RPR MAC Definition and Implementation

Sanjay K. Agrawal

Jason Fan, Raj Sharma

Luminous Networks

sanjay@luminousnetworks.com

Outline

- Class of service support in layered networks
- RPR MAC framework proposal
- RPR MAC hardware implementation
- System level architecture tradeoff
- Conclusion

Class of Service support

- Layer 3: IP DSCP specifies 3 bits for CoS, 3 bits for Drop Presidence, 2bits for ECN
- Layer2: 802.1P/Q specifies 3 bits for CoS
- Layer 2.5: MPLS specifies 3 bits of CoS

Diffserv code points are standard
and poised to be consistent across layers

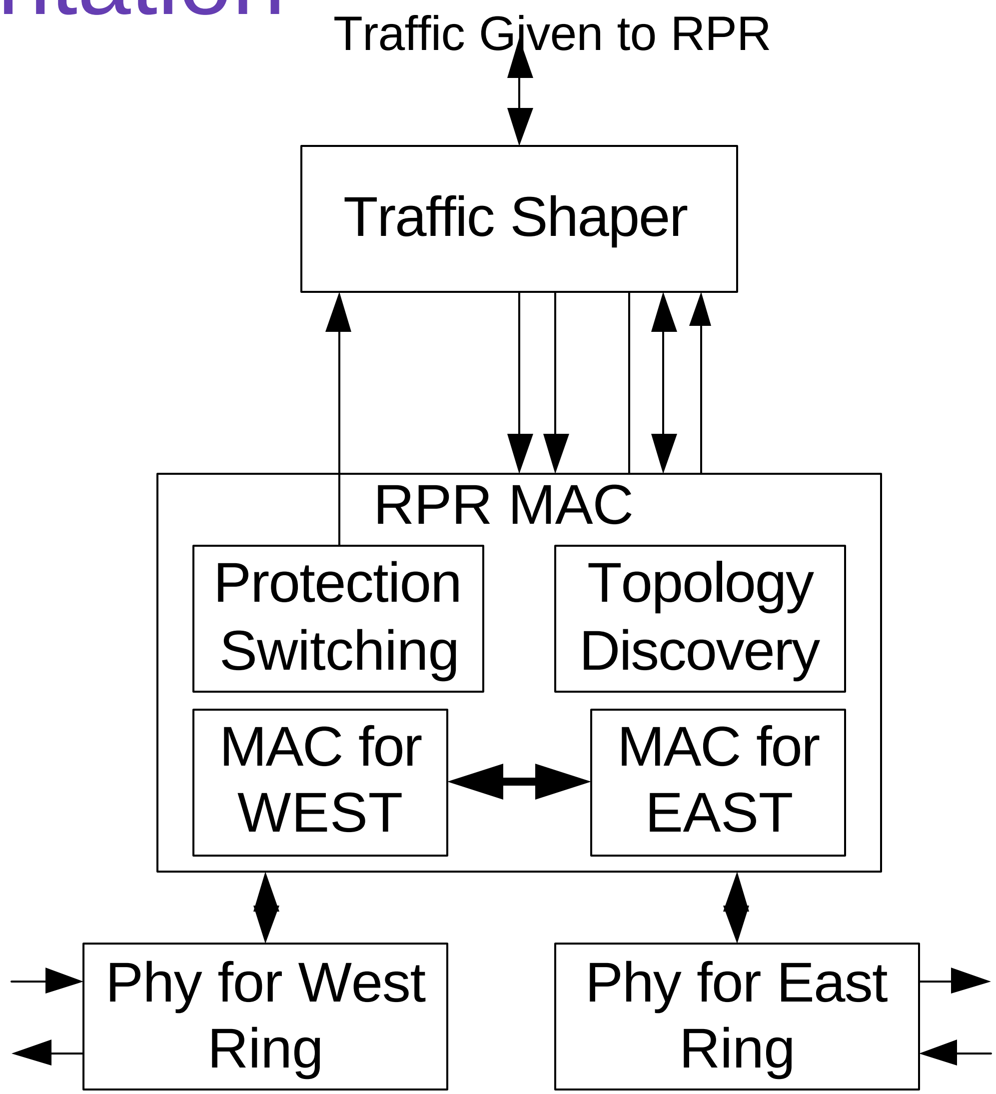
Class of Service

- Express Forwarding (110): Time sensitive committed class
  - Legacy leased line, Voice over IP
- Assured Forwarding 3 (011): Time Sensitive committed class
  - Video
- Assured Forwarding 2(010): Time insensitive committed class
  - Committed data Services, Protected
- Assured Forwarding 1(001) Time insensitive over-committed class
  - Over committed data services
- Best Effort (000): data services

