Interconnection Networks

High Performance Computing / *Computer Architecture*

Tor Skeie
University of Oslo and Simula Research Laboratory

With major presentation contribution from:

José Flich, Universidad Politécnica de Valencia
Timothy Mark Pinkston, University of Southern California



UiO : University of Oslo

Introduction

Reasons to devote attention to interconnection networks – "interconnects"

- They provide connectivity within a single computer system at many levels, and between aggregated computers to offer High Performance Computing (HPC)

- *Trends:* The convergence going on between AI and HPC requires smarter networks and ultra-high connectivity – application trends: Big Data, AI/ML, Internet of Things, Life Science …
  - increased computing power and storage capacity
  - switched networks are replacing buses
- *Computer architects must understand interconnect problems and solutions in order to more effectively design and evaluate systems*

Goal

- *To provide an overview of designing interconnection networks and potential solutions*

Outline

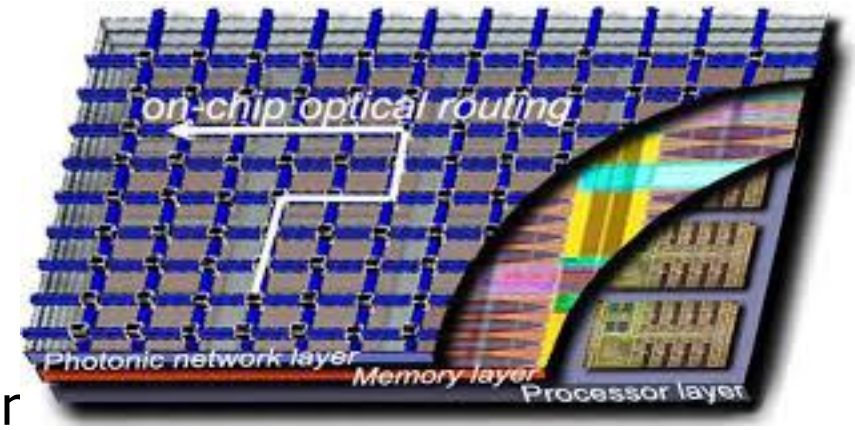
- Introduction
- Interconnecting Two Devices
- Interconnecting Many Devices
- Network Topology
- Network Routing
- Concluding Remarks and References

Introduction

Interconnection Network Domains

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

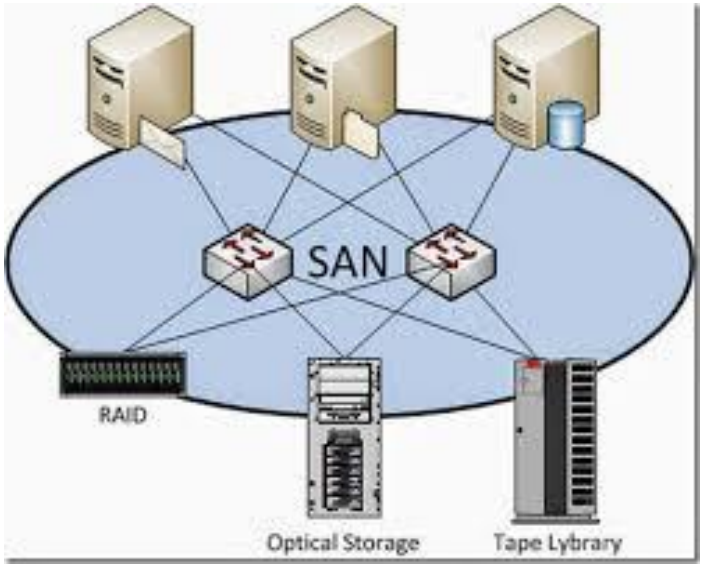
- *On-chip networks* (*OCNs*), a.k.a., network-on-chip (NoC)
  - ➢ Interconnect microarchitecture functional units, register files, caches, compute tiles, processor GPU and CPU cores
  - ➢ Chip or multichip modules
  - ➢ Tens to 100s of devices interconnected
  - ➢ Maximum interconnect distance on the order of centimetres
  - ➢ Examples (custom designed)
    - ✓ Element Interconnect Bus (Cell Broadbar Engine processor chip) 2,400 Gbps (3.2 Ghz processor clock), 12 elements on the chip



Introduction

Interconnection Network Domains

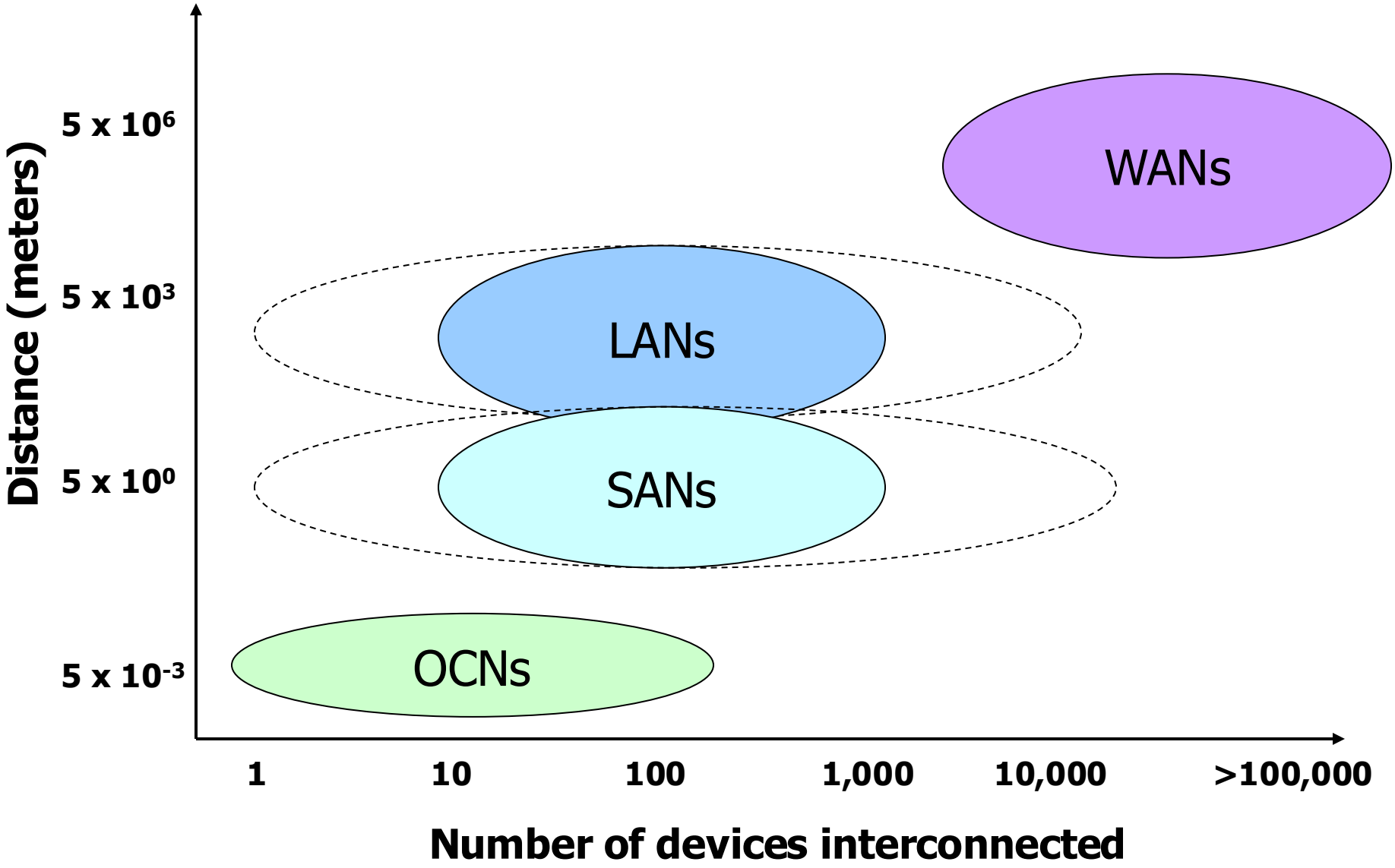
- *System/storage area networks (SANs), but also called off-chip networks*
  - Multiprocessor and multicomputer systems
    - Is about aggregating computing power *in a way that can deliver extreme computing (HPC)*
    - Interprocessor and processor-memory interconnections
  - Data center and AI factory environments
    - CPU, GPU, Storage, and I/O components
  - Hundreds to thousands of devices interconnected
    - El Capitan, located at Lawrence Livermore National Laboratory (USA), is the world´s most powerful computer right now
    - consisting of 11,039,616 CPU and GPU cores in total. The interconnect is HPE´s Slingshot, Capacity: 1.742 ExaFlops. Power: 30 MW
  - 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, Slingshot, CXL, Omni-Path, BXI, Data Centre Ethernet, Ultra Ethernet is also emerging





