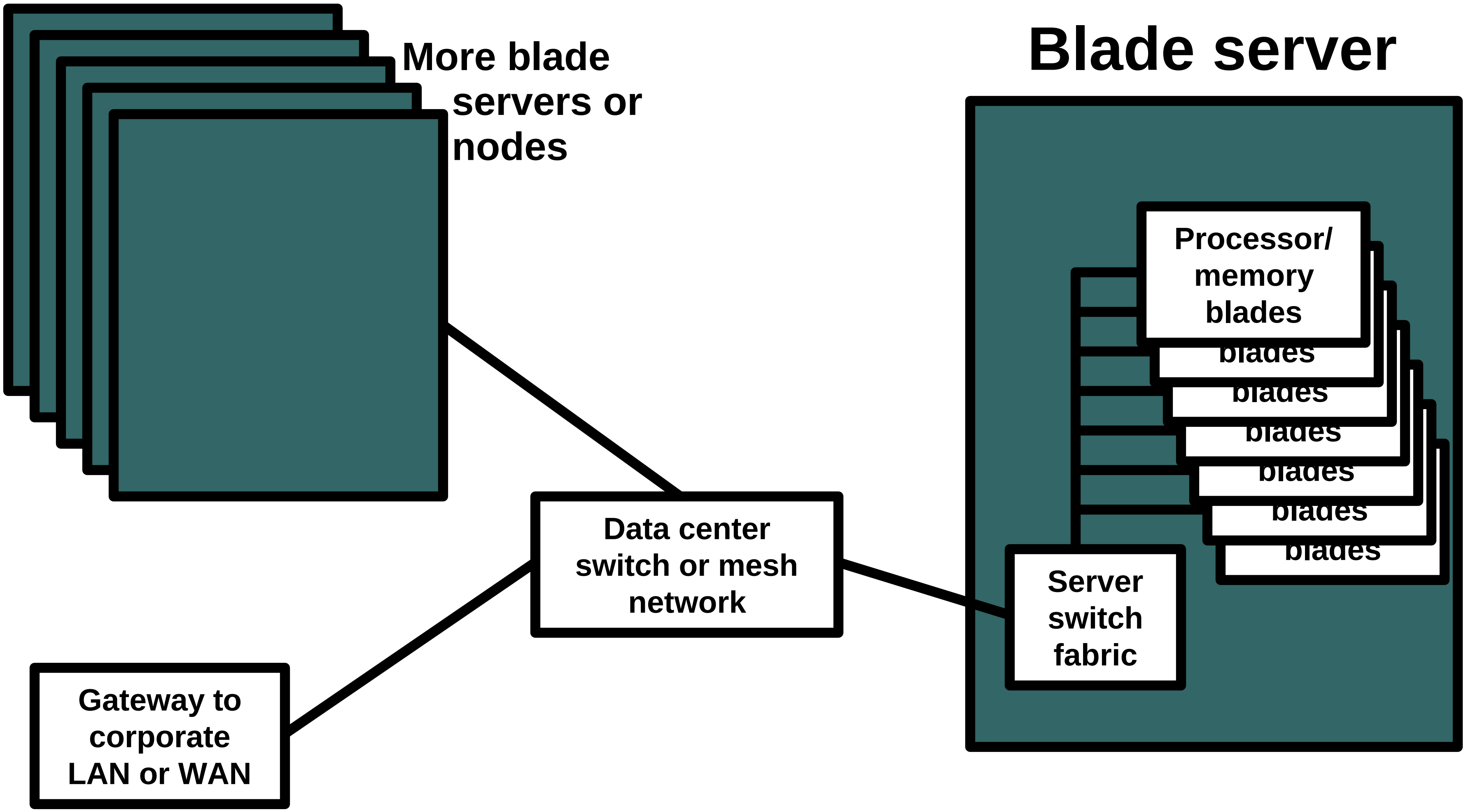
Problem space for Ethernet congestion management

On behalf of Congestion Management Study Group

Agenda

- **Topology and components**
- **Layering**
- **Congestion management**
- **Notification**
- **Conclusions and proposals**

Data center topology



Data center components

- Network interface subsystems (in server blades and other nodes):
  - Includes Ethernet encapsulation
  - May include network & transport layer acceleration
- Blade server switch fabric:
  - Typically <20 blades supported
  - Dedicated uplink ports
- Data center switch or mesh network:
  - Single fabric, up to 100’s ports or multiple fabrics
  - May include multi-path switching (aggregated links etc.)
- Gateway to corporate LAN or WAN:
  - Connects to legacy networks
  - Could be layer 3 or above

Data center component options

Network interface subsystems (in server blades and other nodes):

- Tagging, rate shaping, flow control
- Maybe transport window adjustment, per flow/session state information

Blade server switch fabric:

- Priority queuing, buffer size optimization, congestion tagging, policing
- Maybe rate limiting methods

Data center switch or mesh network:

- Similar to blade server switch fabric
- Assumed to be more “feature rich”

