Interconnection Networks

Computer Architecture: A Quantitative Approach
4th Edition, Appendix E

Timothy Mark Pinkston
University of Southern California
http://ceng.usc.edu/smart/slides/appendixE.html

José Duato
Universidad Politécnica de Valencia
http://www.gap.upv.es/slides/appendixE.html

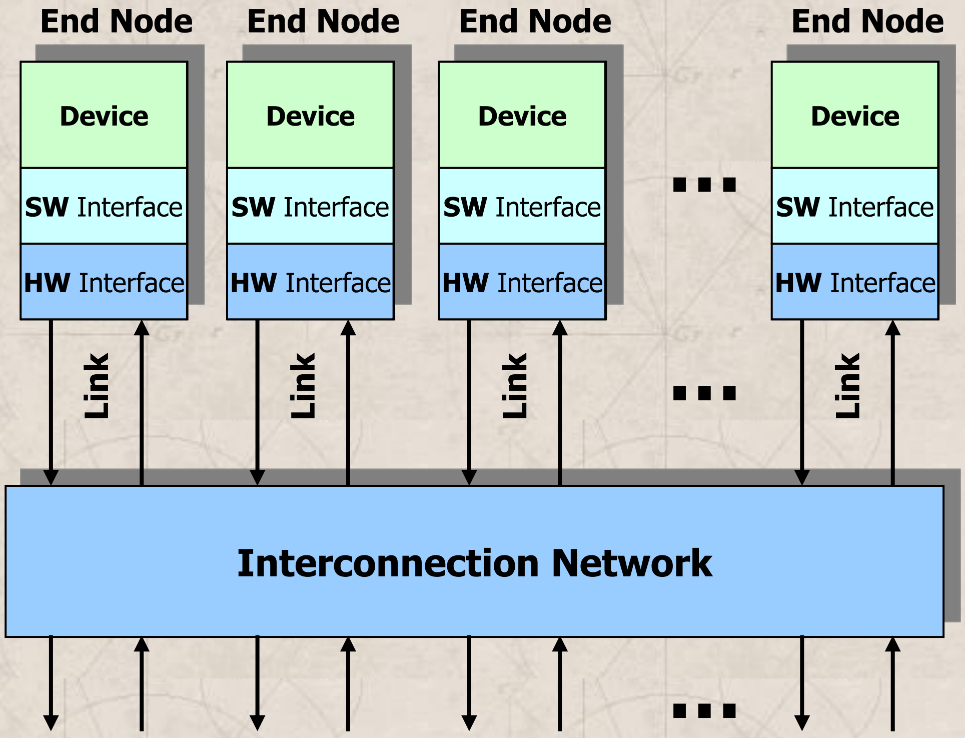
…with major presentation contribution from José Flich, UPV
(and Cell BE EIB slides by Tom Ainsworth, USC)

Outline

- **E.1 Introduction *(Lecture 1)***
- E.2 Interconnecting Two Devices *(Lecture 1)*
- E.3 Interconnecting Many Devices *(Lecture 2)*
- E.4 Network Topology *(Lecture 2)*
- E.5 Network Routing, Arbitration, and Switching *(Lecture 3)*
- E.6 Switch Microarchitecture *(Lecture 4)*
- E.7 Practical Issues for Commercial Interconnection Networks *(Lec 4)*
- E.8 Examples of Interconnection Networks *(Lecture 5)*
- E.9 Internetworking (skipped)
- E.10 Crosscutting Issues for Interconnection Networks (skipped)
- E.11 Fallacies and Pitfalls *(Lecture 5)*
- E.12 Concluding Remarks and References *(Lecture 5)*

Introduction

How to connect individual devices together into a community of communicating devices?



- Device (definition):
  - Component within a computer
  - Single computer
  - System of computers
- Types of elements:
  - end nodes (device + interface)
  - links
  - interconnection network
- Internetworking: interconnection of multiple networks

- ***Interconnection networks*** *should be designed to transfer the* *<u>maximum amount of information</u>* *within the* *<u>least amount of time</u>* *(and cost, power constraints) so as not to bottleneck the system*

Introduction

Interconnection Network Domains

Interconnection networks can be grouped into *four major networking domains*, depending on the *number* and *proximity* of devices to be interconnected: *OCNs, SANs, LANs,* and *WANs*

- *On-chip networks* (*OCNs*), a.k.a., network-on-chip (NoC)
  - – Interconnect microarchitecture functional units, register files, caches, compute tiles, processor and IP cores
  - – Chip or multichip modules
  - – Tens (in future, possibly 100s) of devices interconnected
  - – Maximum interconnect distance on the order of centimeters
  - – Examples (custom designed)
    - › Element Interconnect Bus (Cell Broadband Engine processor chip)
      - » 2,400 Gbps (3.2 Ghz processor clock), 12 elements on the chip
  - – Examples (proprietary designs)
    - › CoreConnect (IBM), AMBA (ARM), Smart Interconnect (Sonic)

Introduction

Interconnection Network Domains

- *System/storage area networks* (*SANs*)
  - Multiprocessor and multicomputer systems
    - Interprocessor and processor-memory interconnections
  - Server and data center environments
    - Storage and I/O components
  - Hundreds to thousands of devices interconnected
    - IBM Blue Gene/L supercomputer (64K nodes, each with 2 processors)
  - Maximum interconnect distance typically on the order of tens of meters, but some with as high as a few hundred meters
    - InfiniBand: 120 Gbps over a distance of 300 m
  - Examples (standards and proprietary)
    - InfiniBand, Myrinet, Quadrics, Advanced Switching Interconnect

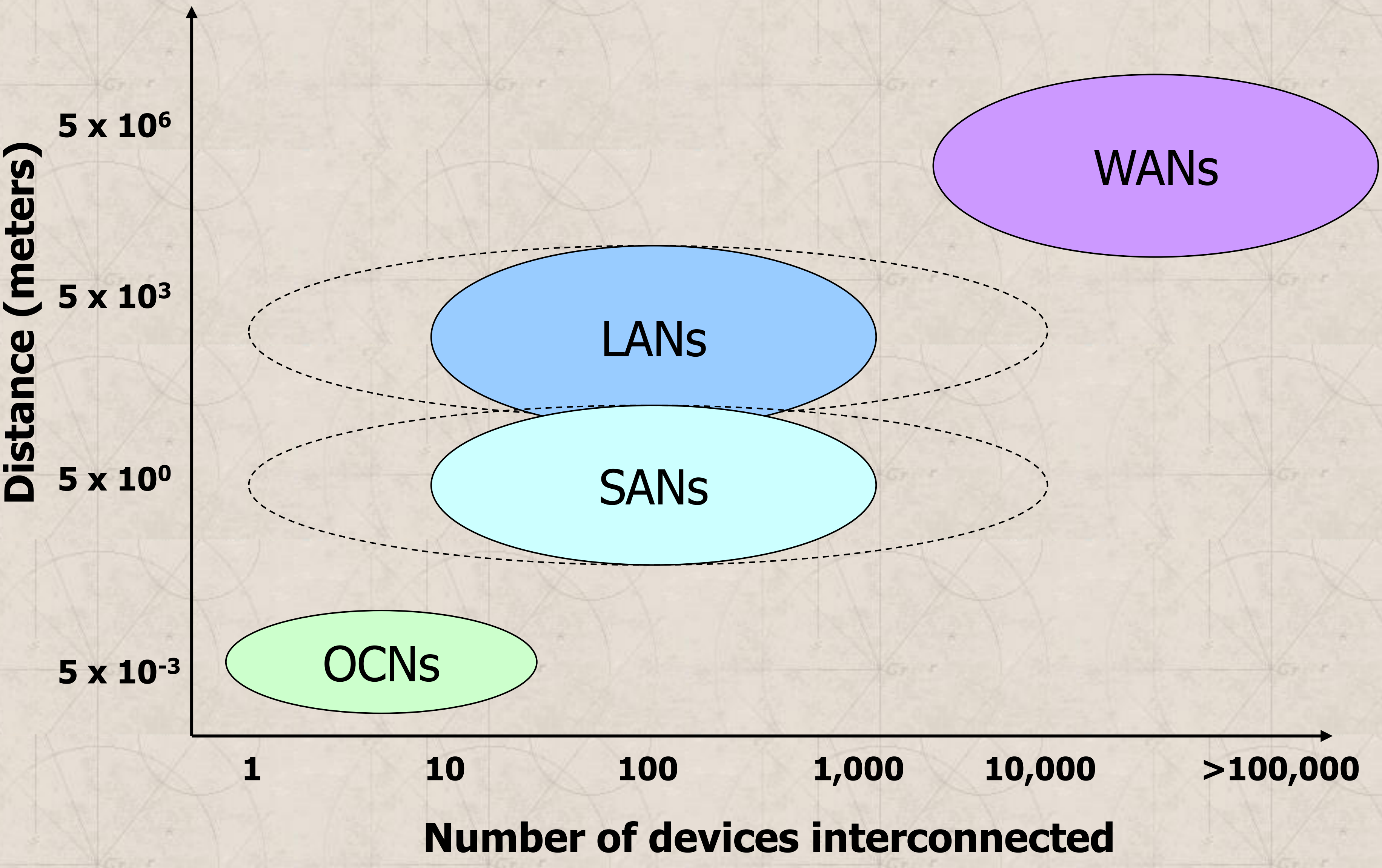
Introduction

Interconnection Network Domains

- *Local area networks* (*LANs*)
  - – Interconnect autonomous computer systems
  - – Machine room or throughout a building or campus
  - – Hundreds of devices interconnected (1,000s with bridging)
  - – Maximum interconnect distance on the order of few kilometers, but some with distance spans of a few tens of kilometers
  - – Hundred devices (thousands with bridging)
  - – Example (most popular): Ethernet, with 10 Gbps over 40Km
- *Wide area networks* (*WANs*)
  - – Interconnect systems distributed across the globe
  - – Internetworking support is required
  - – Many millions of devices interconnected
  - – Maximum interconnect distance of many thousands of kilometers
  - – Example: ATM

Introduction

Interconnection Network Domains



Introduction

Organization

- Top-down Approach
  - We unveil concepts and complexities involved in designing interconnection networks by first viewing the network as an ideal “black box” and then systematically removing various layers of the black box, exposing non-ideal behavior and complexities.
  - We first consider interconnecting only two devices (E.2)
  - We then consider interconnecting many devices (E.3)
  - Other layers of the black box are peeled away, exposing the network topology, routing, arbitration, and switching (E.4, E.5)
  - We then zoom in on the switch microarchitecture (E.6)
  - Next, we consider Practical Issues for Interconnection Networks (E.7)
  - Finally, we look at some examples: OCNs and SANs (E.8)
  - Internetworking (E.9) (skipped)
  - Additional Crosscutting Issues (E.10) (skipped)
  - Fallacies and Pitfalls (E.11)
  - Concluding Remarks (E.12) and References

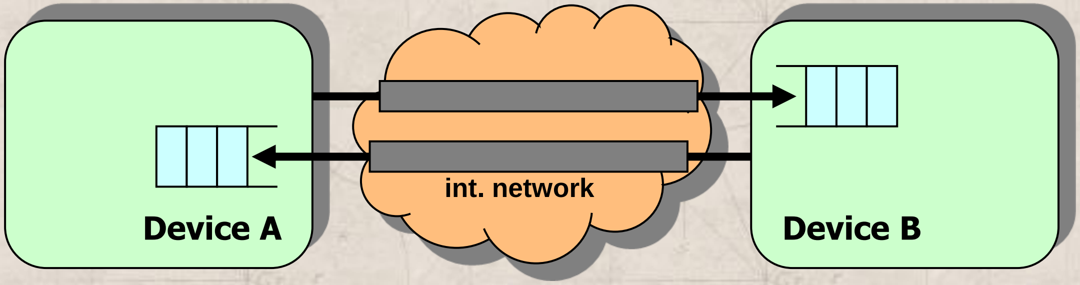
Outline

- E.1 Introduction *(Lecture 1)*
- **E.2 Interconnecting Two Devices** *(Lecture 1)*
  - **An Ideal Network**
  - **Network Interface Functions**
  - **Communication Protocol**
  - **Basic Network Structure and Functions**
  - **Characterizing Performance: Latency & Effective Bandwidth**
  - **Basic Network Characteristics in Commercial Machines**
- E.3 Interconnecting Many Devices *(Lecture 2)*
- E.4 Network Topology *(Lecture 2)*
- E.5 Network Routing, Arbitration, and Switching *(Lecture 3)*
- E.6 Switch Microarchitecture *(Lecture 4)*
- E.7 Practical Issues for Commercial Interconnection Networks *(Lecture 4)*
- E.8 Examples of Interconnection Networks *(Lecture 5)*
- E.9 Internetworking (skipped)
- E.10 Crosscutting Issues for Interconnection Networks (skipped)
- E.11 Fallacies and Pitfalls *(Lecture 5)*
- E.12 Concluding Remarks and References *(Lecture 5)*

Interconnecting Two Devices

An Ideal Network

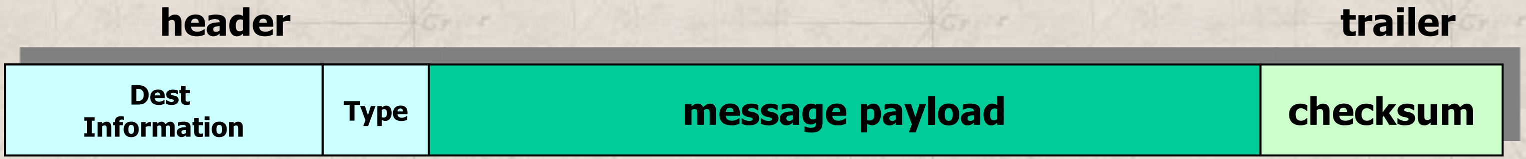
- Two end node devices (A and B)
- Device A (or B) may read an address in device B (or A)
- Interconnection network behaves as *dedicated links* between A, B
  - Unidirectional wires each way dedicated to each device
- Receiving buffers used for staging the transfers at the end nodes
- Communication protocol: *request, reply*
  - basic functions at end nodes to commence and complete comm.



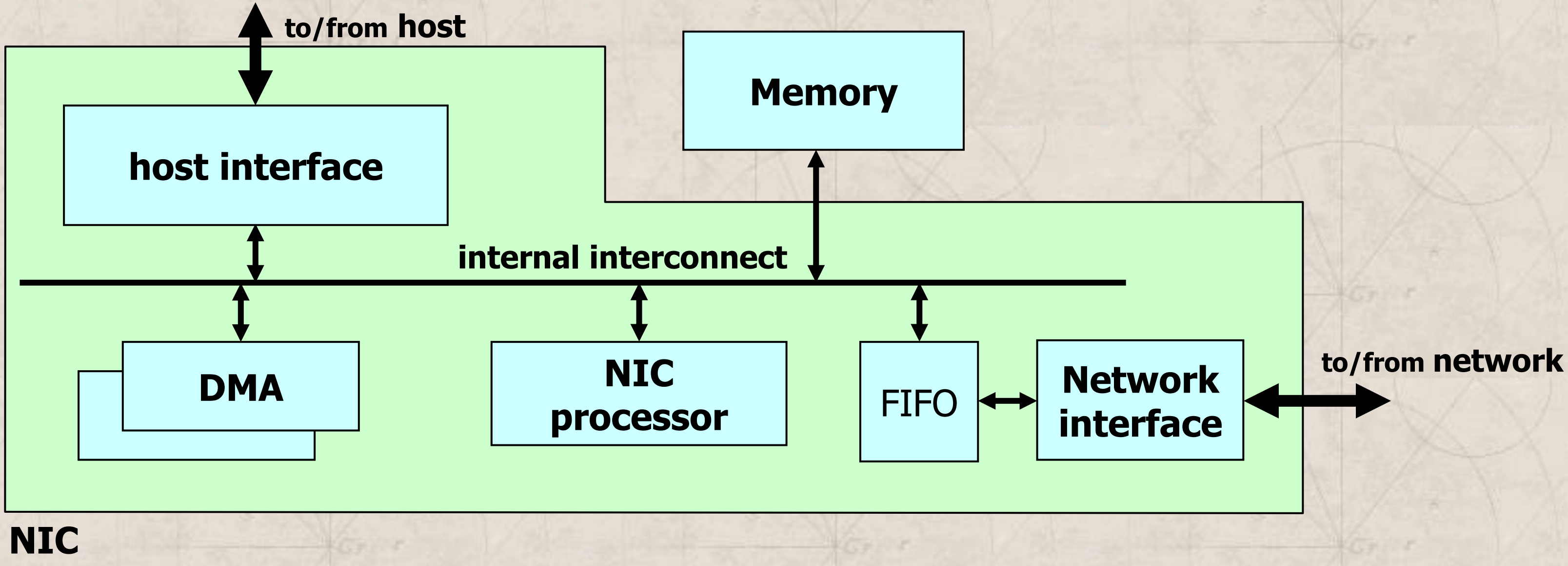
Interconnecting Two Devices

Network Interface Functions

- A *message* is the unit of information: header, payload, trailer



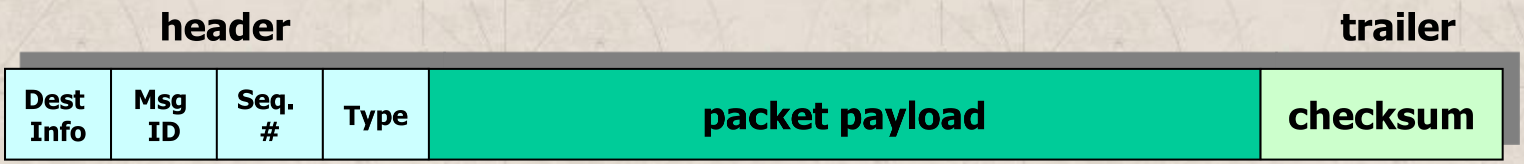
- Interfacing to the network (hardware)
  - Communication device itself (OCNs and some SANs)
  - Additional network interface card or *NIC* (SANs, LANs, WANs)
    - › Embedded processor(s), DMA engine(s), RAM memory



Interconnecting Two Devices

Network Interface Functions

- Interfacing to the network (software, firmware)
  - Translate *requests* and *replies* into messages
  - Tight integration with operating system (OS)
    - › Process isolation, protection mechanisms, etc.
    - › Port (binds sender process with intended receiver process)
  - Packetization
    - › *Maximum transfer unit* (MTU) used for dimensioning resources
    - › Messages larger than MTU divided into *packets* with *message id*
    - › Packet reordering at destination (for message reassembly) is done using *sequence numbers*

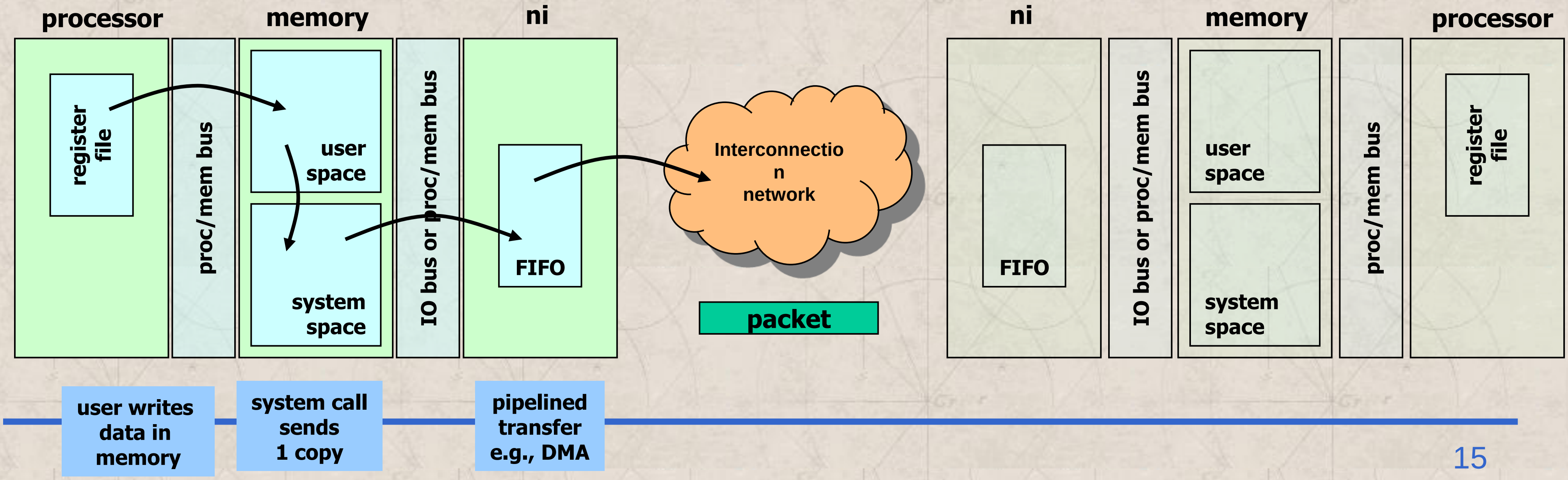


00 = request
01 = reply
10 = request acknowledge
11 = reply acknowledge

Interconnecting Two Devices

Communication Protocol

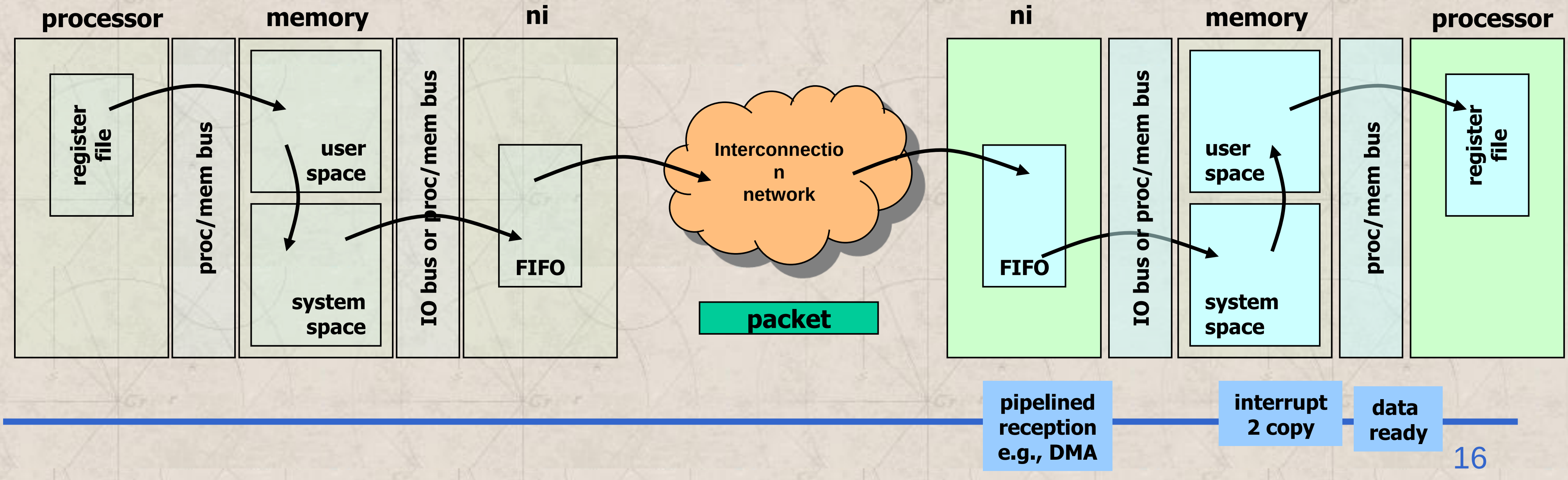
- Typical steps followed by the *sender:*
  1. System call by application
     - Copies the data into OS and/or network interface memory
     - Packetizes the message (if needed)
     - Prepares headers and trailers of packets
  2. Checksum is computed and added to header/trailer
  3. Timer is started and the network interface sends the packets



Interconnecting Two Devices

Communication Protocol

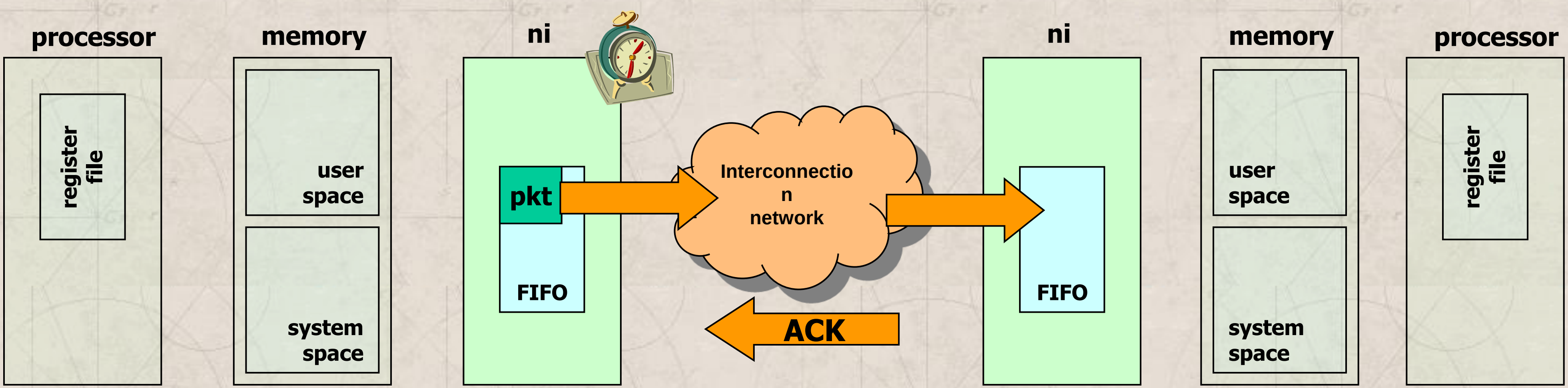
- Typical steps followed by the *receiver:*
  1. NI allocates received packets into its memory or OS memory
  2. Checksum is computed and compared for each packet
     › If checksum matches, NI sends back an ACK packet
  3. Once all packets are correctly received
     › The message is reassembled and copied to user's address space
     › The corresponding application is signalled (via polling or interrupt)



Interconnecting Two Devices

Communication Protocol

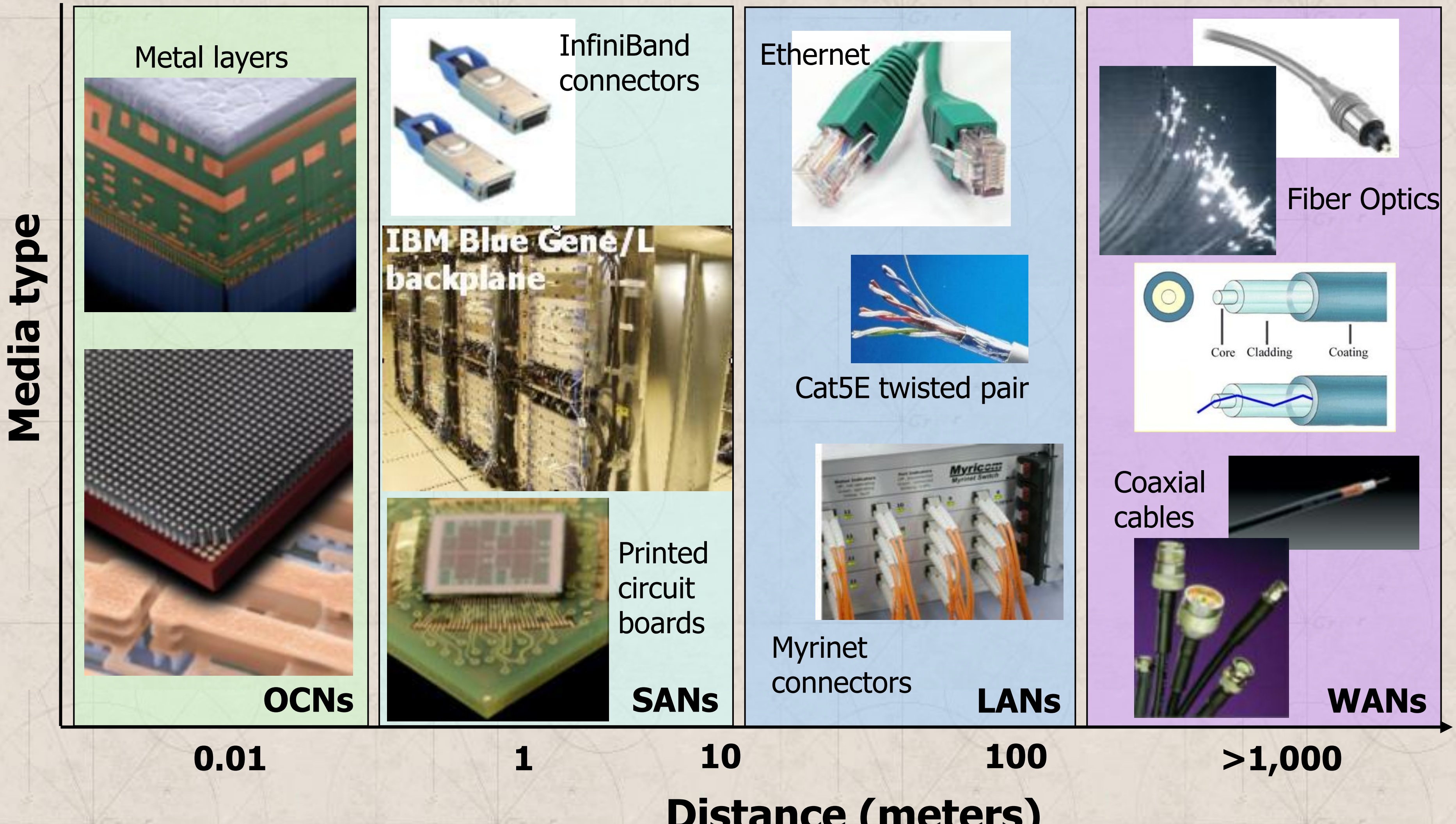
- Additional steps at the sender side:
  - – ACK received: the copy is released and timer is cancelled
  - – Timer expires before receiving an ACK: packet is resent and the timer is started