Introduction

Interconnection Network Domains



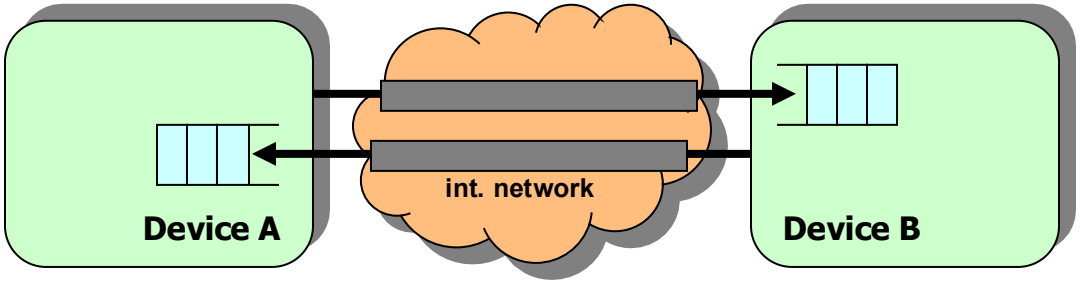
Outline

- Introduction
- **Interconnecting Two Devices** *(Now)*
  - **Network Interface Card Functions**
  - **Basic Network Structure and Functions**
  - **Characterizing Performance: Latency & Effective Bandwidth**
- Interconnecting Many Devices
- Network Topology
- Network Routing
- Concluding Remarks and References