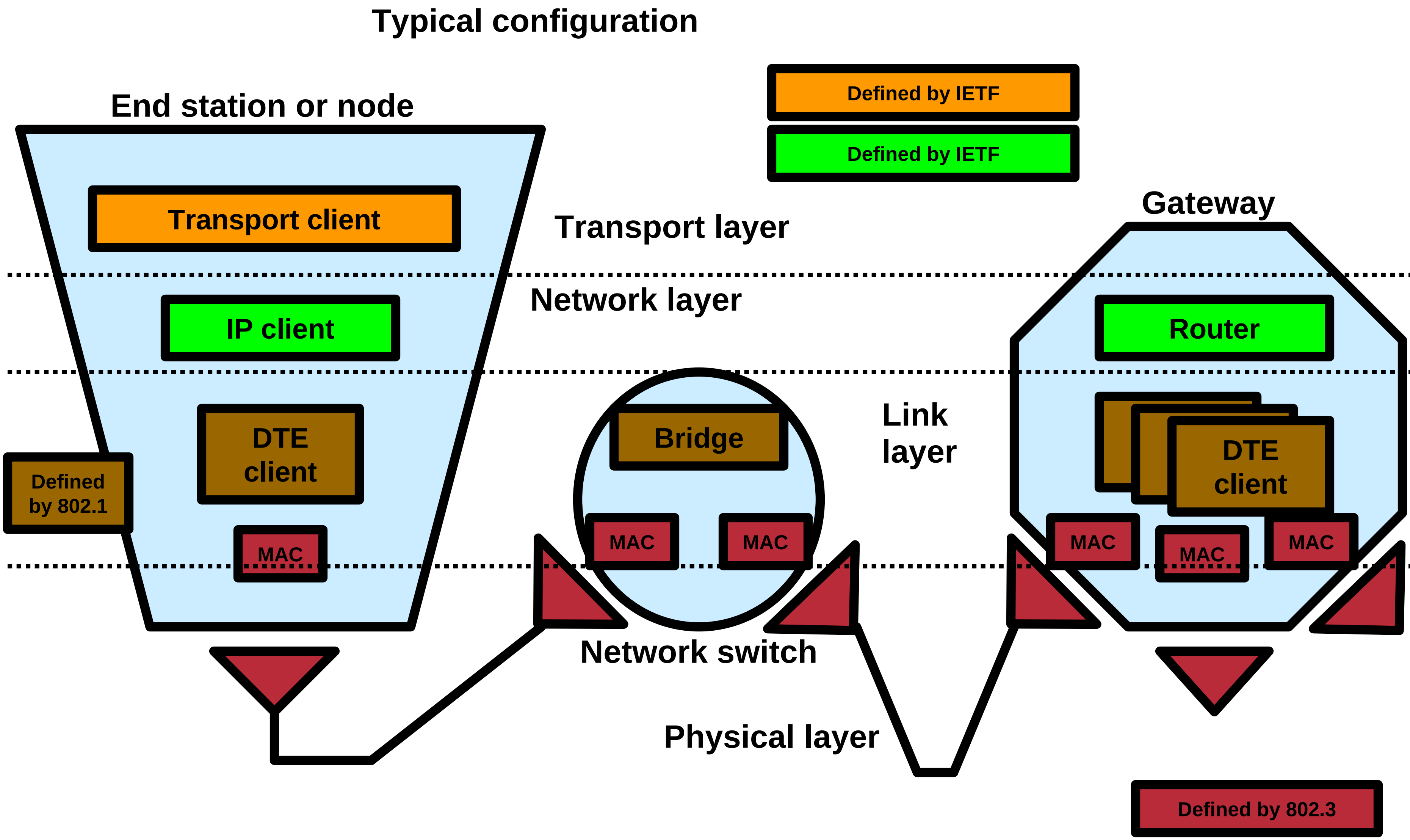
Gateway to corporate LAN or WAN:

- Must accommodate wider application parameters

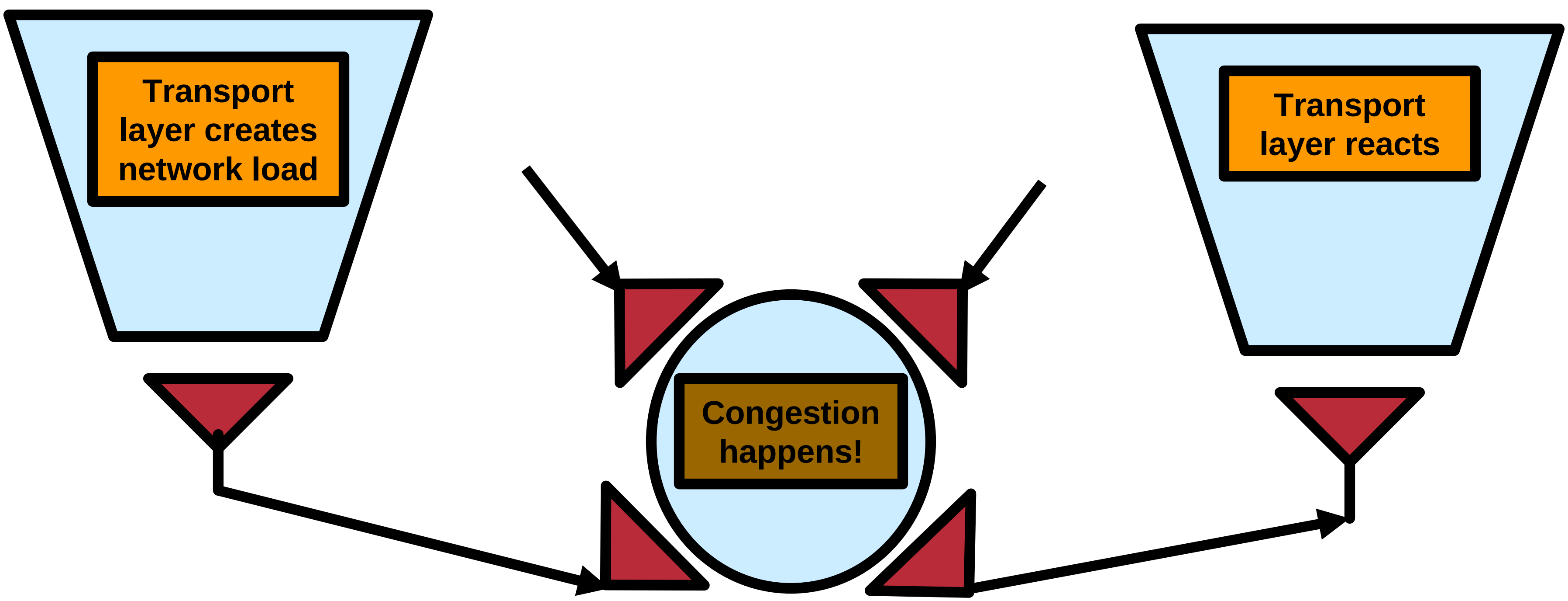
Agenda

- **Topology and components**
- **Layering**
- **Congestion management**
- **Notification**
- **Conclusions and proposals**

ISO layering in data center components



Congestion happens!

