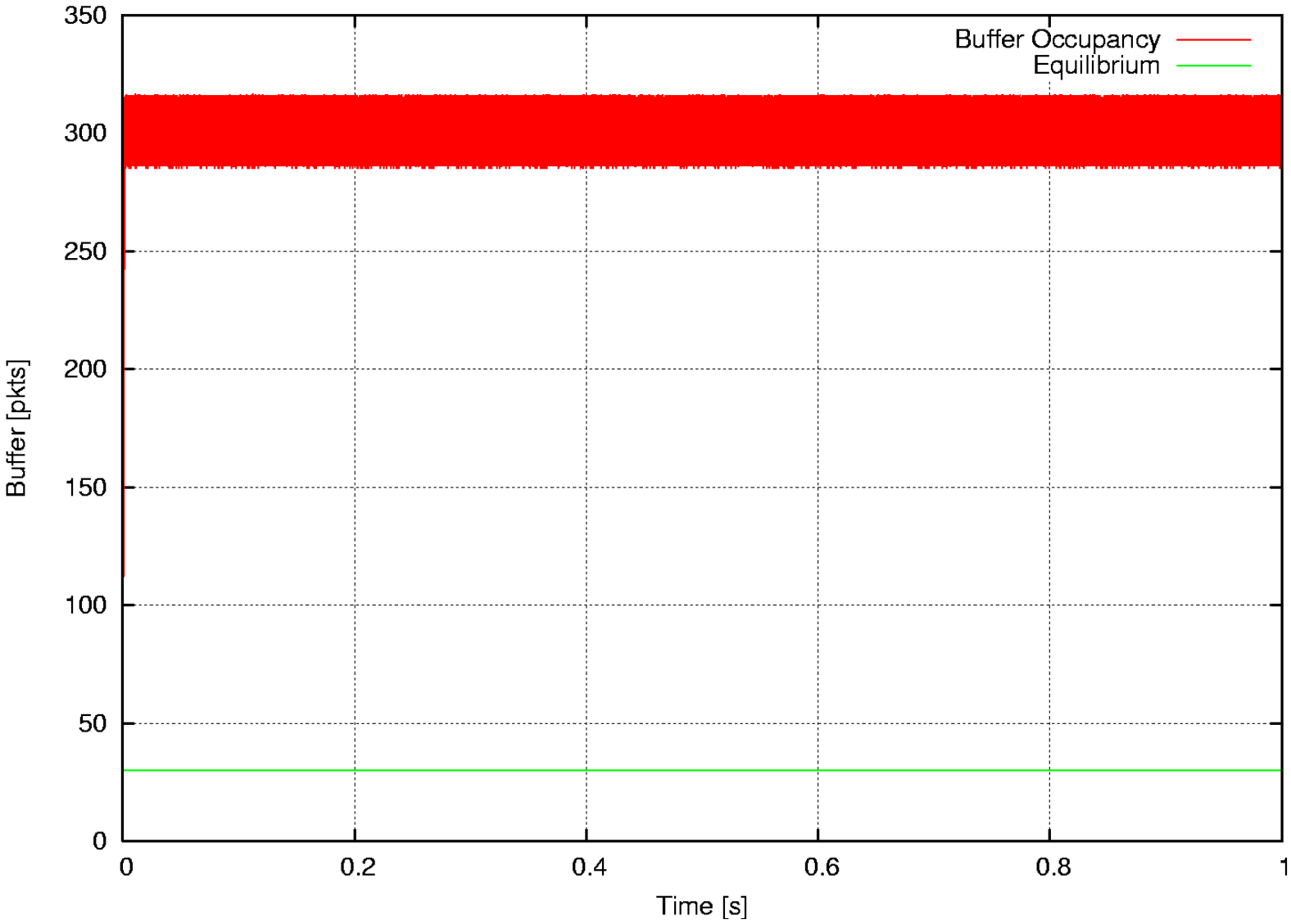
Application Throughput & Latency of Reference Flow (Workload 2: TCP + UDP)