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 communication



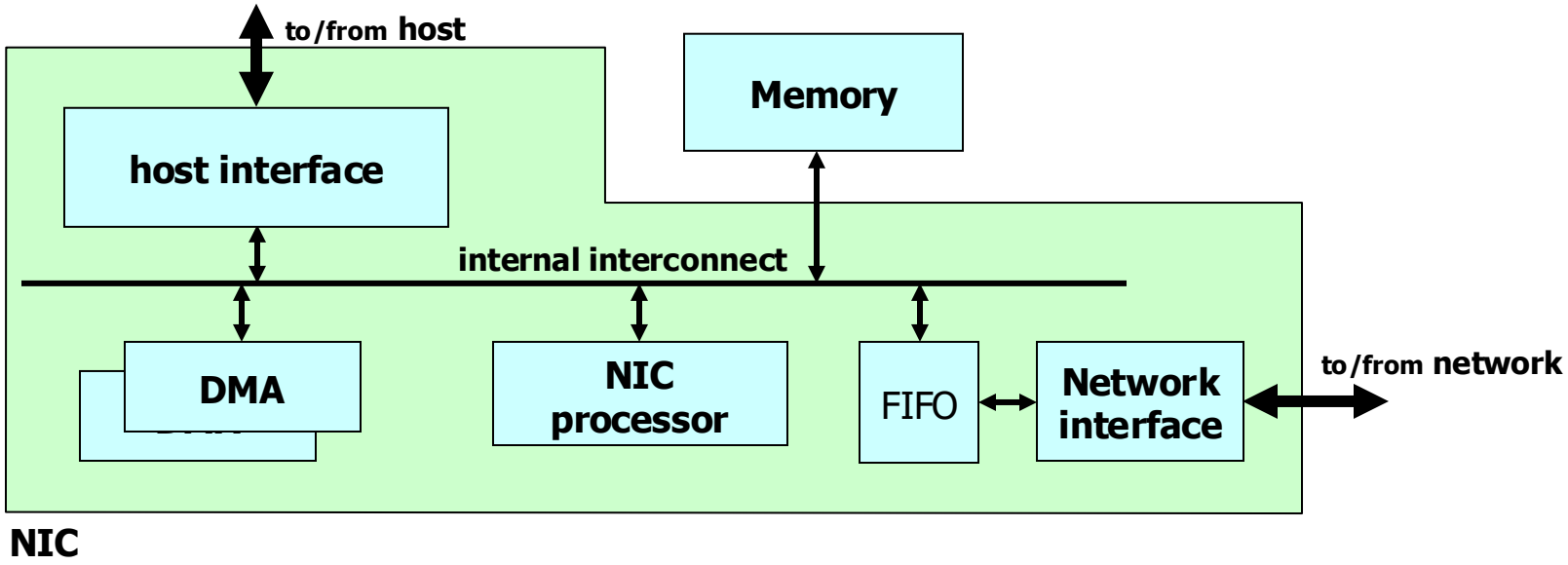
Interconnecting Two Devices

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



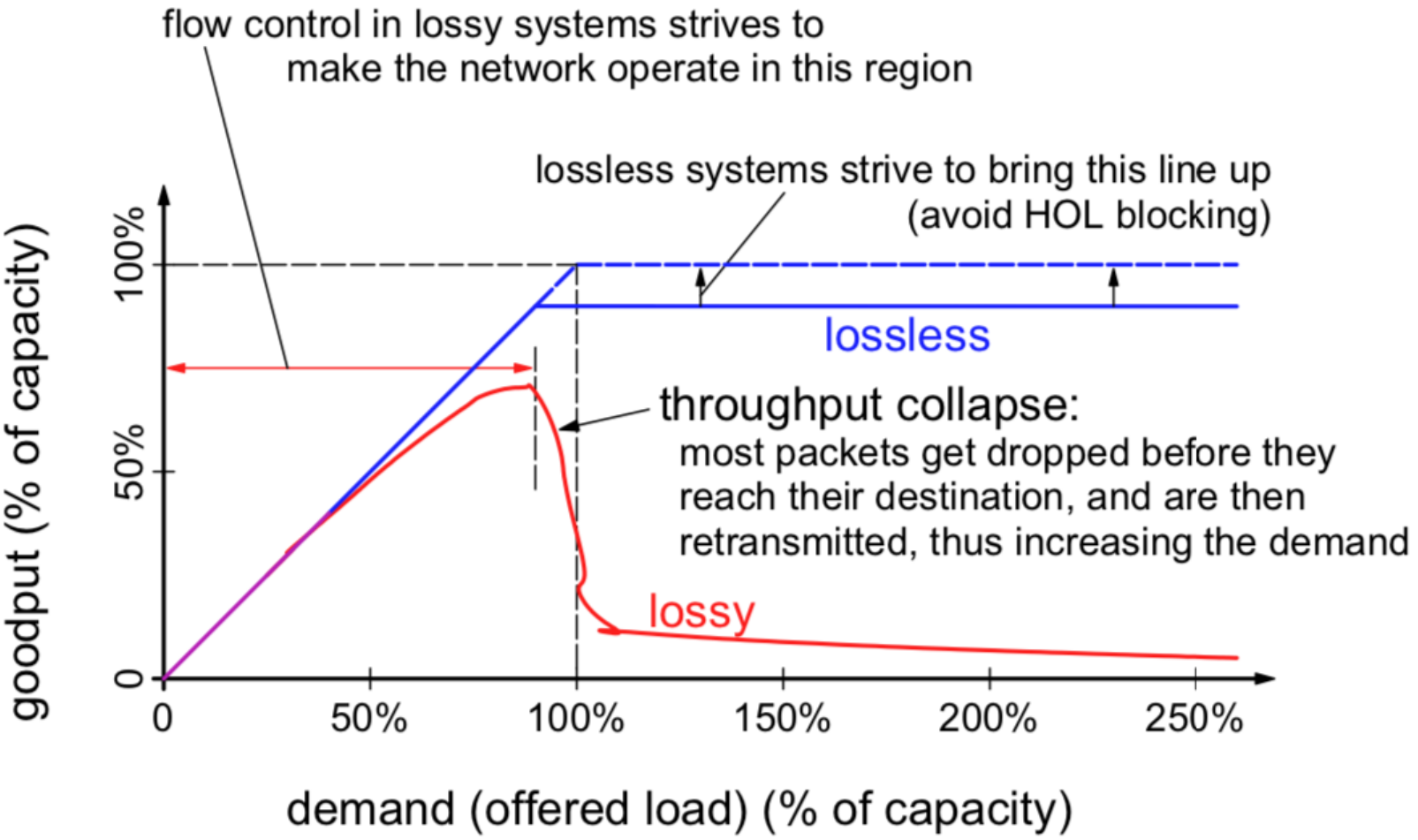
Interconnecting Two Devices

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 network domain dictates which is one used
      - OCN, SAN: typically lossless; LAN, WAN: typically lossy

Interconnecting two devices

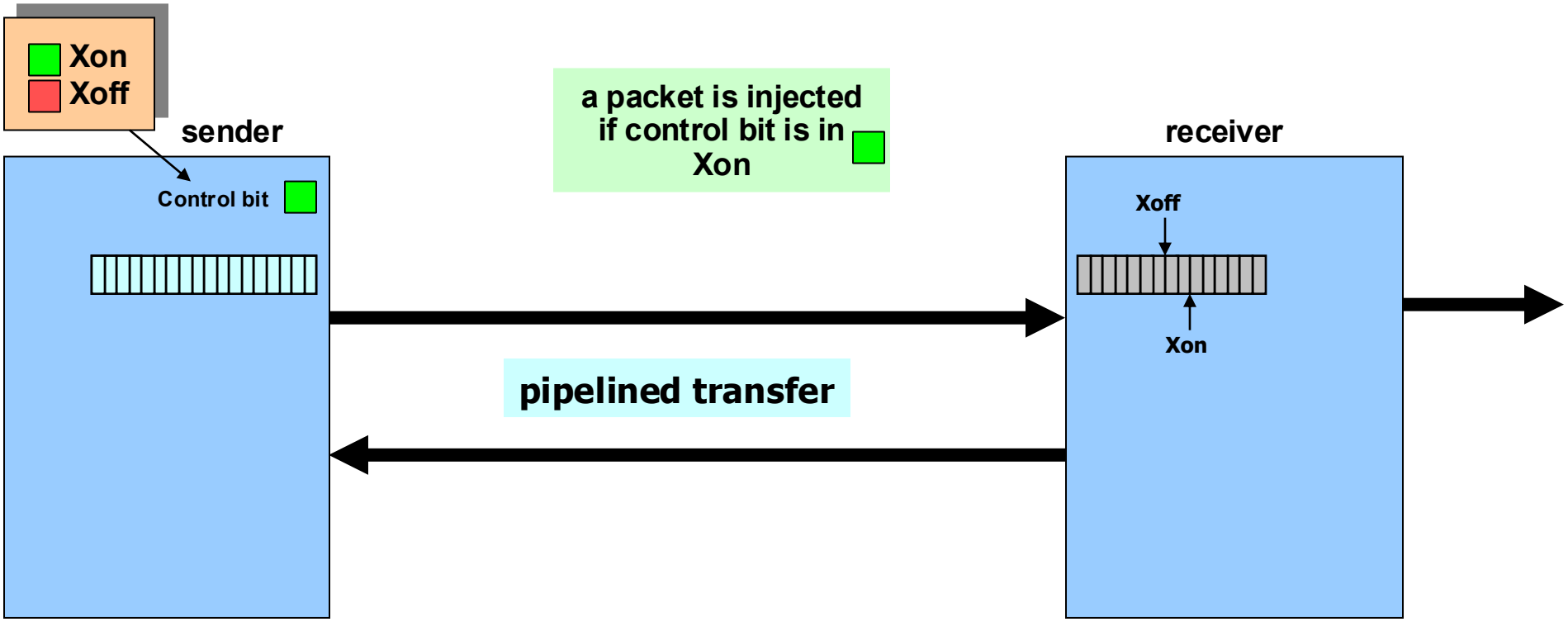
Goodput versus Demand in Lossy and Lossless Flow Control



