# Congestion Management Study Group
# July 2004, Portland, Oregon

# Congestion Management 101

## … or maybe 55 1/2

* **Ethernet networks**
* **What (& when) is congestion**
* **Backpressure problems**
* **Network design for beginners**
* **Exceptions**

# What is Ethernet?

**The age old question...**

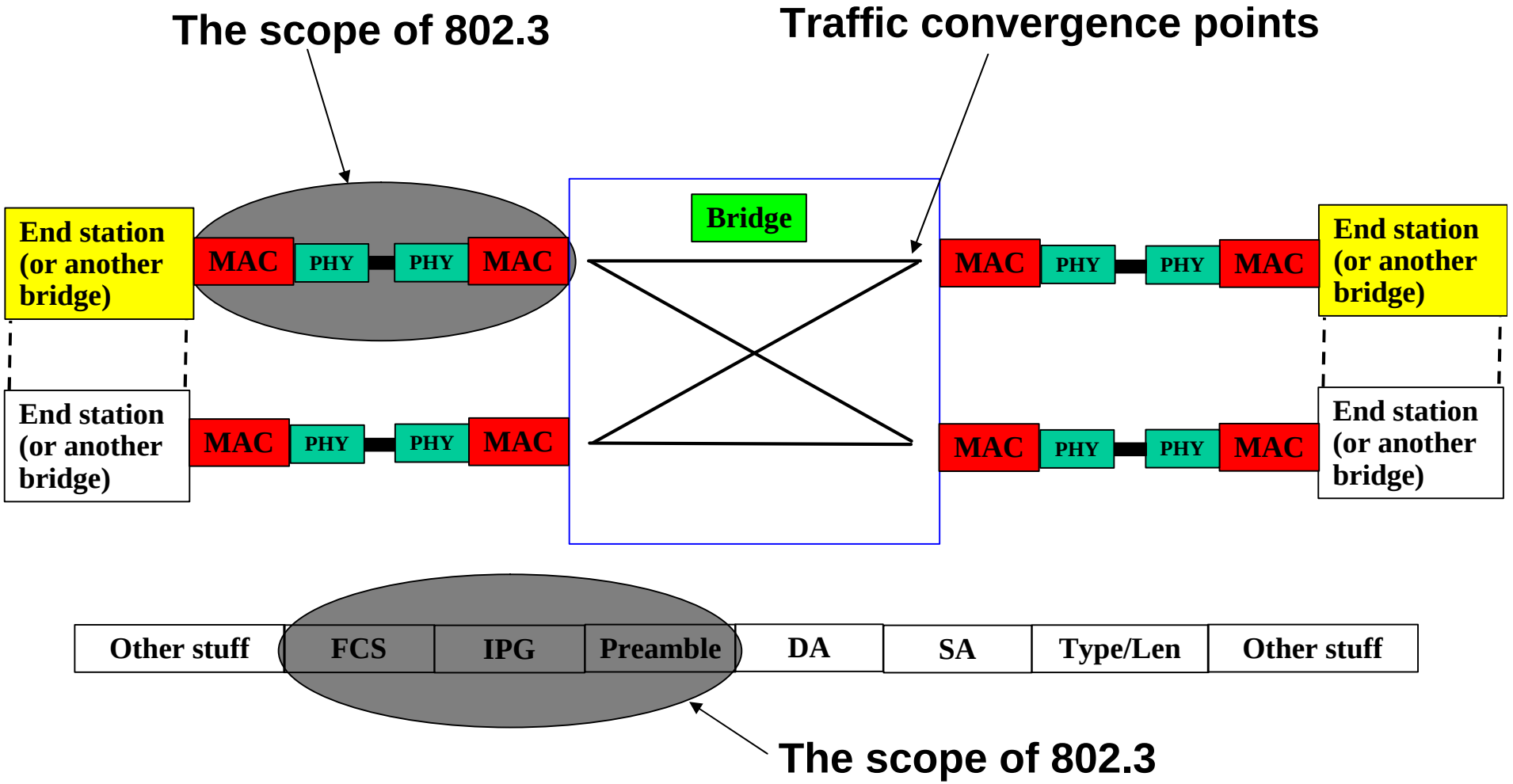
**IEEE 802.3 defines the Ethernet MAC, Ethernet PHYs and some other related stuff**

**Almost all instances of Ethernet today include more than 802.3:**

- **IEEE 802.1 defines bridging, including priority, VLANs, spanning tree etc.**
- **Most Ethernet networks use Internet Protocol (as defined by IETF)**
- **Although TCP is common, many transport protocols are supported**

**“Ethernet Networks” could be used to describe networks using 802.3 links, connected together by 802.1 bridges and running IP.**