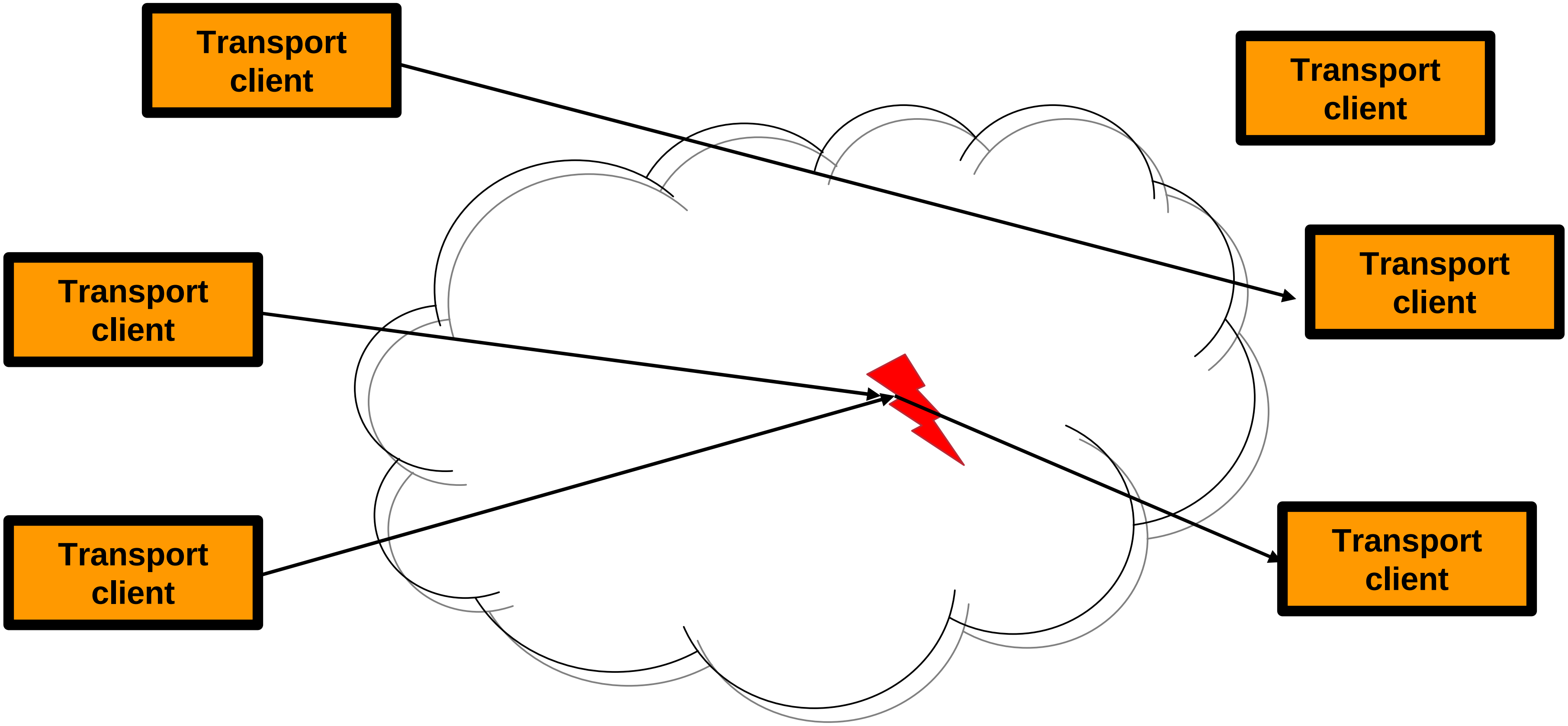


Transport layer sends data into the network,

Congestion happens in the bridge,

Causing a reaction in the transport layer

Congestion in the network cloud



In arbitrary network topology connectivity cannot be assumed

Only by adjusting effected transport can congestion be remedied…

… without perturbing innocent conversations

Problems with transport adjustment mechanisms

- Transport adjustment often relies on packet loss
  - Retries are expensive – timeouts are disastrous!
  - Not only a problem with TCP
- Transport adjustment mechanisms are generally optimized for internet-like topologies
  - Transport windows are very large, requiring large network buffers
  - Reaction times are slow
- Traffic is bursty in time & space
  - Typically clients send bursts to various destinations
  - Causes congestion points to move
  - Needs fast reaction times in transport to avoid "misadjustment"

What is needed for congestion management?

- Lossless transport adjustment
  - Notification to transport clients without causing retries or timeouts
  - For TCP & non-TCP transport
- Fast reaction times for adjustment
  - Low network latency, plus change to optimization mechanisms
  - Removes need to “pre-tune” network
- Method for notifying transport clients of congestion
  - Transport client (at layer 4) must be made aware of congestion happening in (layer 2) bridge
  - Ideally should not rely on layer 4 implementations for network switches
  - Should also be as compatible as possible with legacy devices

Agenda

- Topology and components
- Layering
- **Congestion management**
- Notification
- Conclusions and proposals

Congestion management example

Example solution – Layer 2 ECN (like)

- "Explicit Congestion Notification"
- Marks a layer 2 packet in case of congestion

Set bridge buffer thresholds lower than discard level

- Indication says packet would have been discarded…
- … if my buffer was smaller

Following example uses arbitrary solution for ECN

- NOT intended as a detailed proposal, but as an example
- Uses TCP plus ECN in data center (type) network
- Demonstrates effectiveness of transport client adjustment in a tightly bounded environment

L2– Congestion Indication

Issue:

- Congestion due to oversubscription
- “Reactive” rate control in TCP

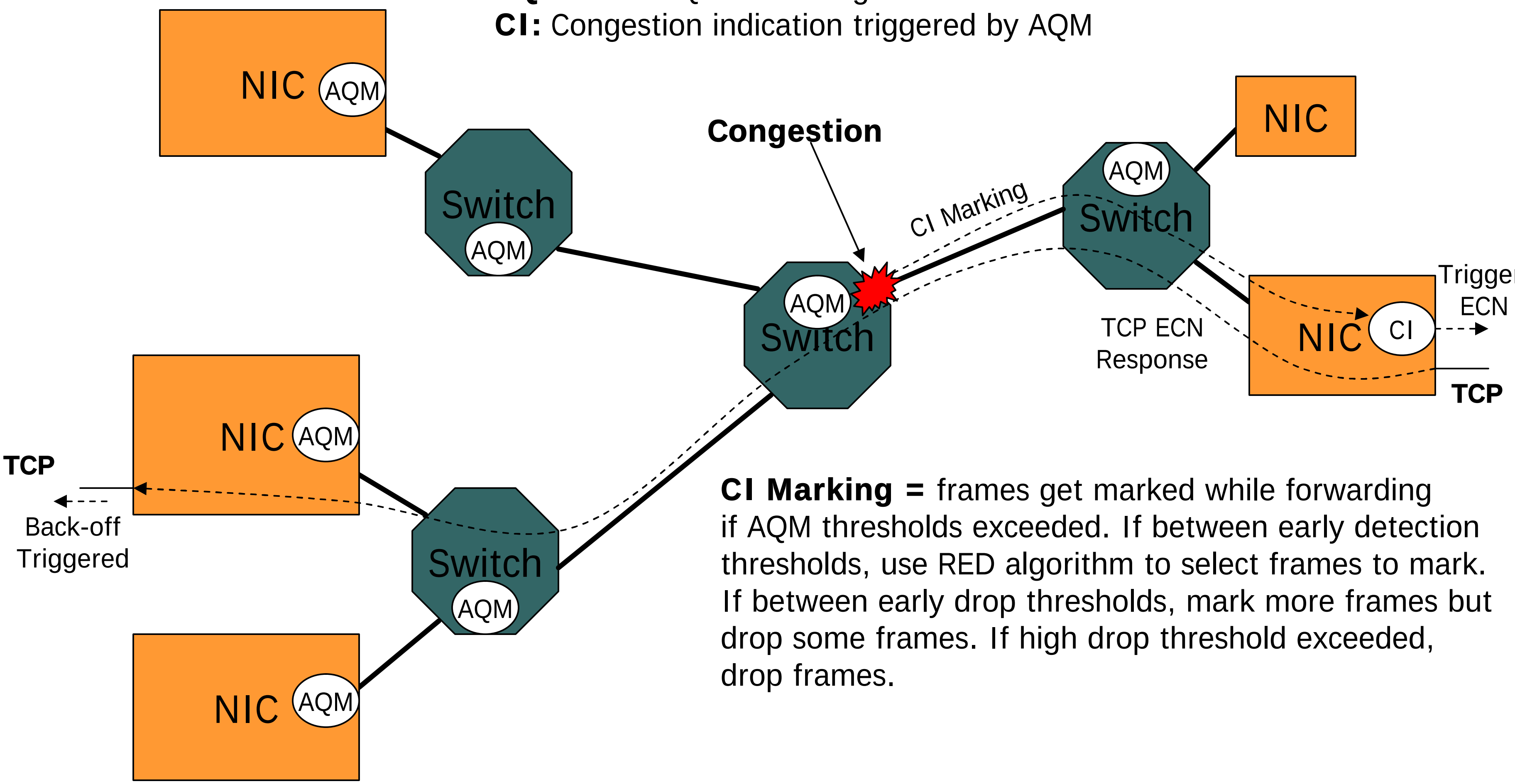
Method:

- “Rate Control” is done at end-points based on congestion information provided by L2 network
  - Provide Congestion Information from the network devices to the edges
  - Standard notification allows end-station drivers to benefit
- Various mechanisms possible for Congestion Indication
  - Marking, control packet, forward/backward/both
- TCP applications can benefit
  - ECN can be triggered even by L2 congestion
  - “Proactive” action by TCP, avoids packet drop
- Non-TCP applications can leverage
  - New mechanism to respond to congestion

Extract from wadekar_2_0904.pdf

Model Implementation: L2 Congestion Indication

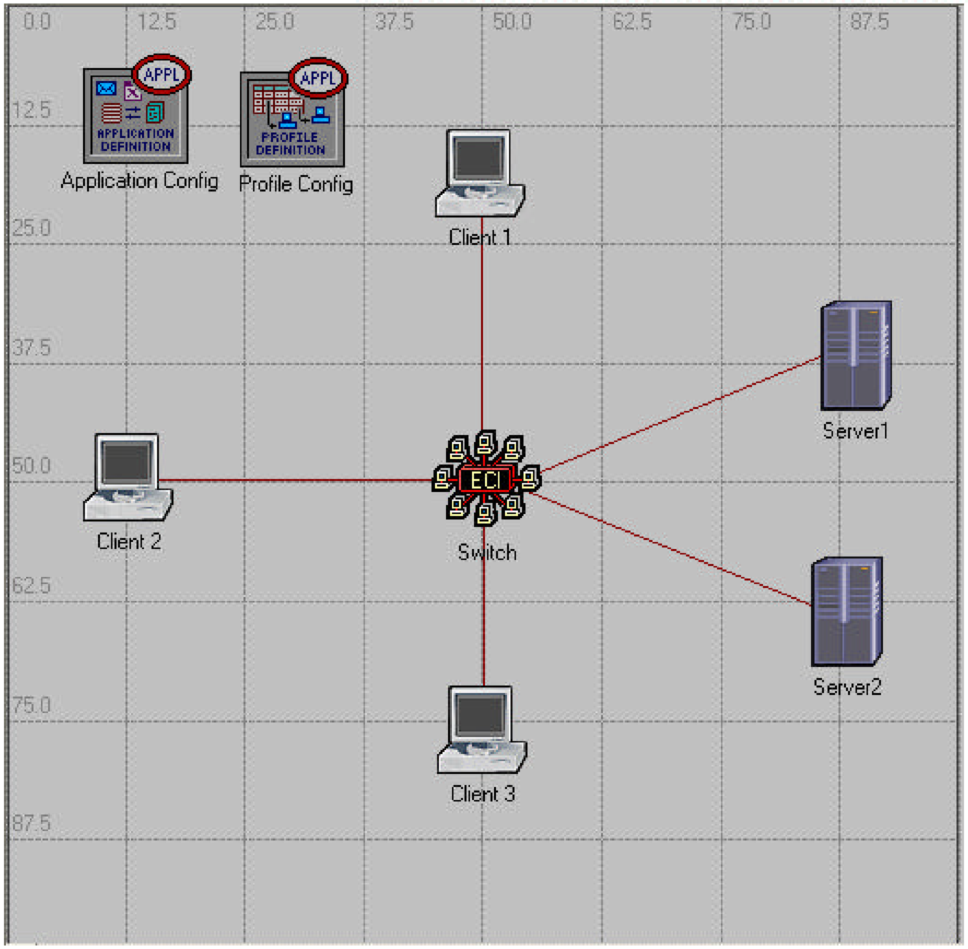
AQM: Active Queue Management
CI: Congestion indication triggered by AQM



CI Marking = frames get marked while forwarding if AQM thresholds exceeded. If between early detection thresholds, use RED algorithm to select frames to mark. If between early drop thresholds, mark more frames but drop some frames. If high drop threshold exceeded, drop frames.

Extract from wadekar_2_0904.pdf

Simple Topology



All Links are 10 Gbs

Shared Memory 150KB

App = Database Entry over full TCP/IP stack

Workload distribution = Exponential (8000)