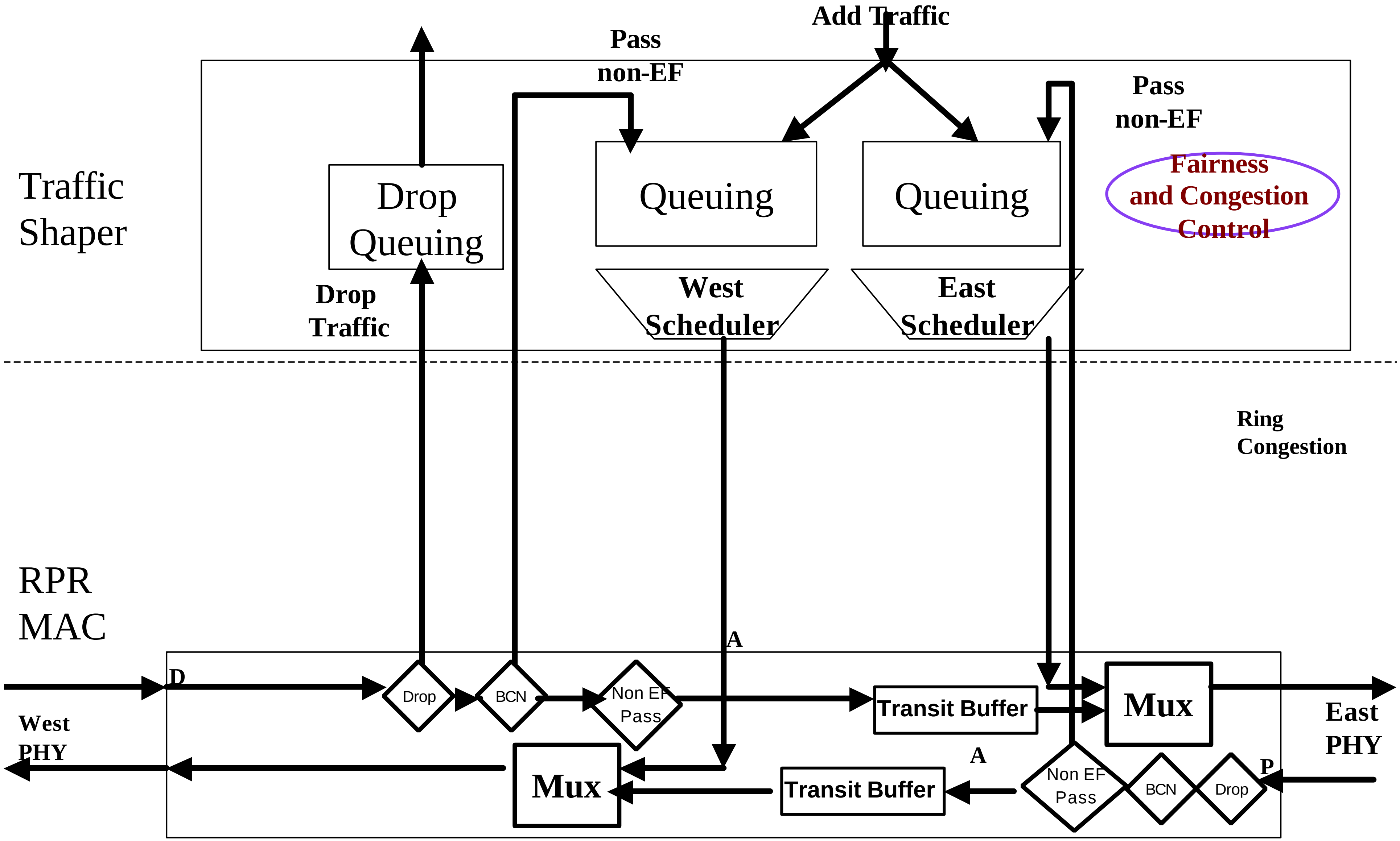
Common RPR MAC functional Requirements

- Class of Service (CoS) support
- Backward Congestion Notification using internode signaling
- Using CoS simultaneous support for
  - Cut through traffic
    - To minimize latency for high priority class
  - Store and forward traffic
    - To allow low priority pass traffic to be stored while high priority add is admitted