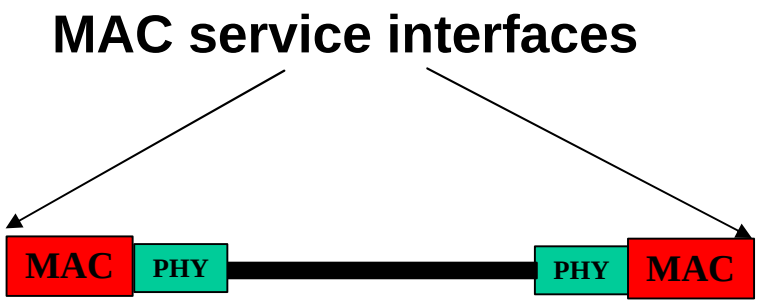
# An Ethernet network



# “Ethernet” never drops a packet!

**For point-to-point Ethernet links**

# The Ethernet Guarantee




**Every single packet put in at one end**

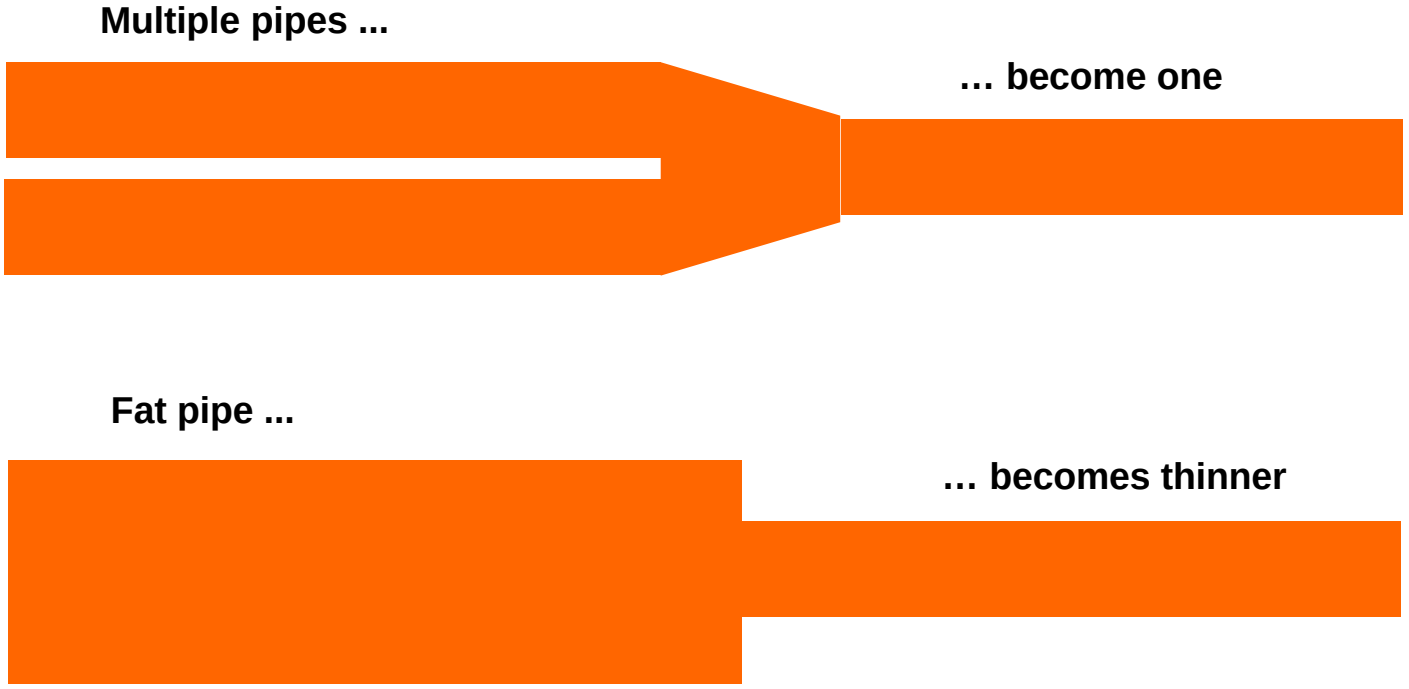
**Comes out at the other end***

***Subject to restrictions imposed by the laws of physics and the Bit Error Ratio**
**All other offers notwithstanding**
**Your mileage may vary**

* Ethernet networks
* What (& when) is congestion
* Backpressure problems
* Network design for beginners
* Exceptions

# Congestion

## Congestion can only happen at a confluence or a constriction




**Congestion will not necessarily occur, depending on circumstances, but…**
**… congestion can never happen in any other case.**

**With a few minor exceptions, there are no confluences or constrictions defined within 802.3**

# Weapons against packet loss

**If the offered load exceeds the ability to carry the load, then packets must be lost! This is an inescapable fact.**

**What can be done to avoid this?**

**Net bandwidth demand must not exceed supply. Things to be adjusted are:**

- **Raw bit rate.**
- **Packet rate (using IPG stretch).**
- **Burst duty cycle (requires buffering to absorb bursts).**

**These can be adjusted:**

- **Pre-emptively (traffic management)**
- **Reactively (backpressure)**
- **Implicitly (transport layer acknowledge mechanism)**

# How are they used with Ethernet

**Pre-emptive management**

**Traffic shaping adjusts the packet rate allowed for each flow**
**Resource reservation ensures that no link is oversubscribed**

- **Works well for pseudo-static flows (e.g. video or voice)**
- **Duration of flow must be >> network latency**
- **Problematic for highly meshed applications**

**Implicit bandwidth management**

**Protocol sends a burst (or requests a burst), waits for response**
**If network is congested, round trip time increases**

- **Burst duty cycle increases, reducing load on congestion point**
- **Supports any number of flows**
- **Insufficient buffering results in packet loss**

# Just how bad is packet loss?

**Some packets will be lost - whether due to congestion or bit errors.**

**What is the effect - in terms of net bandwidth?**

**Assume a burst-acknowledge protocol:**

- **Losing 1 packet is equivalent to wasting one burst of bandwidth.**
- **Therefore the effect of the packet loss is magnified by the ratio of the packet size to the round trip network latency.**
- **For a single network switch, connecting endpoints, this should be ~100.**

**Therefore 1/1000 packet loss is equivalent to ~10% net bandwidth reduction…**

**… or alternatively: 1/1000 packet loss is acceptable if it means that the net bandwidth increases by >10%.**

* Ethernet networks
* What (& when) is congestion
* Backpressure problems
* Network design for beginners
* Exceptions

# Friends don’t let friends use backpressure

**In any network with multiple connections, some may be congested while others are not.**

> **Backpressure (undirected) causes congestion at one point to disturb unrelated communication.**
> **Therefore, in an attempt to alleviate the congestion at one point, effective bandwidth is wasted at other points.**