Interconnecting Two Devices

Basic Network Structure and Functions

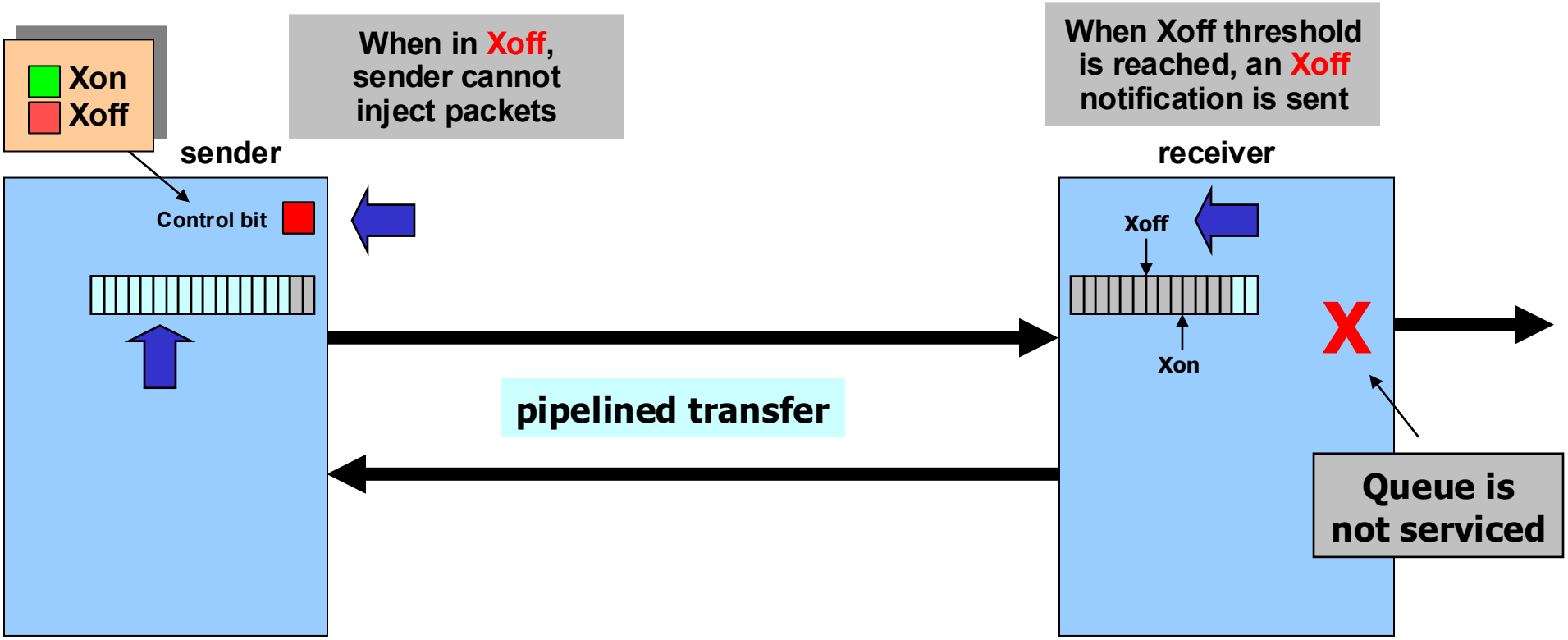
- Xon/Xoff flow control



Interconnecting Two Devices

Basic Network Structure and Functions

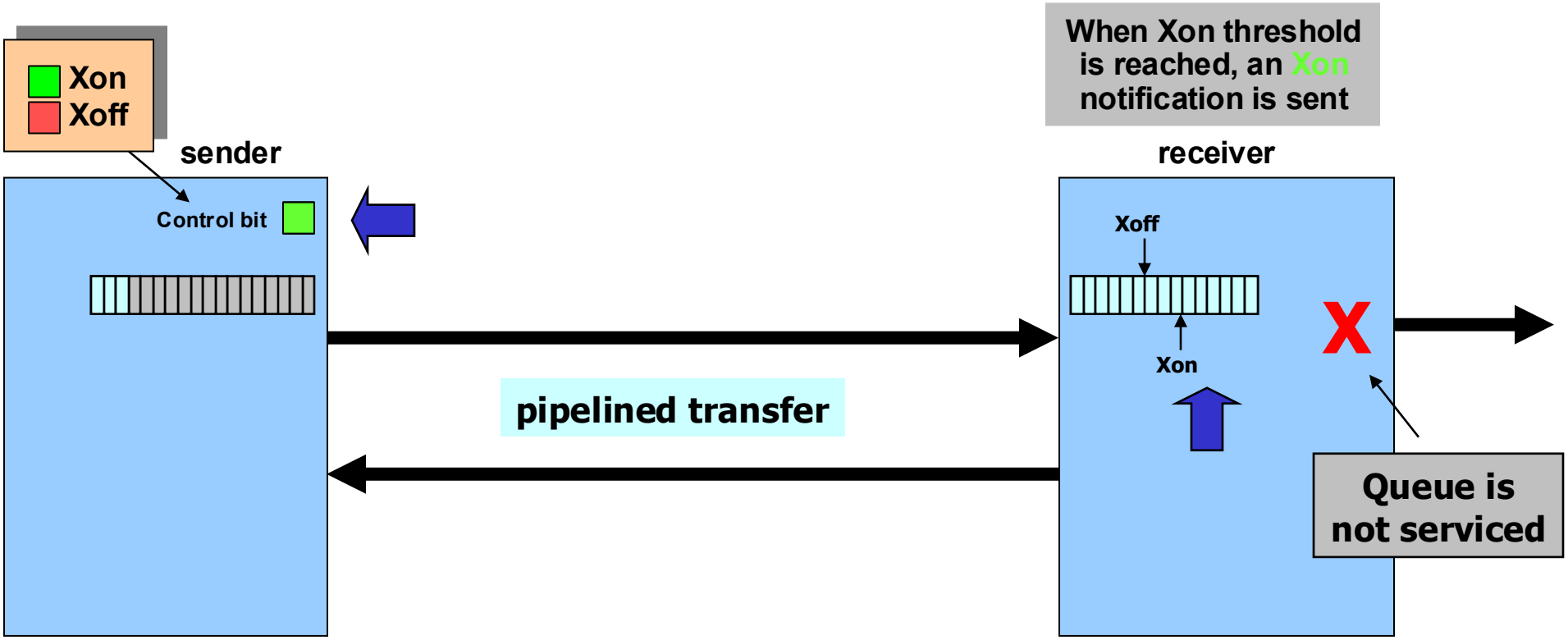
- Xon/Xoff flow control



Interconnecting Two Devices

Basic Network Structure and Functions

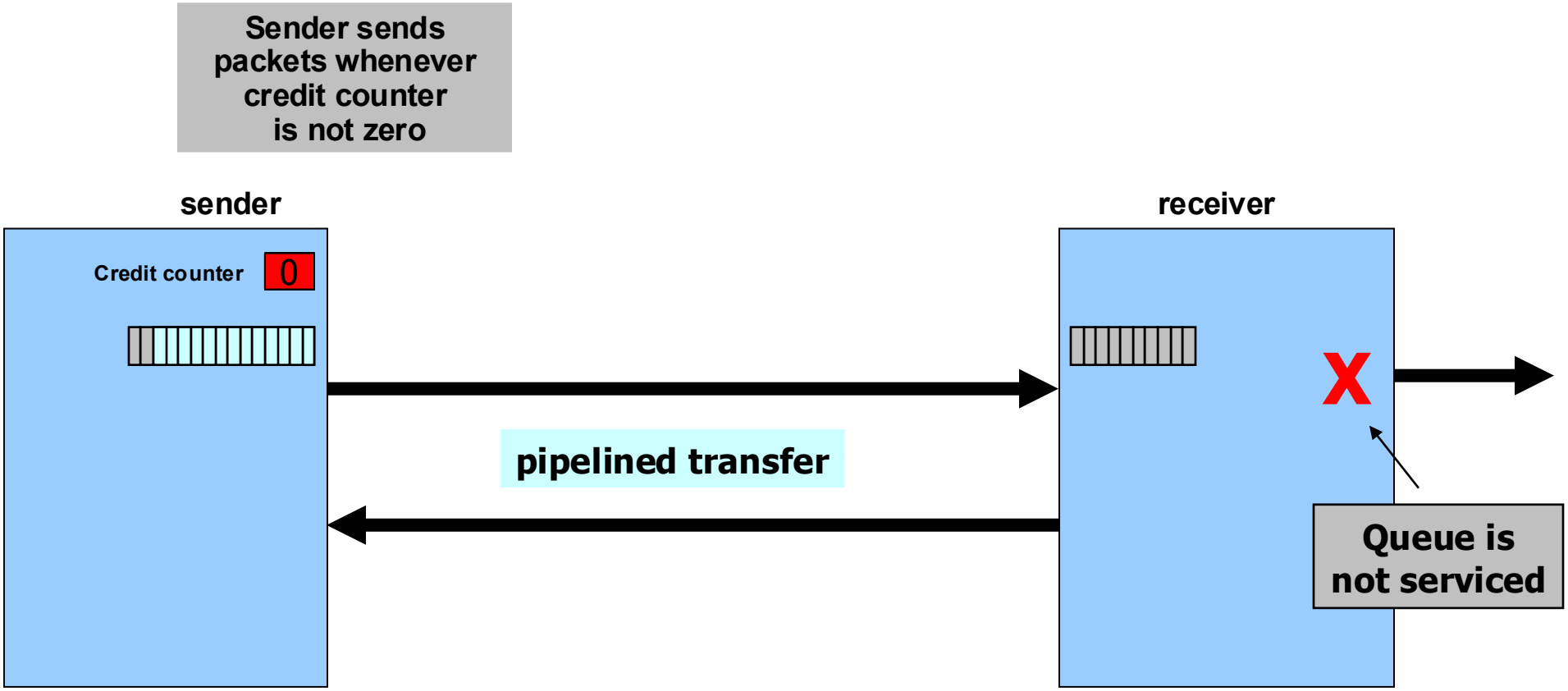
- Xon/Xoff flow control



Interconnecting Two Devices

Basic Network Structure and Functions

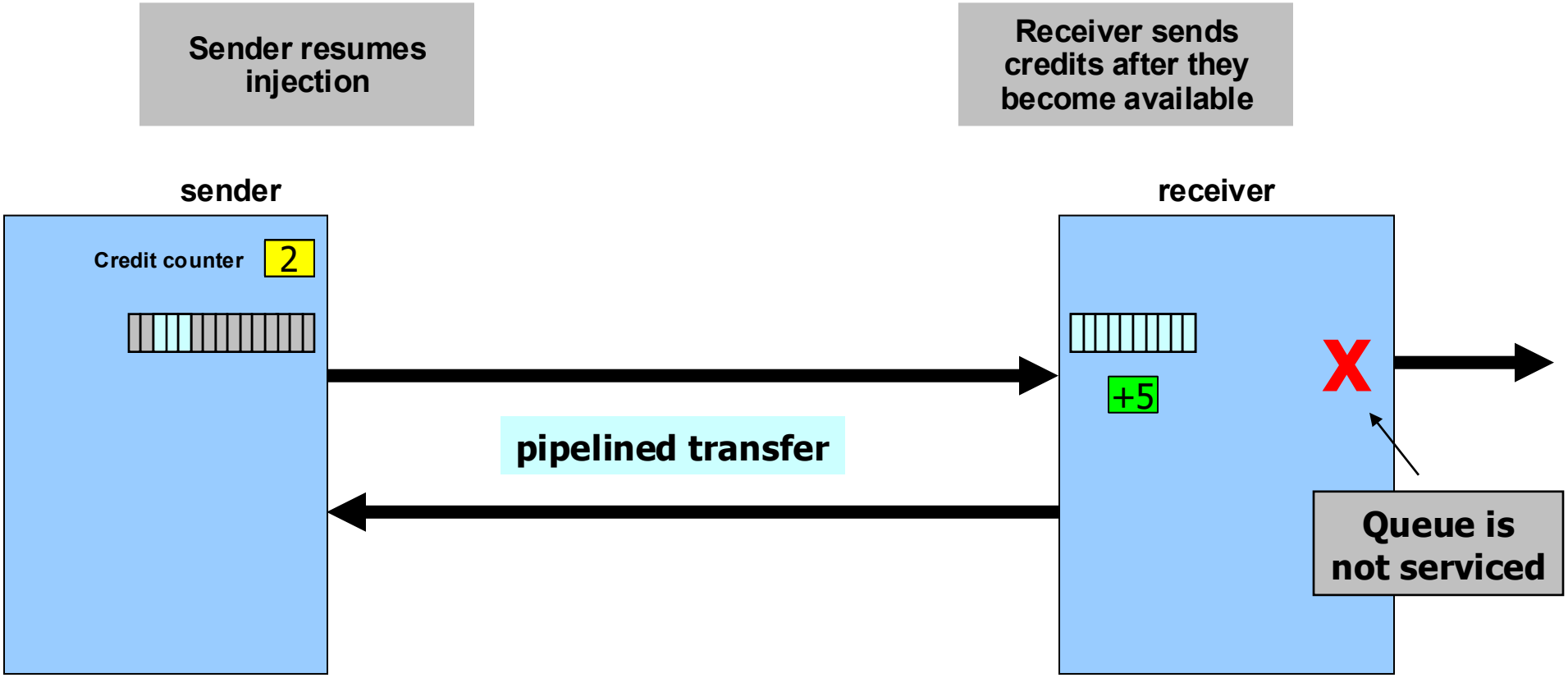
- Credit-based flow control



Interconnecting Two Devices

Basic Network Structure and Functions

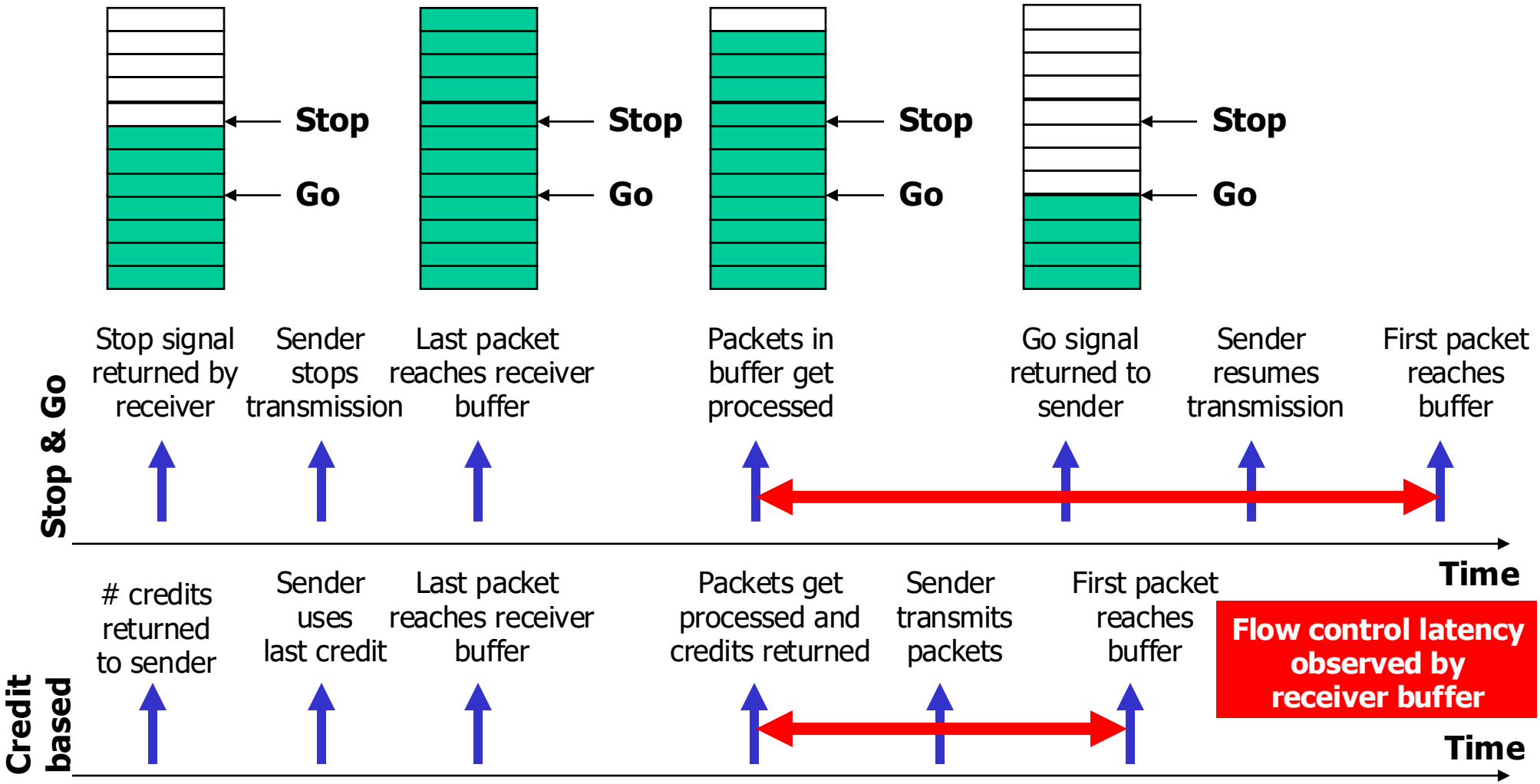
- Credit-based flow control



Interconnecting Two Devices

Basic Network Structure and Functions

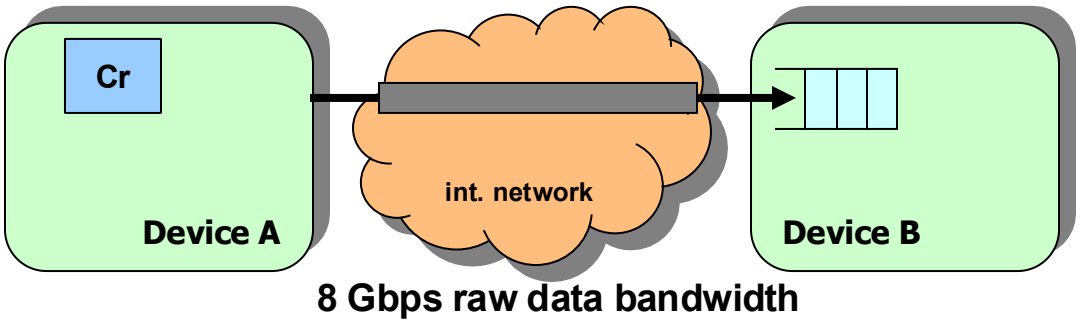