Interconnecting Two Devices

Basic Network Structure and Functions

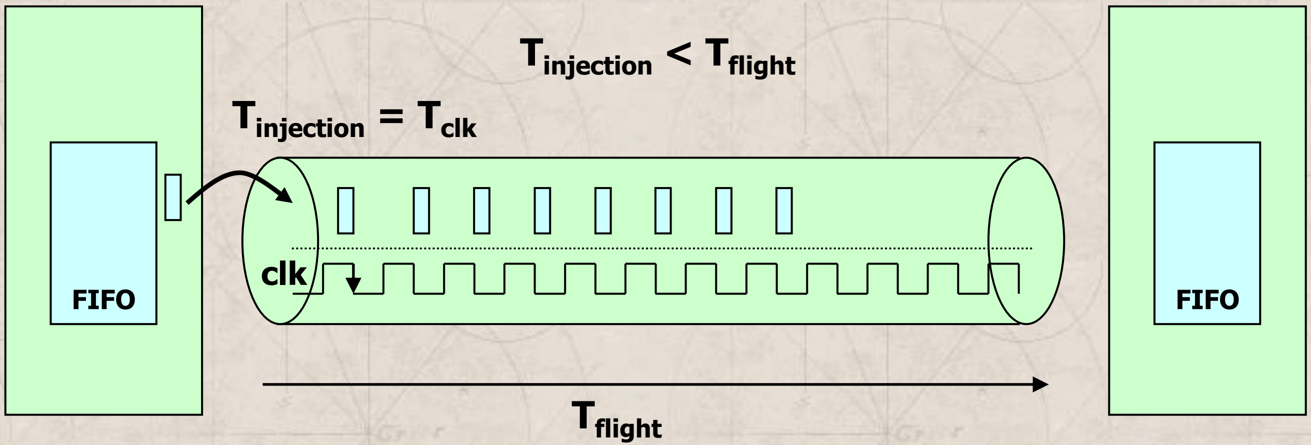
- *Media* and *Form Factor*



Interconnecting Two Devices

Basic Network Structure and Functions

- *Packet Transport*
  - Hardware circuitry needed to drive network links
  - Encoding at the sender and decoding at the receiver
    - Multiple voltage levels, redundancy, data & control rotation (4b/5b)
    - Encoding—along with packet headers and trailers—adds some overhead, which reduces link efficiency
  - Physical layer abstraction:
    - viewed as a long linear pipeline without staging
    - signals propagate as waves through the transmission medium



pipelined transfer

Interconnecting Two Devices

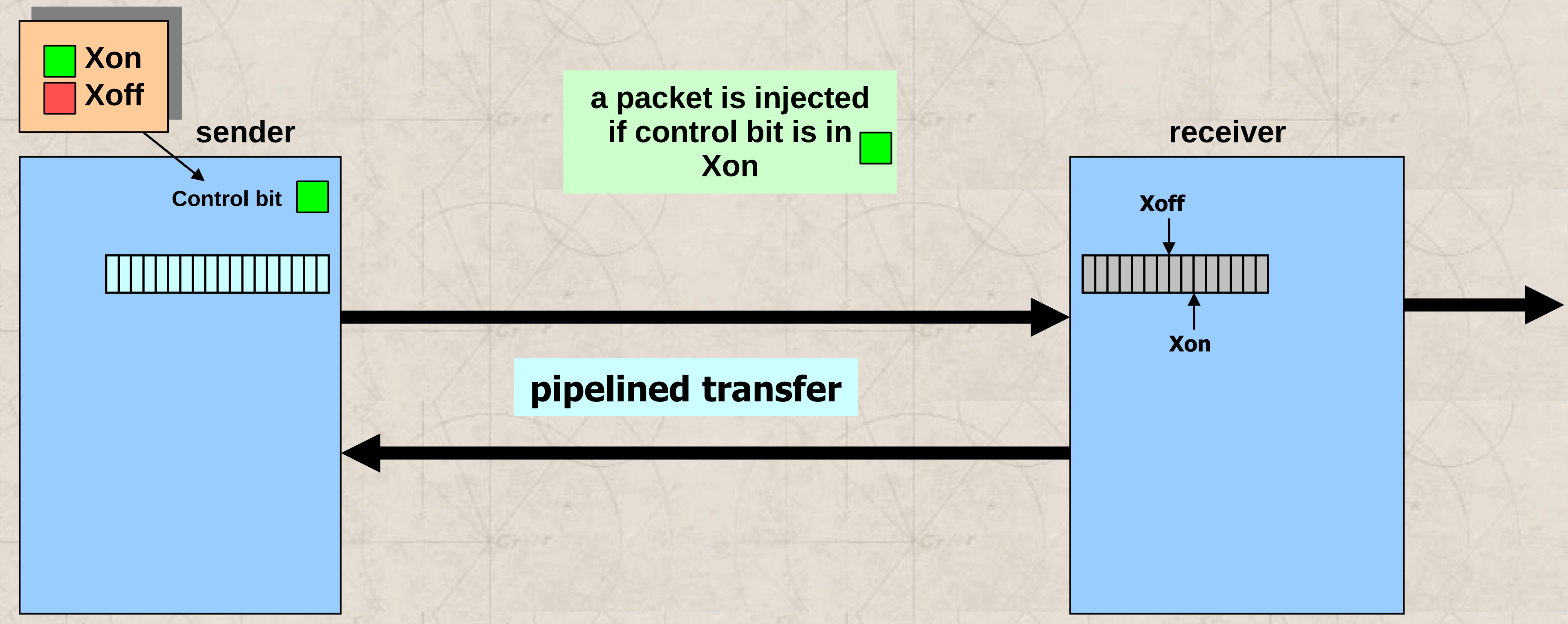
Basic Network Structure and Functions

- *Reliable delivery* of packets: *lossy* versus *lossless* networks
  - Must prevent the sender from sending packets at a faster rate than can be received and processed by the receiver
  - Lossy networks
    - Packets are dropped (discarded) at receiver when buffers fill up
    - Sender is notified to retransmit packets (via time-out or NACK)
  - Lossless networks (flow controlled)
    - Before buffers fill up, sender is notified to stop packet injection
      - *Xon/Xoff* (Stop & Go) flow control
      - *Credit-based* flow control (token or batched modes)
  - Implications of network type (lossless vs. lossy)
    - Constrains the solutions for packet routing, congestion, & deadlock
    - Affects network performance
    - The interconnection network domain dictates which is used
      - OCN, SAN: typically lossless; LAN, WAN: typically lossy

Interconnecting Two Devices

Basic Network Structure and Functions

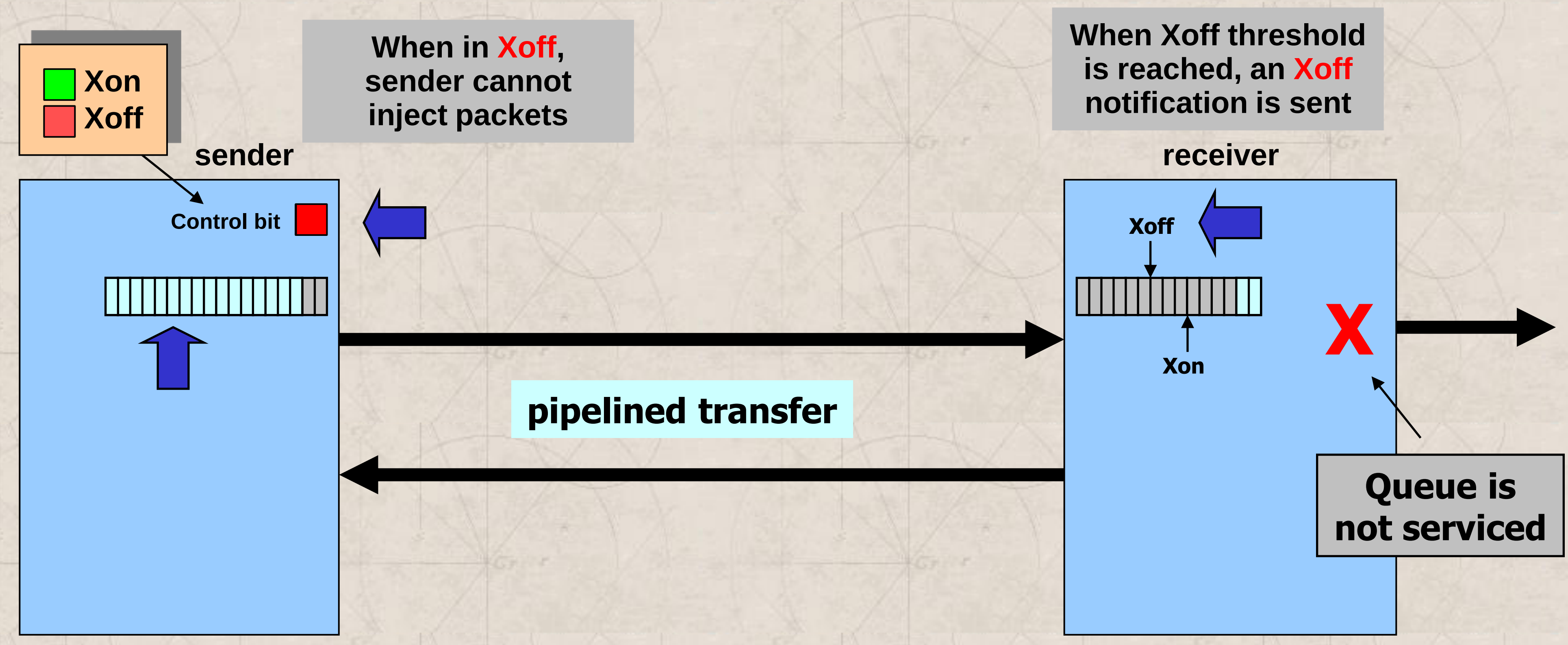
- Xon/Xoff flow control



Interconnecting Two Devices

Basic Network Structure and Functions

- Xon/Xoff flow control

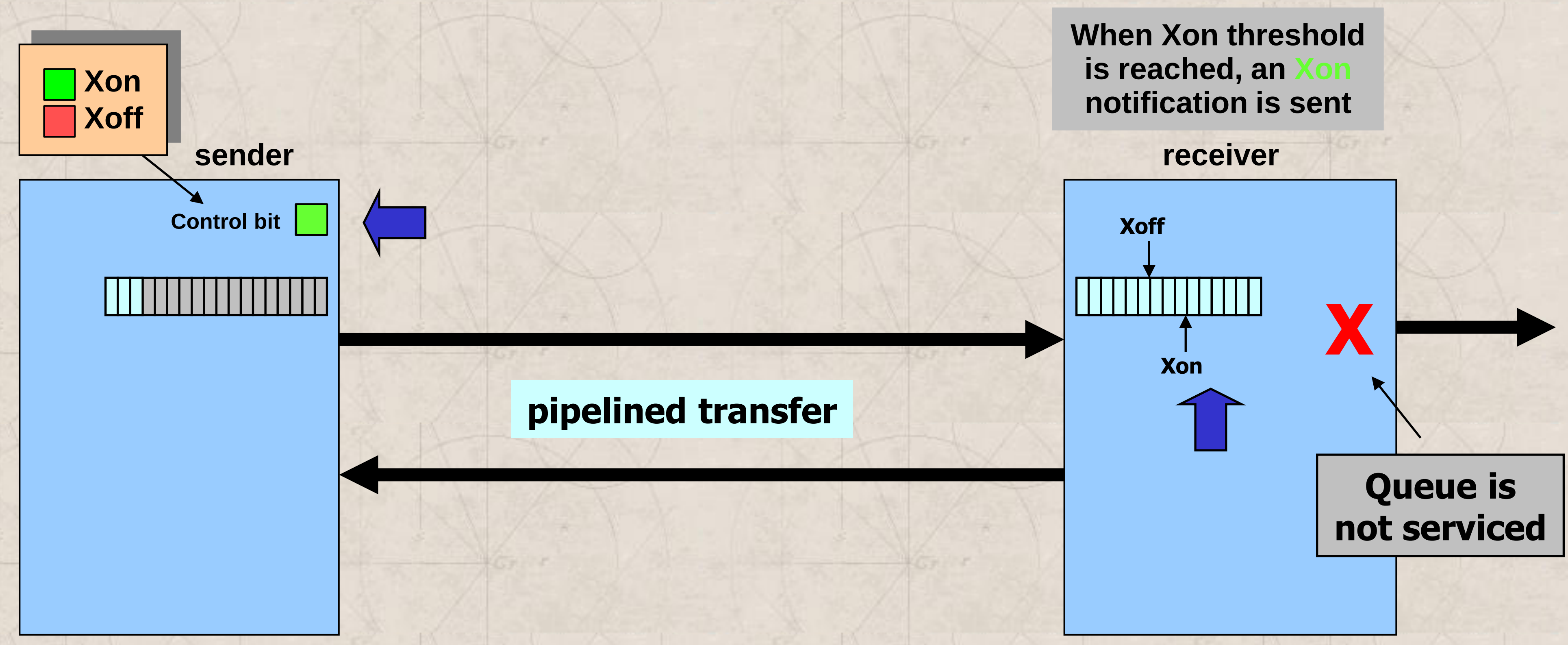


🟩 Xon
🟥 Xoff
sender
When in Xoff, sender cannot inject packets
When Xoff threshold is reached, an Xoff notification is sent
receiver
Control bit
Xoff
Xon
X
pipelined transfer
Queue is not serviced

Interconnecting Two Devices

Basic Network Structure and Functions

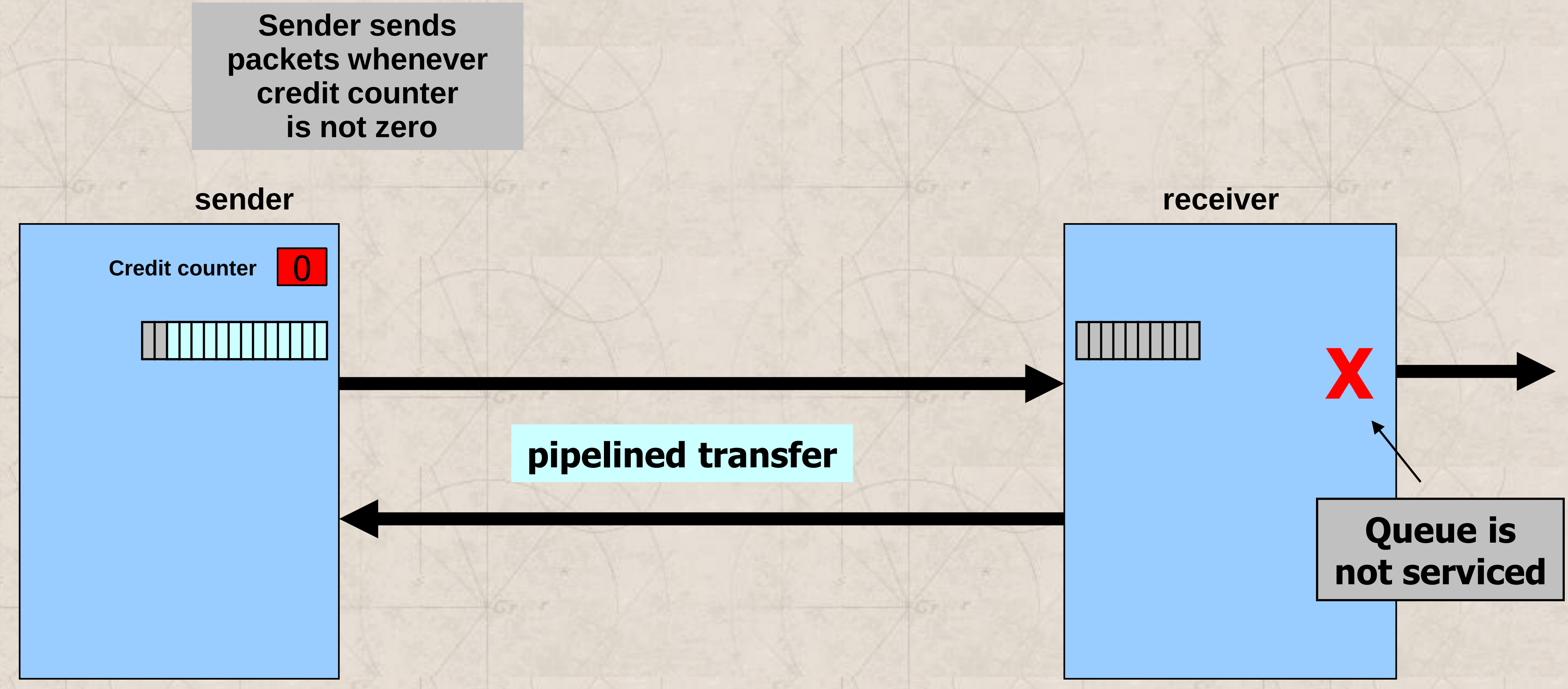
- Xon/Xoff flow control



Interconnecting Two Devices

Basic Network Structure and Functions

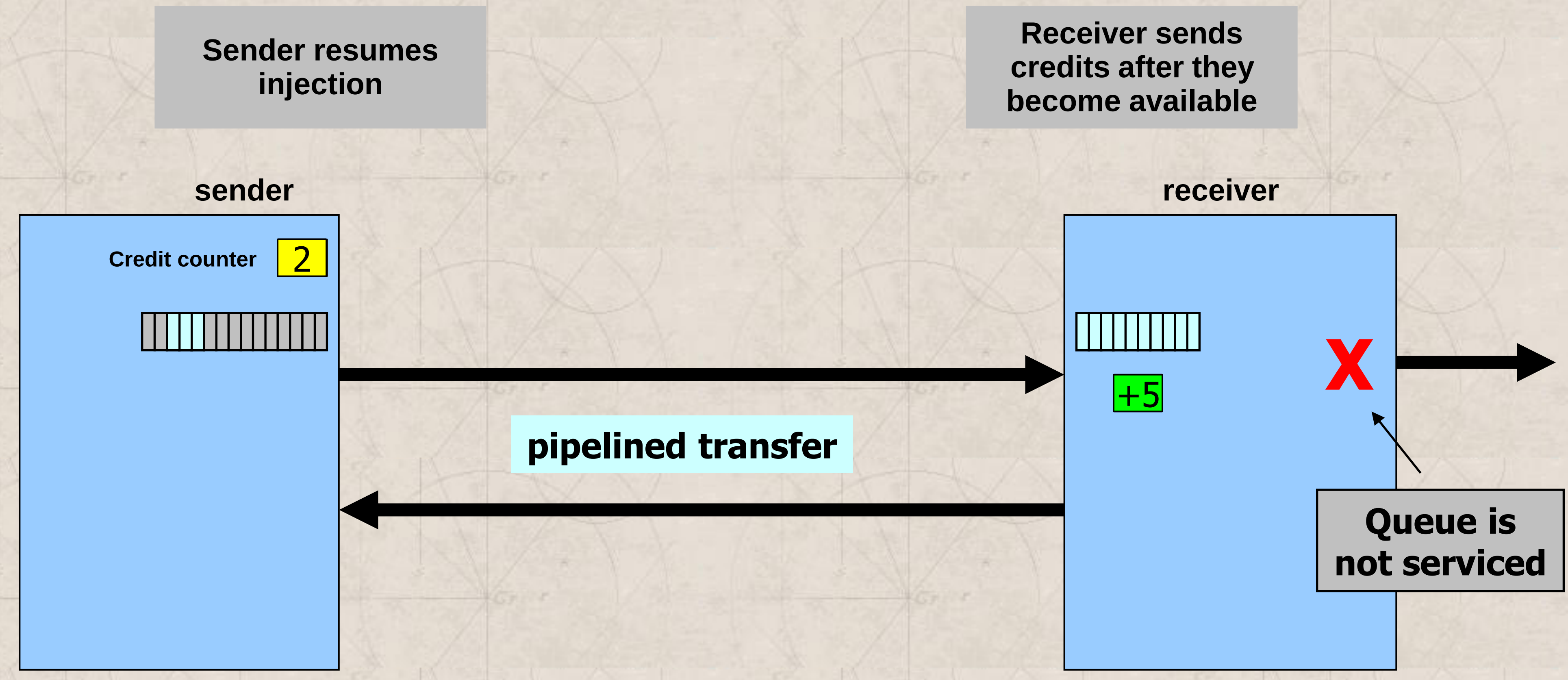
- Credit-based flow control



Interconnecting Two Devices

Basic Network Structure and Functions

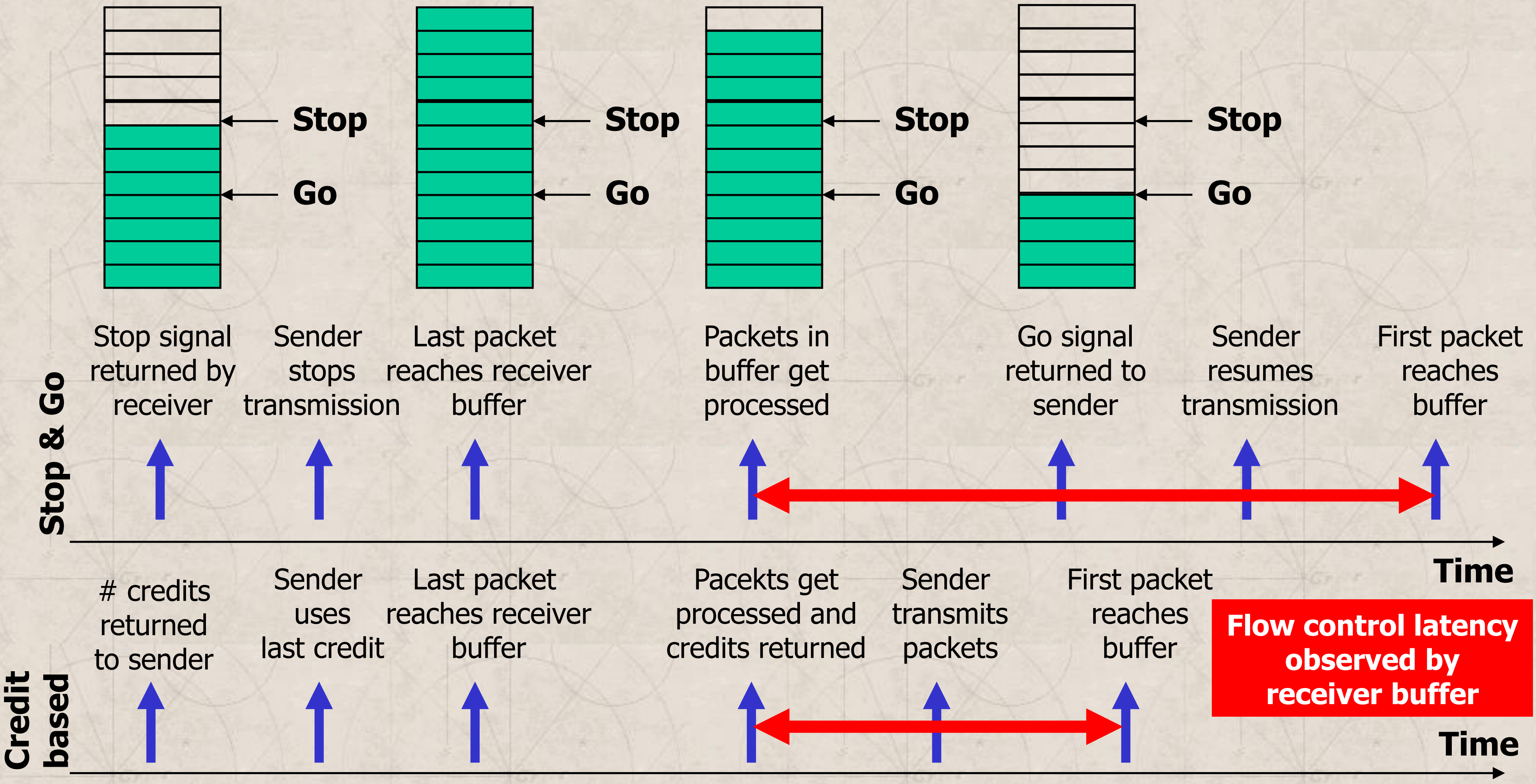
- Credit-based flow control



Interconnecting Two Devices

Basic Network Structure and Functions

- Comparison of Xon/Xoff vs credit-based flow control



Stop

Go

Stop & Go

Stop signal returned by receiver

Sender stops transmission

Last packet reaches receiver buffer

Packets in buffer get processed

Go signal returned to sender

Sender resumes transmission

First packet reaches buffer

Time

Credit based

# credits returned to sender

Sender uses last credit

Last packet reaches receiver buffer

Pacekts get processed and credits returned

Sender transmits packets

First packet reaches buffer

Flow control latency observed by receiver buffer

Time

➔ ***Poor flow control can reduce link efficiency***

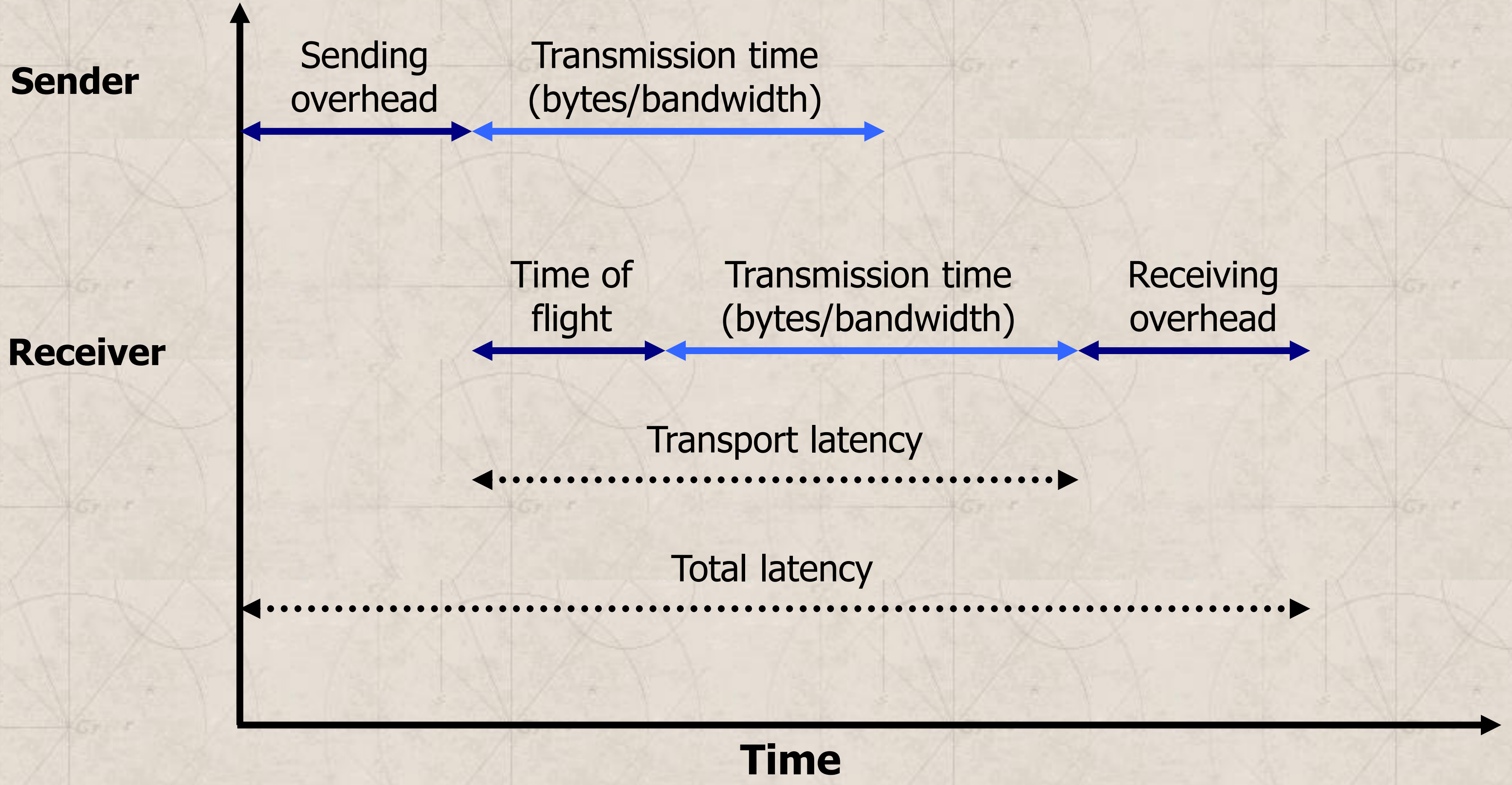
Interconnecting Two Devices

Characterizing Performance: Latency & Effective Bandwidth

- *Transmission time*:
  - The time for a packet to pass through the network, not including the time of flight
  - Equal to the packet size divided by the data bandwidth of the link
- *Transport latency*:
  - Sum of the time of flight and the transmission time
  - Measures the time that a packet spends in the network
- *Sending overhead (latency)*:
  - Time to prepare a packet for injection, including hardware/software
  - A constant term (packet size) plus a variable term (buffer copies)
- *Receiving overhead (latency)*:
  - Time to process an incoming packet at the end node
  - A constant term plus a variable term
  - Includes cost of interrupt, packet reorder and message reassembly

Interconnecting Two Devices

Characterizing Performance: Latency & Effective Bandwidth



Sender

Sending overhead

Transmission time (bytes/bandwidth)

Receiver

Time of flight

Transmission time (bytes/bandwidth)

Receiving overhead

Transport latency

Total latency

Time

$$\textbf{Latency} = \textit{Sending Overhead} + \textit{Time of flight} + \frac{\textit{packet size}}{\textit{Bandwidth}} + \textit{Receiving Overhead}$$

Interconnecting Two Devices

Characterizing Performance: Latency & Effective Bandwidth

- Example (latency):

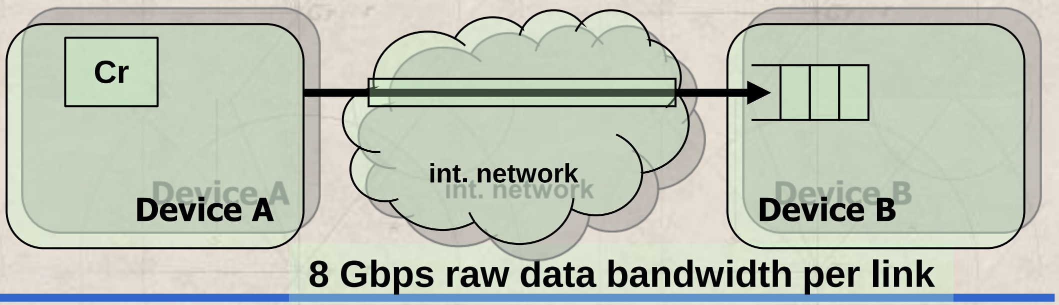
$$Latency = Sending\ overhead + Time\ of\ flight + \frac{Packet\ size}{Bandwidth} + Receiving\ overhead$$

$$Latency_{OCN} = 5\ ns + 0.025\ ns + 100\ ns + 5\ ns = 110.025\ ns$$

$$Latency_{SAN} = 0.305\ \mu s + 0.025\ ns + 0.1\ \mu s + 0.405\ \mu s = 0.835\ \mu s$$

$$Latency_{LAN} = 3.005\ \mu s + 25\ \mu s + 0.1\ \mu s + 4.005\ \mu s = 32.11\ \mu s$$

$$Latency_{WAN} = 20.05\ \mu s + 25\ \mu s + 0.1\ \mu s + 40.05\ \mu s = 25.07\ ms$$

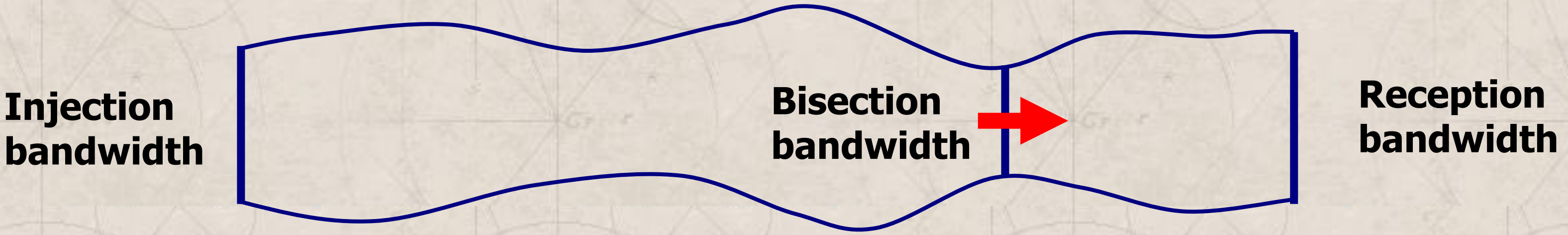


Interconnecting Two Devices

Characterizing Performance: Latency & Effective Bandwidth

A Simple (General) Throughput Performance Model:

- The network can be considered as a "*pipe*" of variable width



- There are three points of interest *end-to-end*:
  – Injection into the pipe
  – Narrowest section within pipe (i.e., minimum network bisection that has traffic crossing it)
  – Reception from the pipe
- *The bandwidth at the narrowest point* ***and utilization of that bandwidth*** *determines the throughput!!!*

Interconnecting Two Devices

Characterizing Performance: Latency & Effective Bandwidth

A Simple (General) Throughput Performance Model:

$$\begin{aligned} \textit{Effective bandwidth} &= \min(BW_{NetworkInjection},\ BW_{Network},\ \boldsymbol{\sigma} \times BW_{NetworkReception}) \\ &= \min(N \times BW_{LinkInjection},\ BW_{Network},\ \boldsymbol{\sigma} \times N \times BW_{LinkReception}) \end{aligned}$$

$$BW_{Network} = \boldsymbol{\rho} \times \frac{\boldsymbol{BW_{Bisection}}}{\gamma}$$

σ is the *ave. fraction of traffic to reception links that can be accepted* (captures contention at reception links due to application behavior)

γ is the *ave. fraction of traffic that must cross the network bisection*

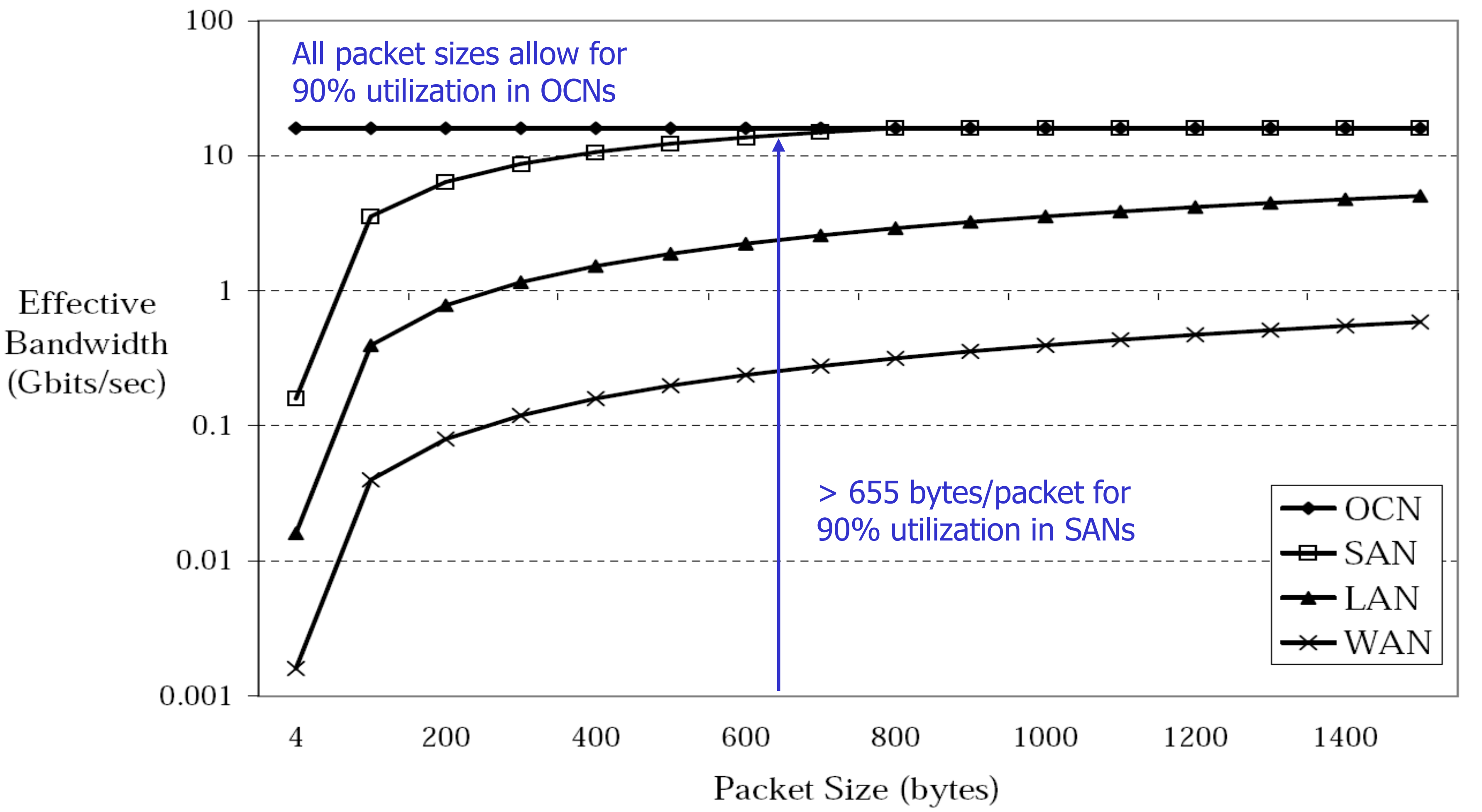
ρ is the *network efficiency*, which mostly depends on other factors: *link efficiency, routing efficiency, arb. efficiency, switching eff., etc.*

$BW_{Bisection}$ *is minimum network bisection that has traffic crossing it*

Interconnecting Two Devices

Characterizing Performance: Latency & Effective Bandwidth

- Example: plot the effective bandwidth versus packet size



All packet sizes allow for 90% utilization in OCNs

> 655 bytes/packet for 90% utilization in SANs

***Transmission time** is the limiter for OCNs; **overhead** limits SANs for packets sizes < 800 bytes*

Interconnecting Two Devices

Basic Network Characteristics of Commercial Machines

Company	System [Network] Name	Intro year	Max. compute nodes [x # CPUs]	System footprint for max. configuration	Packet [header] max. size	Injection [Recept'n] node BW in Mbytes/s	Minimum send/rec overhead	Maximum copper link length; flow control; error
Intel	ASCI Red Paragon	2001	4,510 [x 2]	2,500 sq. feet	1984 B [4 B]	400 [400]	few µsec	handshaking ; CRC+parity
IBM	ASCI White SP Power3 [Colony]	2001	512 [x 16]	10,000 sq. feet	1024 B [6 B]	500 [500]	~3 µsec	25 m; credit-based; CRC
Intel	Thunter Itanium2 Tiger4 [QsNet^{II}]	2004	1,024 [x 4]	120 m^2	2048 B [14 B]	928 [928]	0.240 µsec	13 m;credit-based; CRC for link, dest
Cray	XT3 [SeaStar]	2004	30,508 [x 1]	263.8 m^2	80 B [16 B]	3,200 [3,200]	few µsec	7 m; credit-based; CRC
Cray	X1E	2004	1,024 [x 1]	27 m^2	32 B [16 B]	1,600 [1,600]	~0 (direct LD/ST acc.)	5 m; credit-based; CRC
IBM	ASC Purple pSeries 575 [Federation]	2005	>1,280 [x 8]	6,720 sq. feet	2048 B [7 B]	2,000 [2,000]	~ 1 µsec *	25 m; credit-based; CRC
IBM	Blue Gene/L eServer Sol. [Torus Net]	2005	65,536 [x 2]	2,500 sq. feet (.9x.9x1.9 m^3/1K node rack)	256 B [8 B]	612,5 [1,050]	~ 3 µsec (2,300 cycles)	8.6 m; credit-based; CRC (header/pkt)

*with up to 4 packets processed in parallel

Outline

- E.1 Introduction *(Lecture 1)*
- E.2 Interconnecting Two Devices *(Lecture 1)*
- **E.3 Interconnecting Many Devices** *(Lecture 2)*
  - **Additional Network Structure and Functions**
  - **Shared-Media Networks**
  - **Switched-Media Networks**
  - **Comparison of Shared- versus Switched-Media Networks**
  - **Characterizing Performance: Latency & Effective Bandwidth**
- E.4 Network Topology *(Lecture 2)*
- E.5 Network Routing, Arbitration, and Switching *(Lecture 3)*
- E.6 Switch Microarchitecture *(Lecture 4)*
- E.7 Practical Issues for Commercial Interconnection Networks *(Lec 4)*
- E.8 Examples of Interconnection Networks *(Lecture 5)*
- E.9 Internetworking (skipped)
- E.10 Crosscutting Issues for Interconnection Networks (skipped)
- E.11 Fallacies and Pitfalls *(Lecture 5)*
- E.12 Concluding Remarks and References *(Lecture 5)*

Interconnecting Many Devices

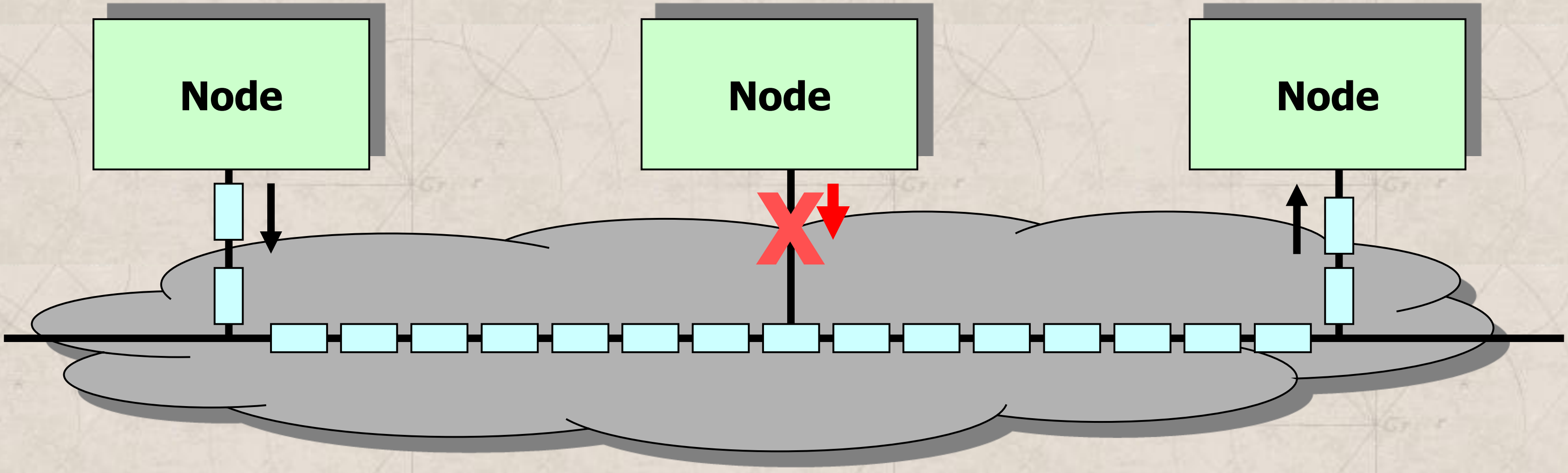
Additional Network Structure and Functions

- Additional functions (routing, arbitration, switching)
  - *Routing*
    - *Which of the possible paths are allowable (valid) for packets?*
    - Provides the set of operations needed to compute a valid path
    - Executed at source, intermediate, or even at destination nodes
  - *Arbitration*
    - *When are paths available for packets?* (along with flow control)
    - Resolves packets requesting the same resources at the same time
    - For every arbitration, there is a winner and possibly many losers
      - Losers are buffered (lossless) or dropped on overflow (lossy)
  - *Switching*
    - *How are paths allocated to packets?*
    - The winning packet (from arbitration) proceeds towards destination
    - Paths can be established one fragment at a time or in their entirety

Interconnecting Many Devices

Shared-media Networks

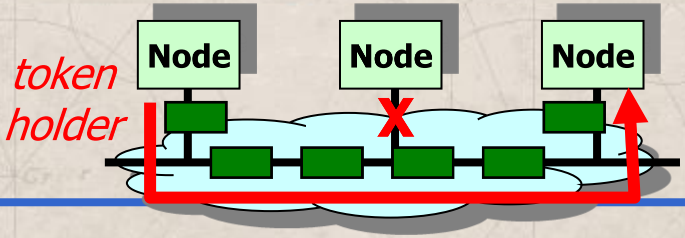
- The network media is shared by all the devices
- Operation: half-duplex or full-duplex



Interconnecting Many Devices

Shared-media Networks

- *Arbitration*
  - *Centralized* arbiter for smaller distances between devices
    - Dedicated control lines
  - *Distributed* forms of arbiters
    - CSMA/CD
      - The device first checks the network (carrier sensing)
      - Then checks if the data sent was garbled (collision detection)
      - If collision, device must send data again (retransmission): wait an increasing exponential random amount of time beforehand
      - Fairness is not guaranteed
    - Token ring—provides fairness
      - Owning the token provides permission to use network media



Interconnecting Many Devices

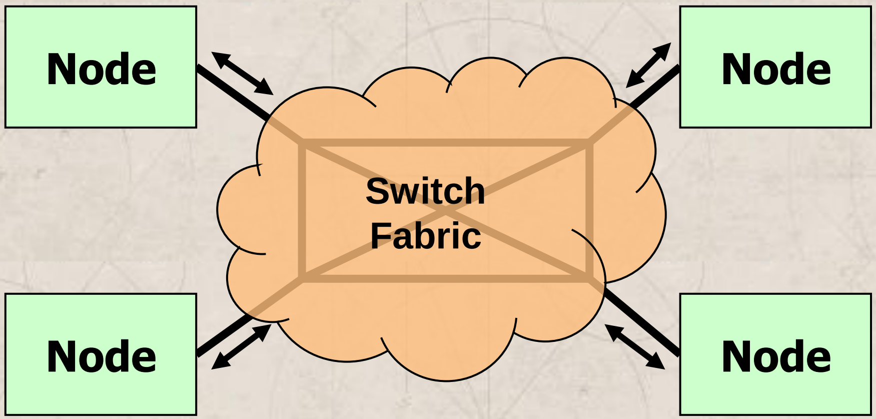
Shared-media Networks

- *Switching*
  - Switching is straightforward
  - The granted device connects to the shared media
- *Routing*
  - Routing is straightforward
  - Performed at all the potential destinations
    - Each end node device checks whether it is the target of the packet
  - Broadcast and multicast is easy to implement
    - Every end node devices sees the data sent on shared link anyway
- Established order: arbitration, switching, and *then* routing

Interconnecting Many Devices

Switched-media Networks

- *Disjoint portions of the media are shared via switching*
- Switch fabric components
  - Passive *point-to-point links*
  - Active *switches*
    - › Dynamically establish communication between sets of source-destination pairs
- Aggregate bandwidth can be many times higher than that of shared-media networks



Interconnecting Many Devices

Switched-media Networks

- *Routing*
  - – Every time a packet enters the network, it is routed
- *Arbitration*
  - – Centralized or distributed
  - – Resolves conflicts among concurrent requests
- *Switching*
  - – Once conflicts are resolved, the network "*switches in*" the required connections
- Established order: routing, arbitration, and *then* switching

Interconnecting Many Devices

Comparison of Shared- versus Switched-media Networks

- Shared-media networks
  - – Low cost
  - – Aggregate network bandwidth does not scale with # of devices
  - – Global arbitration scheme required (a possible bottleneck)
  - – Time of flight increases with the number of end nodes
- Switched-media networks
  - – Aggregate network bandwidth scales with number of devices
  - – Concurrent communication
    - › Potentially much higher network effective bandwidth
  - – *Beware:* inefficient designs are quite possible
    - › Superlinear network cost but sublinear network effective bandwidth

Interconnecting Many Devices

Characterizing Performance: Latency & Effective Bandwidth

$$\text{Latency} = \text{Sending overhead} + (T_{\text{TotalProp}} + T_R + T_A + T_S) + \frac{\text{Packet size}}{\text{Bandwidth}} + \text{Receiving overhead}$$

T_R = routing delay
T_A = arbitration delay
T_S = switching delay
$T_{\text{TotalProp}}$ = propagation delay

lower bound (contention delay not included)

upper bound (contention effects not *fully* included)

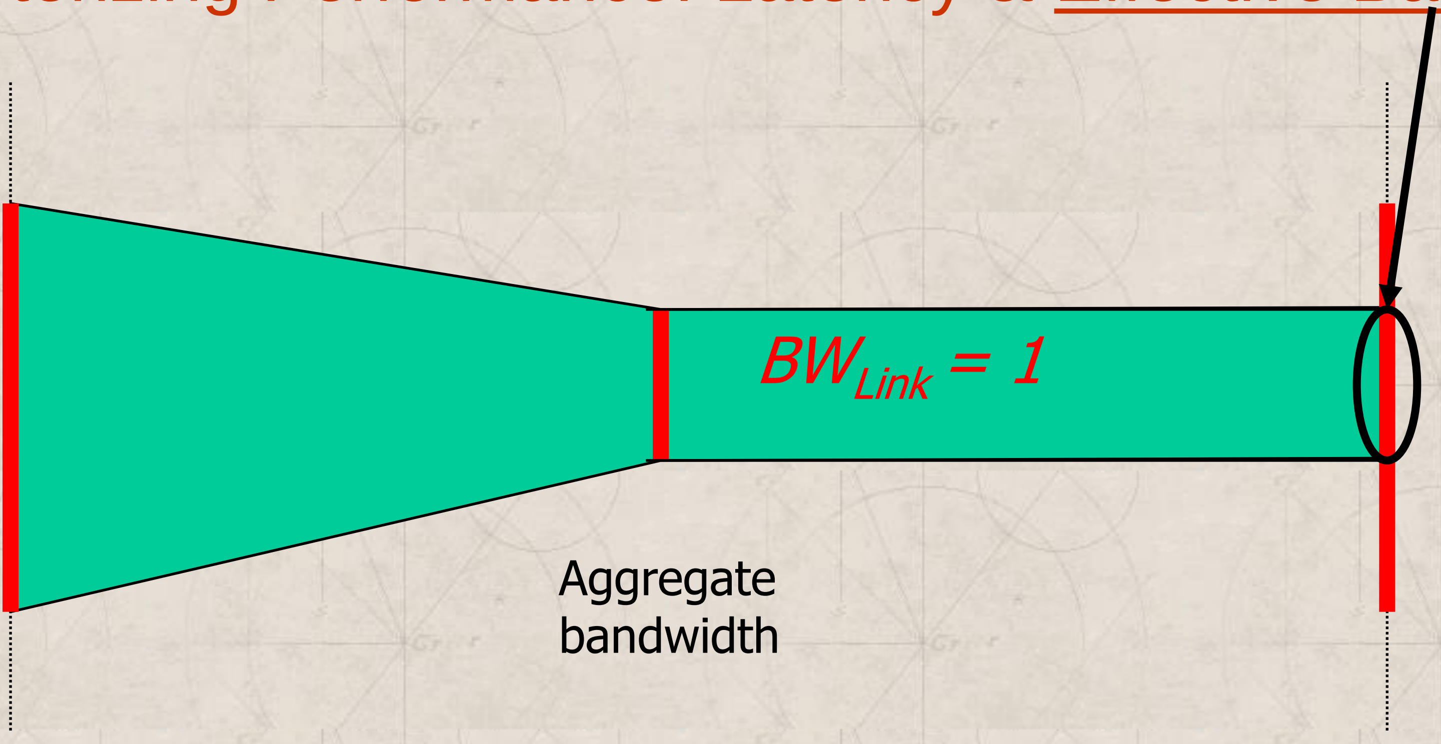
$$\text{Effective bandwidth} = \min(BW_{\text{NetworkInjection}}, BW_{\text{Network}}, \sigma \times BW_{\text{NetworkReception}})$$

$$= \min(N \times BW_{\text{LinkInjection}}, BW_{\text{Network}}, \sigma \times N \times BW_{\text{LinkReception}})$$

σ = average reception factor (contention at reception links due to application behavior)

Interconnecting Many Devices

Characterizing Performance: Latency & Effective Bandwidth



Injection bandwidth (*N times*)

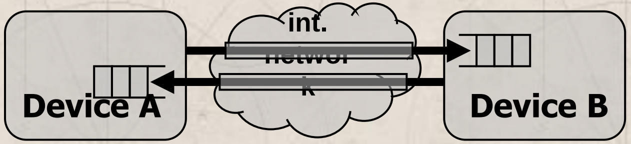
Network injection

Aggregate bandwidth

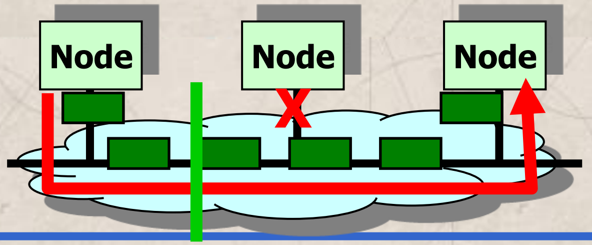
Reception bandwidth (*N times*)

Network reception

$BW_{Network} = (BW_{NetworkInjection}) / N$



Dedicated-link network

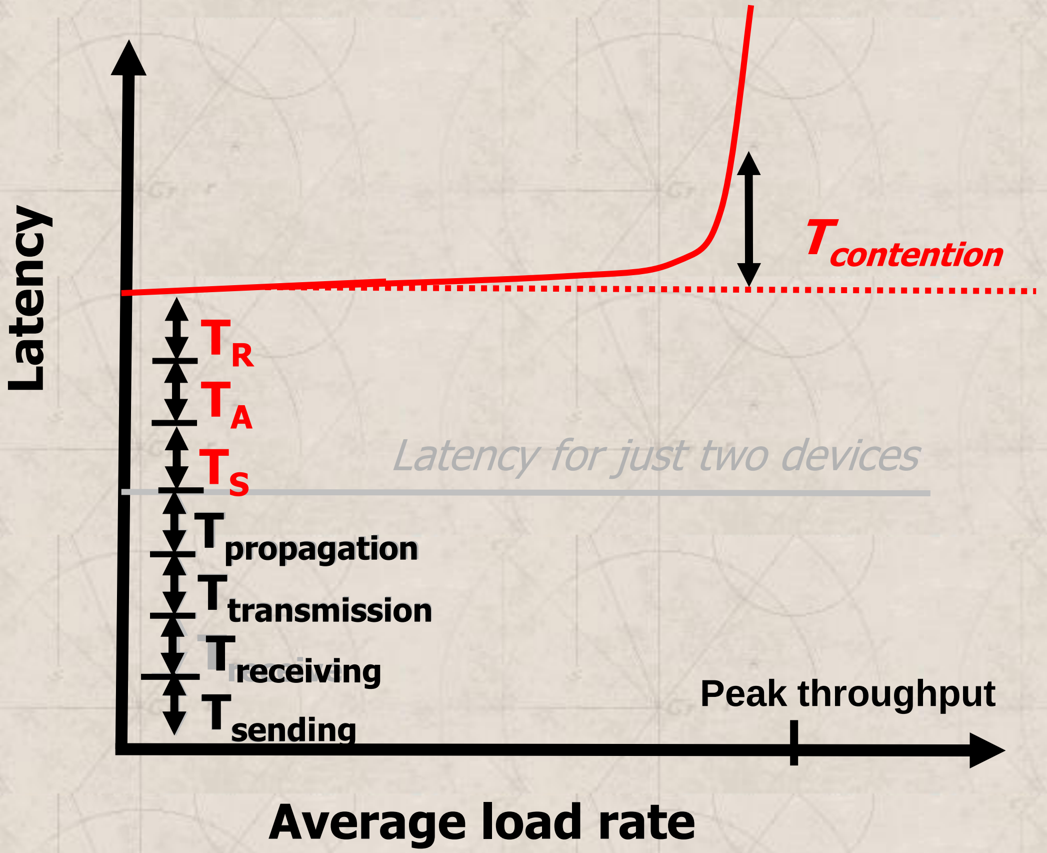


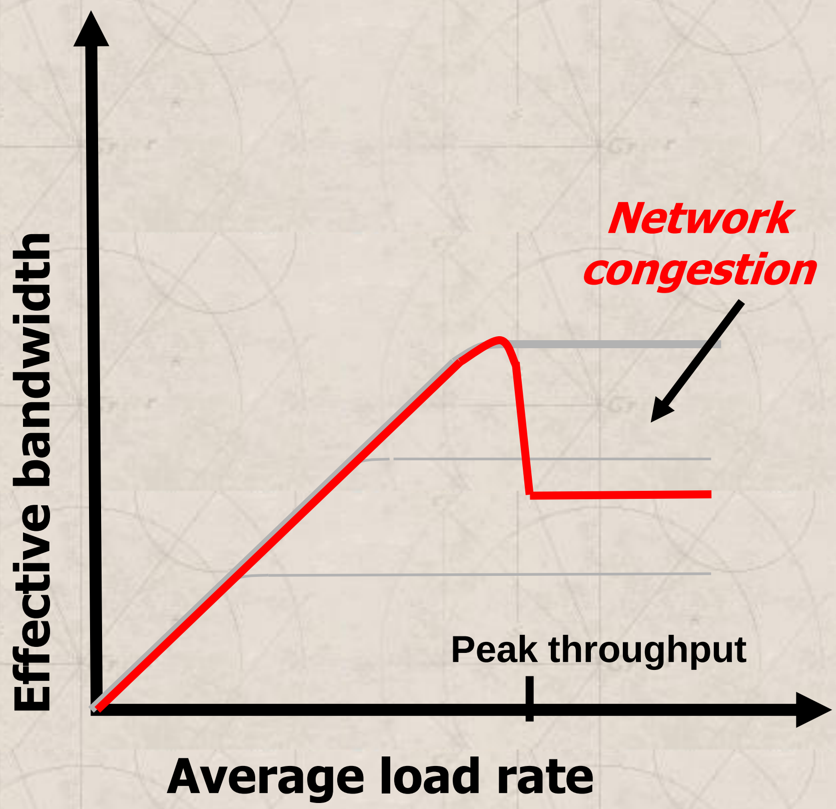
Shared-media network

Interconnecting Many Devices

Characterizing Performance: Latency & Effective Bandwidth

- *Characteristic performance plots*: latency vs. average load rate; throughput (effective bandwidth) vs. average load rate





Outline

- E.1 Introduction *(Lecture 1)*
- E.2 Interconnecting Two Devices *(Lecture 1)*
- E.3 Interconnecting Many Devices *(Lecture 2)*
- **E.4 Network Topology** *(Lecture 2)*
  - **Preliminaries and Evolution**
  - **Centralized Switched (Indirect) Networks**
  - **Distributed Switched (Direct) Networks**
  - **Comparison of Indirect and Direct Networks**
  - **Characterizing Performance: Latency & Effective Bandwidth**
- E.5 Network Routing, Arbitration, and Switching *(Lecture 3)*
- E.6 Switch Microarchitecture *(Lecture 4)*
- E.7 Practical Issues for Commercial Interconnection Networks *(Lecture 4)*
- E.8 Examples of Interconnection Networks *(Lecture 5)*
- E.9 Internetworking (skipped)
- E.10 Crosscutting Issues for Interconnection Networks (skipped)
- E.11 Fallacies and Pitfalls *(Lecture 5)*
- E.12 Concluding Remarks and References *(Lecture 5)*

Network Topology

Preliminaries and Evolution

- One switch suffices to connect a small number of devices
  - Number of switch ports limited by VLSI technology, power consumption, packaging, and other such *cost* constraints
- A *fabric* of interconnected switches (i.e., *switch fabric* or *network fabric*) is needed when the number of devices is much larger
  - The *topology* must make a path(s) available for every pair of devices—property of *connectedness* or *full access* (*What paths?)*
- Topology defines the connection structure across all components
  - *Bisection bandwidth*: the *minimum* bandwidth of all links crossing a network split into two roughly equal halves
  - *Full bisection bandwidth*:
    - Network $BW_{Bisection}$ = Injection (or Reception) $BW_{Bisection}$= *N/2*
  - Bisection bandwidth mainly affects *performance*
- Topology is constrained primarily by local chip/board *pin-outs*; secondarily, (if at all) by global bisection bandwidth

Network Topology

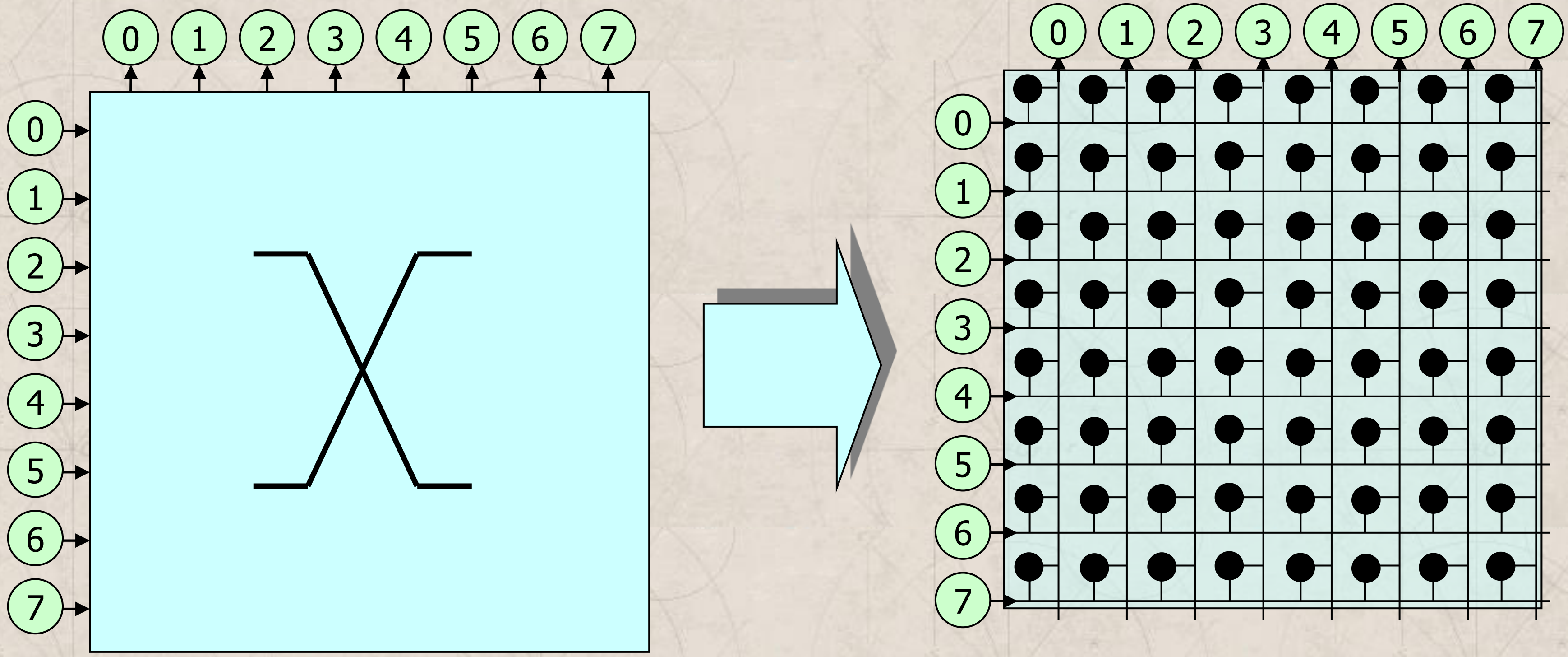
Preliminaries and Evolution

- Several tens of topologies proposed, but less than a dozen used
- 1970s and 1980s
  - – Topologies were proposed to reduce *hop count*
- 1990s
  - – Pipelined transmission and switching techniques
  - – Packet latency became decoupled from hop count
- 2000s
  - – Topology still important (especially OCNs, SANs) when *N* is high
  - – Topology impacts performance and has a major impact on cost