ULP Packet Sizes = 1 Bytes to ~85KB

Client 1 sending to both servers

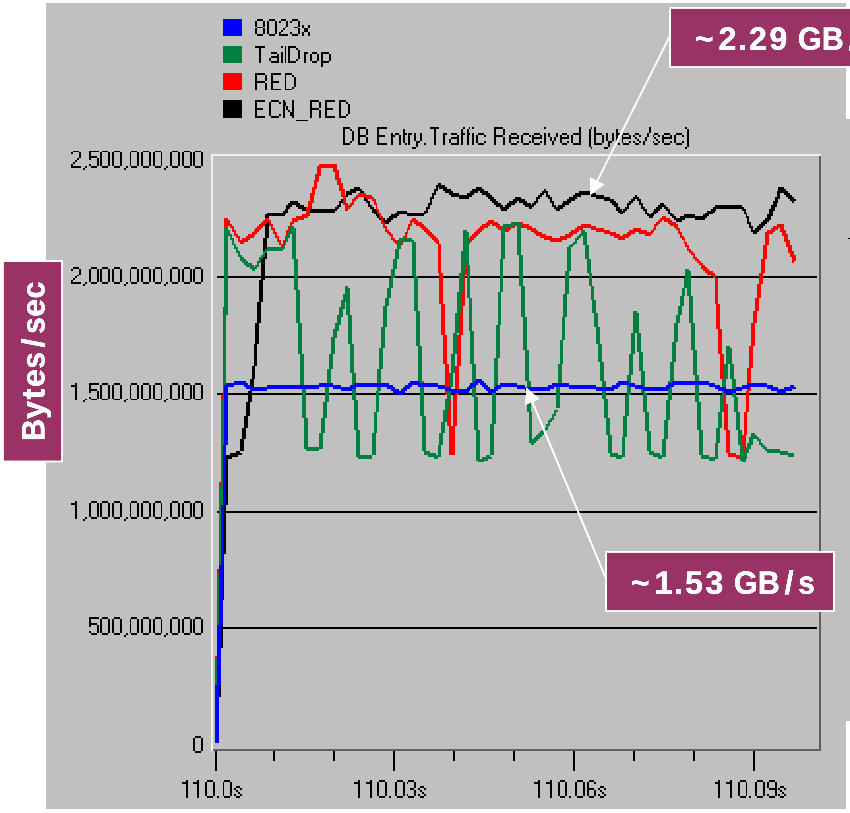
Clients 2 & 3 sending to Server 1

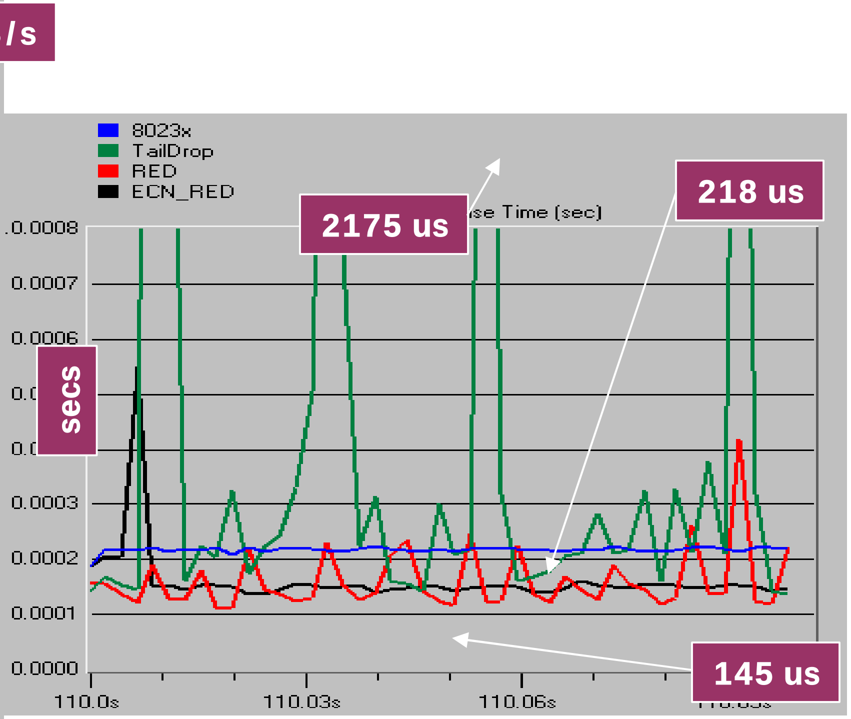
TCP Delay = DB Entry request to completion