- Comparison of Xon/Xoff vs credit-based flow control



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

Basic Network Structure and Functions

- Example: impact of different link lengths for credit-based flow control
  - Calculate the minimum amount of credits and buffer space for interconnect distances of *1 cm*, *1 m*, *100 m*, and *10 km*
  - Assume a dedicated-link network with
    - 8 Gbps (raw) data bandwidth per link (each direction)
  - Device A continuously sends 100-byte packets (header included)
  - Consider only the link propagation delay (no other delays or overheads), and that the speed of the medium is 2/3 x 300 000 km/sec



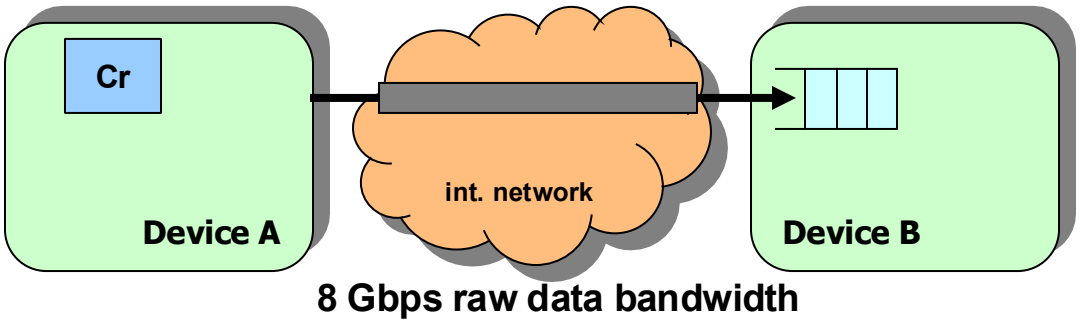
Basic Network Structure and Functions

- Example: comparison of flow control techniques