Network Topology

Centralized Switched (Indirect) Networks

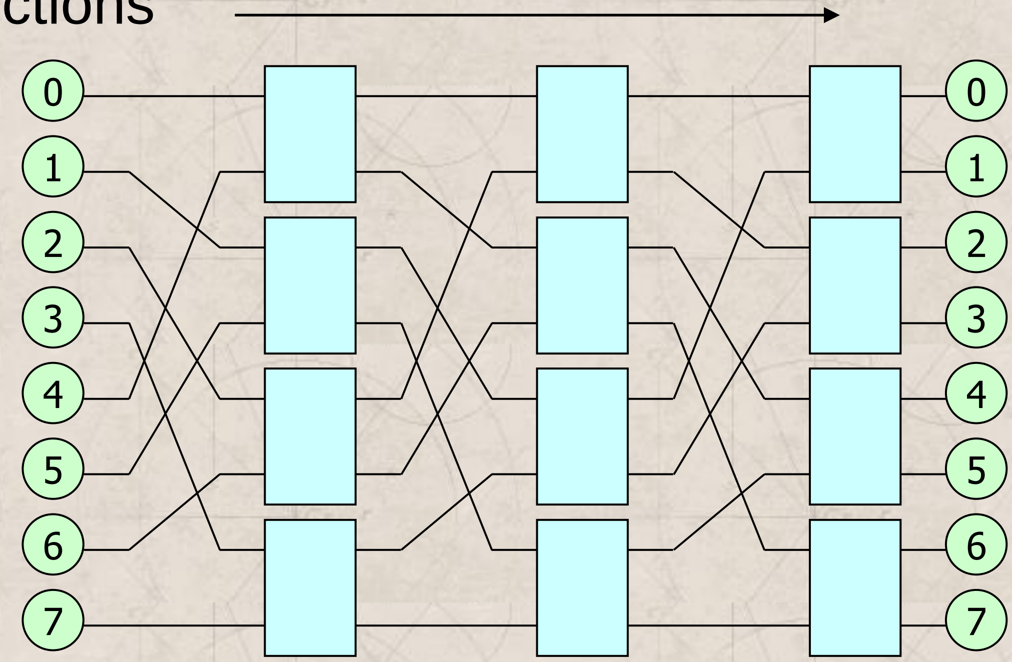
- ***Crossbar network***
  - – Crosspoint switch complexity increases quadratically with the number of crossbar input/output ports, *N*, i.e., grows as O(N^2)
  - – Has the property of being *non-blocking*



Network Topology

Centralized Switched (Indirect) Networks

- ***Multistage interconnection networks (MINs)***
  - Crossbar split into several stages consisting of smaller crossbars
  - Complexity grows as O(*N* × log *N*), where *N* is # of end nodes
  - Inter-stage connections represented by a set of permutation functions

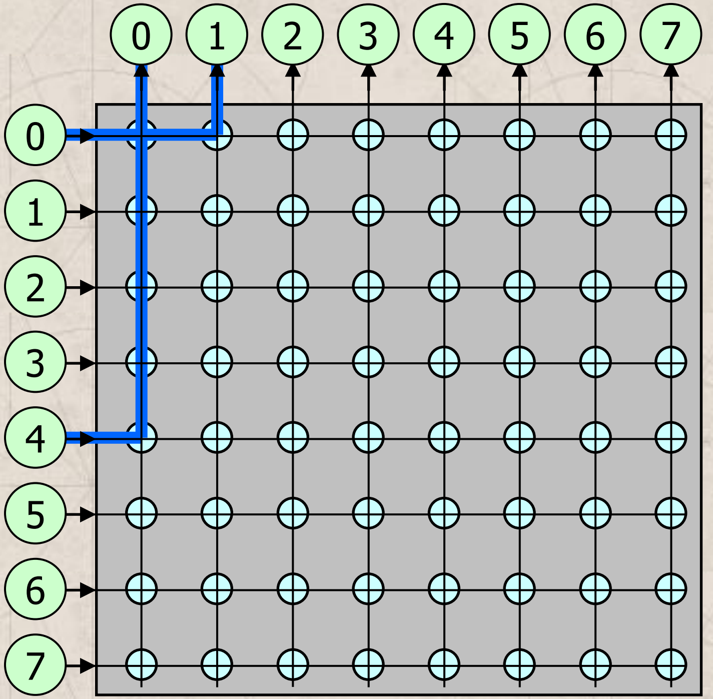


Omega topology, perfect-shuffle exchange

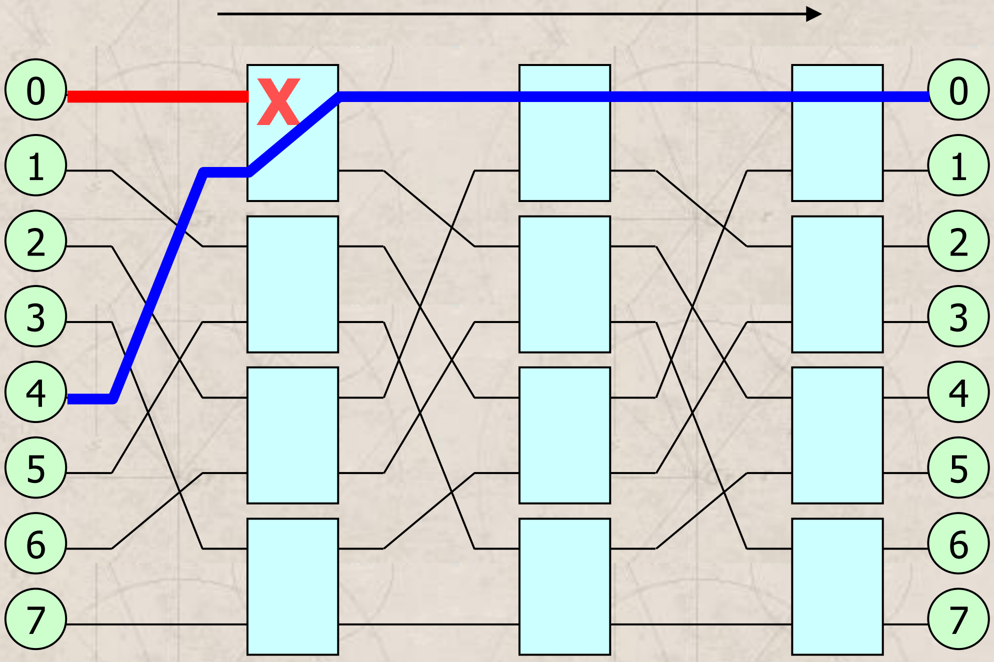
Network Topology

Centralized Switched (Indirect) Networks

- Reduction in MIN switch *cost* comes at the price of *performance*
  - Network has the property of being *blocking*
  - *Contention* is more likely to occur on network links
    - › Paths from different sources to different destinations share one or more links



non-blocking topology



blocking topology

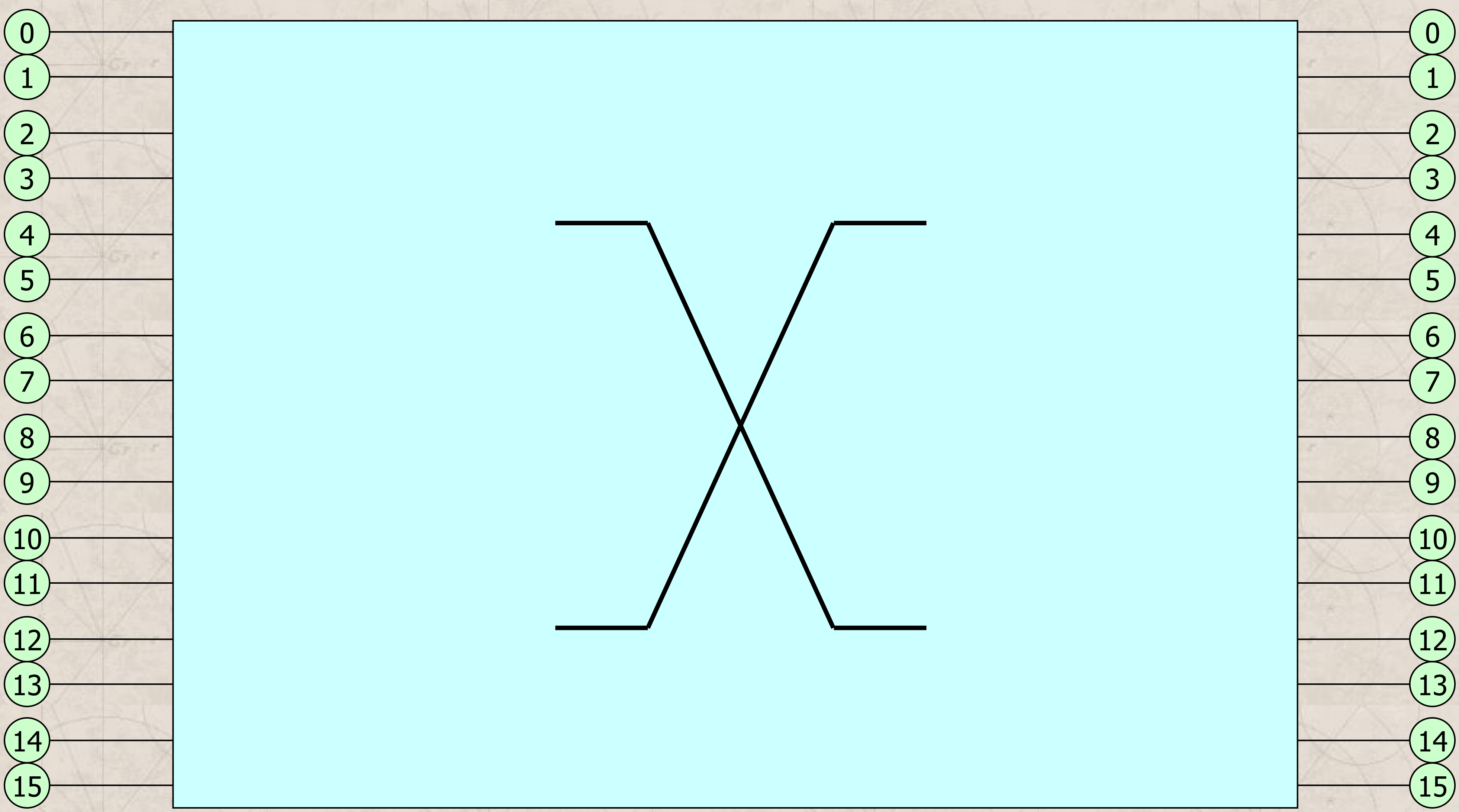
Network Topology

Centralized Switched (Indirect) Networks

- How to reduce blocking in MINs? *Provide alternative paths!*
  - Use larger switches (can equate to using more switches)
    - ***Clos network***: minimally three stages (non-blocking)
      - A larger switch in the middle of two other switch stages provides enough alternative paths to avoid all conflicts
  - Use more switches
    - Add $\log_k N$ - 1 stages, mirroring the original topology
      - *Rearrangeably non-blocking*
      - Allows for non-conflicting paths
      - Doubles network *hop count (distance), d*
      - Centralized control can rearrange established paths
    - ***Benes topology***: $2(\log_2 N) - 1$ stages (rearrangeably non-blocking)
      - Recursively applies the three-stage Clos network concept to the middle-stage set of switches to reduce all switches to 2 x 2

Network Topology

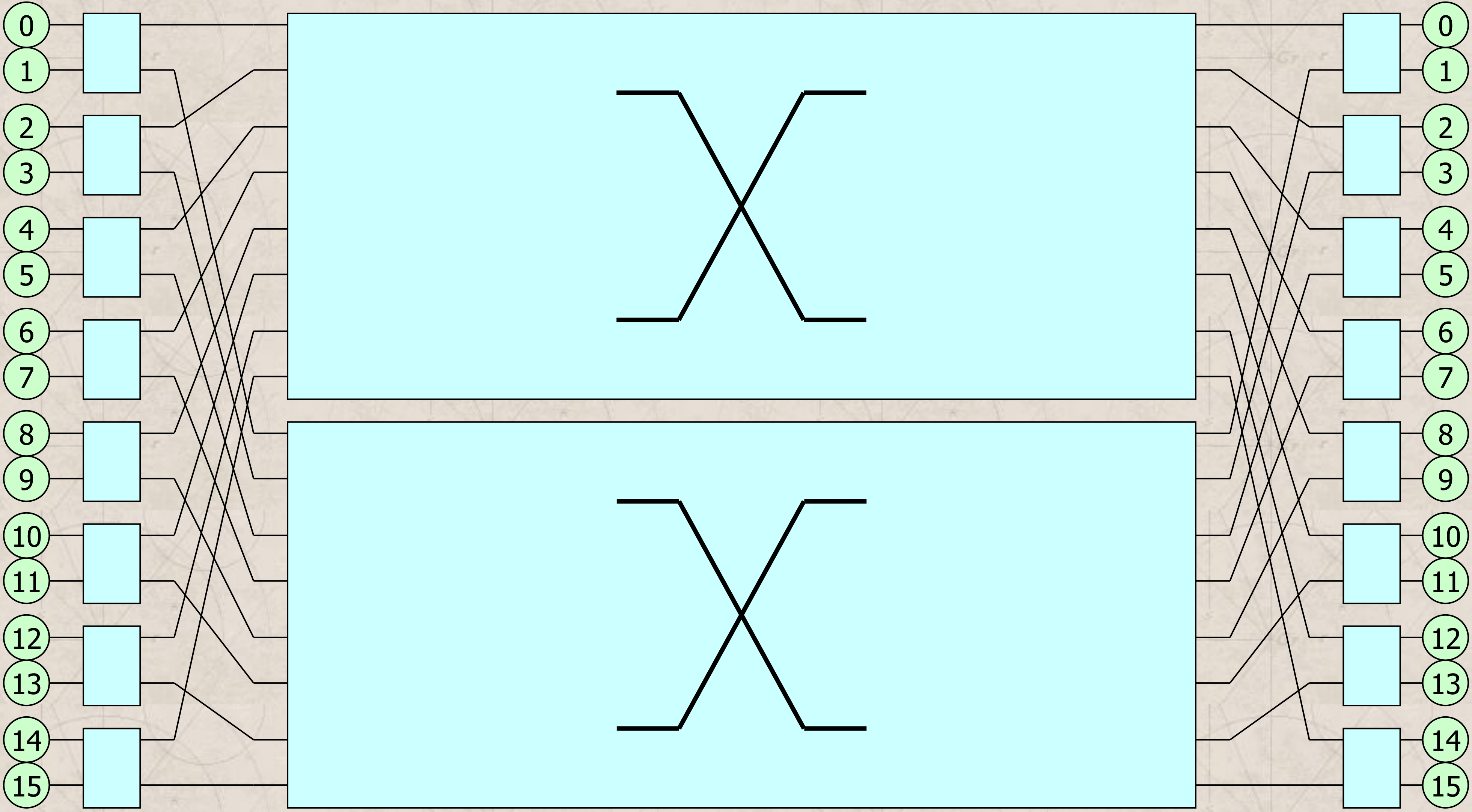
Centralized Switched (Indirect) Networks



16 port ***Crossbar*** network

Network Topology

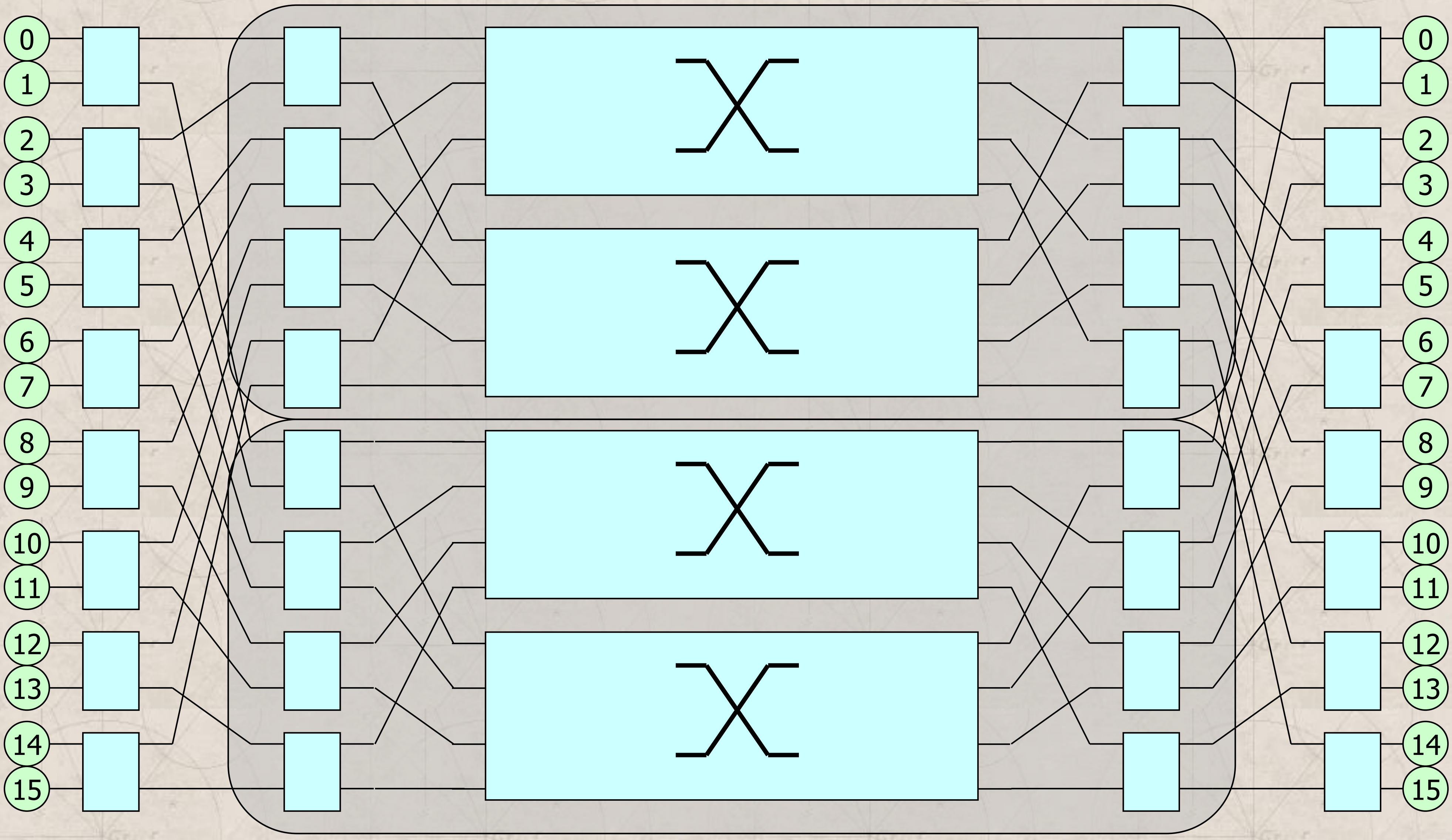
Centralized Switched (Indirect) Networks



16 port, ***3-stage Clos*** network

Network Topology

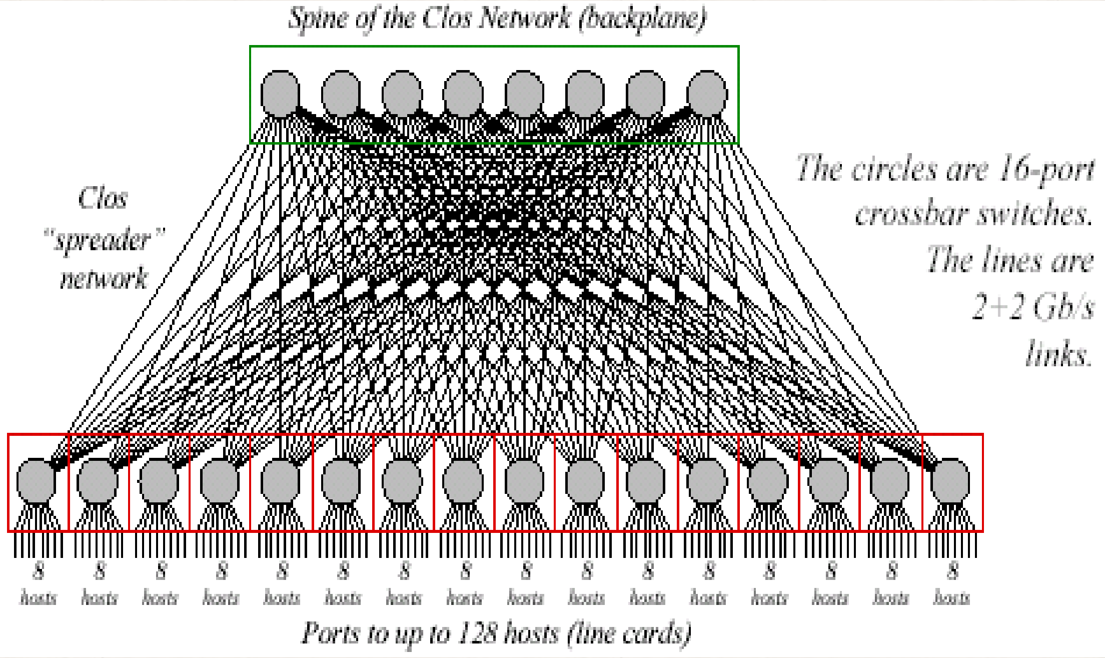
Centralized Switched (Indirect) Networks

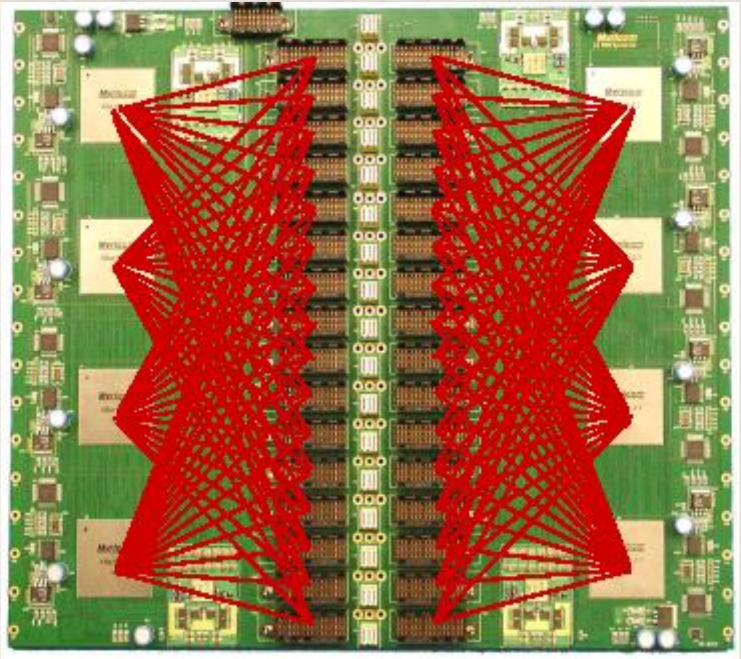


16 port, ***5-stage Clos*** network

Network Topology

Myrinet-2000 Clos Network for 128 Hosts


Spine of the Clos Network (backplane)
Clos "spreader" network
The circles are 16-port crossbar switches. The lines are 2+2 Gb/s links.
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
8 hosts
Ports to up to 128 hosts (line cards)



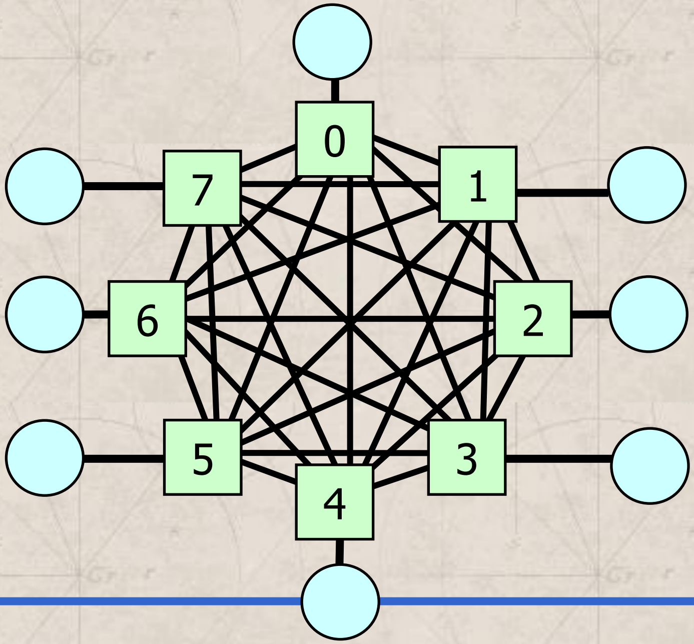
- **Backplane of the M3-E128 Switch**
- **M3-SW16-8F fiber line card (8 ports)**

http://myri.com

Network Topology

Distributed Switched (Direct) Networks

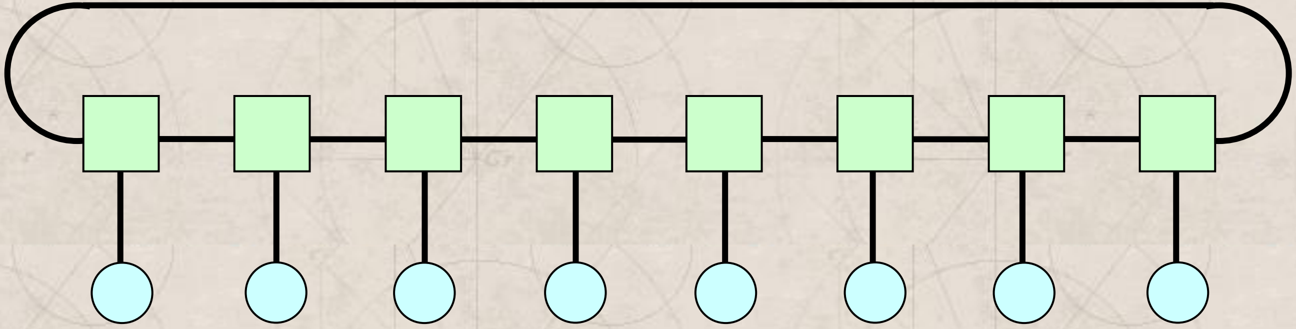
- Tight integration of end node devices with network resources
  - Network switches distributed among end nodes
  - A “node” now consists of a network switch with *one or more end node devices directly connected* to it
  - Nodes are directly connected to other nodes
- ***Fully-connected network***: all nodes are directly connected to all other nodes using bidirectional dedicated links



Network Topology

Distributed Switched (Direct) Networks

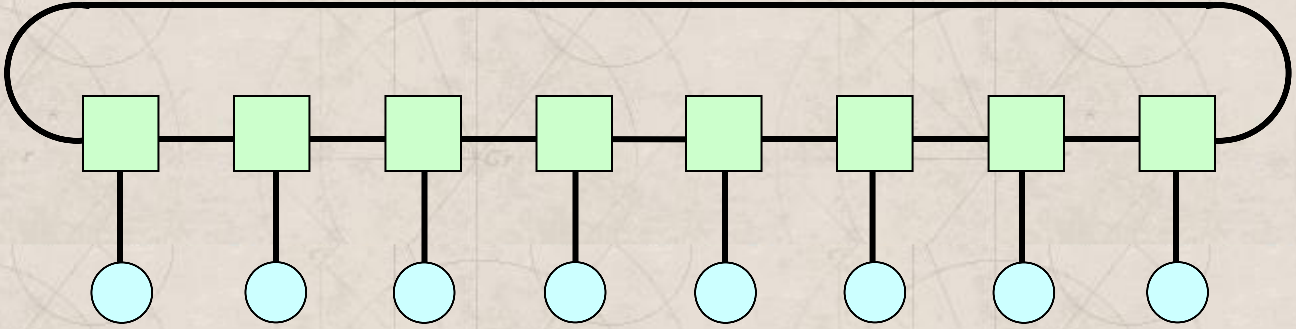
- ***Bidirectional Ring networks***
  - – *N* switches (3 × 3) and *N* bidirectional network links
  - – Simultaneous packet transport over disjoint paths
  - – Packets must hop across intermediate nodes
  - – Shortest direction usually selected (*N*/4 hops, on average)

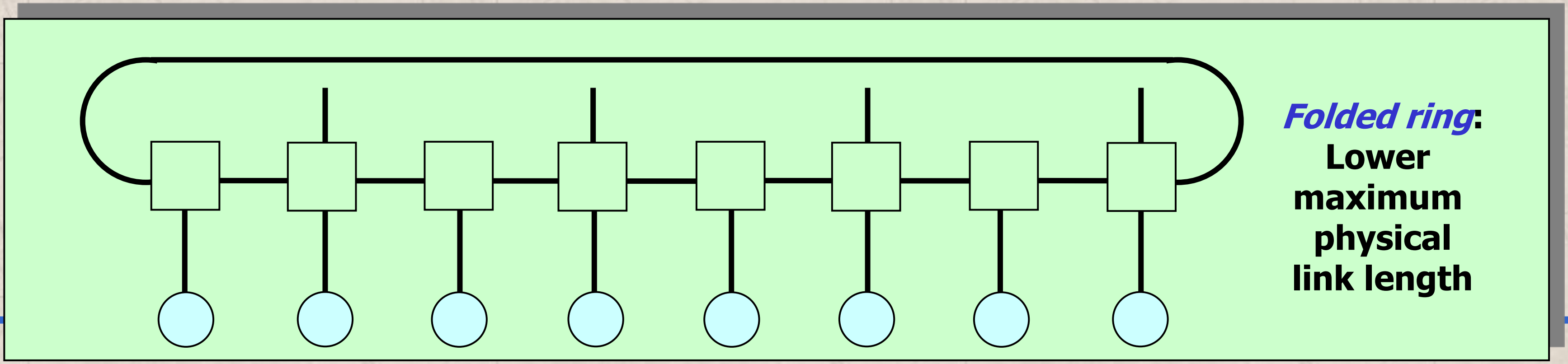


Network Topology

Distributed Switched (Direct) Networks

- ***Bidirectional Ring networks (folded)***
  - *N* switches (3 × 3) and *N* bidirectional network links
  - Simultaneous packet transport over disjoint paths
  - Packets must hop across intermediate nodes
  - Shortest direction usually selected (*N*/4 hops, on average)





Folded ring**: Lower maximum physical link length**

Interconnection Networks: © Timothy Mark Pinkston and José Duato
...with major presentation contribution from José Flich

Network Topology

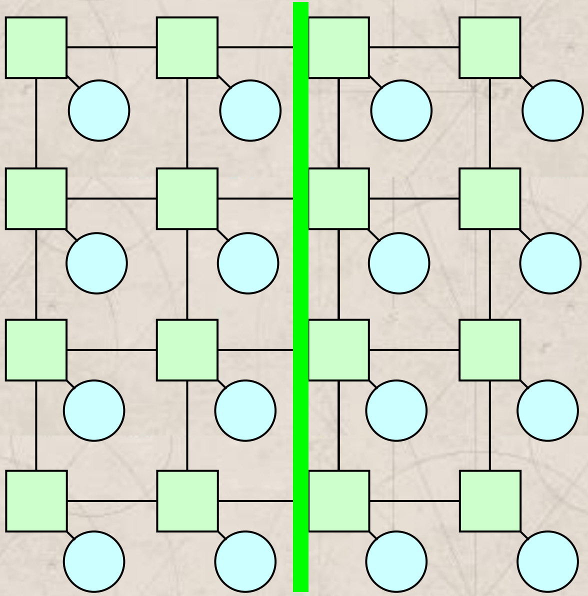
Distributed Switched (Direct) Networks:

- *Fully connected and ring topologies delimit the two extremes*
- ***The ideal topology:***
  - – Cost approaching a ring
  - – Performance approaching a fully connected (crossbar) topology
- More practical topologies:
  - – ***k-ary n-cubes*** (*meshes*, *tori*, *hypercubes*)
    - › ***k*** nodes connected in each dimension, with ***n*** total dimensions
    - › *Symmetry* and *regularity*
      - » network implementation is simplified
      - » routing is simplified

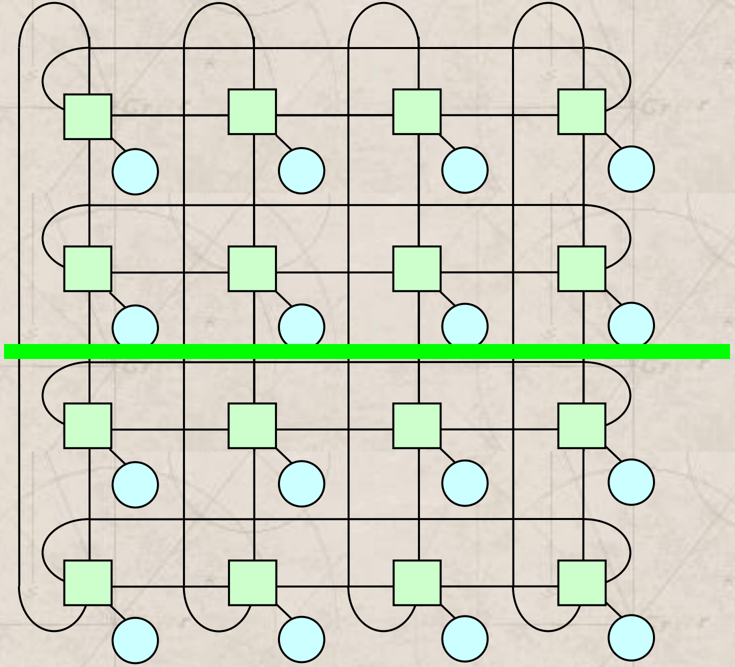
Network Topology

Distributed Switched (Direct) Networks

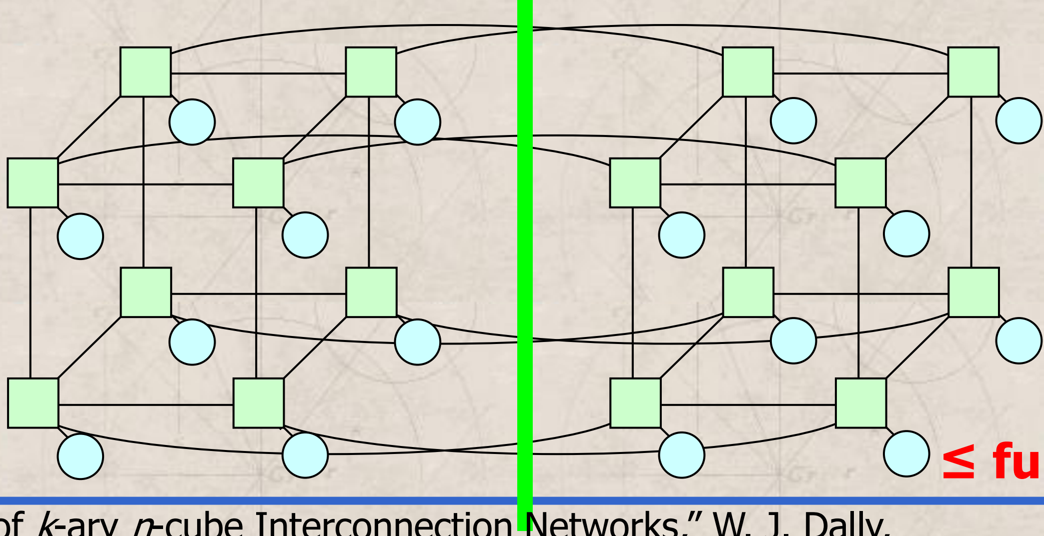
2D ***mesh*** or grid of 16 nodes



2D ***torus*** of 16 nodes



hypercube of 16 nodes
(16 = 2^4, so n = 4)



Network Bisection

≤ full bisection bandwidth!