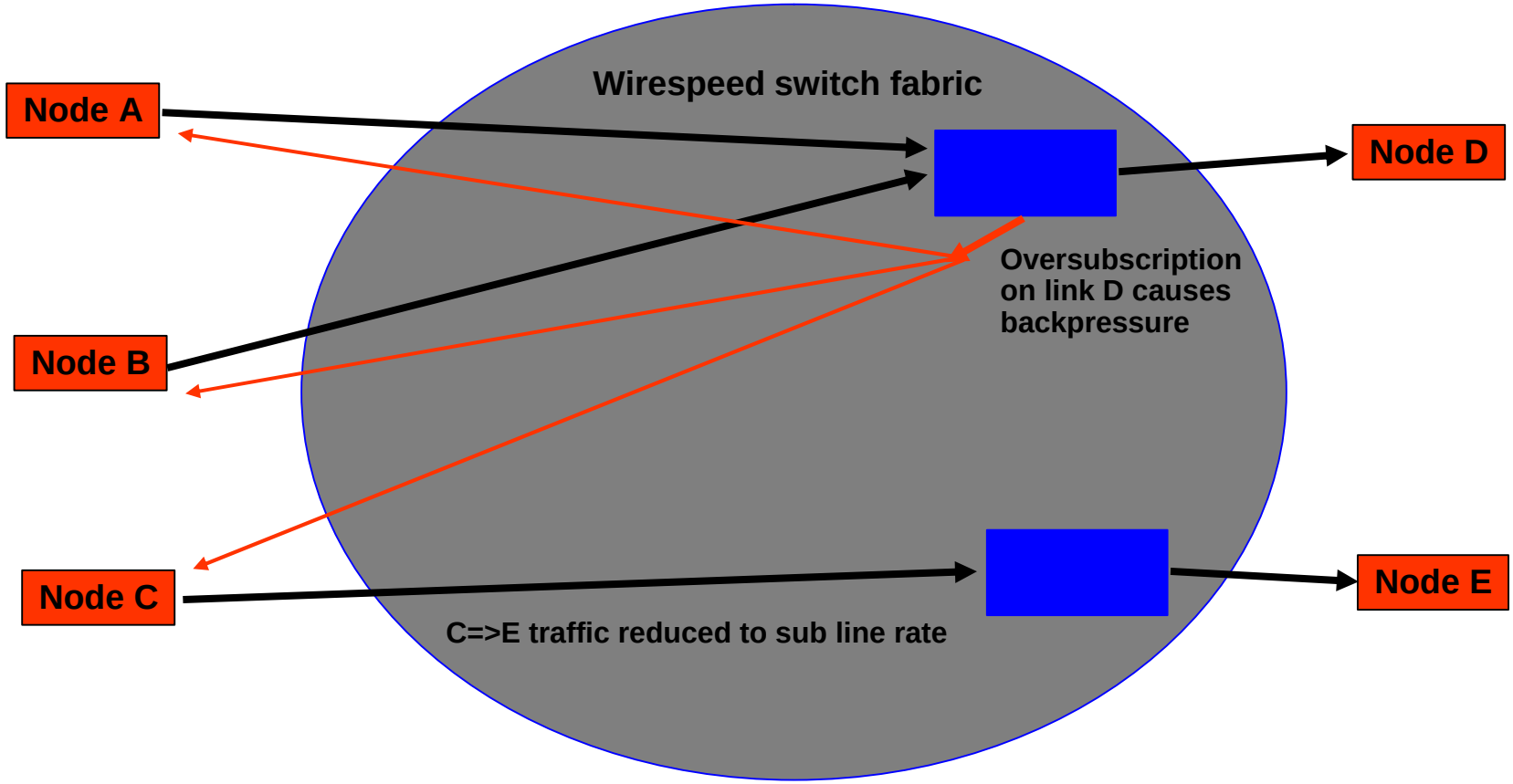
**In a typical packet network, traffic is bursty causing temporary congestion.**

> **Backpressure is reactive and reflects the state of congestion as it was – not as it will be.**
> **Therefore backpressure will cause instability, halting or restricting transmitters that are near the end of a burst.**

**These problems are often ignored by simulations that rely on constant or steady traffic and frames directed randomly or statically.**
**Simulations should use bursts of frames, all the frames of a burst should be directed to the same destination but separate bursts may have different destinations.**