$$\text{Packet propagation delay} + \text{Credit propagation delay} \leq \frac{\text{Packet size}}{\text{Bandwidth}} \times \text{Credit count}$$

$$\left(\frac{\text{distance}}{\text{2/3 x 300,000 km/sec}}\right) \times 2 \leq \frac{\text{100 bytes}}{\text{8 Gbits/sec}} \times \text{Credit count}$$

1 cm → 1 credit
1 m → 1 credit
100 m → 10 credits
10 km → 1,000 credits



Outline

- Introduction
- Interconnecting Two Devices
- **Interconnecting Many Devices** (Now)
  - **Additional Network Structure and Functions**
  - **Switched-based Networks**
  - **Comparison of Shared-media versus Switched-based Networks**
  - **Characterizing Performance: Latency & Effective Bandwidth**
- Network Topology
- Network Routing
- Concluding Remarks and References

Interconnecting Many Devices

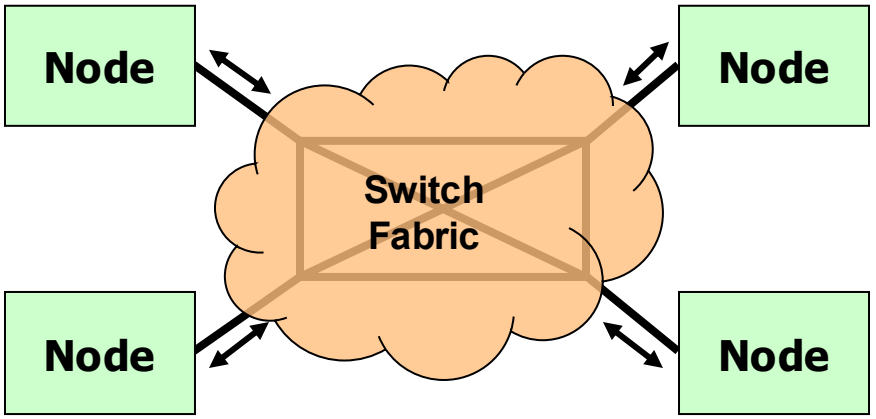
Additional Network Structure and Functions