"Performance Analysis of *k*-ary *n*-cube Interconnection Networks," W. J. Dally, *IEEE Trans. on Computers*, Vol. 39, No. 6, pp. 775–785, June, 1990.

Network Topology

Topological Characteristics of Commercial Machines

Company	System [Network] Name	Max. number of nodes [x # CPUs]	Basic network topology	Injection [Recept'n] node BW in MBytes/s	# of data bits per link per direction	Raw network link BW per direction in Mbytes/sec	Raw network bisection BW (bidir) in Gbytes/s
Intel	ASCI Red Paragon	4,510 [x 2]	2-D mesh 64 x 64	400 [400]	16 bits	400	51.2
IBM	ASCI White SP Power3 [Colony]	512 [x 16]	BMIN w/8-port bidirect. switches (fat-tree or Omega)	500 [500]	8 bits (+1 bit of control)	500	256
Intel	Thunter Itanium2 Tiger4 [QsNet^{II}]	1,024 [x 4]	fat tree w/8-port bidirectional switches	928 [928]	8 bits (+2 control for 4b/5b enc)	1,333	1,365
Cray	XT3 [SeaStar]	30,508 [x 1]	3-D torus 40 x 32 x 24	3,200 [3,200]	12 bits	3,800	5,836.8
Cray	X1E	1,024 [x 1]	4-way bristled 2-D torus (~ 23 x 11) with express links	1,600 [1,600]	16 bits	1,600	51.2
IBM	ASC Purple pSeries 575 [Federation]	>1,280 [x 8]	BMIN w/8-port bidirect. switches (fat-tree or Omega)	2,000 [2,000]	8 bits (+2 bits of control)	2,000	2,560
IBM	Blue Gene/L eServer Sol. [Torus Net]	65,536 [x 2]	3-D torus 32 x 32 x 64	612,5 [1,050]	1 bit (bit serial)	175	358.4

Outline

- E.1 Introduction *(Lecture 1)*
- E.2 Interconnecting Two Devices *(Lecture 1)*
- E.3 Interconnecting Many Devices *(Lecture 2)*
- E.4 Network Topology *(Lecture 2)*
- **E.5 Network Routing, Arbitration, and Switching** *(Lecture 3)*
  - **Routing**
  - **Arbitration**
  - **Switching**
  - **Characterizing Performance: Latency & Effective Bandwidth**
- E.6 Switch Microarchitecture *(Lecture 4)*
- E.7 Practical Issues for Commercial Interconnection Networks *(Lec 4)*
- E.8 Examples of Interconnection Networks *(Lecture 5)*
- E.9 Internetworking (skipped)
- E.10 Crosscutting Issues for Interconnection Networks (skipped)
- E.11 Fallacies and Pitfalls *(Lecture 5)*
- E.12 Concluding Remarks and References *(Lecture 5)*

Routing, Arbitration, and Switching

Routing

- Performed at each switch, regardless of topology
- Defines the “allowed” path(s) for each packet (*Which paths?*)
- Needed to *direct packets through network* to intended destinations
- ***Ideally:***
  - *Supply as many routing options to packets as there are paths provided by the topology, and evenly distribute network traffic among network links using those paths, minimizing contention*
- *Problems:* situations that cause packets never to reach their dest.
  - ***Livelock***
    - Arises from an unbounded number of allowed non-minimal hops
    - *Solution:* restrict the number of non-minimal (mis)hops allowed
  - ***Deadlock***
    - Arises from a set of packets being blocked waiting only for network resources (i.e., links, buffers) held by other packets in the set
    - Probability increases with increased traffic & decreased availability

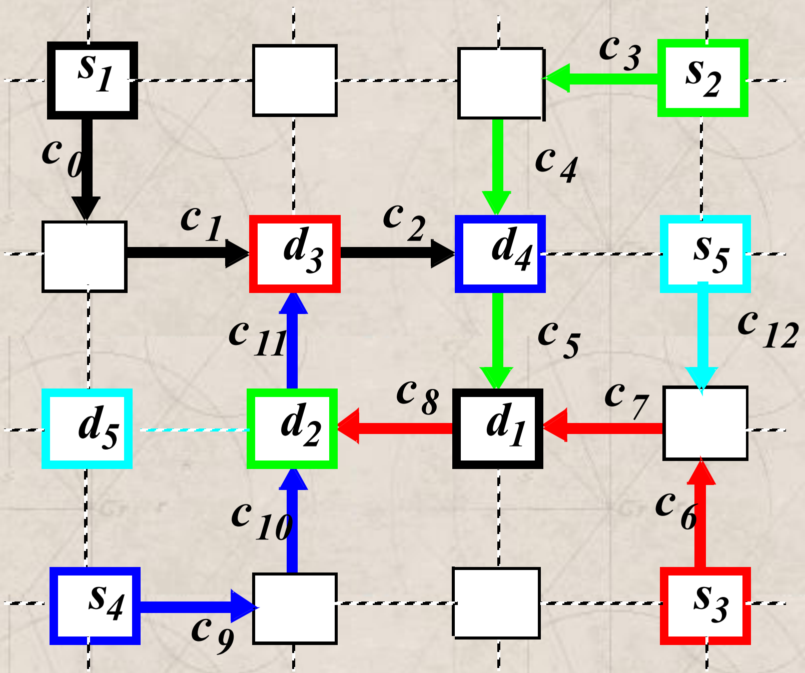
Routing, Arbitration, and Switching

Routing

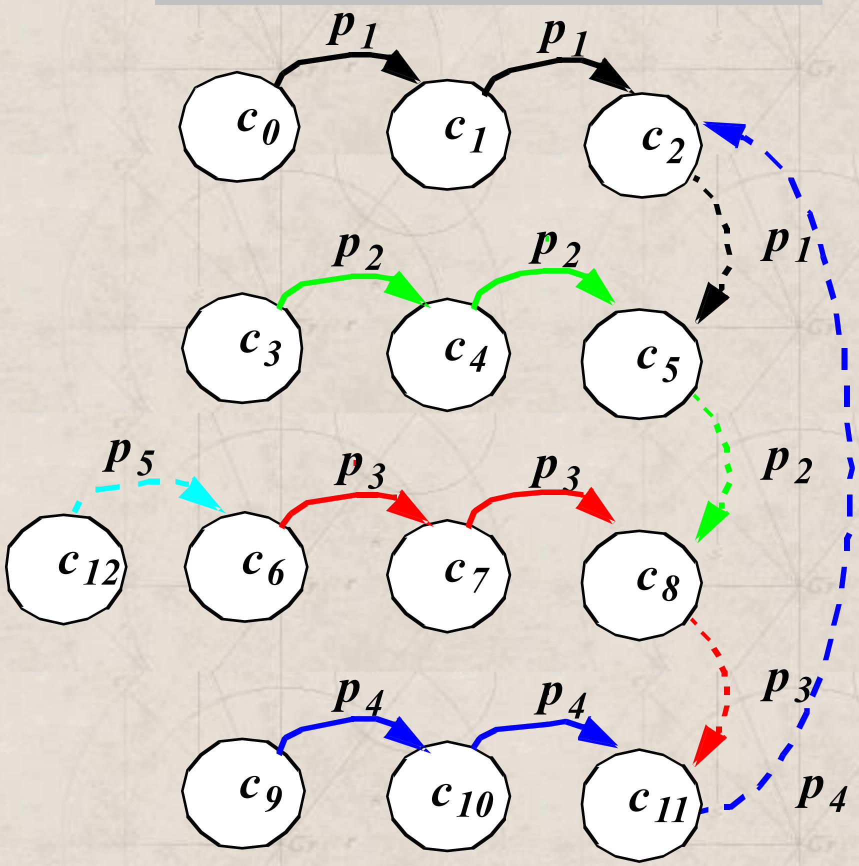
- Common forms of deadlock:
  - – ***Routing-induced deadlock***

c**$_i$* = channel ***i
s**$_i$* = source node ***i
d**$_i$* = destination node ***i
p**$_i$* = packet ***i

Routing of packets in a 2D mesh



Channel dependency graph



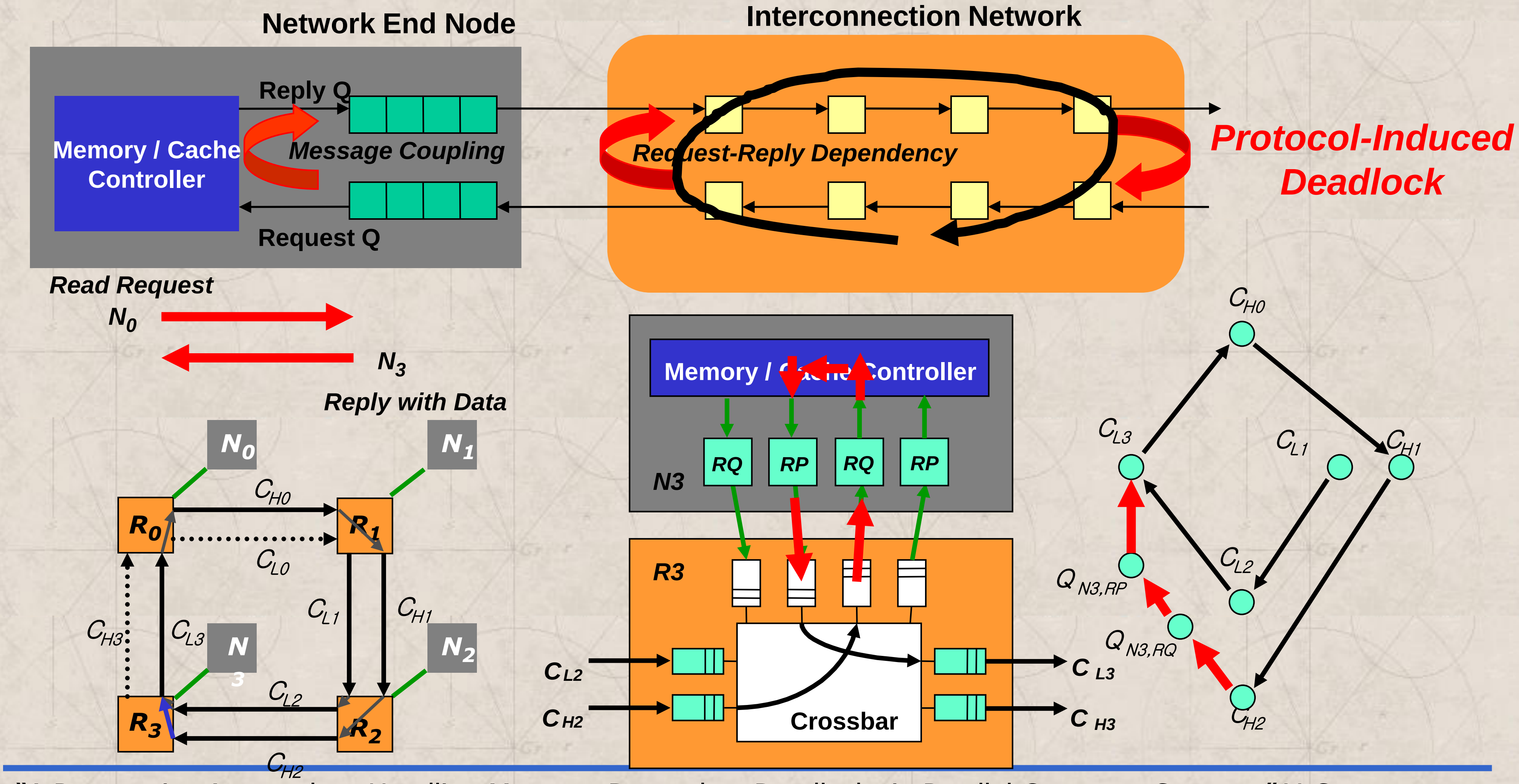
"A Formal Model of Message Blocking and Deadlock Resolution in Interconnection Networks," S. Warnakulasuriya and T. Pinkston, *IEEE Trans. on Parallel and Distributed Systems* , Vol. 11, No. 3, pp. 212–229, March, 2000.

Routing, Arbitration, and Switching

Routing

- Common forms of deadlock:
  - ***Protocol (Message)-induced deadlock***

C_{Hi} = high-ordered channel ***i***
C_{Li} = low-ordered channel ***i***
$Q_{Ni,RQ}$ = node ***i*** Request Q
$Q_{Ni,RP}$ = node ***i*** Reply Q

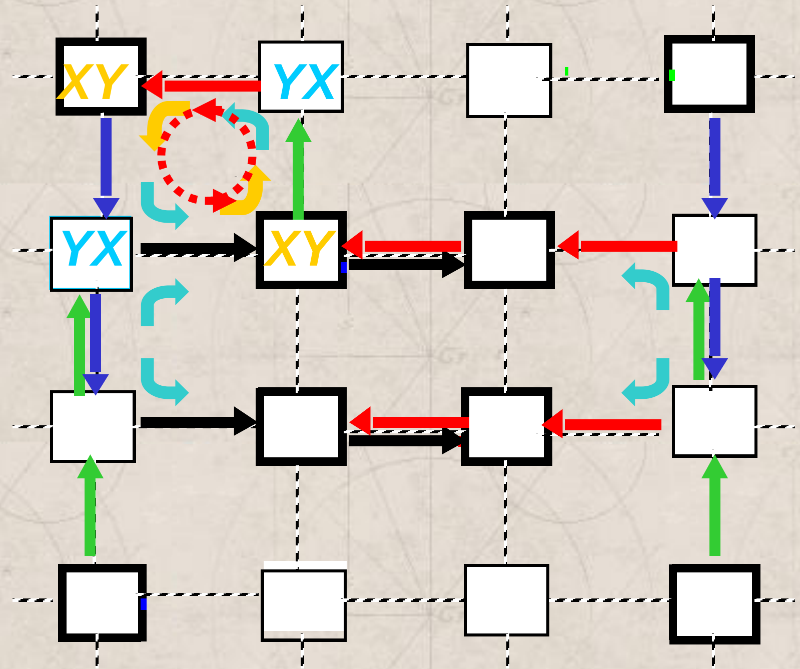


"A Progressive Approach to Handling Message-Dependent Deadlocks in Parallel Computer Systems," Y. Song and T. Pinkston, *IEEE Trans. on Parallel and Distributed Systems* , Vol. 14, No. 3, pp. 259–275, March, 2003.

Routing, Arbitration, and Switching

Routing

- Common forms of deadlock:
  - *Fault (Reconfiguration)-induced deadlock*



- The transition from one routing function (***YX*** routing) to another routing function (***XY*** routing) in order to circumvent faults can create cyclic dependencies on resources that are not present in either routing function alone!

"Part I: A Theory for Deadlock-free Dynamic Reconfiguration of Interconnection Networks," J. Duato, O. Lysne, R. Pang, and T. Pinkston, *IEEE Trans. on Parallel and Distributed Systems* , Vol. 16, No. 5, pp. 412–427, May, 2005.

Routing, Arbitration, and Switching

Routing

- Common strategies to deal with all forms of deadlock
  - *Deadlock avoidance:* restrict allowed paths only to those that keep the global state deadlock-free
    - *Duato's Protocol*: always guarantee an *escape path* from deadlock
      - » Establish ordering only on a minimal (*escape*) set of resources
      - » Grant escape resources in a partial or total order
      - » Cyclic dependencies cannot form on escape resources, although cycles may form on larger set of network resources
    - *DOR* (*dimension-order routing*) on meshes and hypercubes
      - » Establish ordering on ***all*** resources based on network dimension
    - *DOR* on rings and tori (*k*-ary *n*-cubes with wrap-around links)
      - » Ordering on all resources between *and within* each dimension
      - » Apply to multiple *virtual channels* (*VCs*) per physical channel
      - » Alternatively, keep resources along each dimension from reaching full capacity by ensuring the existence of a *bubble(s)*

Routing, Arbitration, and Switching

Routing

- Common strategies to deal with all forms of deadlock
  - *Deadlock recovery:* allow deadlock to occur, but once a potential deadlock situation is detected, break at least one of the cyclic dependencies to gracefully recover
    - A mechanism to detect potential deadlock is needed
    - *Regressive recovery* (*abort-and-retry*): remove packet(s) from a dependency cycle by killing (aborting) and later re-injecting (retry) the packet(s) into the network after some delay
    - *Progressive recovery* (preemptive): remove packet(s) from a dependency cycle by rerouting the packet(s) onto a deadlock-free lane
- *Deterministic routing*: routing function always supplies the same path for a given source-destination pair (e.g., *DOR*)
- *Adaptive routing*: routing function allows alternative paths for a given source-destination pair (e.g., *Duato's Protocol, Bubble Adaptive Routing, Disha Routing*)
  - Increases routing freedom to improve network efficiency, *ρ*

Routing, Arbitration, and Switching

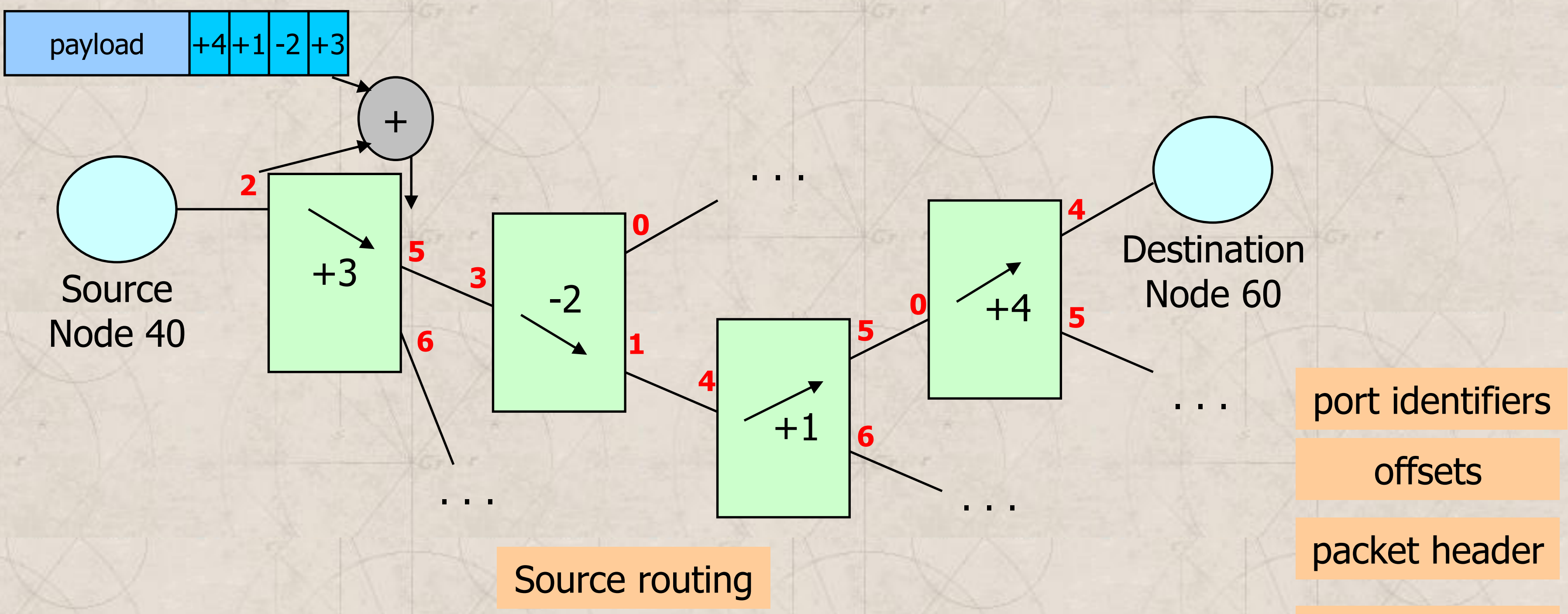
Routing

- Routing in centralized switched (indirect) networks
  - *Least common ancestor* (*LCA*) *routing*
    - Applicable to fat tree and other bidirectional MINs
    - Use resources in some partial order to avoid cycles, deadlock
    - Reach any LCA switch through any one of multiple paths
    - Traverse down the tree to destination through a deterministic path
    - *Self routing property*: switch output port at each hop is given by shifts of the destination node address (least significant bit/digit)
  - *Up*/down* routing*:
    - Universally applicable to *any* topology: map a tree graph onto it
    - Assign "up" and "down" directions to network links (or VCs)
    - Allowed paths to destination consist of zero or more "up" traversals followed by zero or more "down" traversals
    - Up-down traversals impose partial order to avoid cycles, deadlocks

Routing, Arbitration, and Switching

Routing

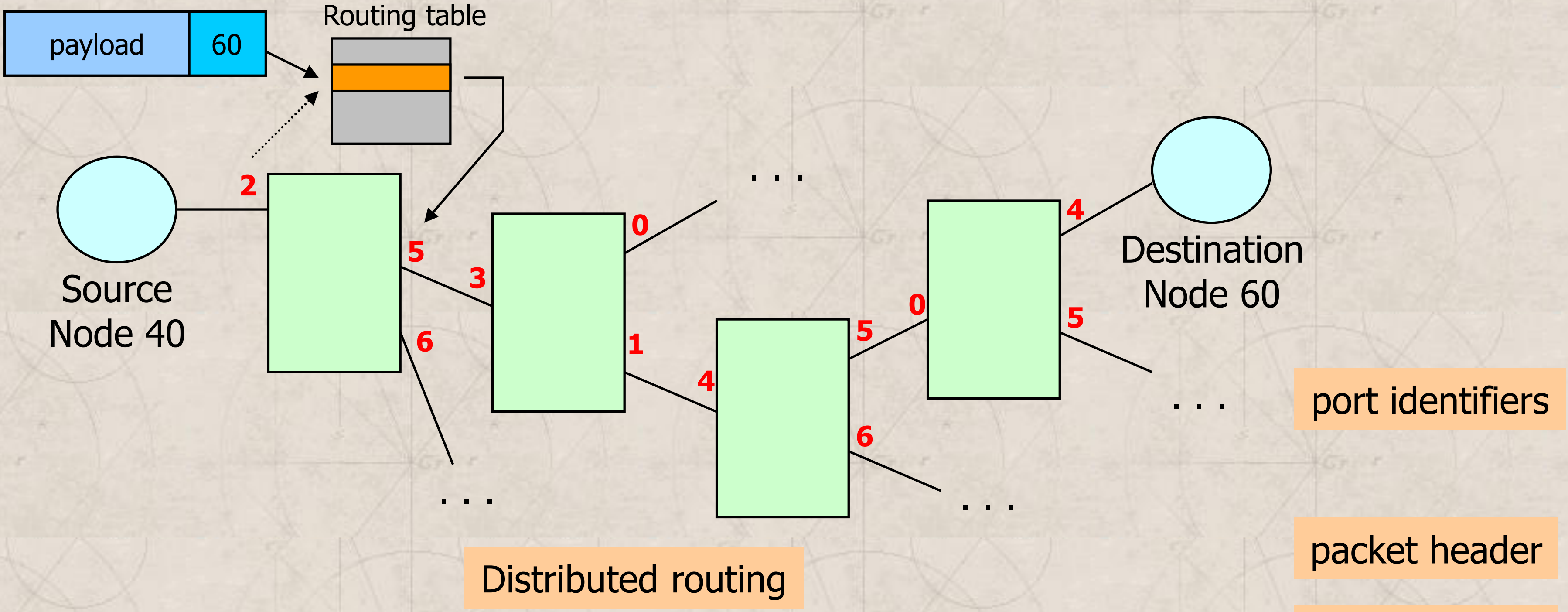
- Implementing the routing: source routing vs distributed routing
  - – *Source routing* (offset-based or could use absolute output port #)
    - › Routing control unit in switches is simplified; computed at source
    - › Headers containing the route tend to be larger → increase overhead



Routing, Arbitration, and Switching

Routing

- Implementing the routing: source routing vs distributed routing
  - *Distributed routing*
    - › Next route computed by *finite-state machine* or by *table look-up*
    - › *Look-ahead routing* is possible: the route one hop away is supplied



Routing, Arbitration, and Switching

Arbitration

- Performed at each switch, regardless of topology
- Determines use of paths supplied to packets (*When allocated?*)
- Needed to *resolve conflicts for shared resources* by requestors
- ***Ideally:***
  - *Maximize the matching between available network resources and packets requesting them*
  - At the switch level, arbiters maximize the matching of free switch output ports and packets located at switch input ports
- *Problems:*
  - ***Starvation***
    - Arises when packets can never gain access to requested resources
    - *Solution:* Grant resources to packets with *fairness*, even if prioritized
- Many straightforward distributed arbitration techniques for switches
  - Two-phased arbiters, three-phased arbiters, and iterative arbiters

Routing, Arbitration, and Switching

Switching

- Performed at each switch, regardless of topology
- Establishes the connection of paths for packets (How allocated?)
- Needed to *increase utilization of shared resources* in the network
- ***Ideally:***
  - *Establish or "switch in" connections between network resources (1) only for as long as paths are needed and (2) exactly at the point in time they are ready and needed to be used by packets*
  - Allows for efficient use of network bandwidth to competing flows
- *Switching techniques:*
  - Circuit switching
    - pipelined circuit switching
  - Packet switching
    - Store-and-forward switching
    - Cut-through switching: virtual cut-through and wormhole

Routing, Arbitration, and Switching

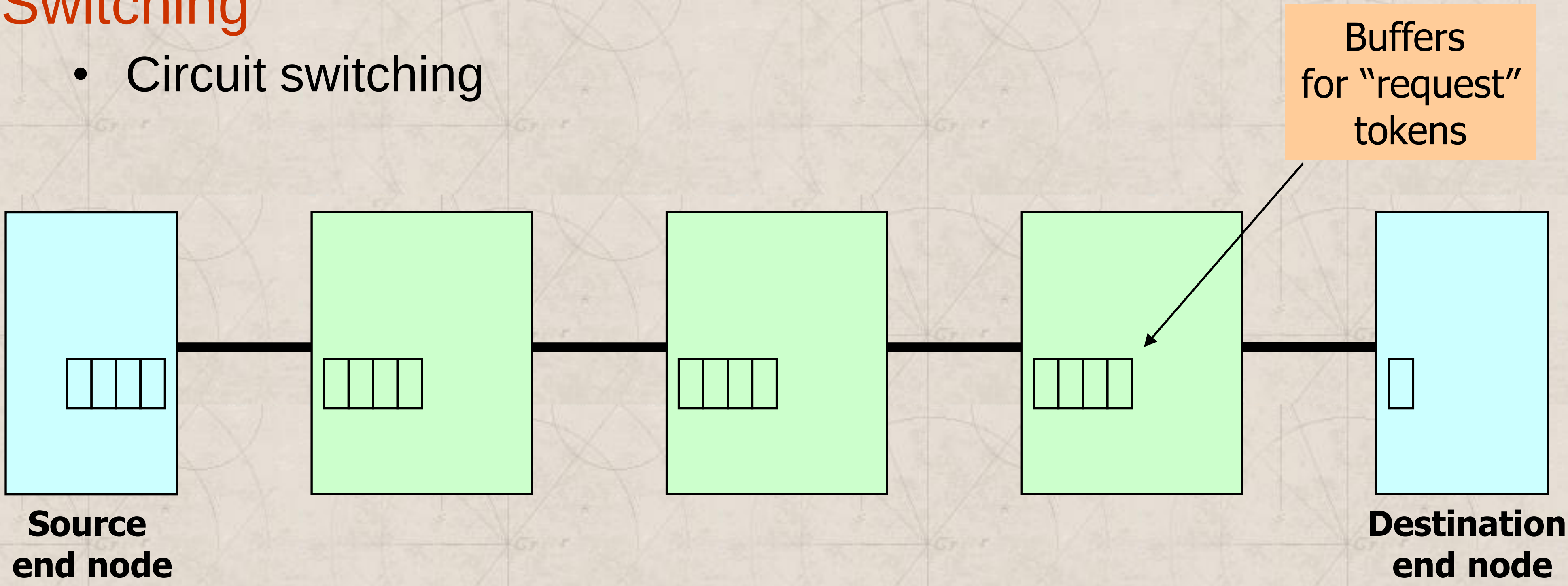
Switching

- *Circuit switching*
  - A “circuit” path is established *a priori* and torn down after use
  - Possible to pipeline the establishment of the circuit with the transmission of multiple successive packets along the circuit
    - *pipelined circuit switching*
  - Routing, arbitration, switching performed once for train of packets
    - Routing bits not needed in each packet header
    - Reduces latency and overhead
  - Can be highly wasteful of scarce network bandwidth
    - Links and switches go under utilized
      - during path establishment and tear-down
      - if no train of packets follows circuit set-up