	Reference Flow 1			Reference Flow 2		
Congestion Management Mechanism	Throughput (Tps)	Throughput (Gbps)	Latency (μs)	Throughput (Tps)	Throughput (Gbps)	Latency (μs)
None	625	0.05383	1,582	5,282	0.45493	188.301
RED	317	0.02757	3,071	31,536	2.71622	30.708
ECN	2	0.00017	10,552	32,216	2.77470	30.040
BCN	4,457	0.38388	272.065	31,588	2.72065	30.657

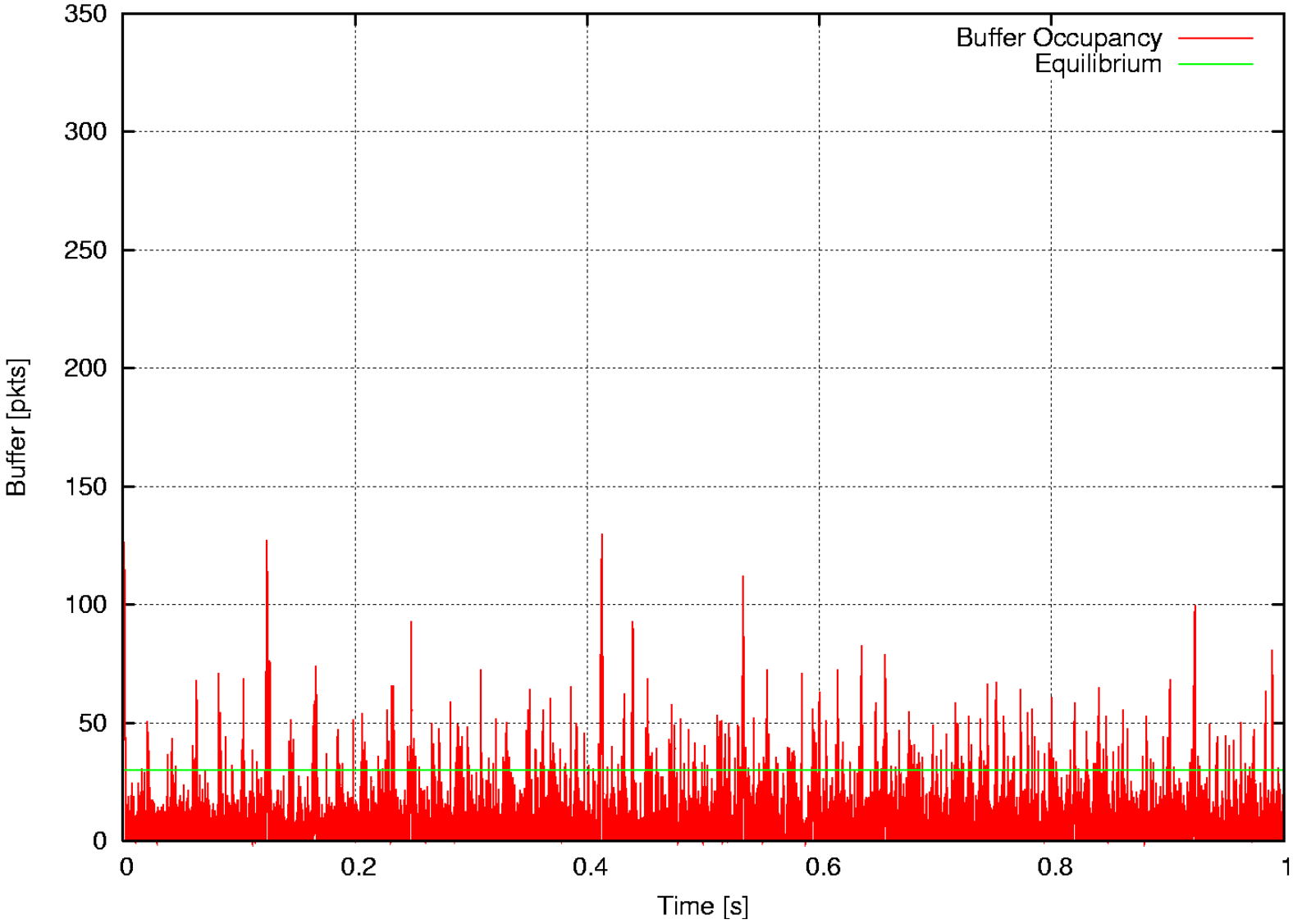
Best Worst N/A

- BCN is the best at protecting fragile congested flow

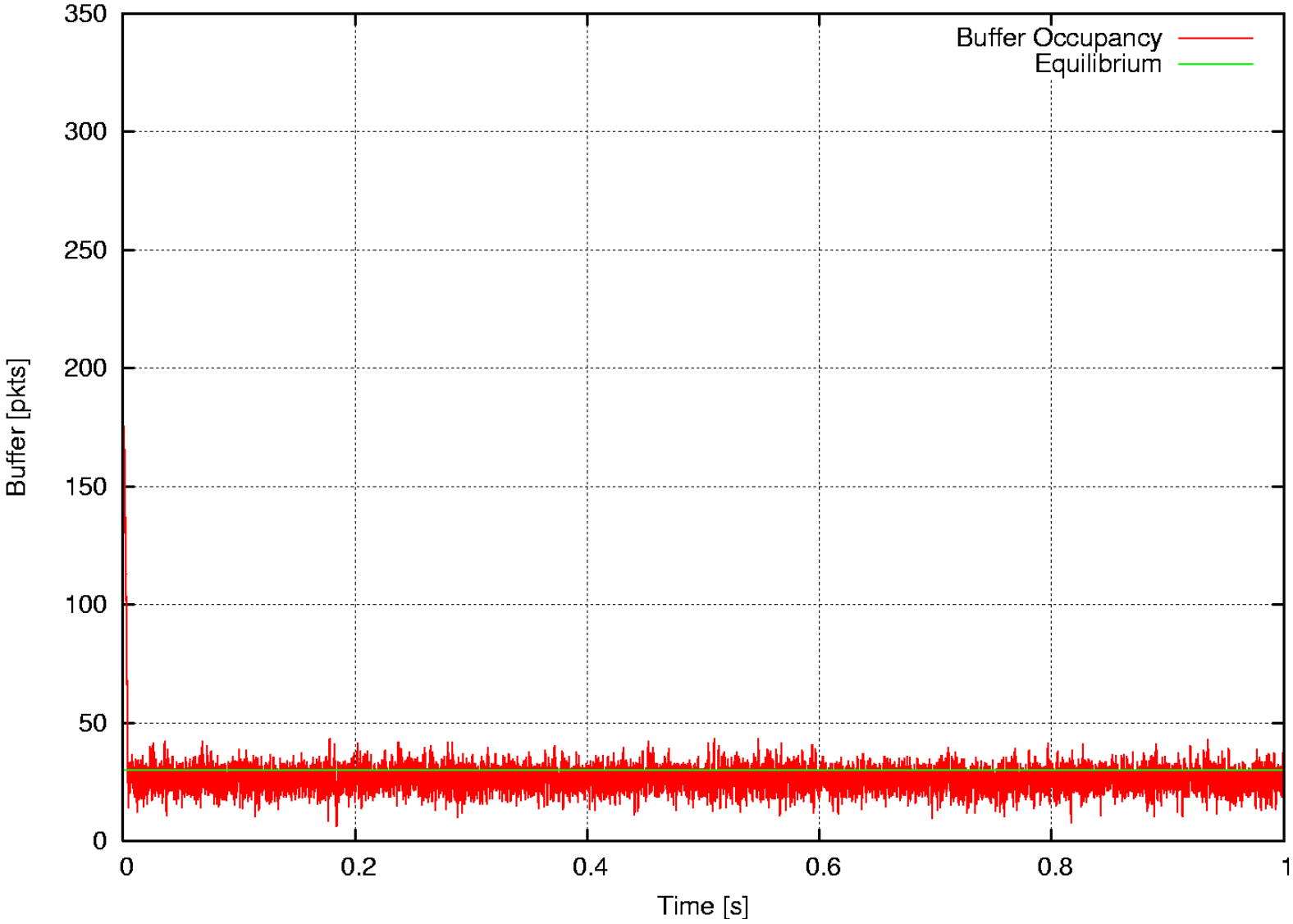