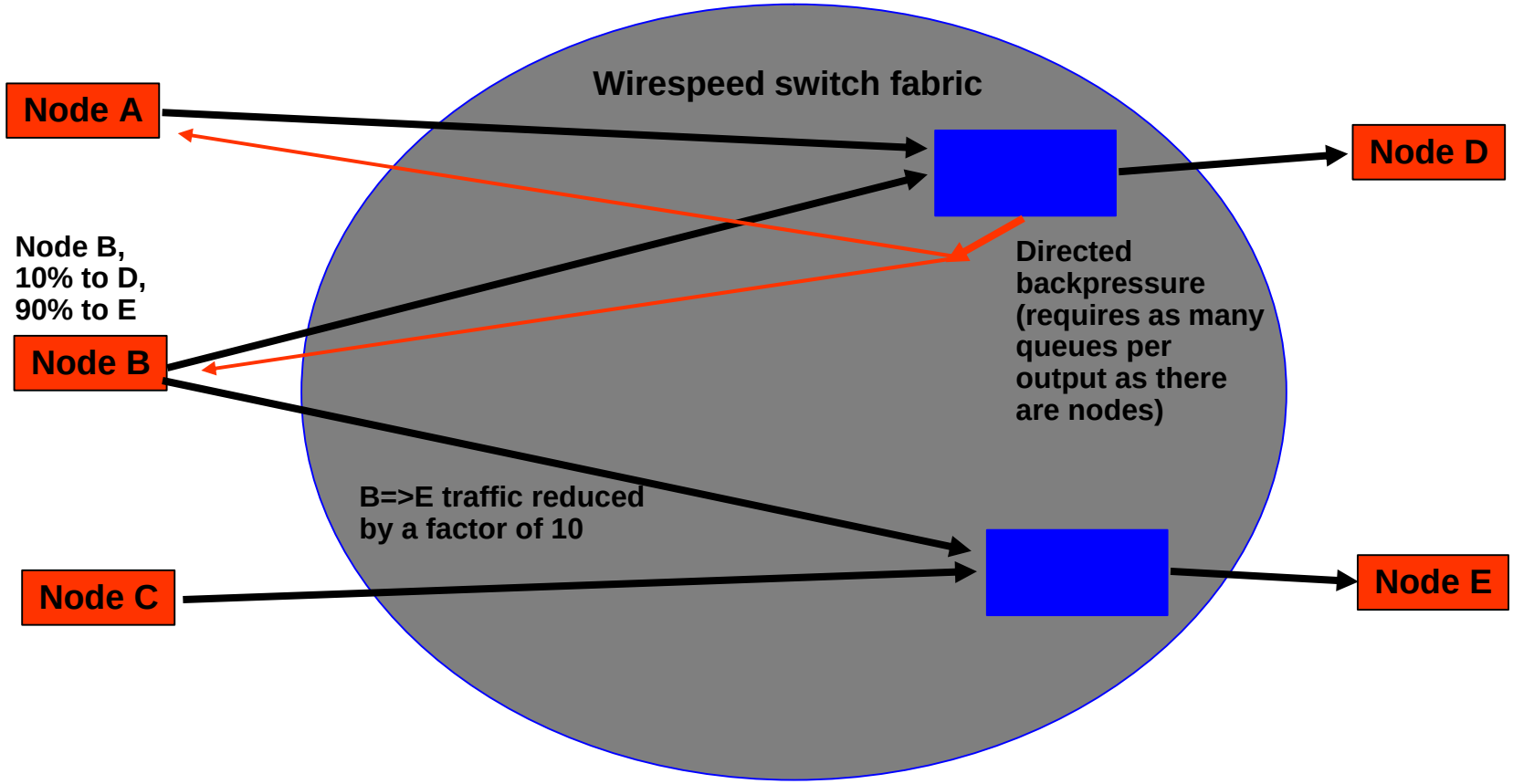
# Topological displacement

**Consider a simple network, nodes A, B, C, D, E.**



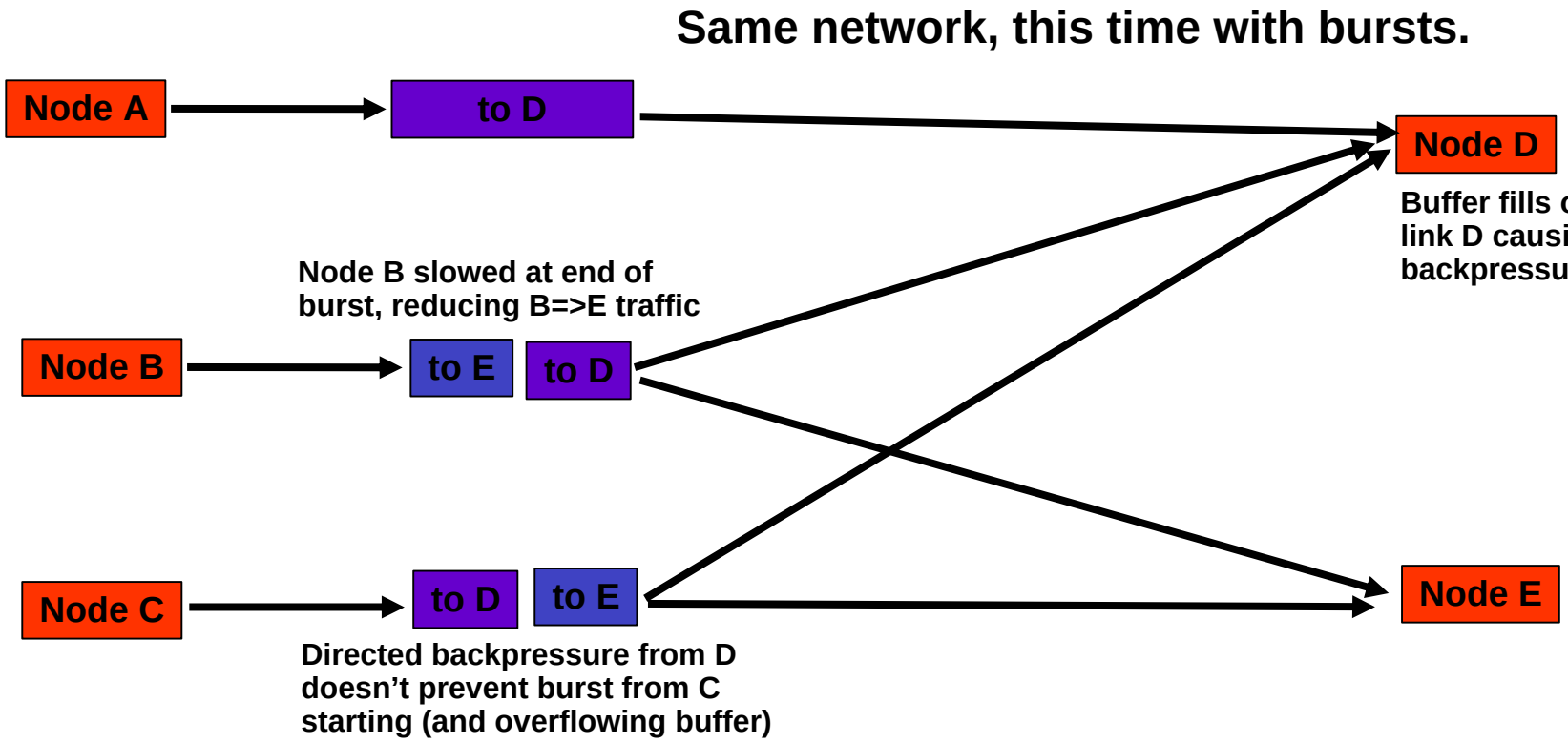
# It gets worse…

**What happens if link D is slower (say 1/10 speed)?**




**Head of line blocking – hits you where it hurts most**

# Temporal displacement



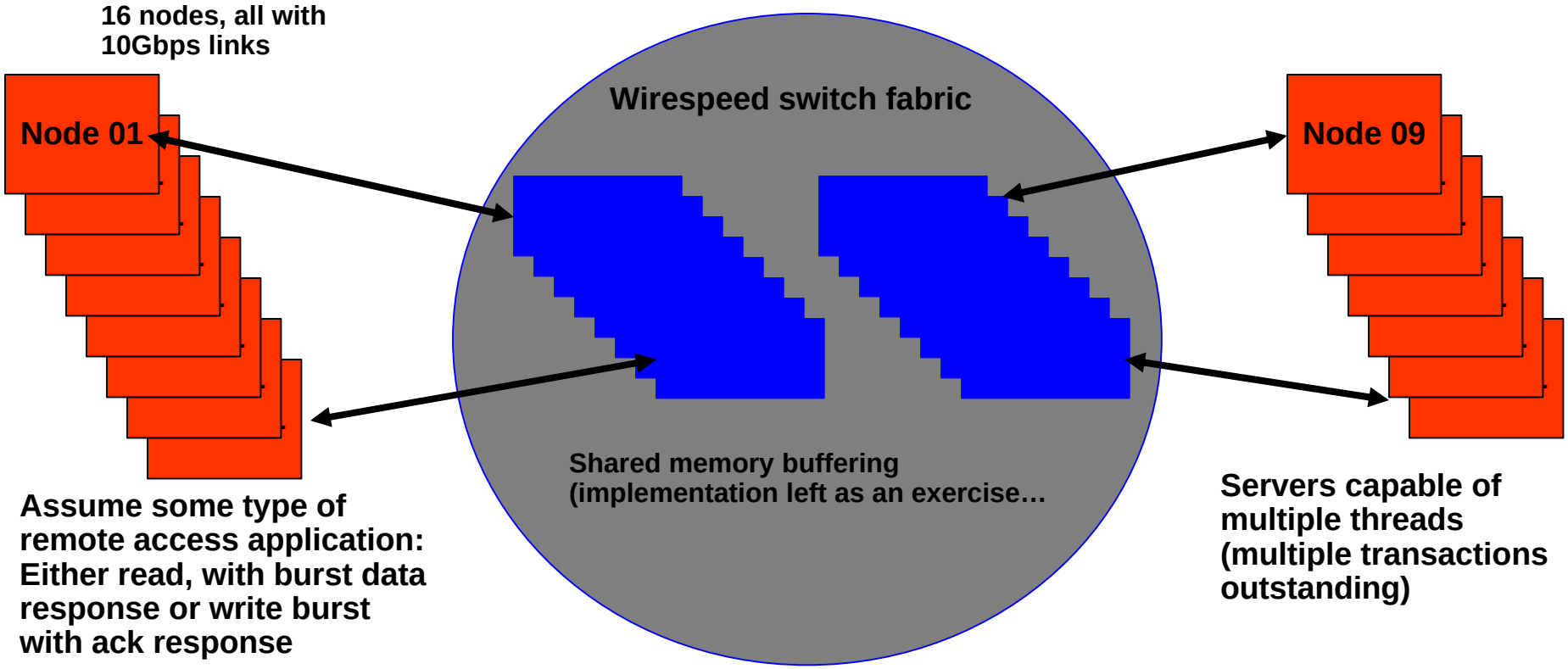

**Bursty nature of traffic causes additional problems with backpressure**
**Directed backpressure disastrous when bursts are directed differently**

**Temporal displacement is worst when network is just reaching saturation (the key time between happy operation and network upgrade)**

* Ethernet networks
* What (& when) is congestion
* Backpressure problems
* Network design for beginners
* Exceptions

# Simple network

**Should be representative of bladeserver (or cluster) applications**




**Look at the parameters to control…**

# Defining the network

**Key parameters:**

**Uncongested latency through the fabric.**

- **Assume 25uS.**