Routing, Arbitration, and Switching

Switching

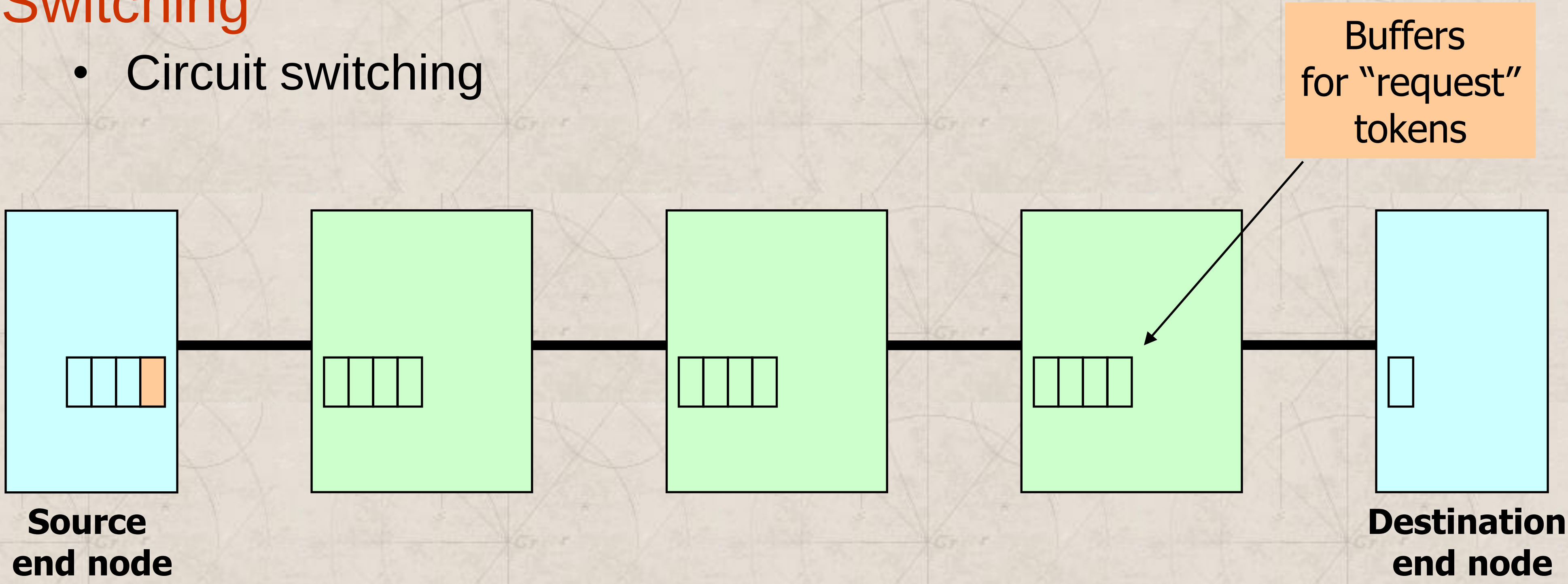
- Circuit switching



Routing, Arbitration, and Switching

Switching

- Circuit switching

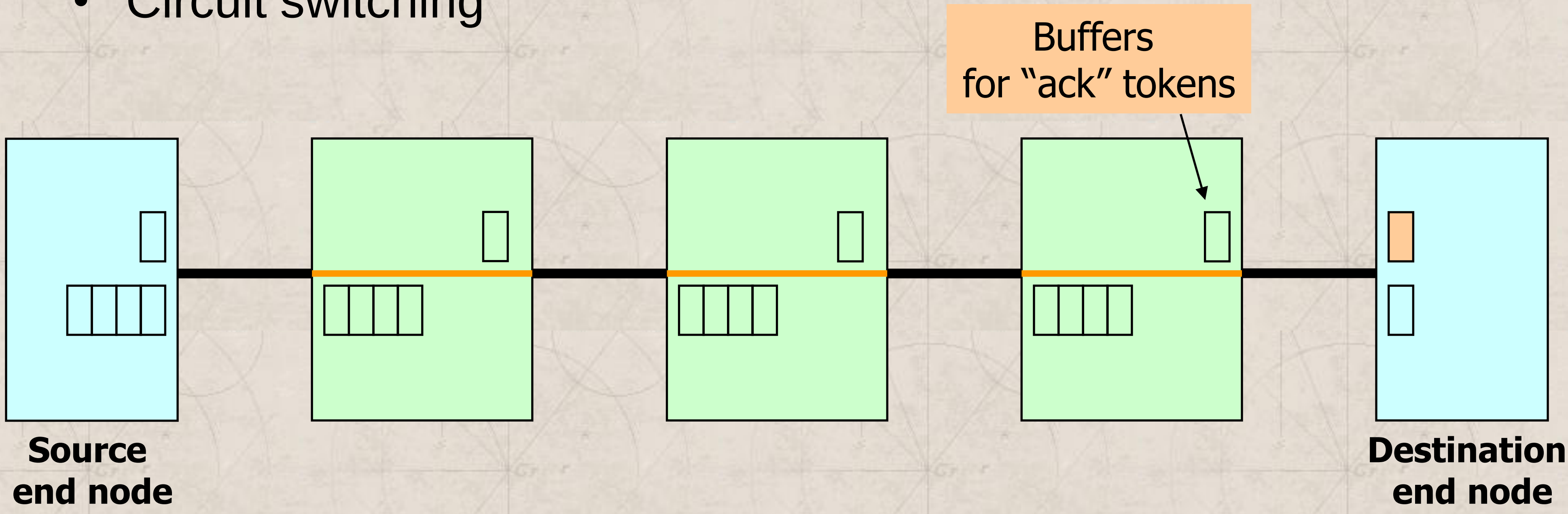


Request for circuit establishment (routing and arbitration is performed during this step)

Routing, Arbitration, and Switching

Switching

- Circuit switching



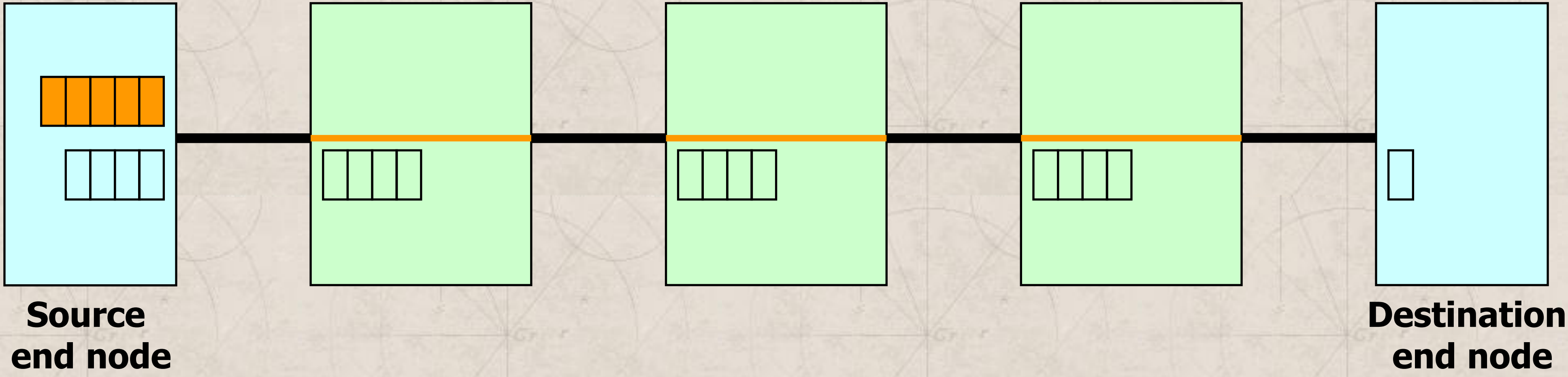
Request for circuit establishment

Acknowledgment and circuit establishment
(as token travels back to the source, connections are established)

Routing, Arbitration, and Switching

Switching

- Circuit switching



Request for circuit establishment

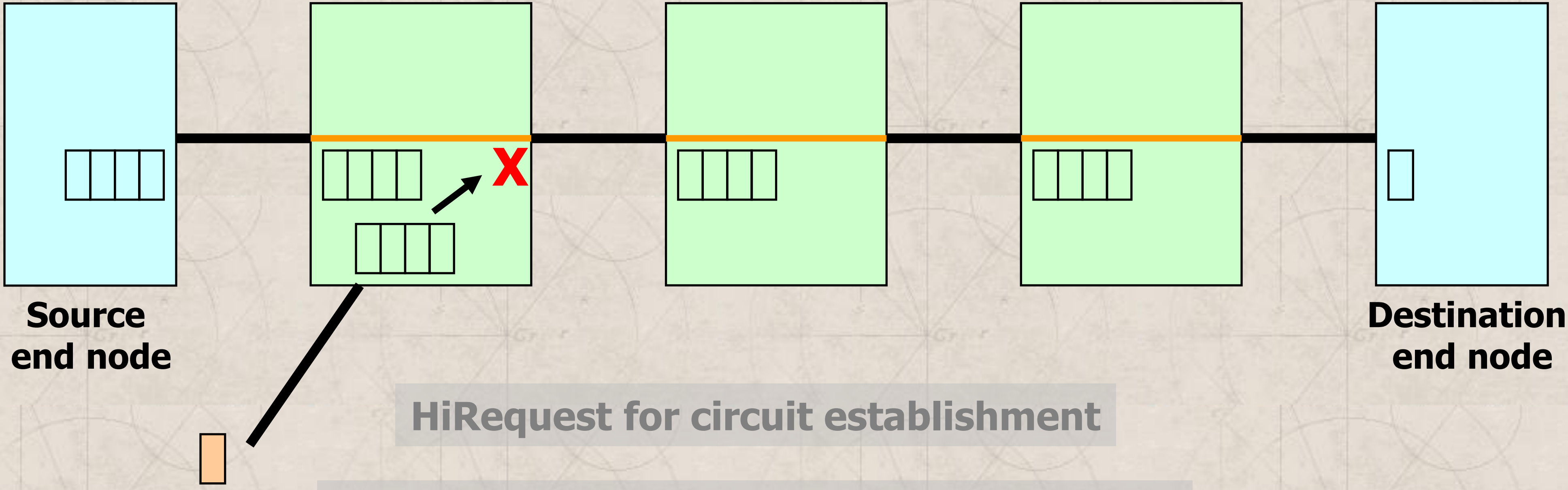
Acknowledgment and circuit establishment

Packet transport
(neither routing nor arbitration is required)

Routing, Arbitration, and Switching

Switching

- Circuit switching



HiRequest for circuit establishment

Acknowledgment and circuit establishment

Packet transport

High contention, low utilization (*ρ*) → *low throughput*

Routing, Arbitration, and Switching

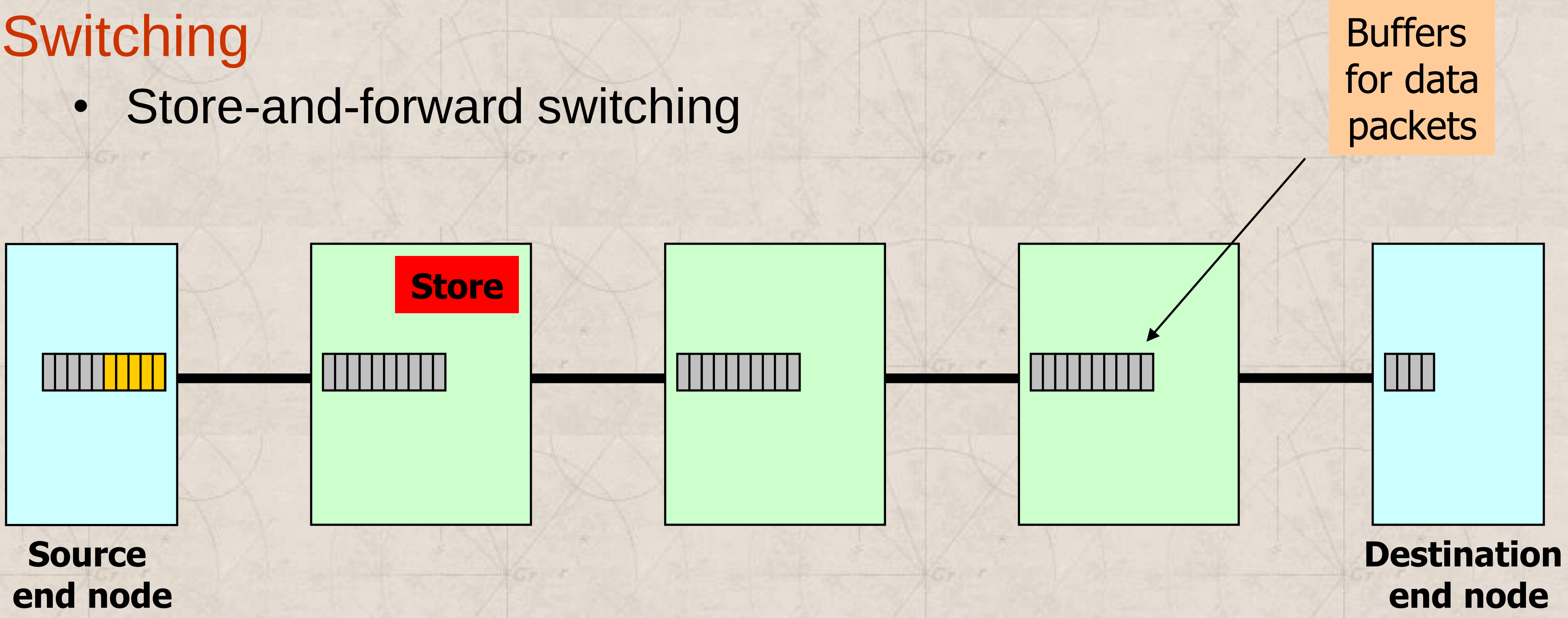
Switching

- *Packet switching*
  - Routing, arbitration, switching is performed on a per-packet basis
  - Sharing of network link bandwidth is done on a per-packet basis
  - More efficient sharing and use of network bandwidth by multiple flows if transmission of packets by individual sources is more intermittent
  - *Store-and-forward* switching
    - Bits of a packet are forwarded only after entire packet is first stored
    - Packet transmission delay is *multiplicative* with hop count, *d*
  - *Cut-through* switching
    - Bits of a packet are forwarded once the header portion is received
    - Packet transmission delay is *additive* with hop count, *d*
    - *Virtual cut-through*: flow control is applied at the packet level
    - *Wormhole*: flow control is applied at the *flow unit* (*flit*) level
    - *Buffered wormhole*: flit-level flow control with centralized buffering

Routing, Arbitration, and Switching

Switching

- Store-and-forward switching

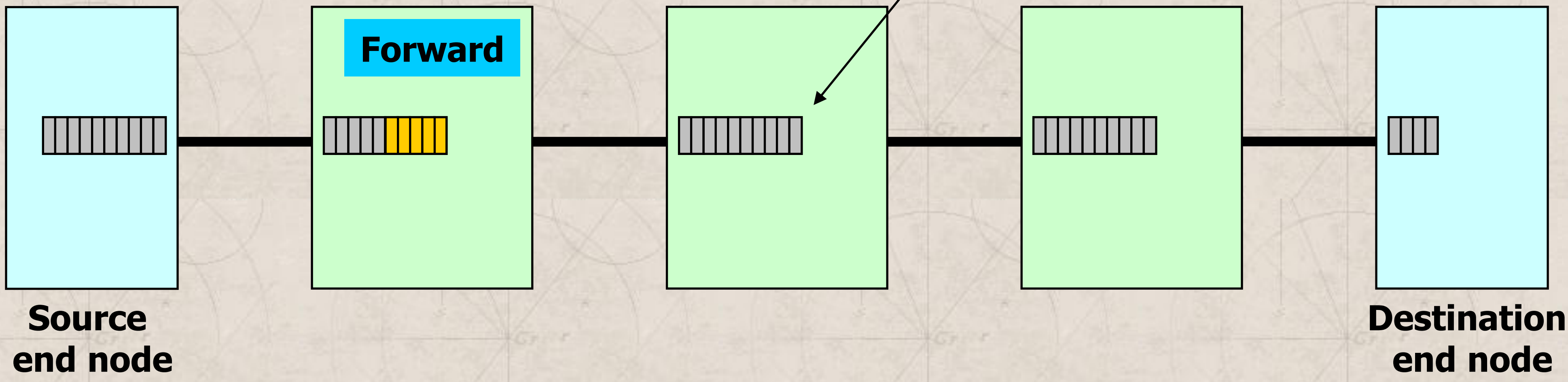


Packets are completely stored before any portion is forwarded

Routing, Arbitration, and Switching

Switching

- Store-and-forward switching



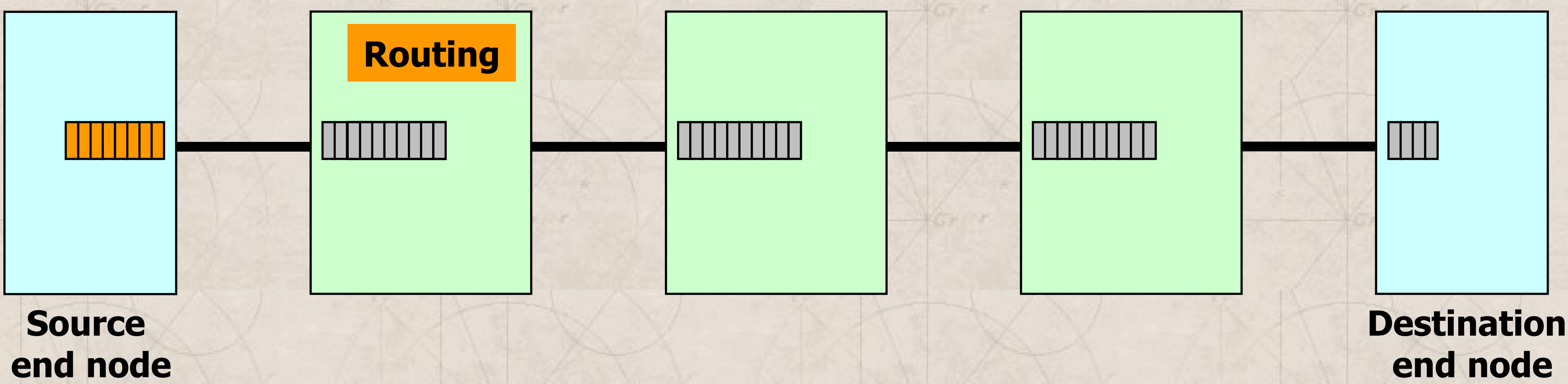
Requirement: buffers must be sized to hold entire packet (MTU)

Packets are completely stored before any portion is forwarded

Routing, Arbitration, and Switching

Switching

- Cut-through switching

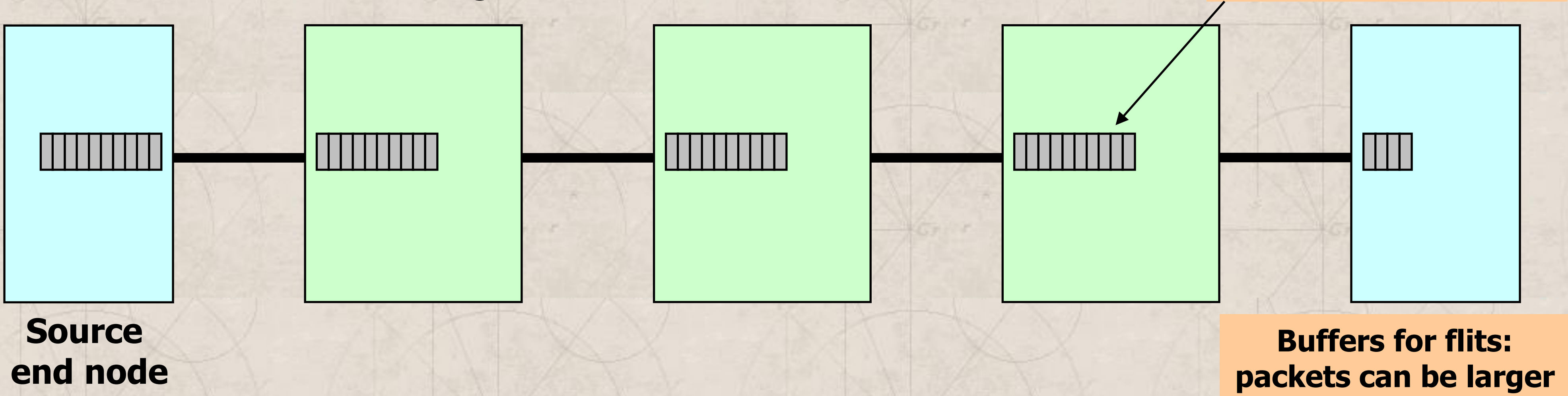


Portions of a packet may be forwarded ("cut-through") to the next switch before the entire packet is stored at the current switch

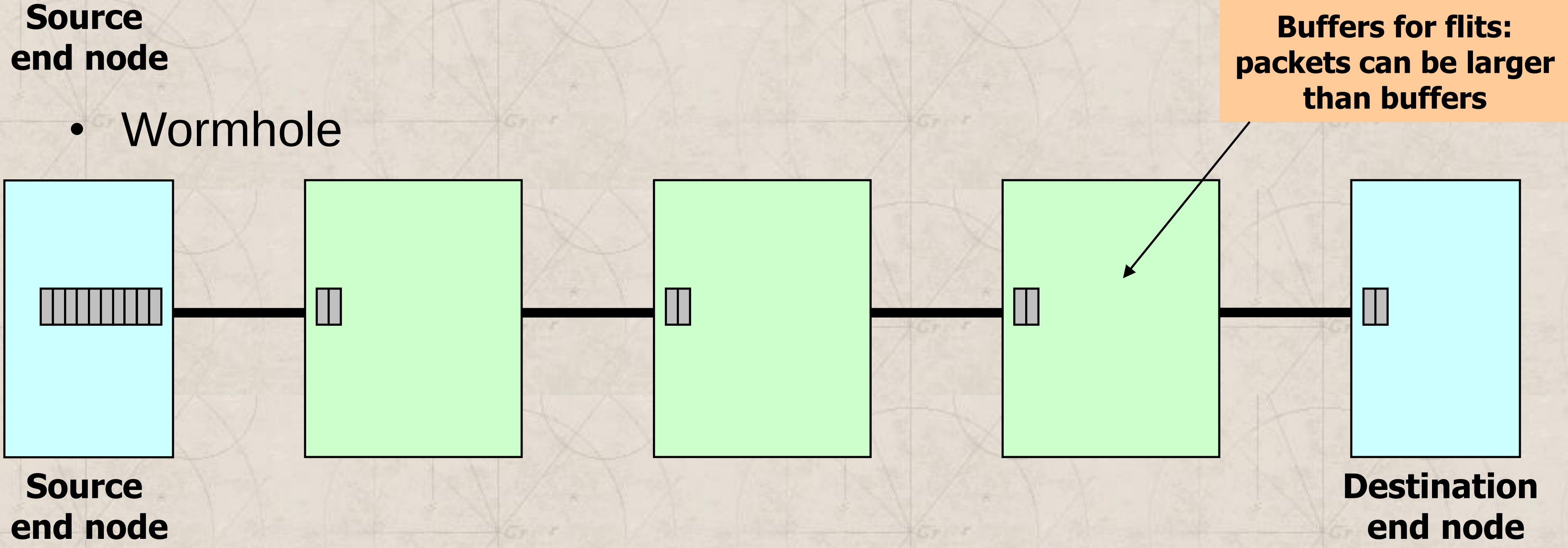
Routing, Arbitration, and Switching

Switching

- Virtual cut-through


Buffers for data packets
Requirement: buffers must be sized to hold entire packet (MTU)
Source end node

- Wormhole

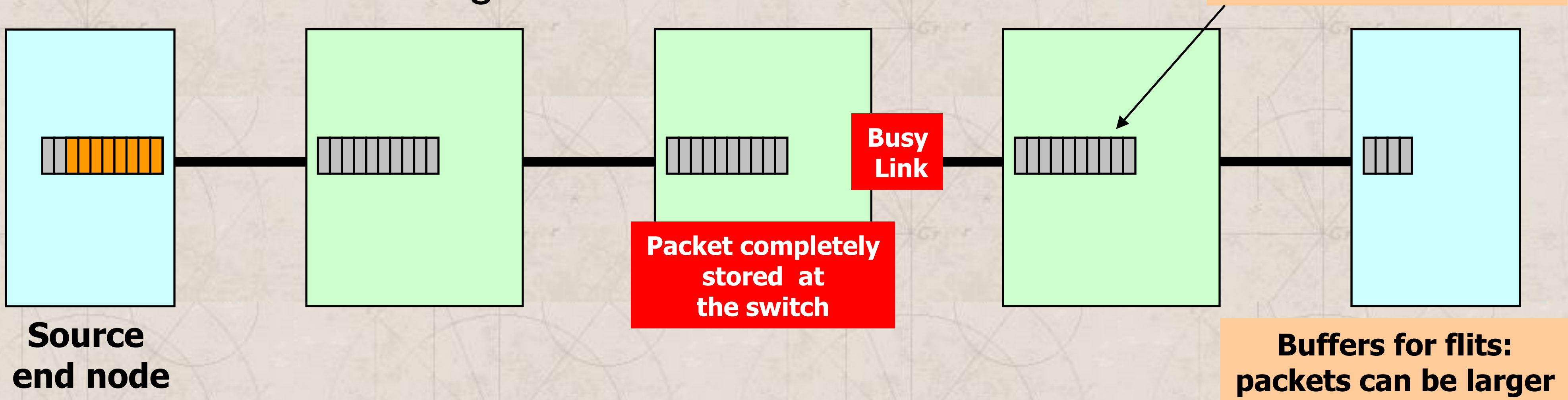


"Virtual Cut-Through: A New Computer Communication Switching Technique," P. Kermani and L. Kleinrock, *Computer Networks*, 3, pp. 267–286, January, 1979.

Routing, Arbitration, and Switching

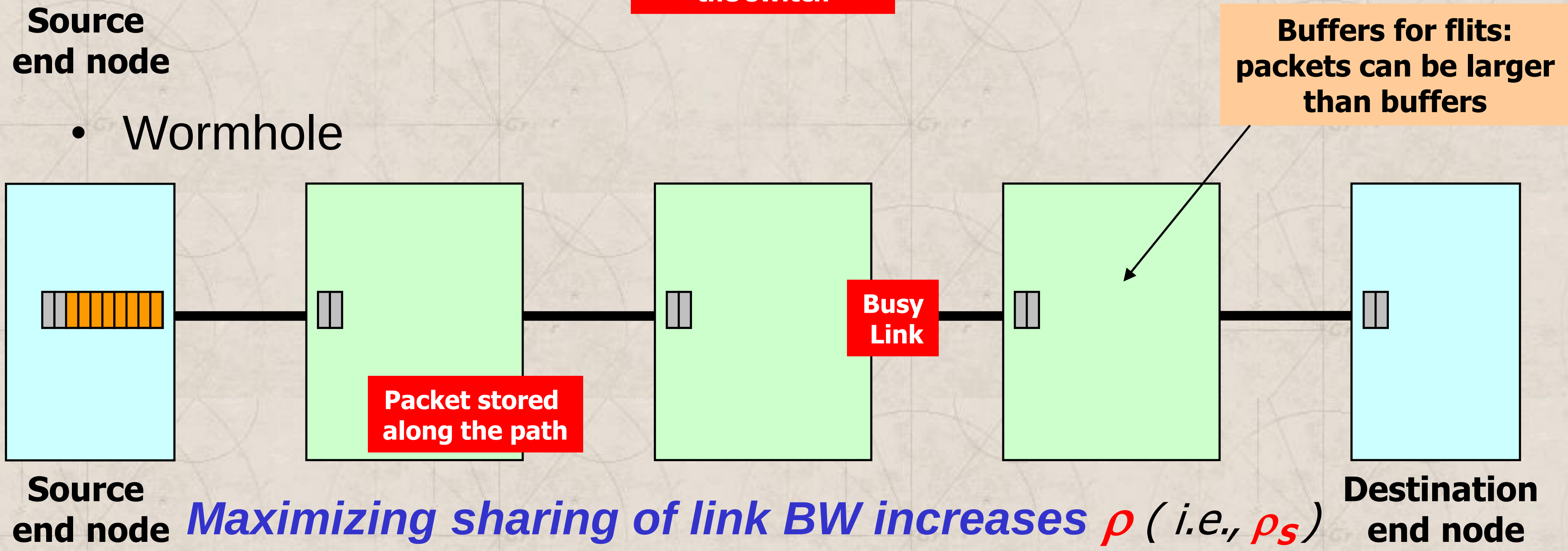
Switching

- Virtual cut-through



Buffers for data packets
Requirement: buffers must be sized to hold entire packet (MTU)
Busy Link
Packet completely stored at the switch
Source end node

- Wormhole



Buffers for flits: packets can be larger than buffers
Busy Link
Packet stored along the path
Source end node
Destination end node

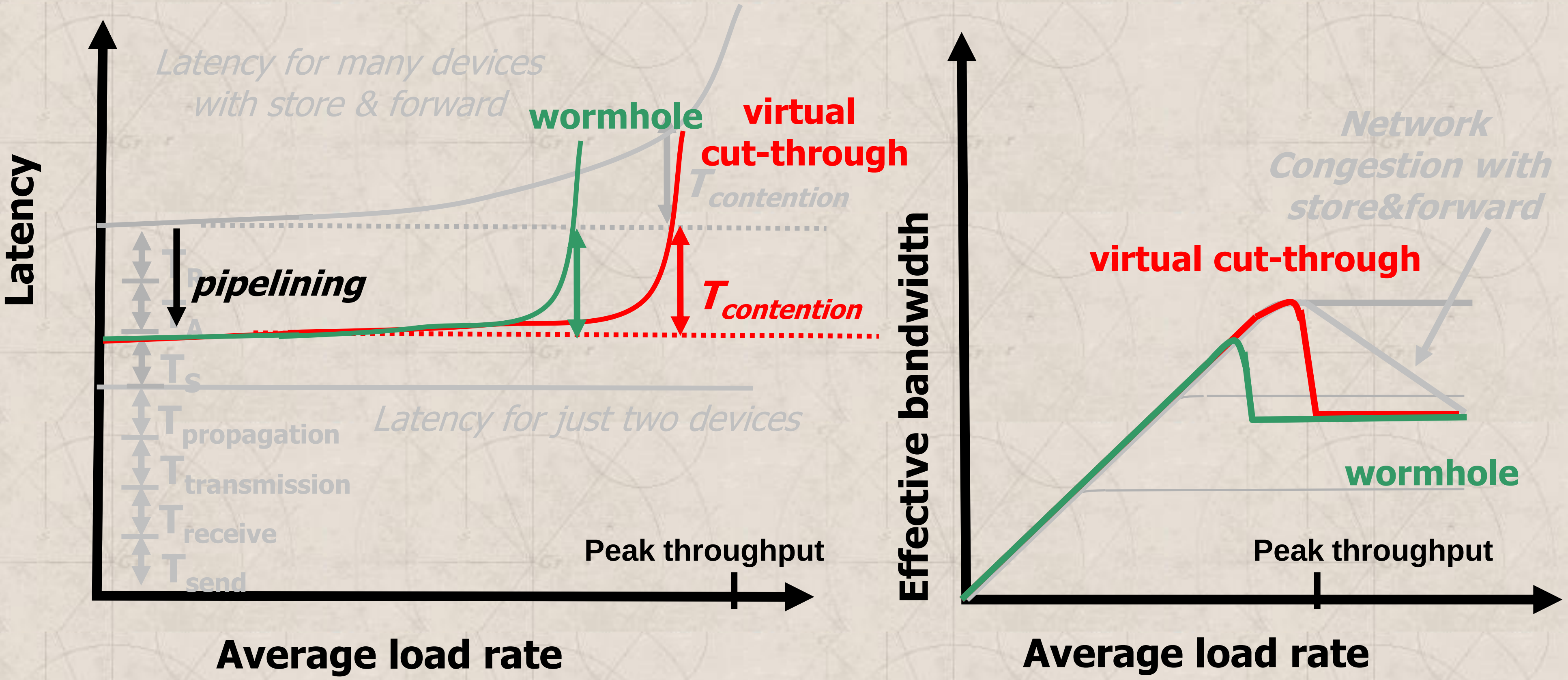
Maximizing sharing of link BW increases ρ *(i.e.,* ρ_s *)*

"Virtual Cut-Through: A New Computer Communication Switching Technique," P. Kermani and L. Kleinrock, *Computer Networks*, 3, pp. 267–286, January, 1979.