- *Network topology – being a regular structure*
  - *What paths are possible for packets?*
  - Networks usually share paths among different pairs of devices
- *Routing*
  - *Which of the possible paths are allowable (valid) for packets?*
  - Provides the set of operations needed to compute a valid path
  - A routing decision is taken at each intermediate node, but the routing tables is programmed by centralized controller, aka SDN (Software Defined Networking)
- *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
  - Complexity and applied order depends on the category of the topology: *shared-media* or *switched media*

Interconnecting Many Devices

Switched-based Networks

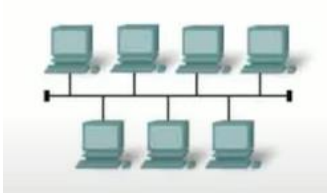
- *Disjoint portions of the media are shared via switching*
- Switch fabric components
  - Passive *point-to-point links*
  - Active *switches*
    - Dynamically establish communication between sets of source-destination pairs
- Aggregate bandwidth can be many times higher than that of shared-media networks

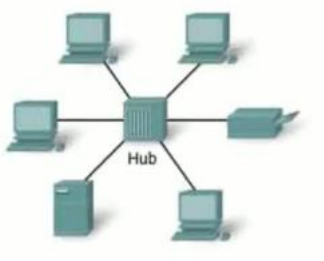


Interconnecting Many Devices

Comparison of Shared-media versus Switched-based 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-based networks
  - Aggregate network bandwidth scales with number of devices
  - Concurrent communication
    - Potentially much higher network effective bandwidth
  - *Beware:* inefficient designs are possible
    - Superlinear network cost but sublinear network effective bandwidth

Characterizing Performance: Latency & Effective Bandwidth

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

lower bound (contention delay not included)

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

upper bound (contention effects not *fully* included)

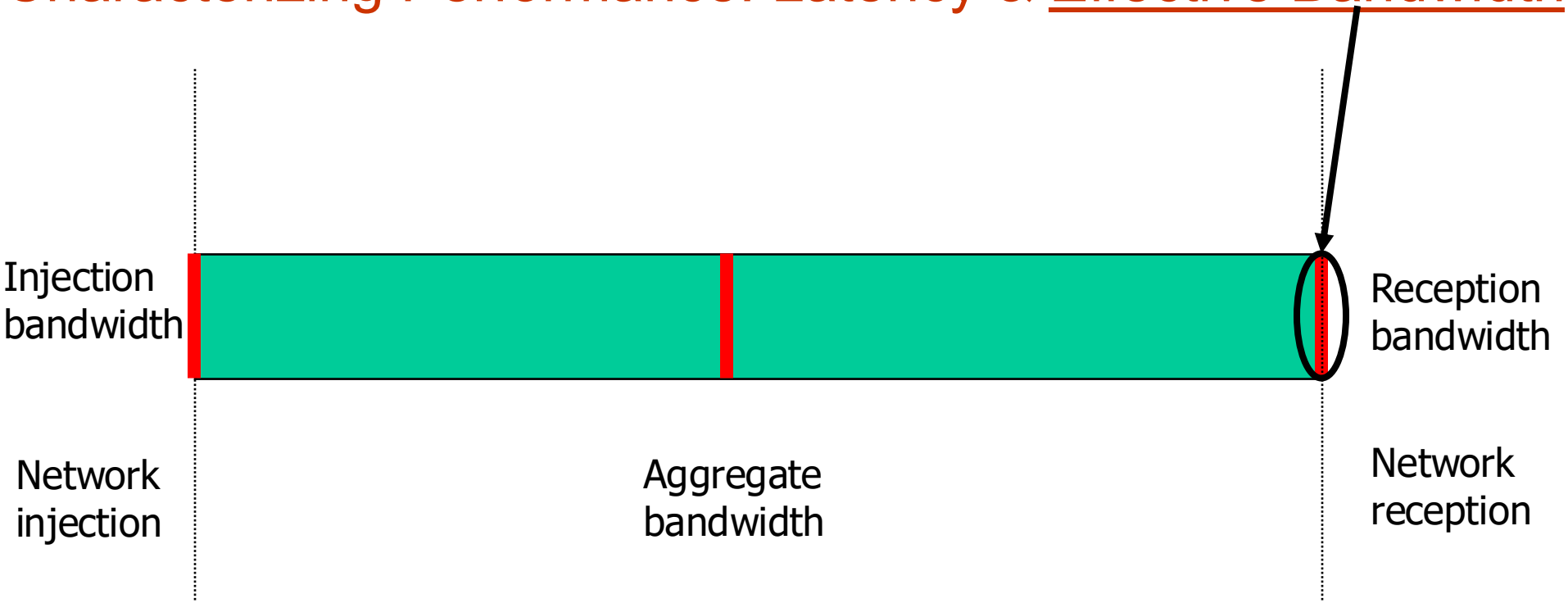
$$\text{Effective bandwidth} = \min(BW_{\text{NetworkInjection}}, BW_{\text{Network}}, \sigma \times BW_{\text{NetworkReception}})$$

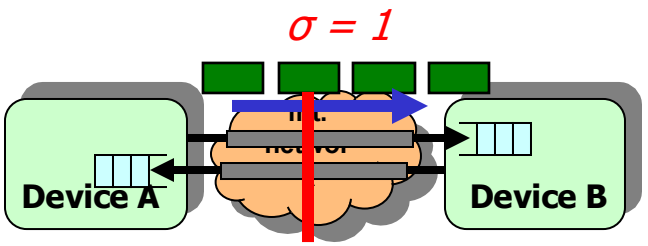
$$= \min(N \times BW_{\text{LinkInjection}}, BW_{\text{Network}}, \sigma \times N \times BW_{\text{LinkReception}})$$