Buffer Utilization: No CM



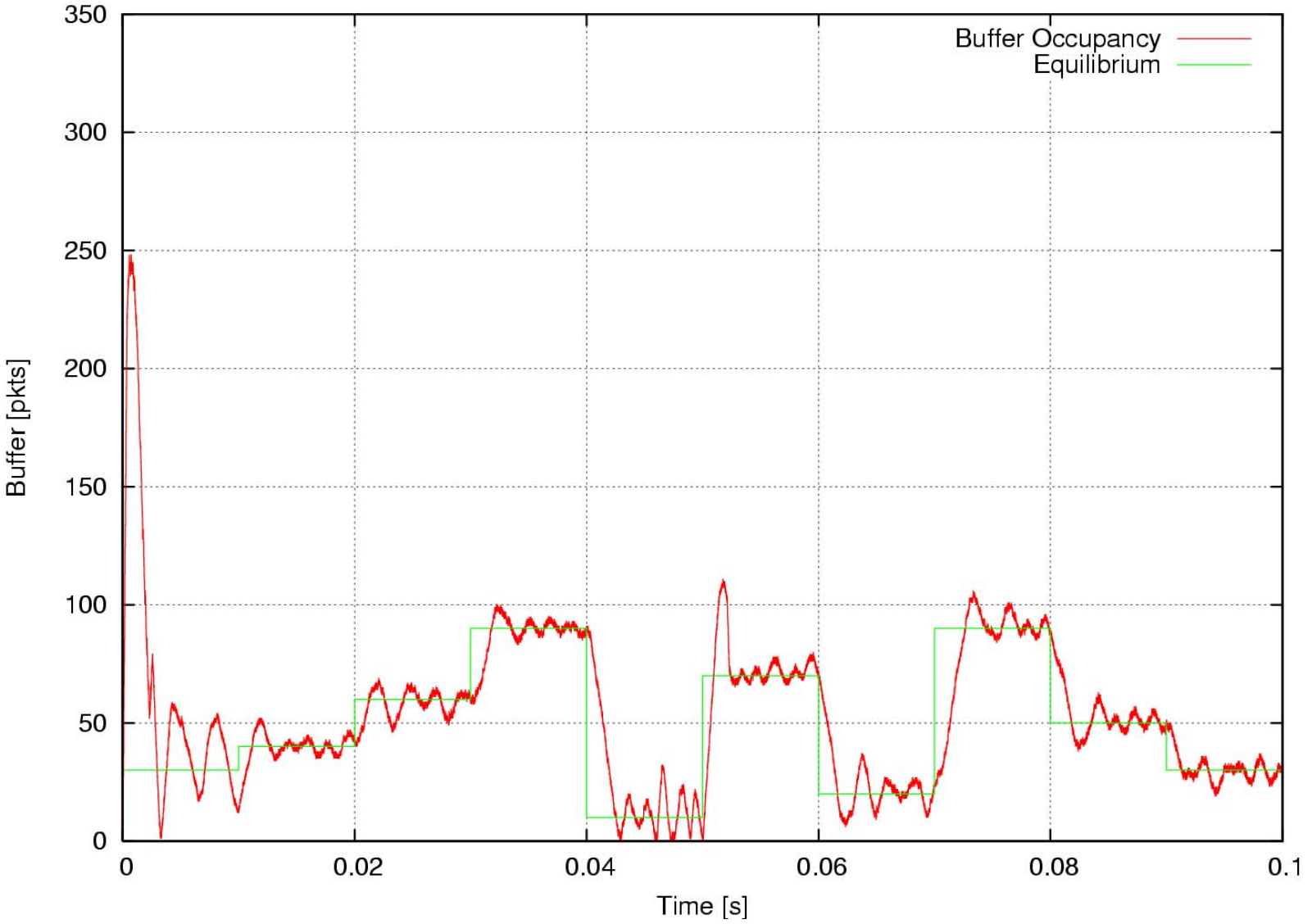
Buffer Utilization: ECN



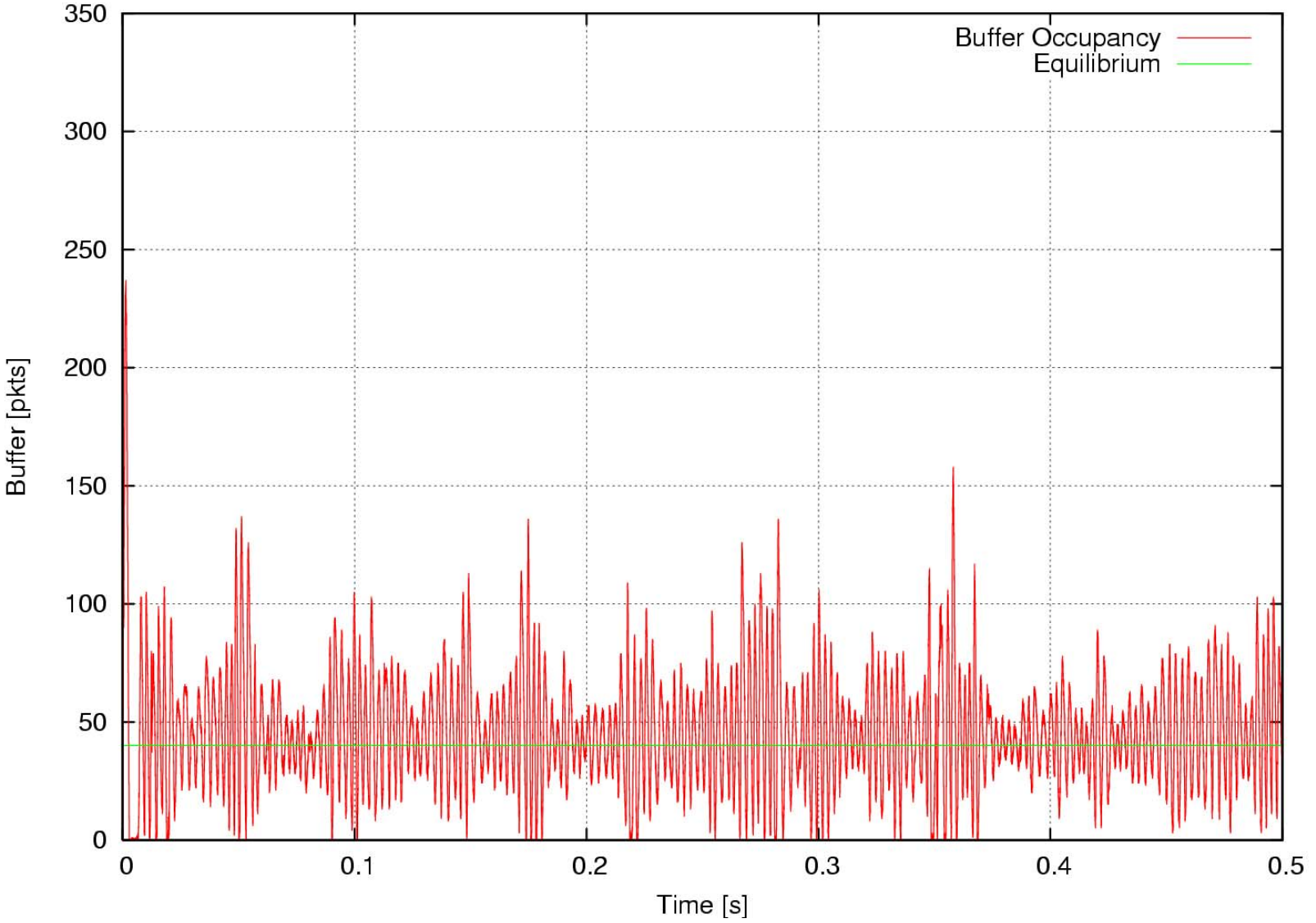
Buffer Utilization: BCN



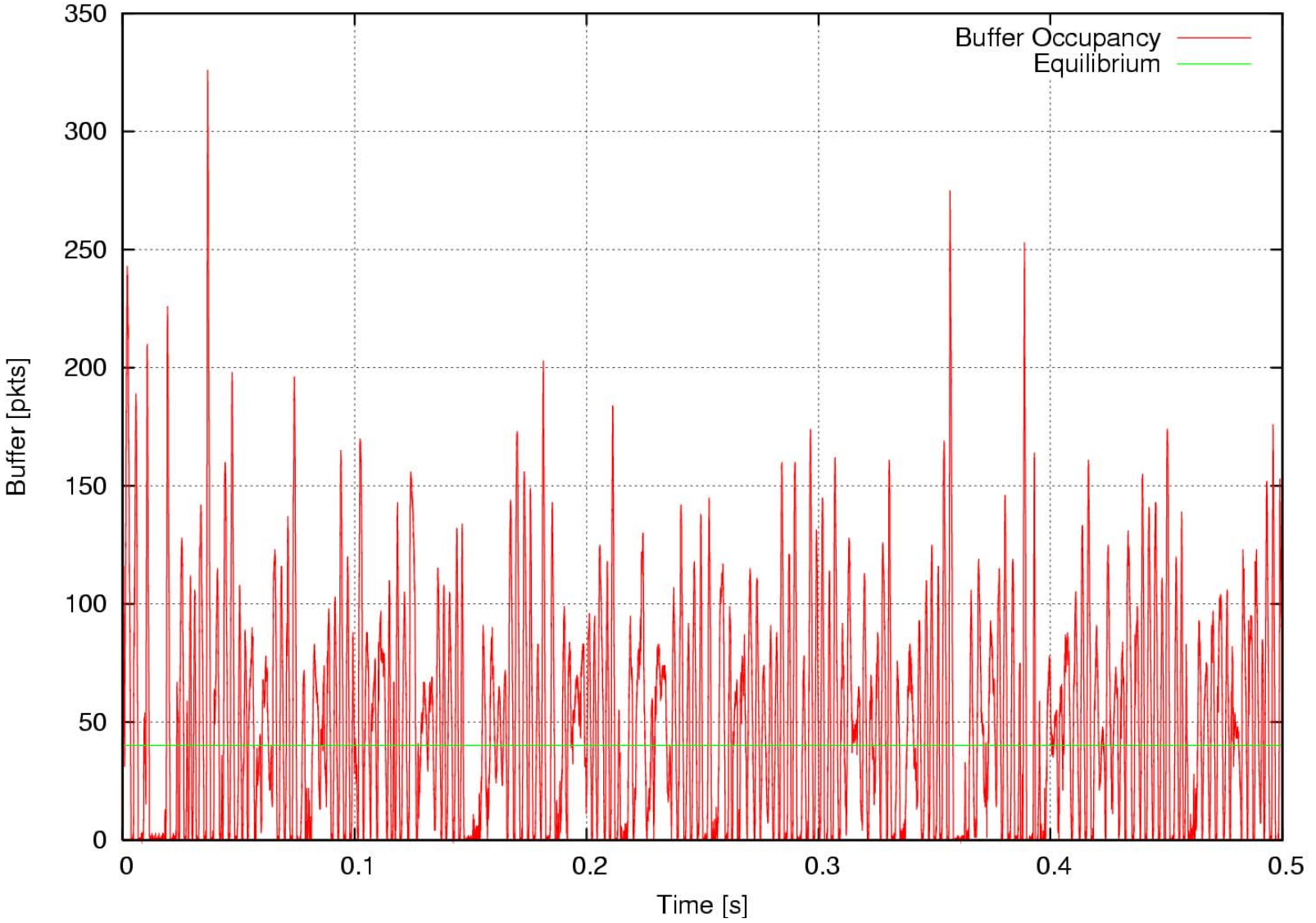
Buffer Utilization: BCN stability



Where BCN starts having trouble: 10Km links



Where BCN breaks down: 20Km links



Summary & Next Steps

- **Backward Congestion Notification (BCN) has a number of advantages:**
  - **Effectiveness (tight control loop)**
  - **L3/4 Protocol agnosticism**
  - **Fairness**
- **And some disadvantage:**
  - **Traffic overhead in backward direction (0.311%)**
- **Consider BCN for core-to-edge congestion management in IEEE 802 short range networks**
- **Study the effect of ECN on the rate-limiter queues to reduce latency**