Proposed RPR MAC Implementation



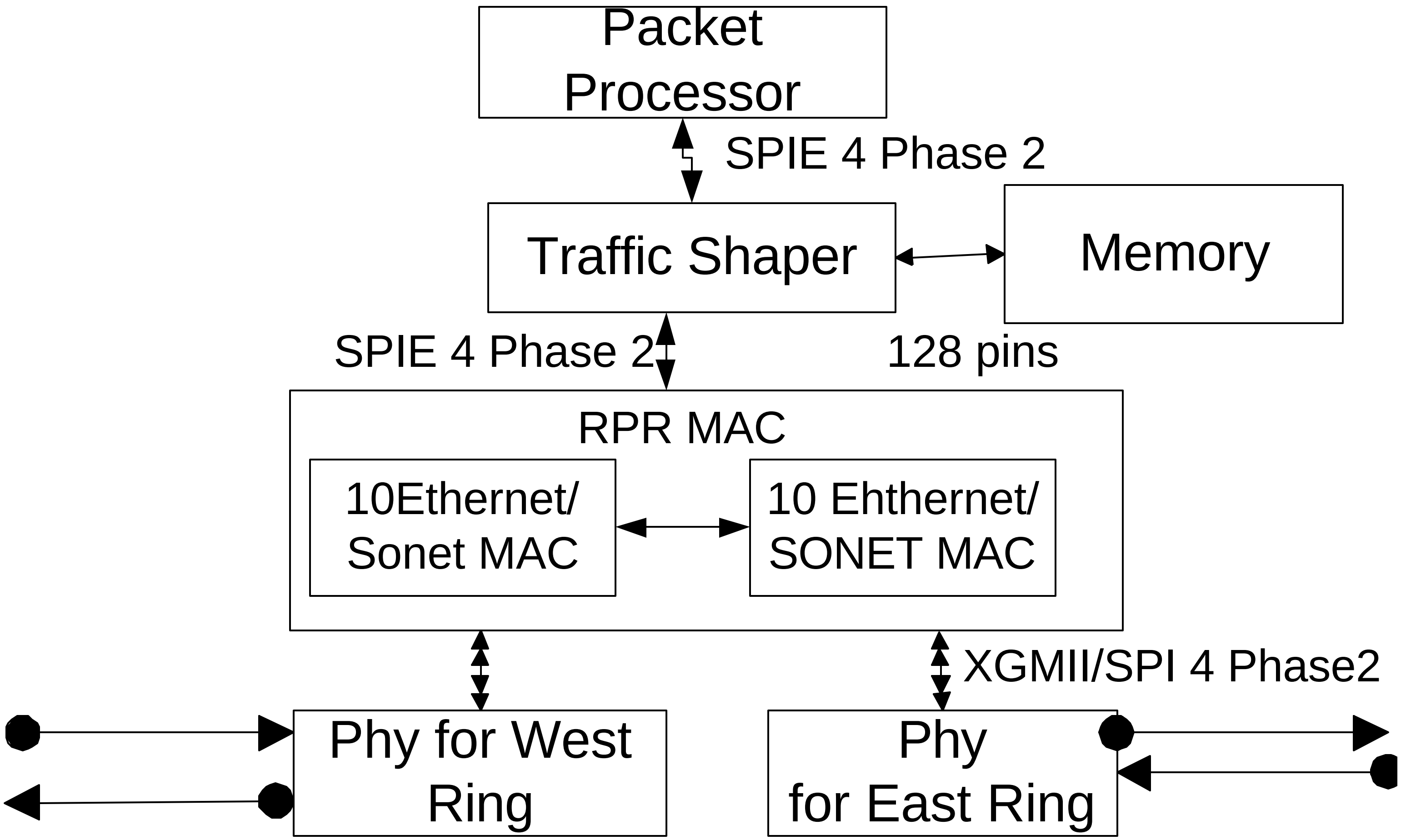
RPR System Architecture



RPR MAC Hardware Implementation Requirements

- Rate adaptation for the drop traffic
- Minimal buffering in the MAC chip
  - Only on board buffers ~ 8Mbits (.8msec@10Gig)
  - External memory interface increases the MAC pin count by 128 pins
- Use of standard interfaces high speed interfaces

Proposed RPR MAC Hardware Implementation



Traffic Shaper Architecture

- Rate adaptation for the drop traffic.
- BCN based architecture
  - Avoids buffering in the intermediate nodes in the rings.
  - Propagates congestion to source nodes.
  - Need per RPR node queuing to minimize BCN based head of the line blocking
  - Add traffic requires 50-100ms buffering
  - For Class based queuing architecture
    - Class based queuing for the add traffic for each node in the ring: 64*8 = 512 queues
    - Class based queuing for drop traffic
  - For Per flow queuing architecture
    - Queue/virtual queue for each SLA based flow: millions of flows
    - Queue for each flow for drop: millions of flows

Traffic Shaper Architecture (cont…)

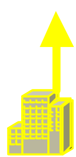
- Class based Queuing Architecture
  - Supports cut through for certain classes, and store and forward for others
  - Avoids inter-node signaling, can accept BCN but may/may not generate it.
  - Single Class based queuing structure for add and pass traffic
  - Single class based queuing structure for drop traffic
  - Requires 50-100ms buffering
  - Flow based fairness addressed in class queues using:
    - Second level of scheduling on per service aggregate
    - Congestion control on per service aggregate
  - No issues with multicast

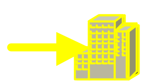
Conclusion

- Layered architecture allows vendor differentiation while insuring interoperability
- RPR MAC proposal simple and not tied to a particular implementation that addresses only a set of needs. Supports both
  - Cut through
  - Store and forward
- Hardware proposal supports most of the proposals with minimum cost of implementation
- Traffic shaper architecture based on class based queuing supports for most of the service scenarios