HOL Blocking at Client1 for Client1-Server2 traffic

Extract from wadekar_2_0904.pdf

Application Throughput & Response Time



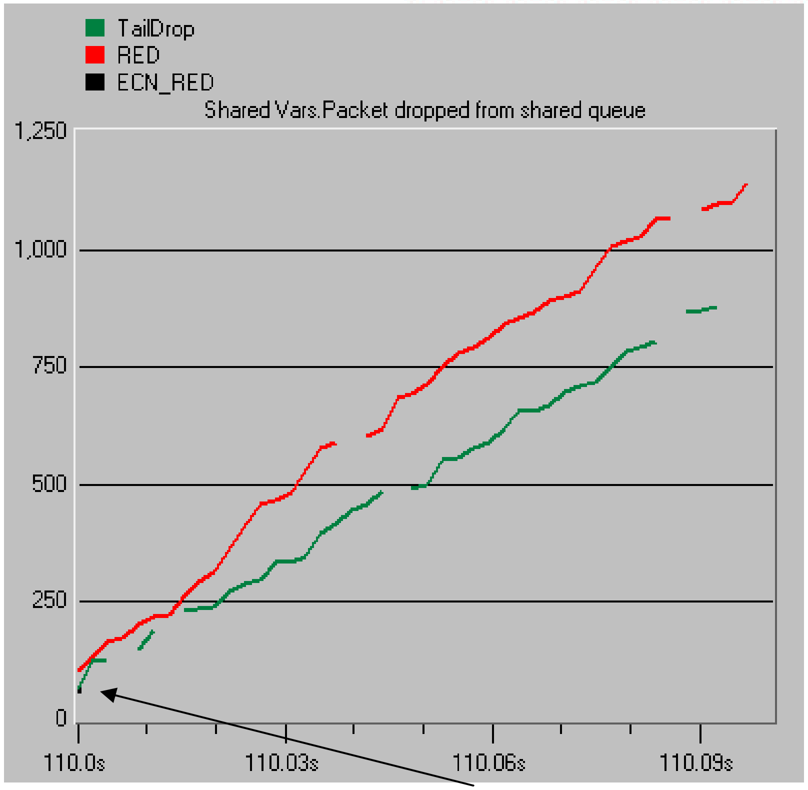


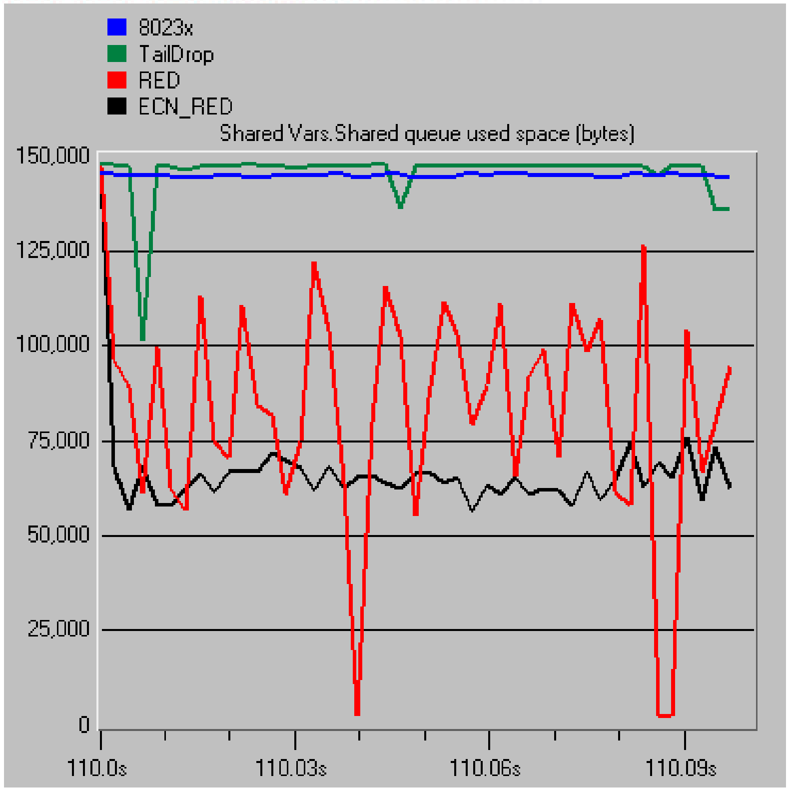
L2-CI with ECN improves TCP Performance

Extract from wadekar_2_0904.pdf

Shared Memory Utilization and Packet Drop at the Switch

Number of drops



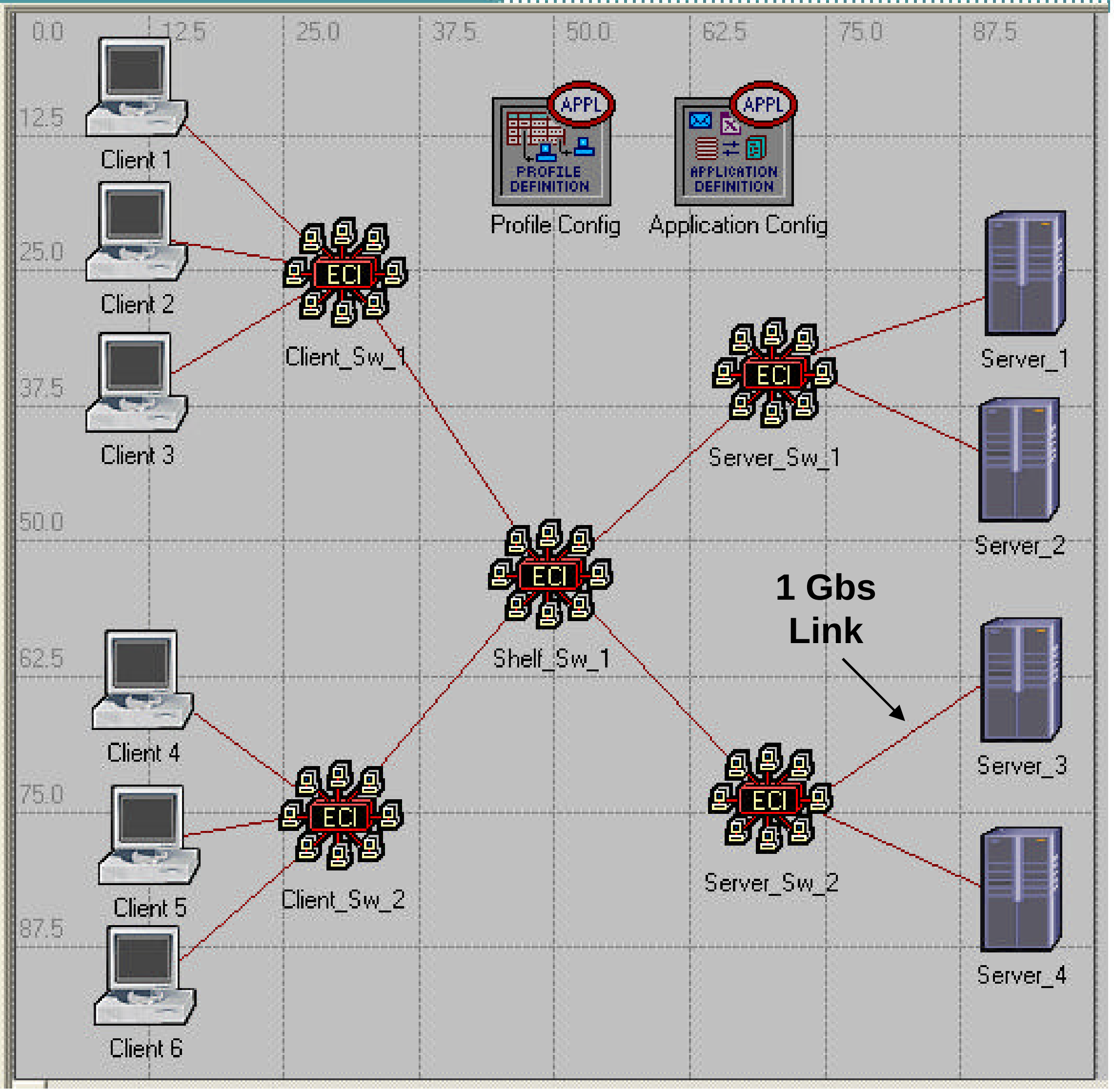


Some initial drops with ECN when it is stabilizing its average Q size

L2-CI can significantly reduce packet drops & reduce buffer requirements

Extract from wadekar_2_0904.pdf

Multi-stage system w/ mixed link speeds



All Links except one are 10 Gbs

Peak Throughput = 2.434 Gigabytes / Sec

App = Database Entry over the full TCP/IP stack

Workload distribution = Exponential (8000)