- **Also supports prioritization.**

**Switch buffer size - TBD.**

- **Assume ideal buffer sharing.**

**Server burst size - TBD.**

- **Server sets data priority low, request/ack priority high.**

**Number of outstanding transactions - TBD**

# Nail down the TBDs…

**Rule of thumb, shared memory switch buffer size should be the sum of ingress b/w times twice the network latency – for no packet loss.**

**160Gbps x 50uS => 8Mbit, 1Mbyte.**
**Well within practical limits, but real solutions may need more.**

**Server burst size needs to be less than network latency (e2e).**

**10Gbps x 25uS => 250kbit, 32kbyte.**

**Number of outstanding transactions should fill, but not overfill the pipe.**

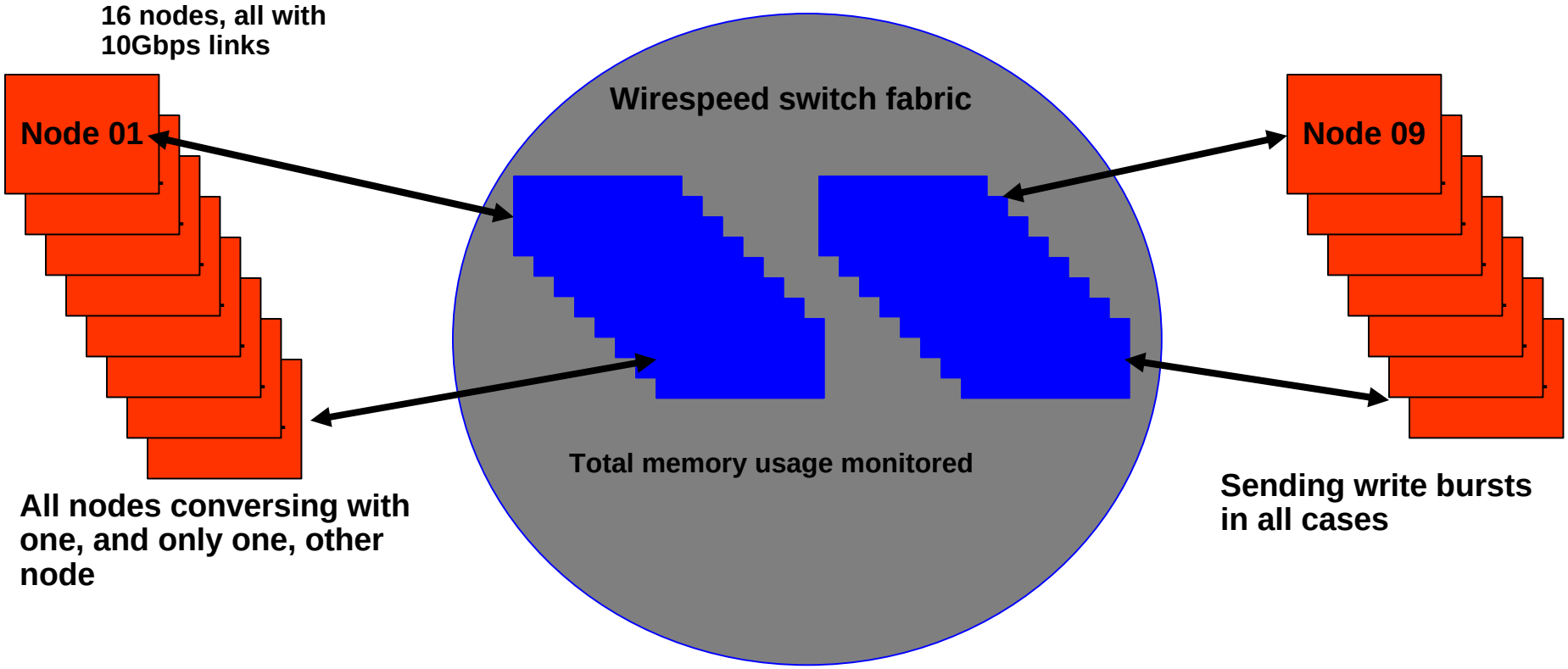
**2 transactions will leave no gaps in an ideal and uncongested network.**