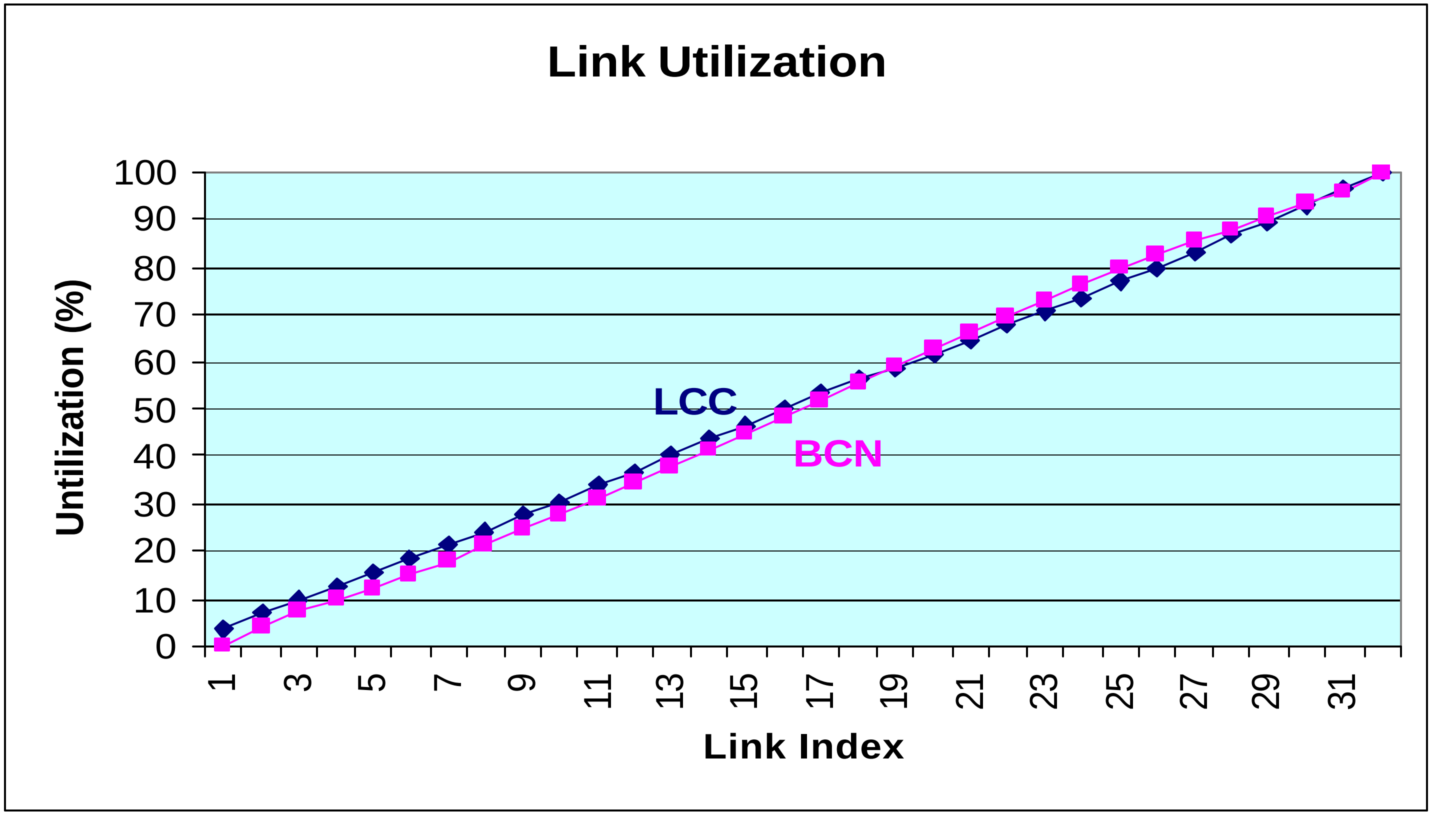
BCN vs. No BCN Traffic Simulation

tcp flows:
Node 1-2 = 3 tcpflows
Node 3-17 = 2 flows
Node 18-32 = 1 flow

Source: 