Routing, Arbitration, and Switching

Characterizing Performance: Latency & Effective Bandwidth

- Characteristic performance plots: latency vs. average load rate; throughput (effective bandwidth) vs. average load rate



Routing, Arbitration, and Switching

R, A, & S Characteristics of Commercial Machines

Company	System [Network] Name	Max. compute nodes [x #CPUs]	Basic network topology	Network routing algorithm	Switch arbitration scheme	Network switching technique
Intel	ASCI Red Paragon	4,510 [x 2]	2-D mesh 64 x 64	distributed dimension-order routing	2-phased RR, distributed across switch	wormhole w/ no virtual channels
IBM	ASCI White SP Power3 [Colony]	512 [x 16]	BMIN w/8-port bidirect. switches (fat-tree or Omega)	source-based LCA adaptive, shortest-path routing	2-phased RR, centralized & distributed at outputs for bypass paths	buffered WH & VCT for multicasting, no VCs
Intel	Thunter Itanium2 Tiger4 [QsNet^{II}]	1,024 [x 4]	fat tree w/8-port bidirectional switches	source-based LCA adaptive, shortest path routing	2-phased RR, priority, aging, distributed at output ports	WH with 2 VCs
Cray	XT3 [SeaStar]	30,508 [x 1]	3-D torus 40 x 32 x 24	distributed table-based dimension-order	2-phased RR, distributed at output ports	VCT with 4 VCs
Cray	X1E	1,024 [x 1]	4-way bristled 2-D torus (~ 23 x 11) with express links	distributed table-based dimension-order	2-phased RR, distributed at output ports	VCT with 4 Vcs
IBM	ASC Purple pSeries 575 [Federation]	>1,280 [x 8]	BMIN w/8-port bidirect. switches (fat-tree or Omega)	source and distrib. table-based LCA adapt. shortest path	2-phased RR, centralized & distributed at outputs for bypass paths	buffered WH & VCT for multicasting, 8 VCs
IBM	Blue Gene/L eServer Sol. [Torus Net]	65,536 [x 2]	3-D torus 32 x 32 x 64	distributed adaptive with bubble escape Duato's Protocol	2-phased SLQ, distributed at input & output	VCT with 4 VCs

Outline

- E.1 Introduction *(Lecture 1)*
- E.2 Interconnecting Two Devices *(Lecture 1)*
- E.3 Interconnecting Many Devices *(Lecture 2)*
- E.4 Network Topology *(Lecture 2)*
- E.5 Network Routing, Arbitration, and Switching *(Lecture 3)*
- **E.6 Switch Microarchitecture** *(Lecture 4)*
    - **Basic Switch Microarchitecture**
    - **Buffer Organizations**
    - **Routing and Arbitration Unit**
    - **Pipelining the Switch Microarchitecture**
    - **Characterizing Performance: Latency & Effective Bandwidth**
- E.7 Practical Issues for Commercial Interconnection Networks *(Lecture 4)*
- E.8 Examples of Interconnection Networks *(Lecture 5)*
- E.9 Internetworking (skipped)
- E.10 Crosscutting Issues for Interconnection Networks (skipped)
- E.11 Fallacies and Pitfalls *(Lecture 5)*
- E.12 Concluding Remarks and References *(Lecture 5)*

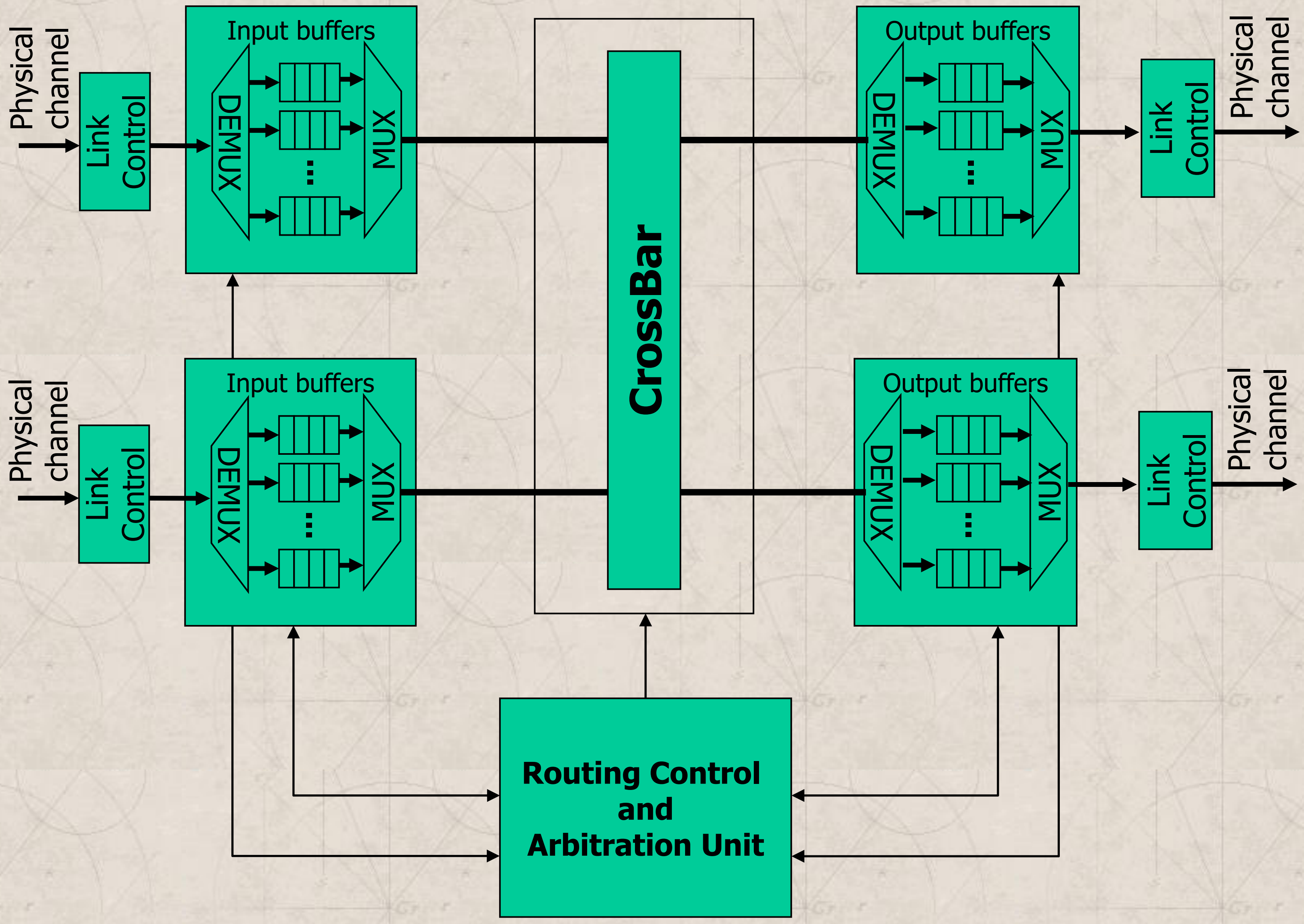
Switch Microarchitecture

Basic Switch Microarchitecture

- Internal data path
  - Implements flow control, routing, arbitration, and switching
  - Provides connectivity between switch input and output ports
  - A crossbar is commonly used to provide internal connectivity
    - Non-blocking, concurrent connectivity
  - Other components along the internal datapath consist of
    - link (flow) control units, I/O buffers, routing and arbitration unit
- *Speedup:* ratio of provided bandwidth to required bandwidth
  - Implemented within the internal data path of a switch by
    - Increased clock frequency (time) or internal datapath width (space)
    - Multiple datapaths via increased # of crossbar access points
    - Alternatively, multiple datapaths via a buffered crossbar switch
      - Arbitration made simpler (independent, distributed arbiters)
      - Expensive architecture

Switch Microarchitecture

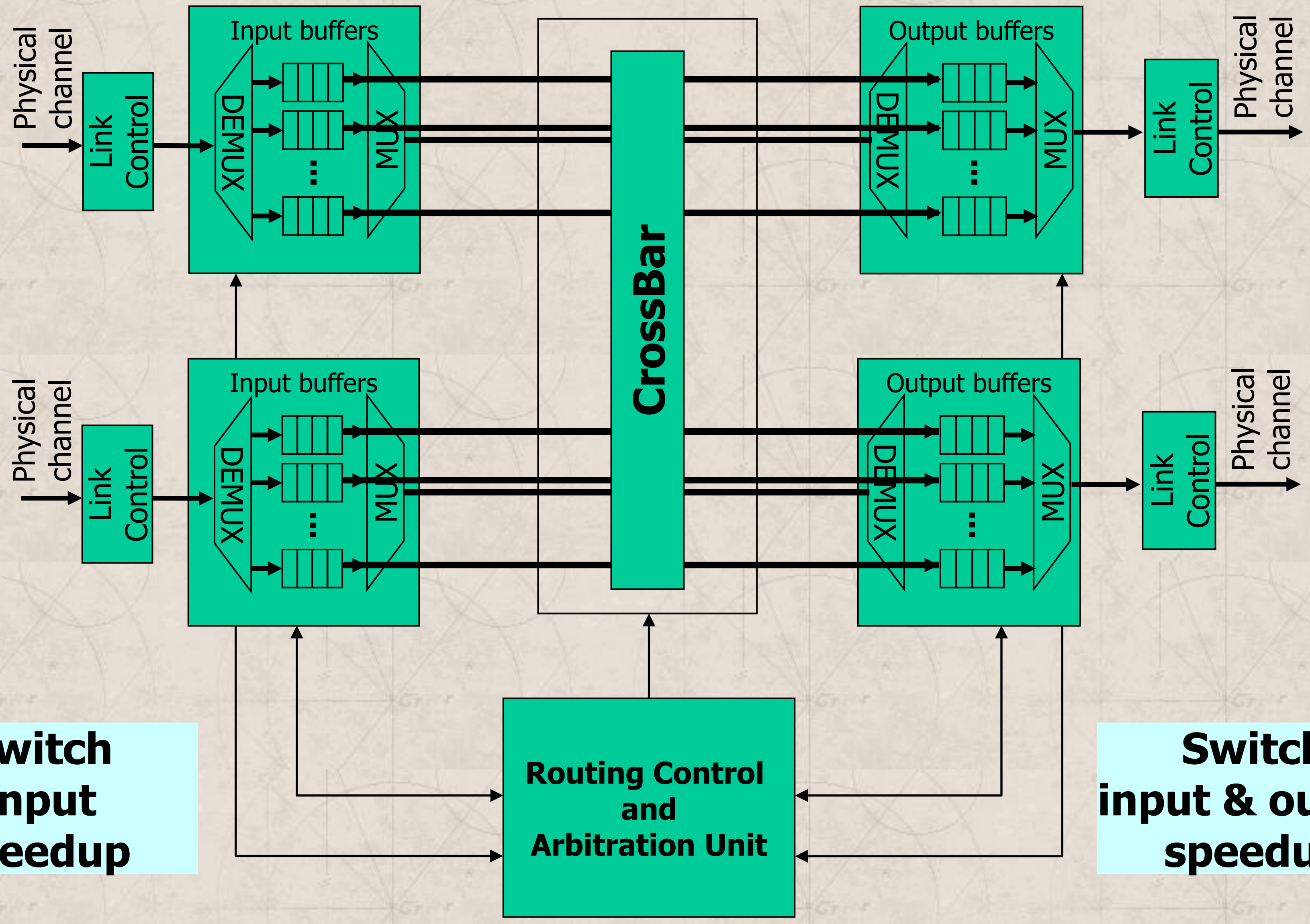
Basic Switch Microarchitecture



Interconnection Networks: © Timothy Mark Pinkston and José Duato
...with major presentation contribution from José Flich

Switch Microarchitecture

Basic Switch Microarchitecture



Switch input speedup

Switch input & output speedup

Maximizing use of internal switch datapath can increase ρ ***(i.e.,*** $\rho_{\mu Arch}$ ***)***

Switch Microarchitecture

Buffer Organizations

- Implemented as FIFOs, circular queues, central memory, or dynamically allocated multi-queues (*DAMQs*) in SRAMs
    - › Input ports (*input-buffered switch*)
    - › Output ports (*output-buffered switch*)
    - › Centrally within switch (*centrally-buffered switch* or *buffered Xbar*)
    - › At both input and output ports (*input-output-buffered switch*)
- Must guard against *head-of-line* (*HOL*) *blocking*
    - – Arises from two or more packets buffered in the same queue
    - – A blocked packet at the head of the queue prevents other packets in the queue from advancing that would otherwise be able to advance if they were at the queue head
    - – Output-buffered switches eliminate HOL blocking within switch
        - › *k*-way speedup required for a *k* x *k* output-buffered switch
        - › Implementations with moderate (< *k*) speedup must drop packets

Switch Microarchitecture

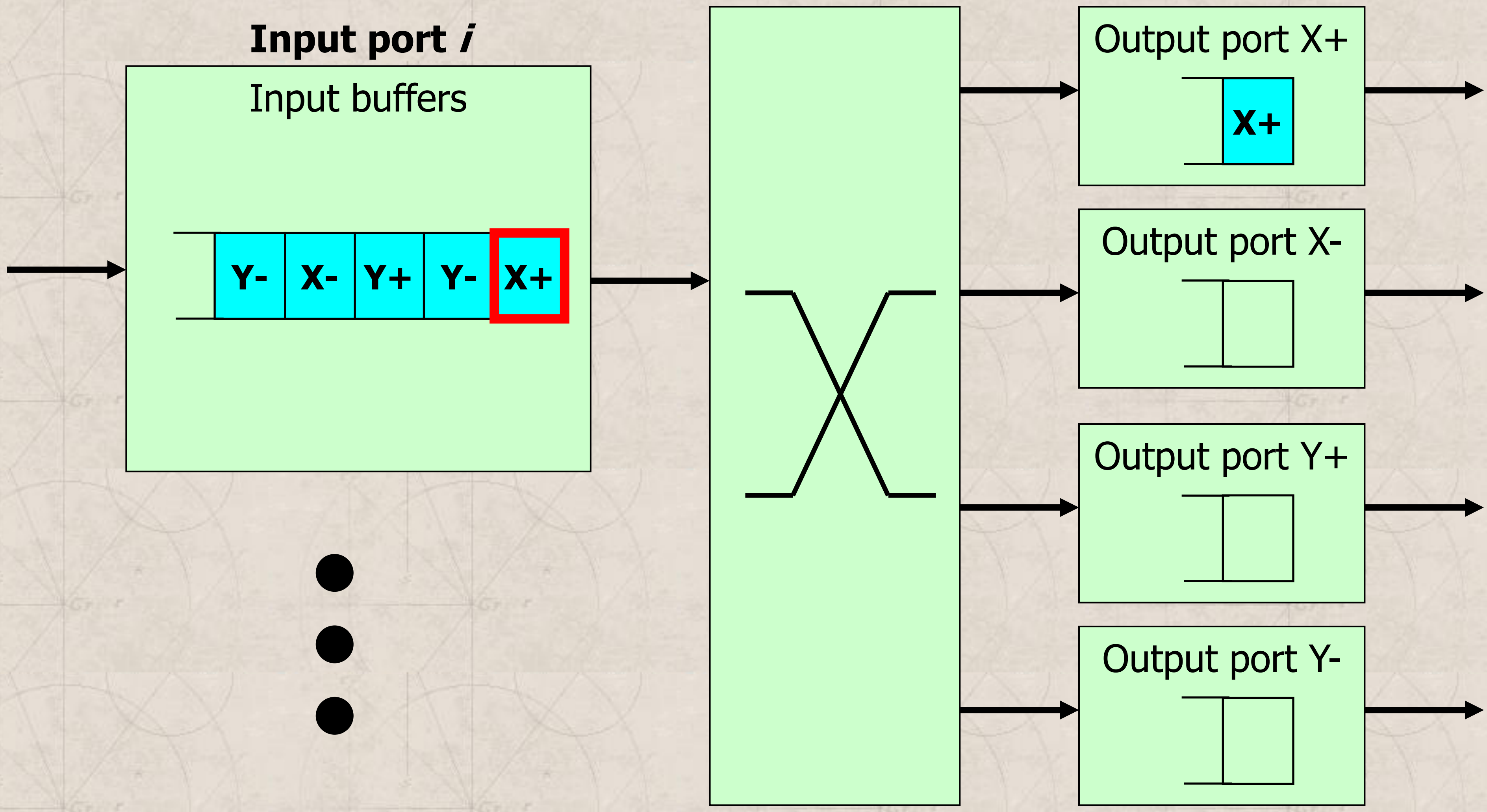
Buffer Organizations

- Head-of-line (HOL) blocking (continued)
  - Input-buffered switches do not require speedup
    - HOL blocking may appear: <60% switch efficiency w/ uniform traffic
    - *Virtual channels* can *mitigate*, but do not *eliminate*, HOL blocking
    - *Virtual Output Queues* (*VOQs*) avoid HOL blocking *within a switch*
      - As many queues in each input port as there are output ports
      - Costly, not scalable: # of queues grow quadratically w/ # ports
      - Does not eliminate HOL blocking of flows that span across multiple switches (unless as many VOQs as there are dest's)
  - Combined input-output-buffered switches
    - Reduces (but not eliminates) HOL blocking and required speedup
    - Decouples packet transmission through links and internal crossbar
  - Buffered crossbar switch
    - HOL blocking is eliminated *within a switch*
    - Again, a very expensive architecture

Switch Microarchitecture

Buffer Organizations

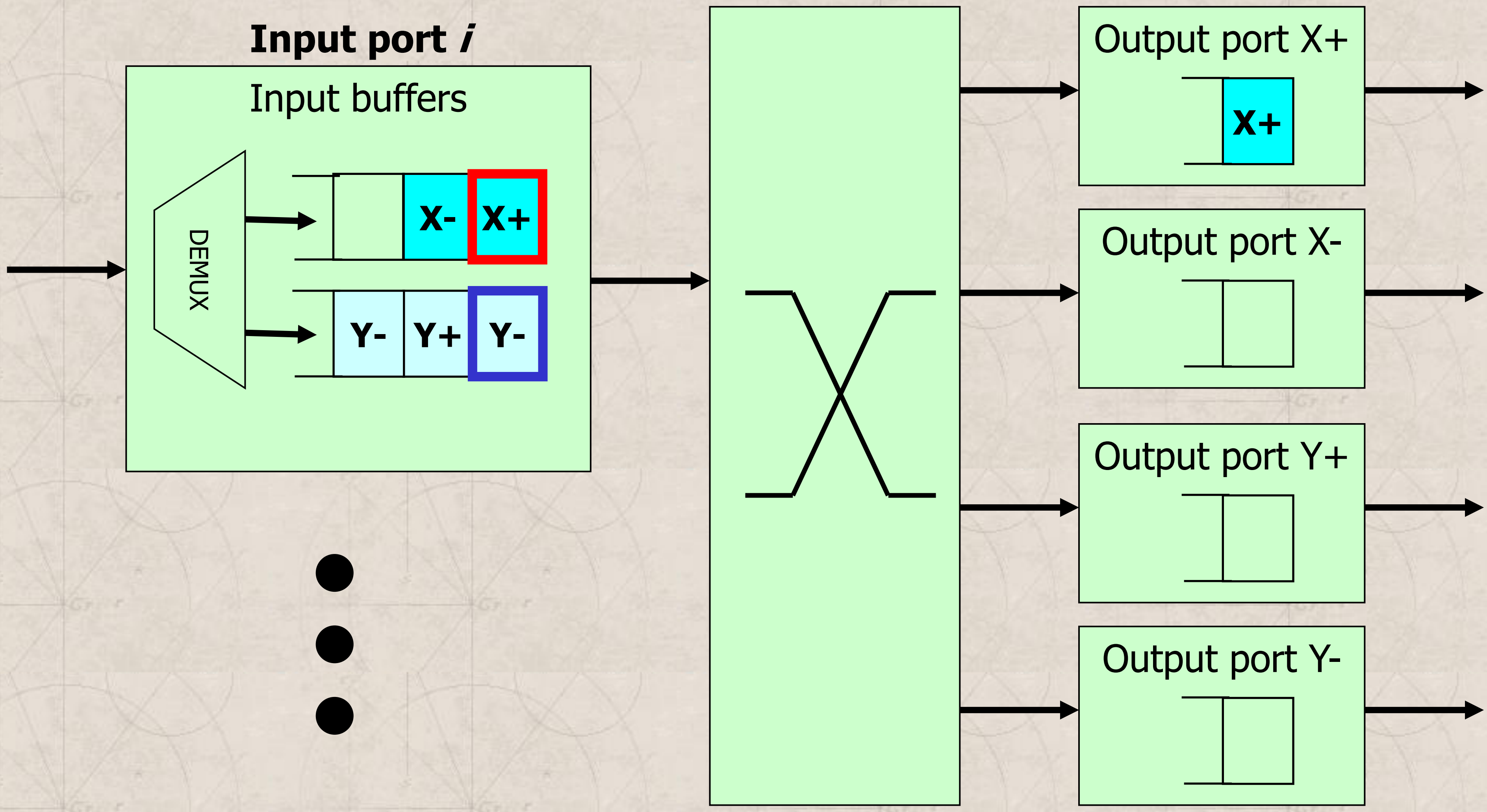
- HOL blocking at an input port



Switch Microarchitecture

Buffer Organizations

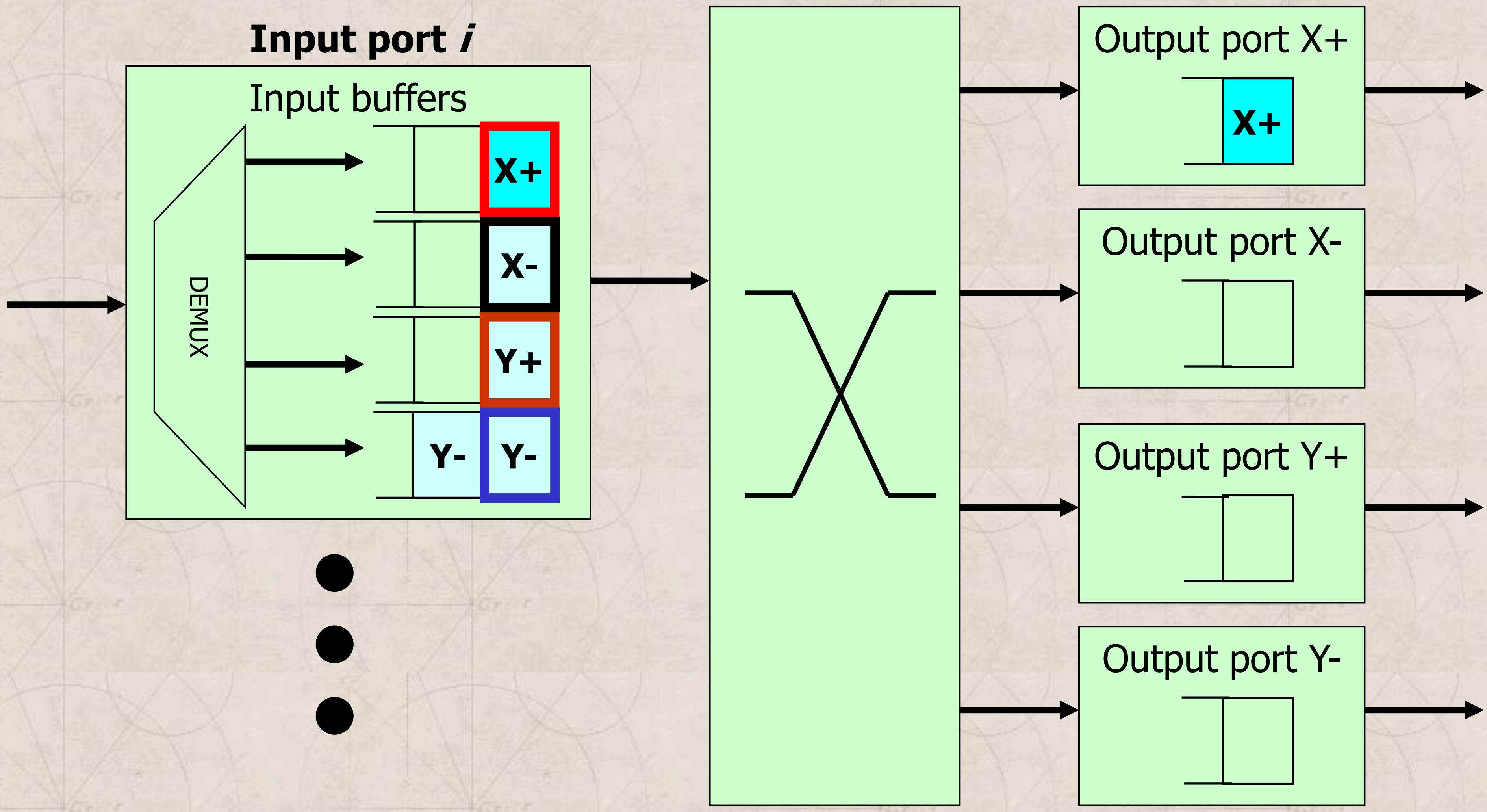
- HOL blocking is *reduced* when using *virtual channels* (2 queues)



Switch Microarchitecture

Buffer Organizations

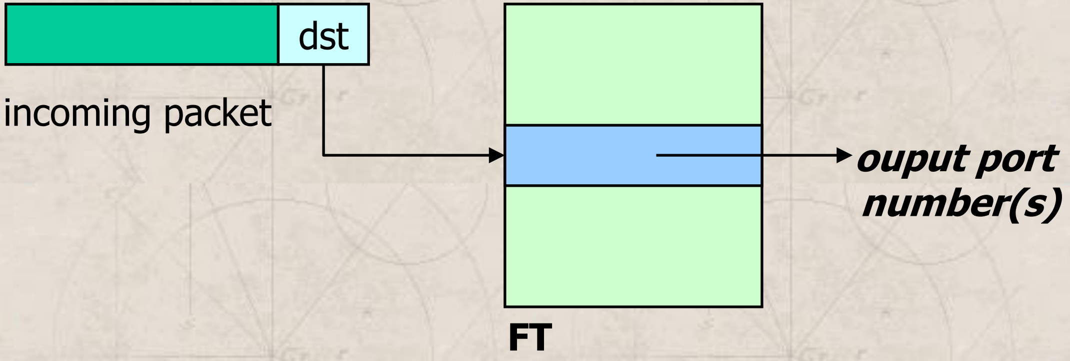
- HOL blocking is *avoided at switch* using *VOQs* (need *k* queues)



Switch Microarchitecture

Routing and Arbitration Unit

- Usually is implemented as a centralized resource
  - Routing done on a per-packet basis
- Finite-state machine (FSM)
  - Based on routing information in the header, FSM computes the output port(s) (several if adaptive routing)
  - Routing info at header is usually stripped off or modified
- Forwarding table (FT)
  - Routing info used as an address to access the forwarding table
  - FT must be preloaded into switches



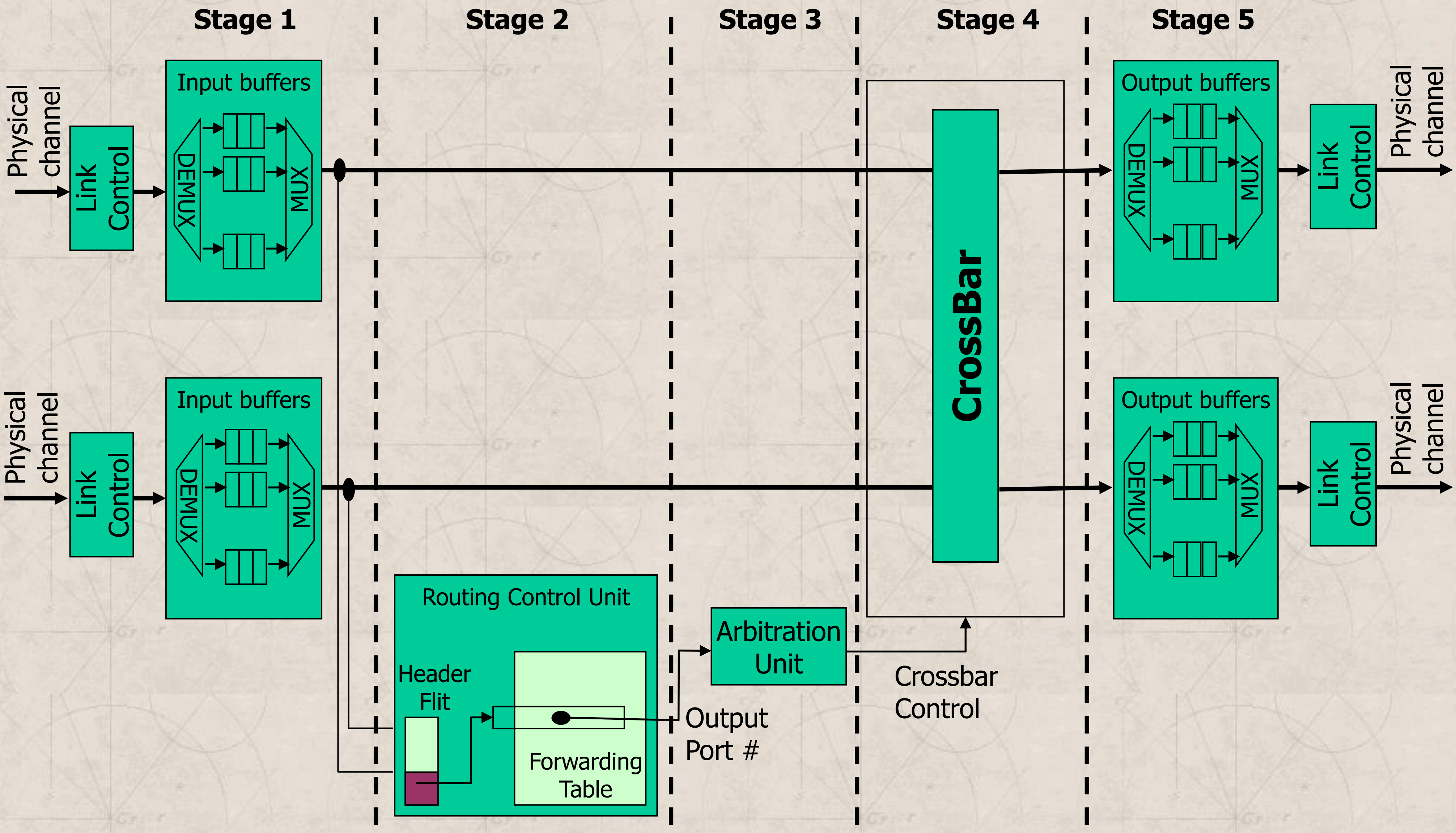
Switch Microarchitecture

Pipelining the Switch Microarchitecture

- Similarities with vector processors
  - Packet header indicates how to process the *ph*ysical un*its* (*phits*)
- Packets at different input ports are independent
  - Parallelism