**Then to be realistic, add some slack:**

**Increase buffer size and number of outstanding transactions.**
**4Mbyte and 4 x 64kbyte transactions – allows for inefficiencies.**

# Performance

## First a simple point to point model



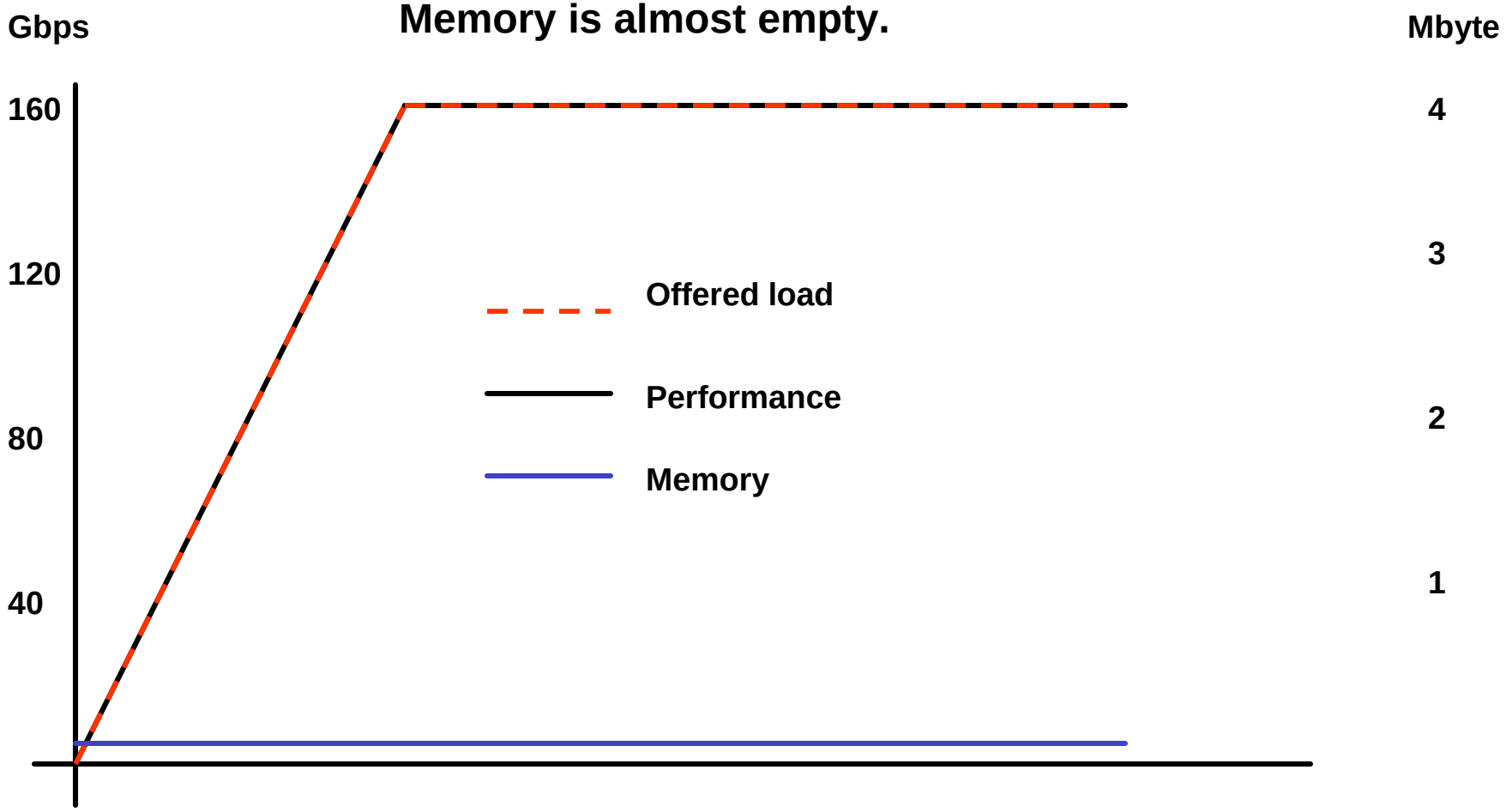