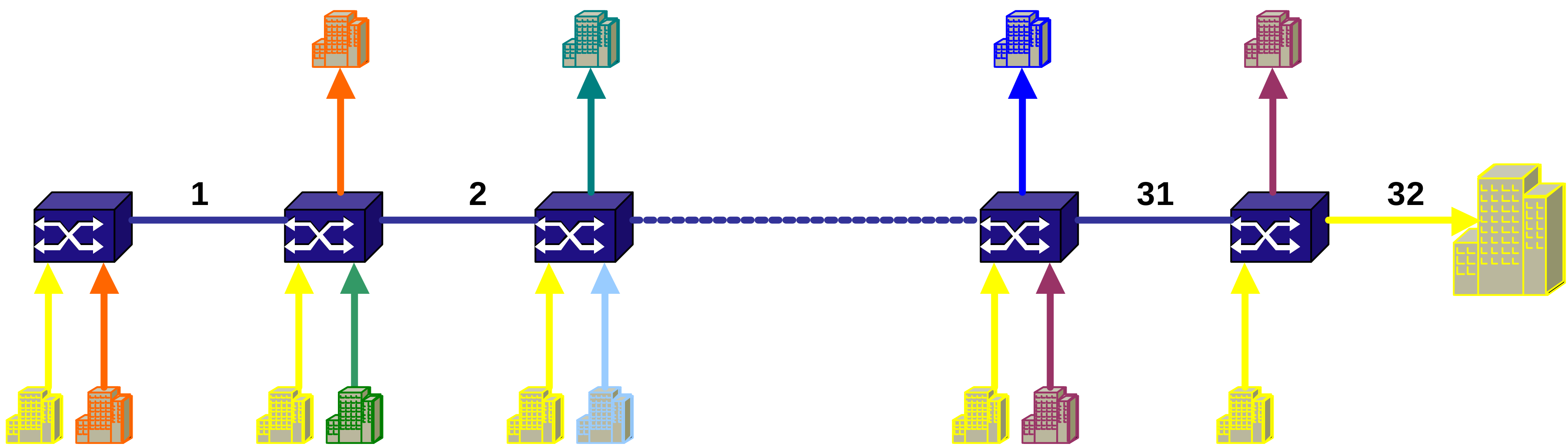
Sink: 

BCN vs. No BCN Traffic Simulation



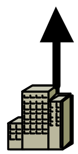
BCN = Backward Congestion Notification
LCC = Local Congestion Control