ULP Packet Sizes = 1 Byte to ~85KB

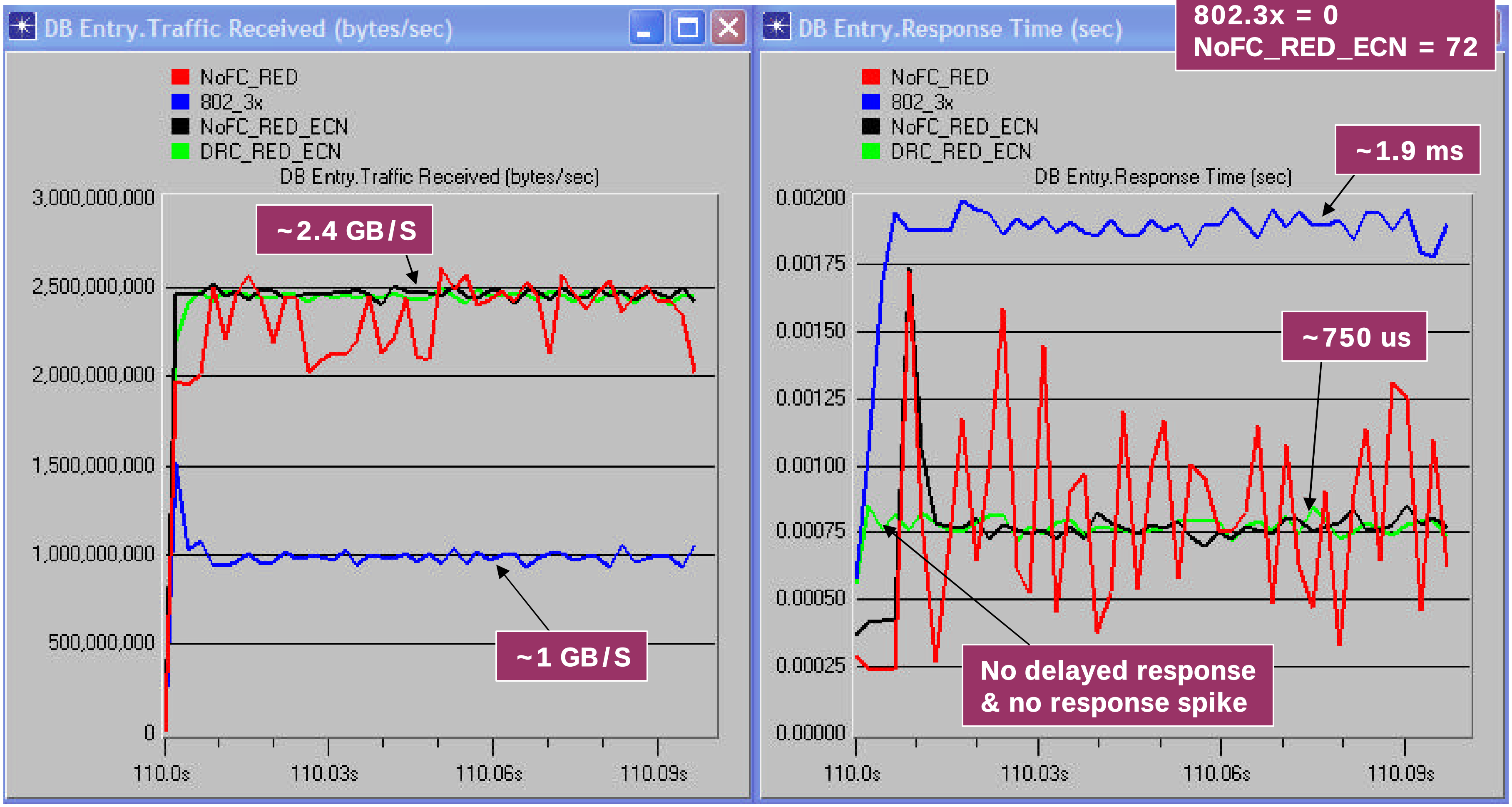
TCP Window size = 64KB

All clients sending database entries to all servers

Extract from wadekar_2_0904.pdf

Application Throughput & Response Time (Buffer = 64 KB per Switch Port)

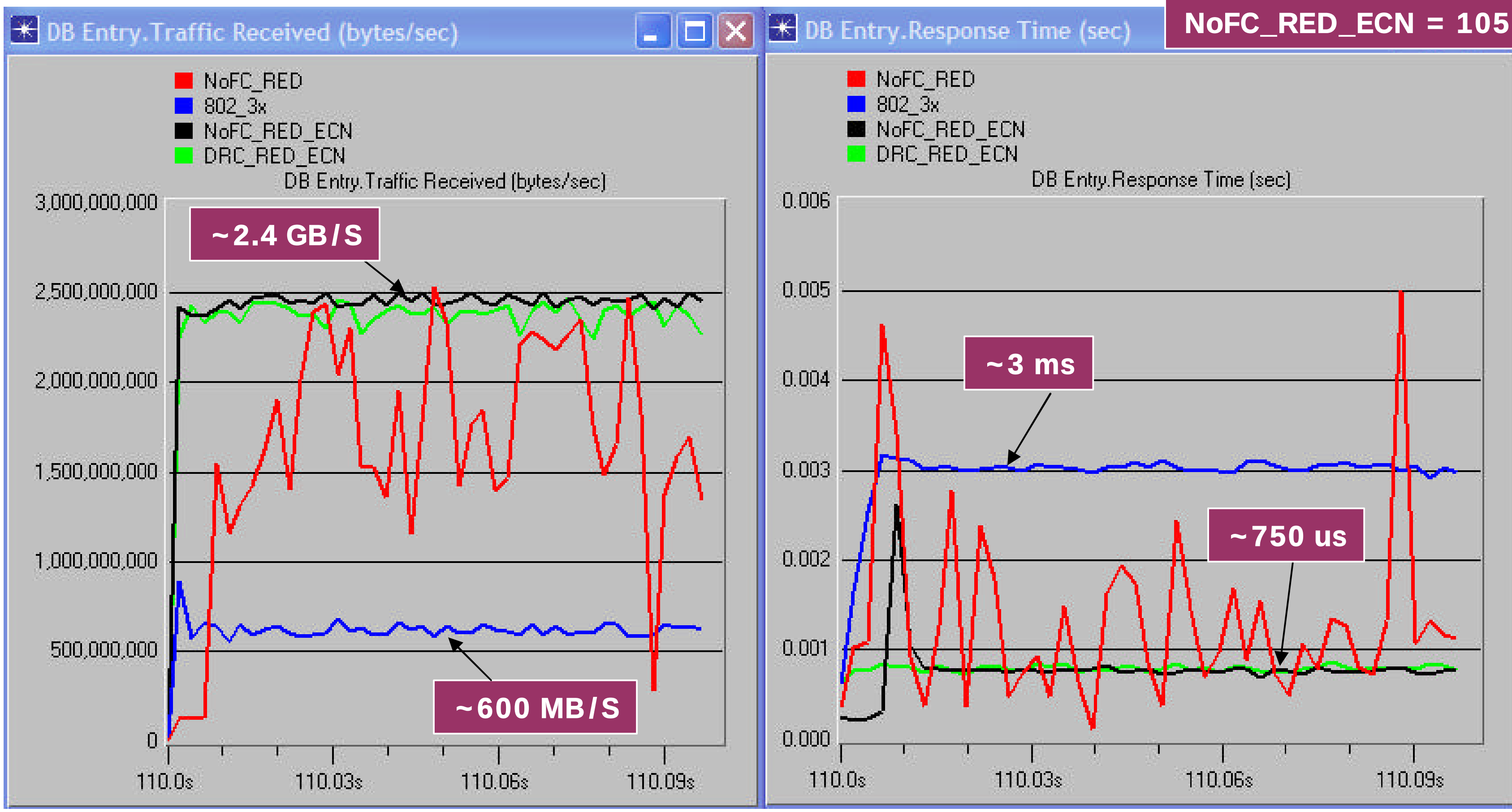
Drops:
NoFC_RED = 2554
802.3x = 0
NoFC_RED_ECN = 72