**The results are not surprising…**

# Non-blocking case results

**No surprises:**

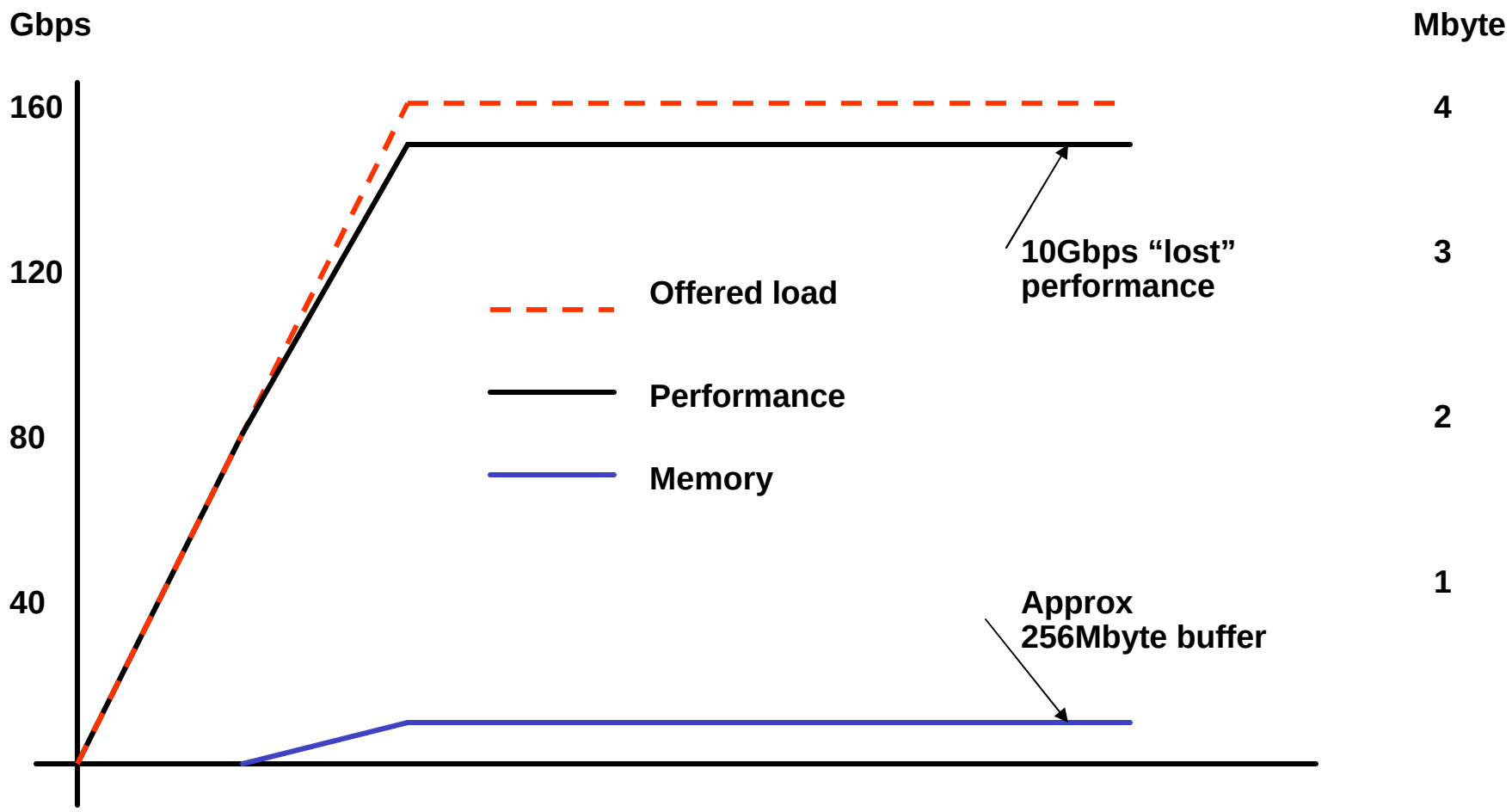
**Performance matches offered load**

**Memory is almost empty.**



# Next with limited blocking

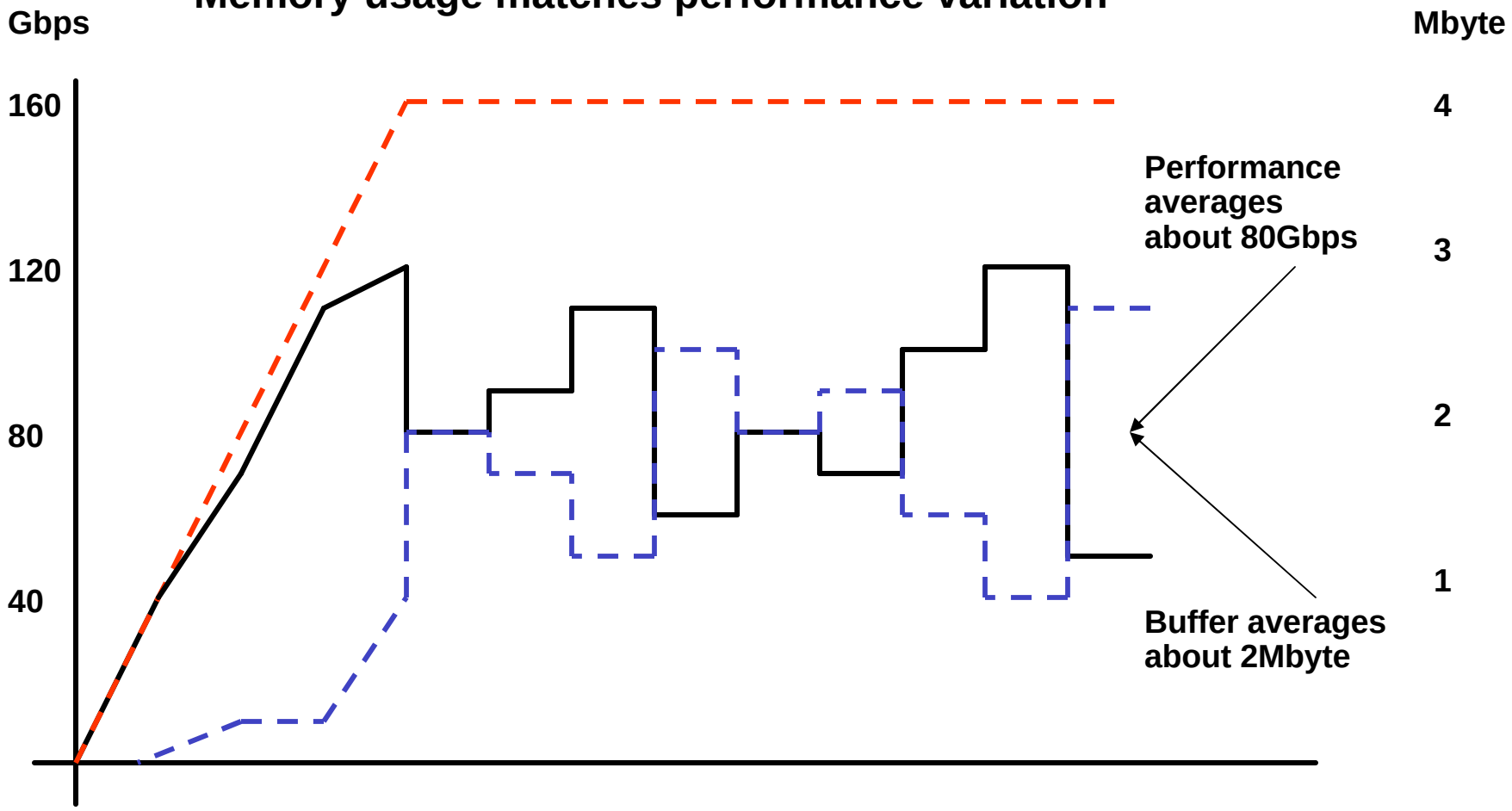
**One port sends to “clashing” address**
**Reduces the performance because of destination link limit**