BCN vs. No BCN Traffic Simulation

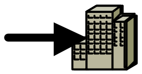


tcp flows:
Node 1-32 = 5 tcpflows

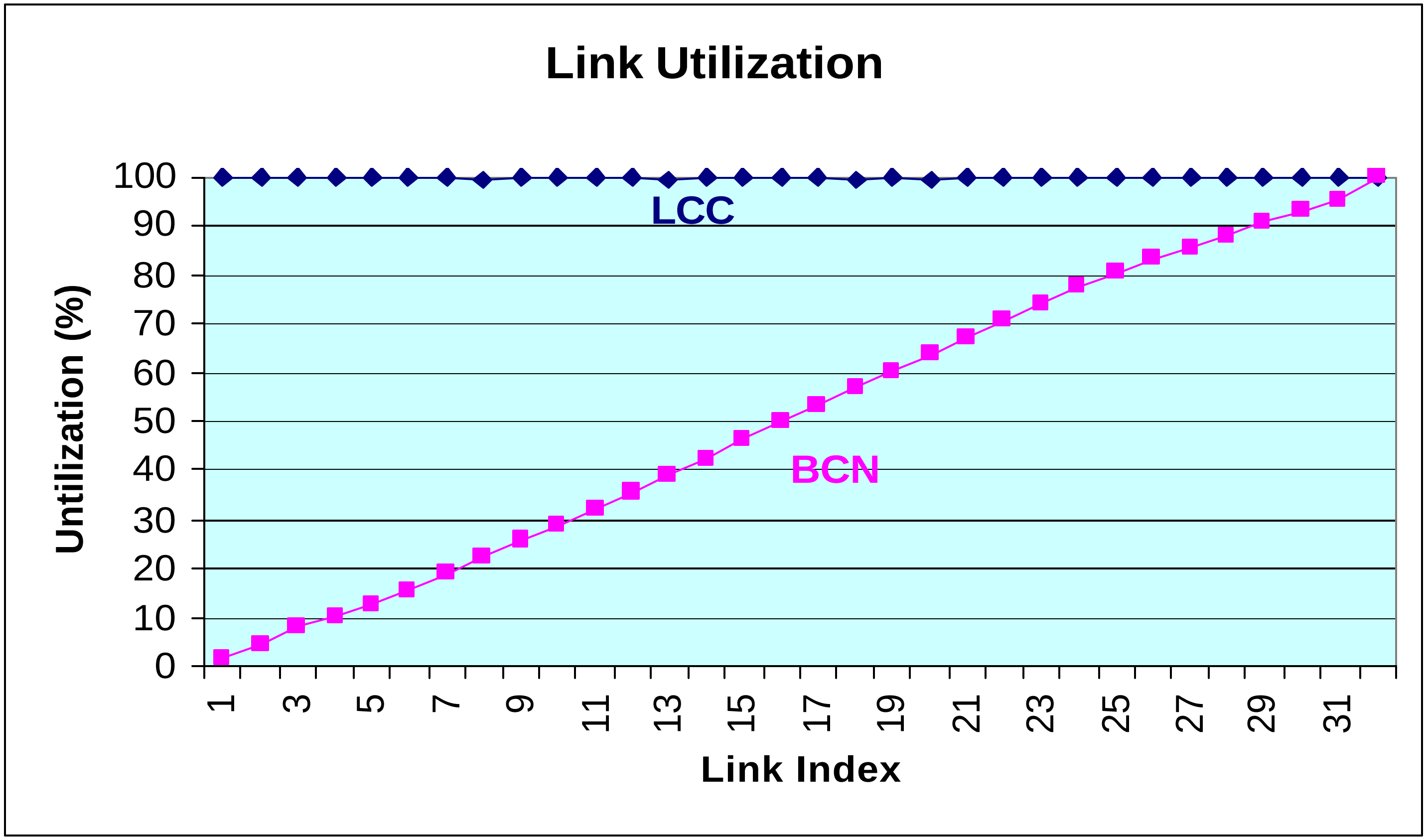
Source:



Sink:

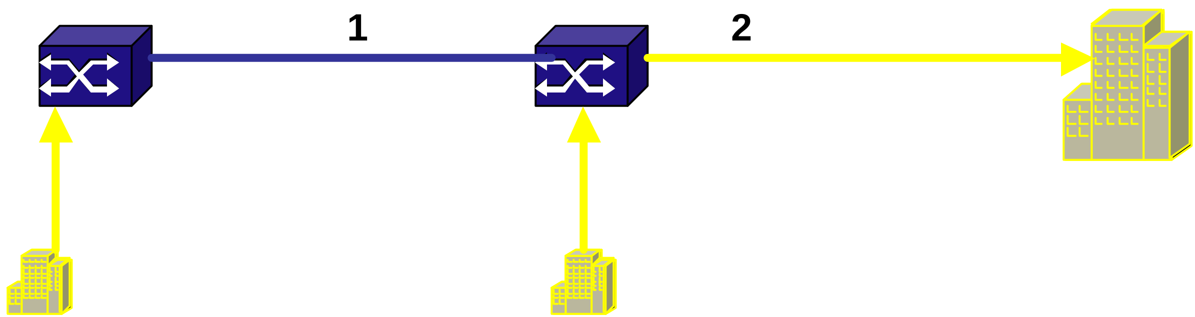


BCN vs. No BCN Traffic Simulation



BCN = Backward Congestion Notification
LCC = Local Congestion Control

Cut through vs. Store and Forward



- Link 2 is congested
- Measure Max Q Delay for Host Traffic in Link 2
- Variable: Span Propagation Delay
- Number TCP flows such that link saturated
    - 2ms = 21 TCP flows

Cut through vs. Store and Forward High priority Add Max Q Delay