L2-CI/ECN shows excellent characteristic for short range TCP.

Application Throughput & Response Time (Buffer = 32 KB per Switch Port)

Drops:
NoFC_RED = 2373
802.3x = 0
NoFC_RED_ECN = 105



L2-CI/ECN maintains performance even with small switch buffers

Extract from wadekar_2_0904.pdf

Summary

- **Examples presented show "technical feasibility" of Congestion Management in Ethernet**
- **Can allow MAC Clients to take proactive actions based on congestion information via 802.3**
- **Facilitate & take advantage of higher layer CM mechanisms**
- **Simulations show significant comparative improvements**

Extract from wadekar_2_0904.pdf

Agenda

- Topology and components
- Layering
- Congestion management
- **Notification**
- Conclusions and proposals

Notification

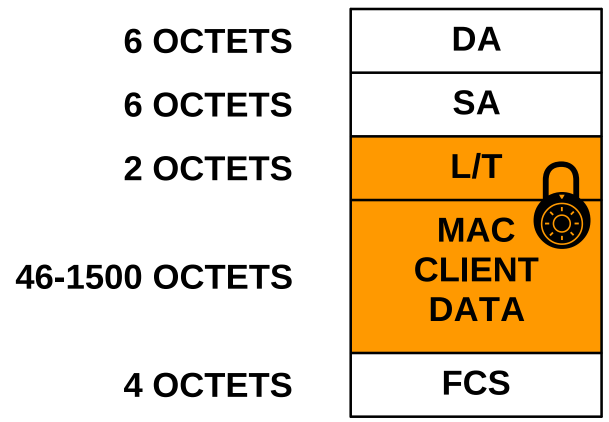
- Requires that layer 2 devices (bridges) notify congestion
  - Must be orthogonal to transport protocol
  - Notification at layer 2, independent of transport
- Should be transparent for legacy bridges or end stations
  - Some network elements may not notify or react to notification
  - Hippocratic oath, “First do no harm.”
- Transfer of information from L2 to higher layer
  - Must be the domain of higher layer devices:
  - Either multilayer switches or end stations
  - Requires new definitions for transport mechanisms to use notification

Ethertype stacking & frame extension

- Example solution – NOT a proposal…
  - … just a illustration
- Use the new definitions for generic encapsulation (stacked Ethertypes)
  - Request Ethertype for CN information
  - Will be ignored by non-cognizant devices
- Other options can be explored
  - Markdown
  - Extra header bits

Basic MAC frame

- No prefix
- No suffix
- 64-1518 octets



Illustrative example

802.1Q Tagged frame

- Prefix=4
- Suffix=0

- 68-1522 octets

6 OCTETS	DA
6 OCTETS	SA
2 OCTETS	EType
2 OCTETS	TCI
2 OCTETS	L/T
46-1500 OCTETS	MAC CLIENT DATA
4 OCTETS	FCS

Illustrative example

802.1ad Tagged frame

- Prefix=8
- Suffix=0
- 72-1526 octets

6 OCTETS	DA
6 OCTETS	SA
2 OCTETS	S-tag EType
2 OCTETS	S-tag TCI
2 OCTETS	C-tag Etype
2 OCTETS	C-tag TCI
2 OCTETS	L/T
46-1500 OCTETS	MAC CLIENT DATA
4 OCTETS	FCS

Illustrative example

802.1Q Tagged CMSG frame

- Prefix=8
- Suffix=0
- 72-1526 octets

6 OCTETS	DA
6 OCTETS	SA
2 OCTETS	Q-tag EType
2 OCTETS	Q-tag TCI
2 OCTETS	CN Etype
2 OCTETS	CN Tag
2 OCTETS	L/T
46-1500 OCTETS	MAC CLIENT DATA
4 OCTETS	FCS

Illustrative example

802.1ad Tagged CMSG frame

- **Prefix=12**
- **Suffix=0**
- **76-1530 octets**

6 OCTETS	DA
6 OCTETS	SA
2 OCTETS	S-tag EType
2 OCTETS	S-tag TCI
2 OCTETS	C-tag Etype
2 OCTETS	C-tag TCI
2 OCTETS	CN Etype
2 OCTETS	CN Tag
2 OCTETS	L/T
46-1500 OCTETS	MAC CLIENT DATA
4 OCTETS	FCS

Illustrative example

(probably) not needed in the definition

- Buffer management algorithms
  - Early tail drop, RED or variations
  - Relationship to priority queuing
  - Define congestion notification NOT detection
- Transport layer definitions
  - Leave to IETF
  - Possibly make recommendations from 802 TF
- Gateway (layer 3 or higher) device behavior
  - Gateway may choose to ignore, pass or react to notification
  - Beware that wider system is not tightly bound

Agenda

- Topology and components
- Layering
- Congestion management
- Notification
- **Conclusions and proposals**

Work required for 802.1

- Define congestion notification mechanism
  - Will be ignored by non-cognizant devices
  - Non conformant devices or non reactive flows
  - Investigate behavior in mixed environment
- Also need to consider backward indication
  - To work with any transport protocol
  - Especially for unidirectional transport protocols
  - Significantly more complex, requires research

Possible 802.1 PAR (Purpose)

To improve the performance of 802.1 bridged networks in the presence of congestion.

In bounded environments, higher layer protocols may significantly improve their behavior in the presence of congestion if they are notified of the congestion occurring at convergence points in the bridged network.

This project will define a mechanism for notifying higher layers that congestion has been detected in the path that a packet has followed through the network.

Question

Is there support for a new project and Task Force?

- **Mechanism to enable congestion management in bridged networks – 802.1~~**

Share work between 802.1 & 802.3 members

- **Including joint balloting**