Switch Microarchitecture

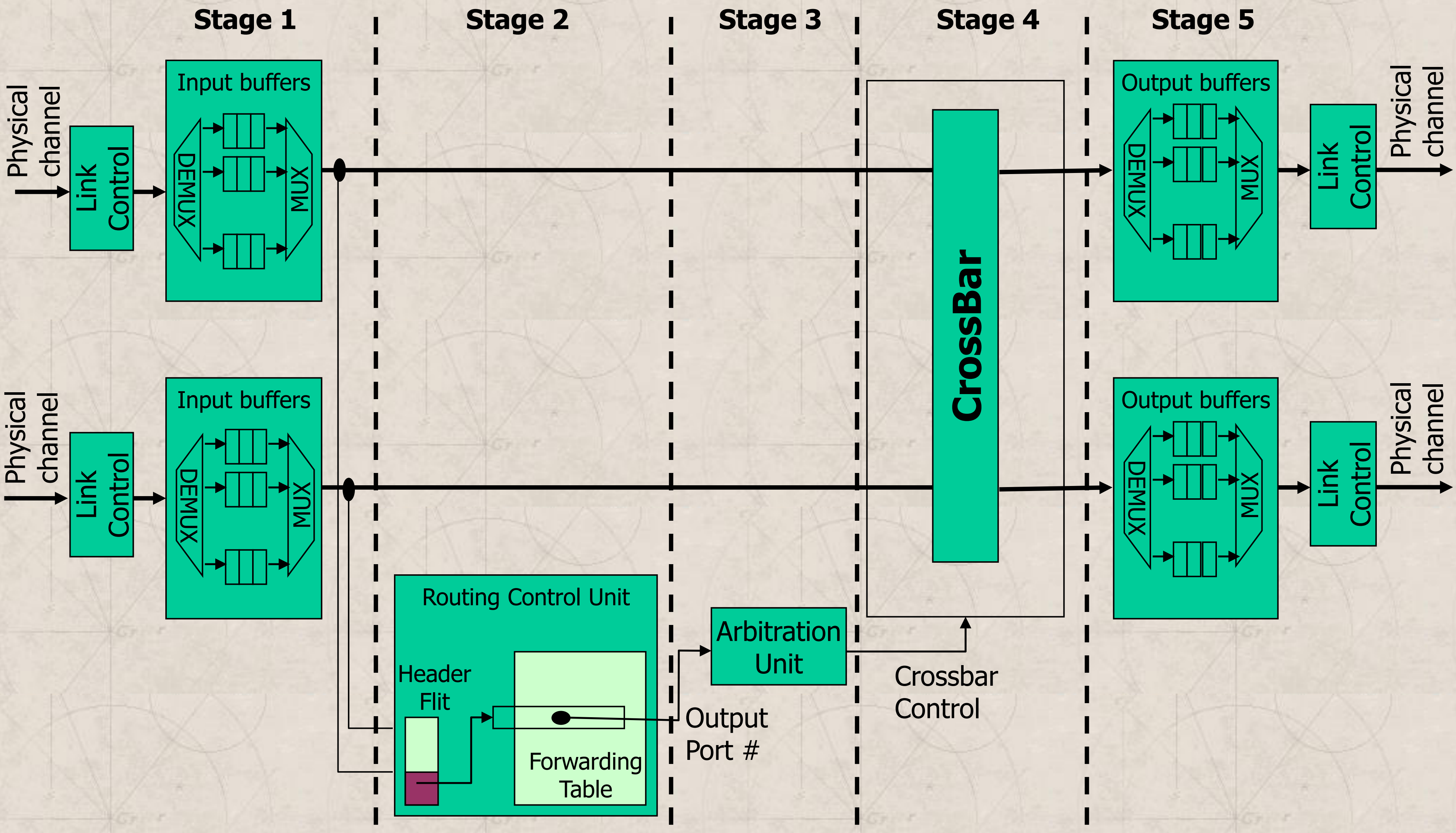
Pipelining the Switch Microarchitecture



IB (Input Buffering)	**RC (Route Computation)**	**SA (Switch Arb) - VCA (VC Arb) -**	**ST (Switch Trasv)**	**OB (Output Buffering)**

Switch Microarchitecture

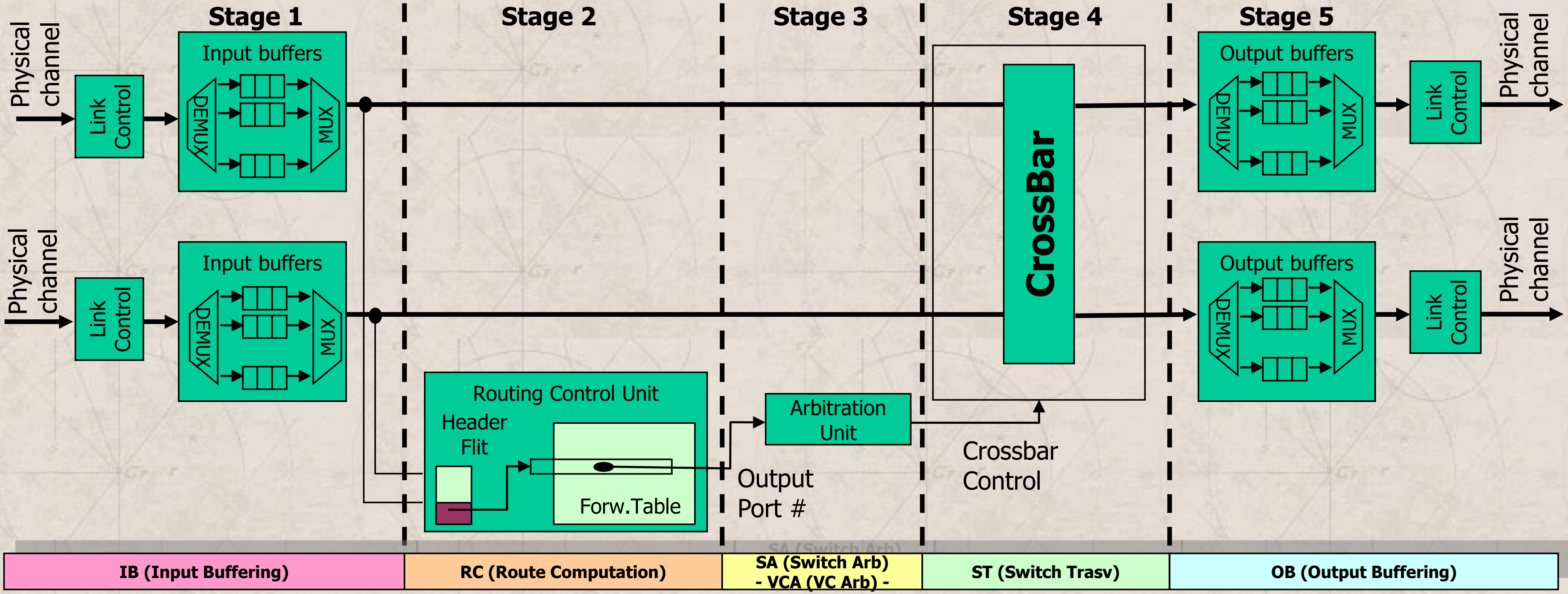
Pipelining the Switch Microarchitecture

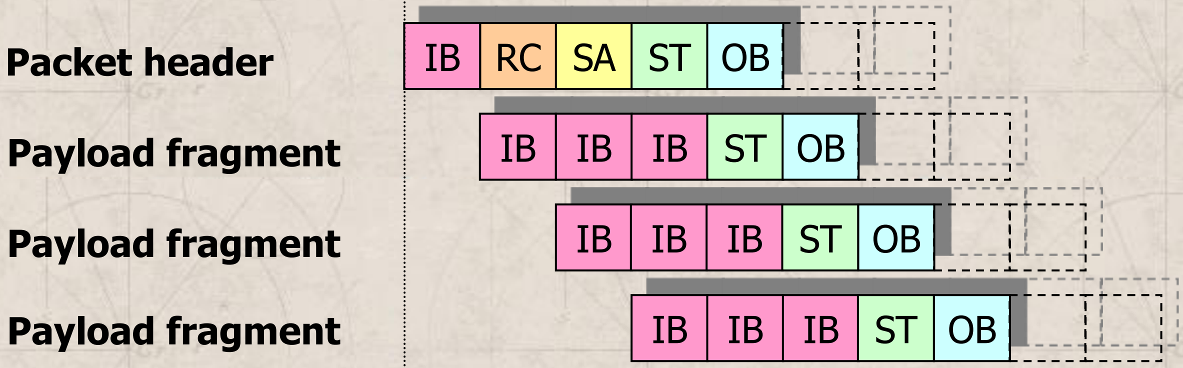


IB (Input Buffering)	**RC (Route Computation)**	**SA (Switch Arb) - VCA (VC Arb) -**	**ST (Switch Trasv)**	**OB (Output Buffering)**

Switch Microarchitecture

Pipelining the Switch Microarchitecture





Matching the throughput of the internal switch datapath to the external link BW can increase ρ (i.e., $\rho_{\mu Arch}$)

"A Delay Model and Speculative Architecture for Pipelined Routers," L. S. Peh and W. J. Dally, *Proc. of the 7th Int'l Symposium on High Performance Computer Architecture*, Monterrey, January, 2001.

Outline

- E.1 Introduction *(Lecture 1)*
- E.2 Interconnecting Two Devices *(Lecture 1)*
- E.3 Interconnecting Many Devices *(Lecture 2)*
- E.4 Network Topology *(Lecture 2)*
- E.5 Network Routing, Arbitration, and Switching *(Lecture 3)*
- E.6 Switch Microarchitecture *(Lecture 4)*
- **E.7 Practical Issues for Commercial Interconnection Networks** *(Lecture 4)*
  - Connectivity (skipped)
  - Standardization (skipped)
  - **Congestion Management**
  - **Fault Tolerance**
- E.8 Examples of Interconnection Networks *(Lecture 5)*
- E.9 Internetworking (skipped)
- E.10 Crosscutting Issues for Interconnection Networks (skipped)
- E.11 Fallacies and Pitfalls *(Lecture 5)*
- E.12 Concluding Remarks and References *(Lecture 5)*

Practical Issues for Interconnection Networks

Fault Tolerance

- Three categories of techniques to deal with permanent failures
  - Resource sparing (redundancy)
    - Faulty resources are bypassed and spare ones are switched in
      - ServerNet, IBM Blue Gene/L (healthy resources be removed)
  - Fault-tolerant routing
    - Alternative paths are incorporated into routing function from start
      - Cray T3E
    - May not account for all (many) possible fault combinations
  - Network reconfiguration
    - A more general, less costly technique
      - Myrinet, Quadrics, InfiniBand, Advanced Switching, etc.
    - Routing function (i.e., forwarding tables) reconfigured either statically or dynamically (hot swapping) to reconnect the network
    - Must guard against reconfiguration-induced deadlocks
      - More than one routing function may be active (conflict) at a time

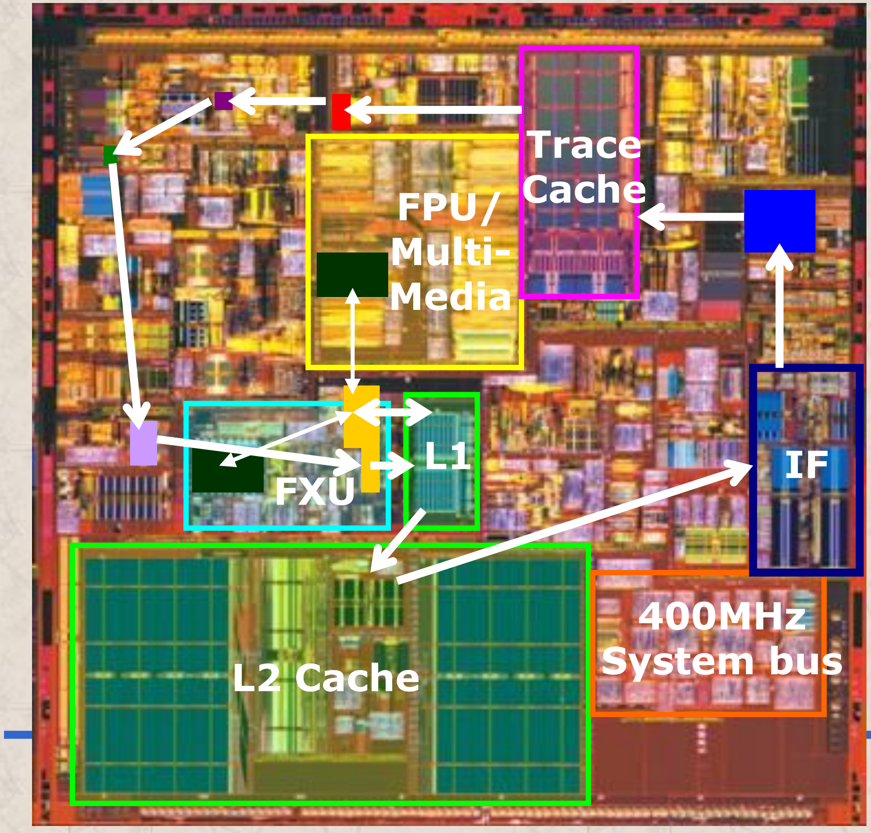
Outline

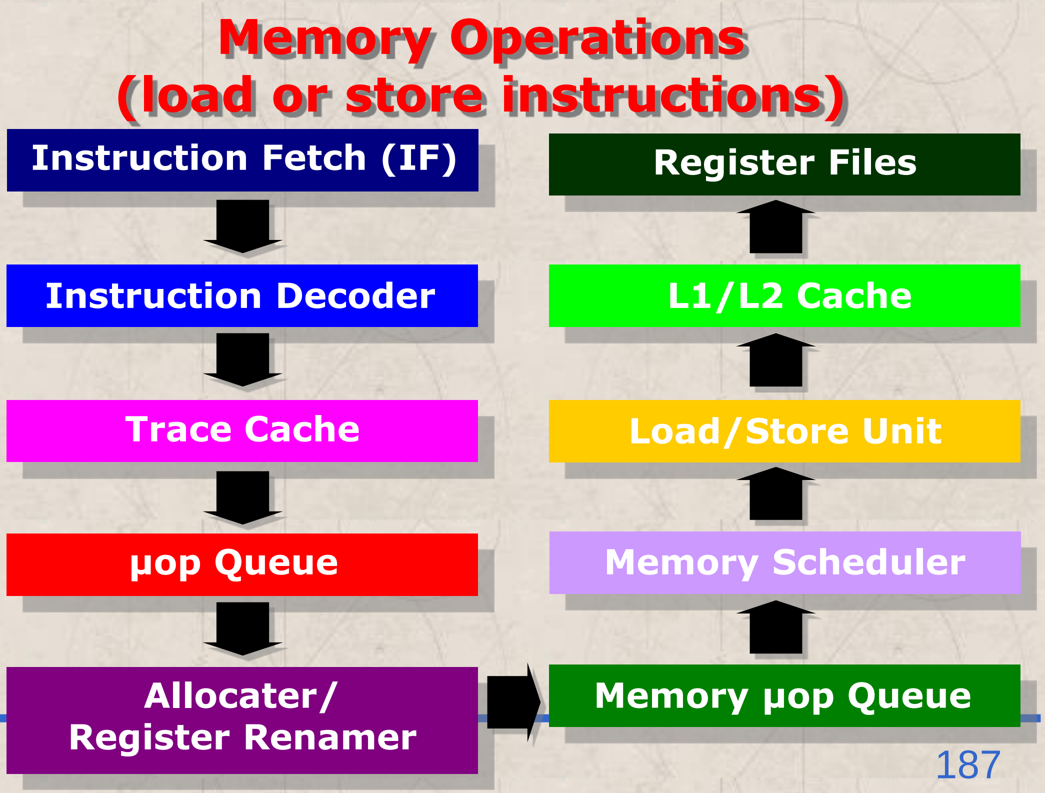
- E.1 Introduction *(Lecture 1)*
- E.2 Interconnecting Two Devices *(Lecture 1)*
- E.3 Interconnecting Many Devices *(Lecture 2)*
- E.4 Network Topology *(Lecture 2)*
- E.5 Network Routing, Arbitration, and Switching *(Lecture 3)*
- E.6 Switch Microarchitecture *(Lecture 4)*
- E.7 Practical Issues for Commercial Interconnection Networks *(Lec 4)*
- **E.8 Examples of Interconnection Networks** *(Lecture 5)*
  - **On-chip Networks (OCNs)**
  - **Cell Broadband Engine Element Interconnect Bus**
  - **System Area Networks (SANs)**
  - **Blue Gene/L 3D Torus Network**
- E.9 Internetworking (skipped)
- E.10 Crosscutting Issues for Interconnection Networks (skipped)
- E.11 Fallacies and Pitfalls *(Lecture 5)*
- E.12 Concluding Remarks and References *(Lecture 5)*

Examples of Interconnection Networks

On-Chip Networks (OCNs)

- *Multicore* architectures are displacing monolithic single cores
  - Power/area/performance-efficiency: better with multiple simpler cores than with fewer (single) more complex cores
  - Memory wall: cache miss latencies hidden with multiple threads
  - Interconnect: wire delay more scalable (fewer chip-crossings)





Examples of Interconnection Networks

On-Chip Networks (OCNs)

Institution & Processor [Network] name	Year built	Number of network ports [cores or tiles + other ports]	Basic network topology	# of data bits per link per direction	Link bandwidth [link clock speed]	Routing; Arbitration; Switching	# of chip metal layers; flow control; # VCs
MIT Raw [General Dynamic Network]	2002	16 port [16 tiles]	2-D mesh 4 x 4	32 bits	0.9 GBps [225 MHz, clocked at proc speed]	XY DOR w/ request-reply deadlock recovery; RR arbitration; wormhole	6 layers; credit-based; no VCs
IBM POWER5	2004	7 ports [2 PE cores + 5 other ports]	Crossbar	256 b Inst fetch; 64 b for stores; 256 b LDs	[1.9 GHz, clocked at proc speed]	Shortest-path; non-blocking; circuit switch	7 layers; handshaking; no virtual channels
U.T. Austin TRIPS EDGE [Operand Network]	2005	25 ports [25 execution unit tiles]	2-D mesh 5 x 5	110 bits	5.86 GBps [533 MHz clk scaled by 80%]	XY DOR; distributed RR arbitration; wormhole	7 layers; on/off flow control; no VCs
U.T. Austin TRIPS EDGE [On-Chip Network]	2005	40 ports [16 L2 tiles + 24 network interface tile]	2-D mesh 10 x 4	128 bits	6.8 GBps [533 MHz clk scaled by 80%]	XY DOR; distributed RR arbitration; VCT switched	7 layers; credit-based flow control; 4 VCs
Sony, IBM, Toshiba Cell BE [Element Interconnect Bus]	2005	12 ports [1 PPE and 8 SPEs + 3 other ports for memory, I/&O interface]	Ring 4 total, 2 in each direction	128 bits data (+16 bits tag)	25.6 GBps [1.6 GHz, clocked at half the proc speed]	Shortest-path; tree-based RR arb. (centralized); pipelined circuit switch	8 layers; credit-based flow control; no VCs
Sun UltraSPARC T1 processor	2005	Up to 13 ports [8 PE cores + 4 L2 banks + 1 shared I/O]	Crossbar	128 b both for the 8 cores and the 4 L2 banks	19.2 GBps [1.2 GHz, clocked at proc speed]	Shortest-path; age-based arbitration; VCT switched	9 layers; handshaking; no VCs

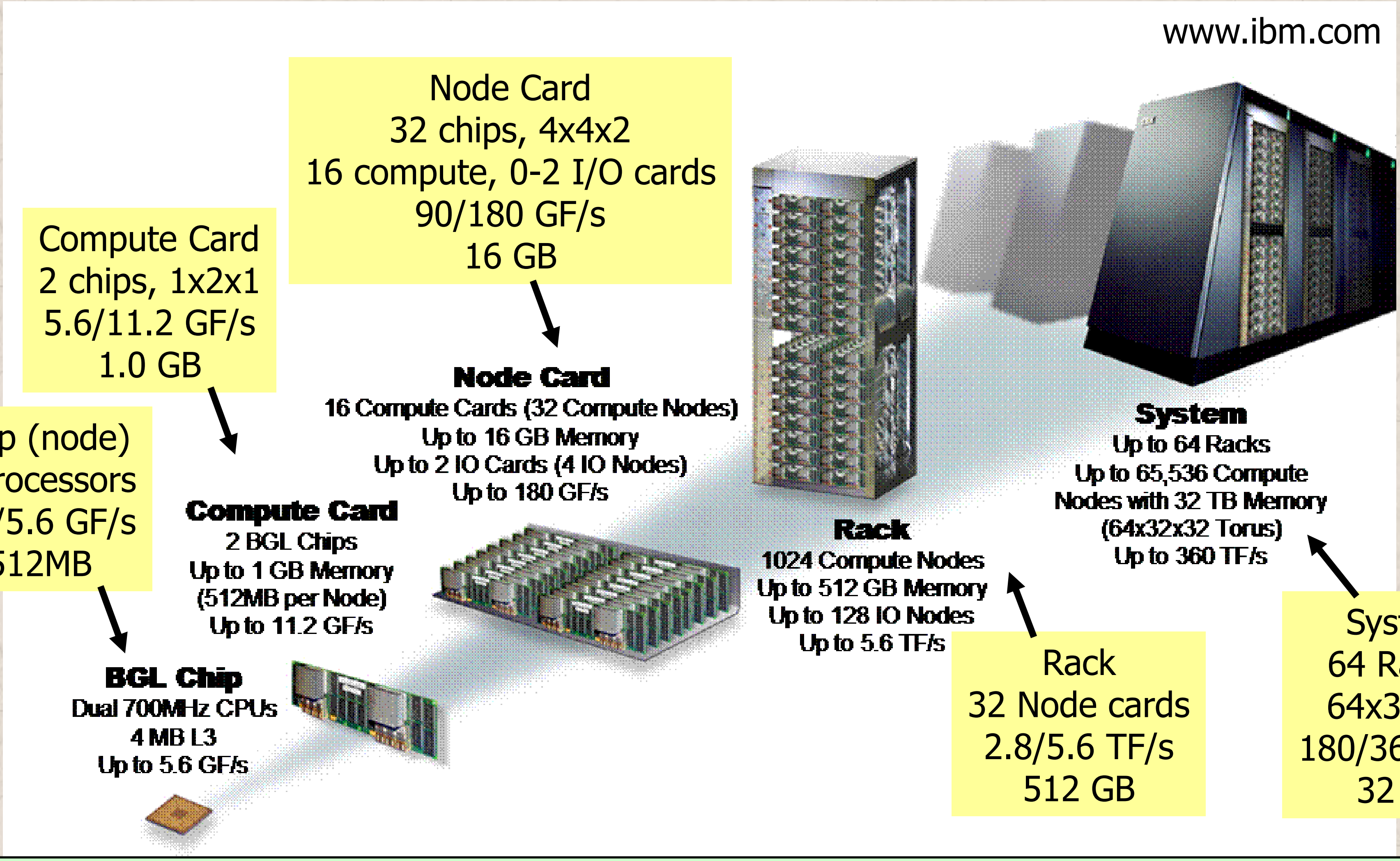
Examples of Interconnection Networks

Blue Gene/L 3D Torus Network

- 360 TFLOPS (peak)
- 2,500 square feet
- Connects 65,536 dual-processor nodes and 1,024 I/O nodes
  - One processor for computation; other meant for communication

Examples of Interconnection Networks

Blue Gene/L 3D Torus Network



Chip (node)
2 processors
2.8/5.6 GF/s
512MB

Compute Card
2 chips, 1x2x1
5.6/11.2 GF/s
1.0 GB

Node Card
32 chips, 4x4x2
16 compute, 0-2 I/O cards
90/180 GF/s
16 GB

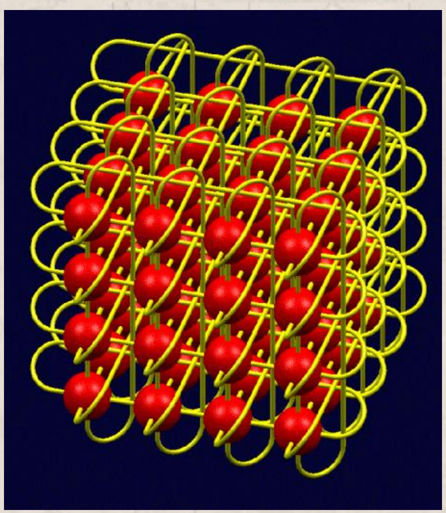
Rack
32 Node cards
2.8/5.6 TF/s
512 GB

System
64 Racks,
64x32x32
180/360 TF/s
32 TB

Node distribution: Two nodes on a 2 x 1 x 1 compute card, 16 compute cards + 2 I/O cards on a 4 x 4 x 2 node board, 16 node boards on an 8 x 8 x 8 midplane, 2 midplanes on a 1,024 node rack, 8.6 meters maximum physical link length

Examples of Interconnection Networks

Blue Gene/L 3D Torus Network



- Main network: 32 x 32 x 64 3-D torus
  - Each node connects to six other nodes
  - Full routing in hardware
- Links and Bandwidth
  - 12 bit-serial links per node (6 in, 6 out)
  - Torus clock speed runs at 1/4th of processor rate
  - Each link is 1.4 Gb/s at target 700-MHz clock rate (175 MB/s)
  - High internal switch connectivity to keep all links busy
    - External switch input links: 6 at 175 MB/s each (1,050 MB/s aggregate)
    - External switch output links: 6 at 175 MB/s each (1,050 MB/s aggregate)
    - Internal datapath crossbar input links: 12 at 175 MB/s each
    - Internal datapath crossbar output links: 6 at 175 MB/s each
    - Switch injection links: 7 at 175 MBps each (2 cores, each with 4 FIFOs)
    - Switch reception links: 12 at 175 MBps each (2 cores, each with 7 FIFOs)

Examples of Interconnection Networks

Blue Gene/L 3D Torus Network

- Routing
  - Fully-adaptive deadlock-free routing based on bubble flow control and Duato’s Protocol
    - DOR and bubble mechanism are used for escape path
  - Hint (direction) bits at the header
    - “100100” indicates the packet must be forwarded in X+ and Y-
    - Neighbor coordinate registers at each node
      - A node cancels hint bit for next hop based on these registers
  - A bit in the header allows for broadcast
  - Dead nodes or links avoided with appropiate hint bits

Outline

- E.1 Introduction *(Lecture 1)*
- E.2 Interconnecting Two Devices *(Lecture 1)*
- E.3 Interconnecting Many Devices *(Lecture 2)*
- E.4 Network Topology *(Lecture 2)*
- E.5 Network Routing, Arbitration, and Switching *(Lecture 3)*
- E.6 Switch Microarchitecture *(Lecture 4)*
- E.7 Practical Issues for Commercial Interconnection Networks *(Lec 4)*
- E.8 Examples of Interconnection Networks *(Lecture 5)*
- E.9 Internetworking (skipped)
- E.10 Crosscutting Issues for Interconnection Networks (skipped)
- **E.11 Fallacies and Pitfalls** *(Lecture 5)*
  - **Fallacies**
  - **Pitfalls**
- E.12 Concluding Remarks and References *(Lecture 5)*

Fallacies and Pitfalls

Fallacies

- The interconnection network is very fast and does not need to be improved
- Bisection bandwidth is an accurate cost constraint of a network
- Zero-copy protocols do not require copying messages or fragments from one buffer to another
- MINs are more cost-effective than direct networks
- Direct networks are more performance-effective than MINs

Fallacies and Pitfalls

Fallacies

- Low-dimensional direct networks achieve higher performance than high-dimensional networks such as hypercubes
- Wormhole switching achieves better performance than other switching techniques
- Implementing a few virtual channels always increases throughput by allowing packets to pass through blocked packets ahead
- Adaptive routing causes out-of-order packet delivery, thus introducing too much overhead to re-order packets
- Adaptive routing by itself is sufficient to tolerate network faults

Fallacies and Pitfalls

Pitfalls

- Using bandwidth (in particular, bisection bandwidth) as the *only* measure of network performance

- Not providing sufficient reception link bandwidth

- Using high-performance NICs, but forgetting the I/O subsystem

- Ignoring software overhead when determining performance

- Providing features only within the network versus end-to-end

Outline

- E.1 Introduction *(Lecture 1)*
- E.2 Interconnecting Two Devices *(Lecture 1)*
- E.3 Interconnecting Many Devices *(Lecture 2)*
- E.4 Network Topology *(Lecture 3)*
- E.5 Network Routing, Arbitration, and Switching *(Lecture 3)*
- E.6 Switch Microarchitecture *(Lecture 4)*
- E.7 Practical Issues for Commercial Interconnection Networks *(Lec 4)*
- E.8 Examples of Interconnection Networks *(Lecture 5)*
- E.9 Internetworking (skipped)
- E.10 Crosscutting Issues for Interconnection Networks (skipped)
- E.11 Fallacies and Pitfalls *(Lecture 5)*
- **E.12 Concluding Remarks and References** *(Lecture 5)*

Concluding Remarks and References

Concluding Remarks

- Interconnect design is an exciting area of computer architecture
  - on-chip networks between cores on within a chip
  - off-chip networks between chips and boards within a system
  - external networks between systems
- Interconnection networks should be designed to transfer the maximum amount of information within the least amount of time (and cost, power constraints) so as not to bottleneck the system
- The design of interconnection networks is end-to-end
  - injection links/interface, network fabric, reception links/interface
  - topology, routing, arbitration, switching, and flow control are among key concepts in realizing high-performance designs
  - *a simple, general throughput model can be used to guide design*
- Improving the performance of interconnection networks is critical to advancing our information- and communication-centric world