# Finally, random destinations

**All ports sending data “randomly”**

- **Performance goes up & down according to number of “clashes”**
- **Memory usage matches performance variation**



# In conclusion…

**The network performed “perfectly”.**

- **No packets were lost.**
- **Performance matched the ideal in all cases*.**

**It should be expected that this behavior could be extended to a larger network.**

- **As long as buffer and burst sizes are controlled.**

**Note that some constraints will need to be monitored for scalability.**

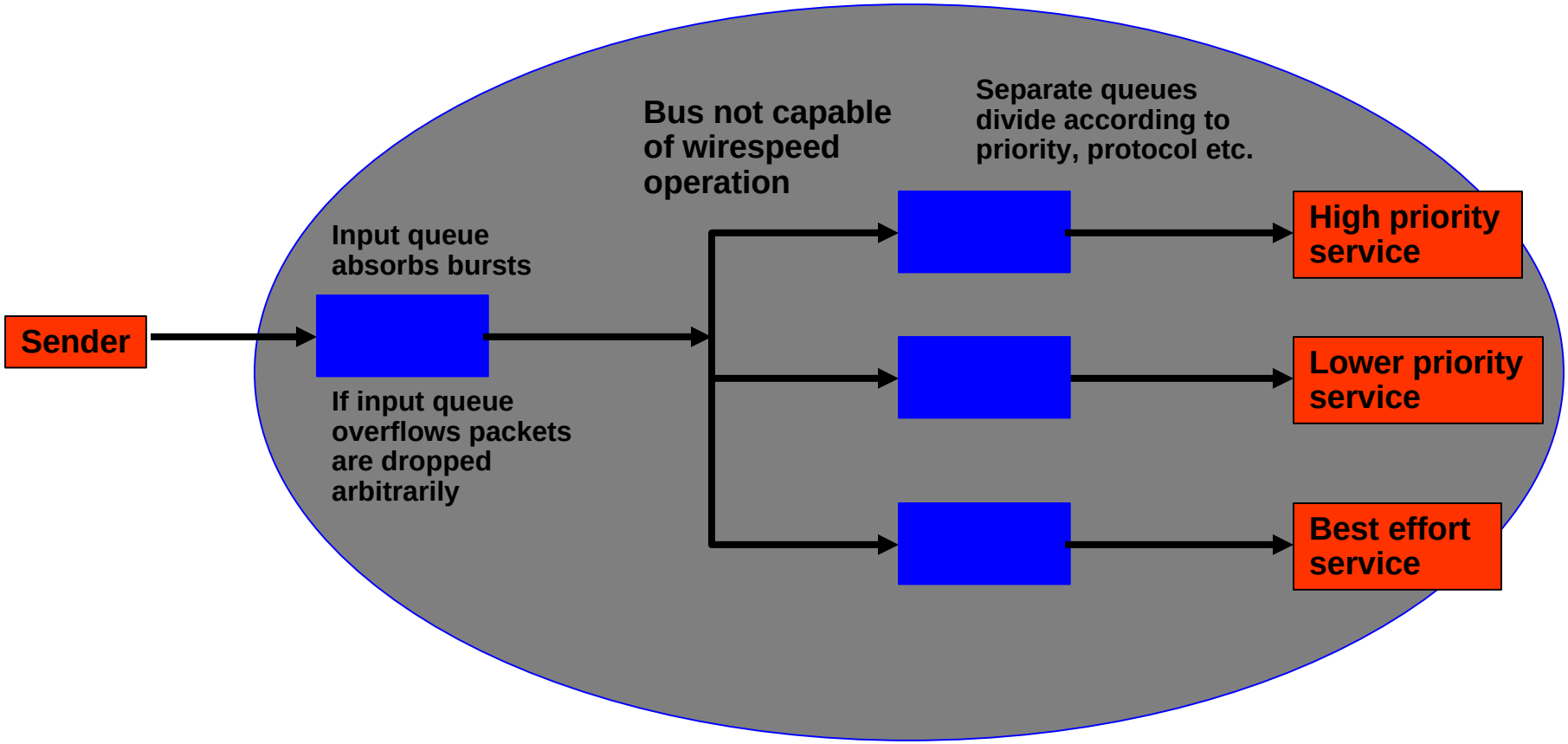
- **Network latency governs burst size...**
- **… burst size and number of nodes govern buffer size.**
- **Faster switch elements could have less buffering!**

*** Within acceptable tolerances**

* Ethernet networks
* What (& when) is congestion
* Backpressure problems
* Network design for beginners
* Exceptions

# A case for flow control

**In an example, a network device has an input queue because it cannot process incoming frames at line rate**




**System needs to limit sender rate so that input queue does not overflow**

# (Maybe) A second case

**Many access networks limit the rate for subscriber frames coming into the network.**

- **This is generally achieved by policing at the ingress port.**
- **This causes customer packets to be dropped if the application is able to fill the ingress buffer.**
- **Generally, there is no packet classification within the ingress queue therefore frames are dropped without regard to COS.**

**A protocol for telling the subscriber device that the rate is limited would be much more efficient.**

- **As long as it is widely adopted.**

**There is still the problem that the customer must deal with the limited rate link.**

- **The subscriber gateway device can use priority queuing on its output at the correct rate.**
- **Possibility for other smart solutions – the gateway knows what the upstream limits are.**

# So what would it take?

**The control of rate limiting could take many forms:**

**It could be a MAC control frame (like PAUSE).**
**It could be an SNMP request.**
**It could be something completely different.**
**Defined by 802.3, 802.1, or even IETF.**

**It won't need to change dynamically.**

**But may need to change on a human timescale.**
**(time of day variation, changes of policy, etc.)**

**It should operate by changing the IPG ...**

**… IPGstretch or pacing - not the burst duty cycle.**
**(PAUSE already does this)**


Cisco Systems