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

Interconnecting Many Devices

Characterizing Performance: Latency & Effective Bandwidth

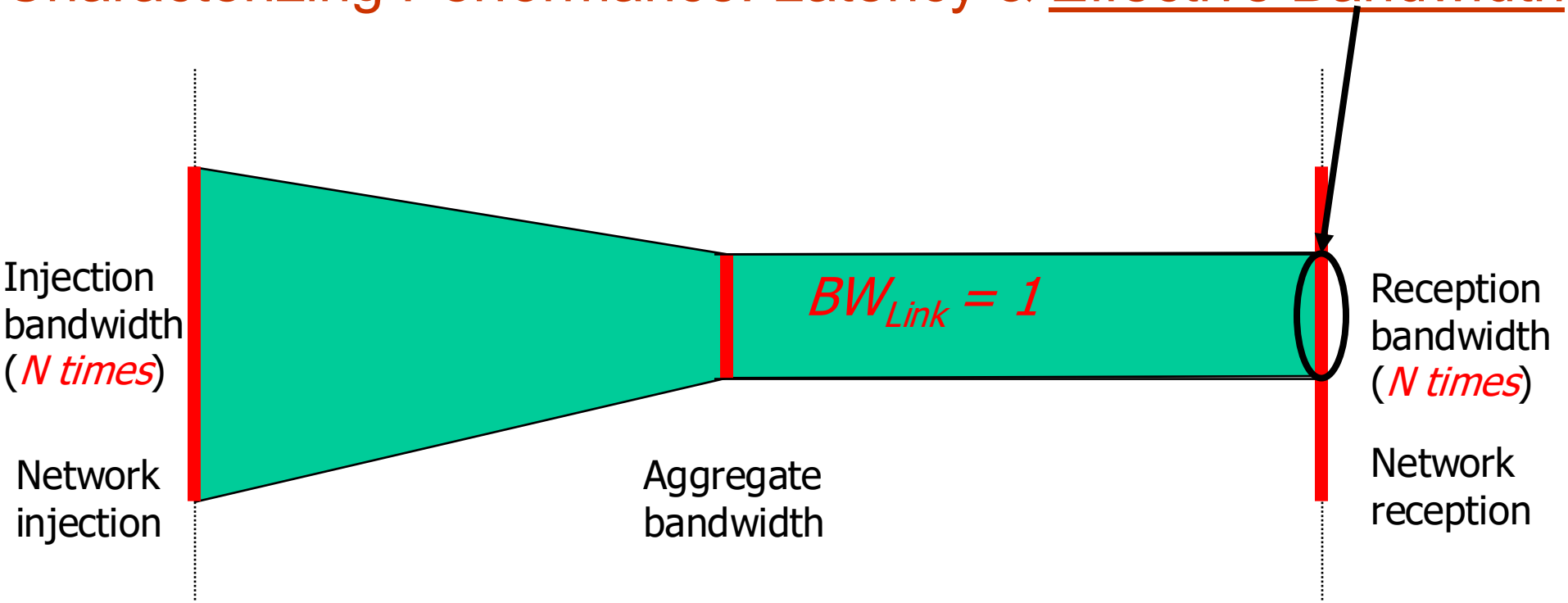




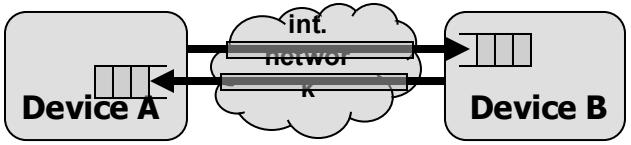
Dedicated-link network

Interconnecting Many Devices

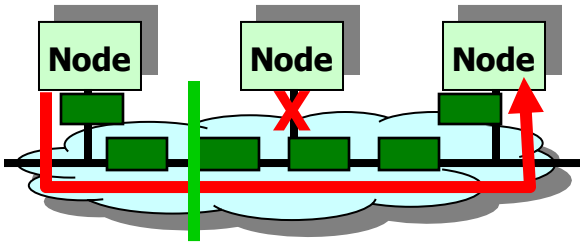
Characterizing Performance: Latency & Effective Bandwidth



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



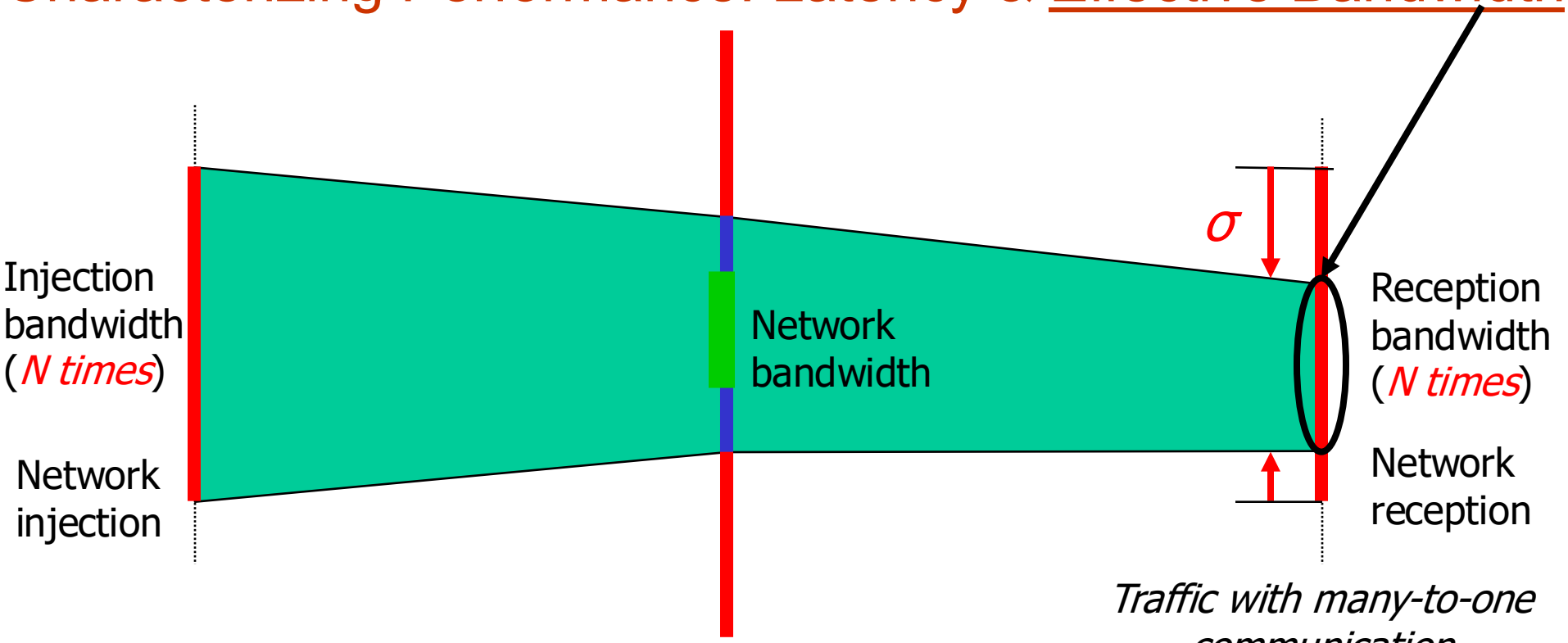
Dedicated-link network



Shared-media network

Interconnecting Many Devices

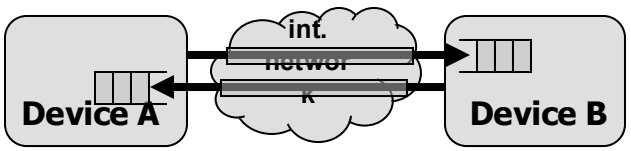
Characterizing Performance: Latency & Effective Bandwidth



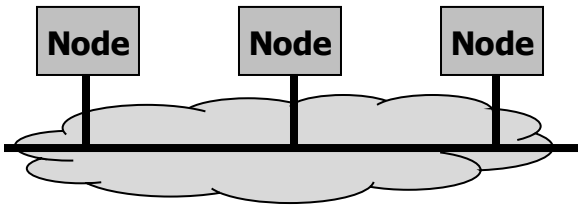
Aggregate bandwidth (≥ *N times*)

Traffic with many-to-one communication

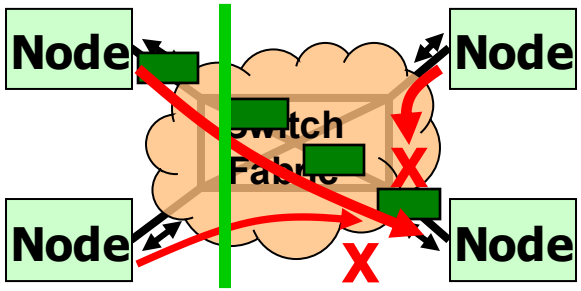
σ < 1



Dedicated-link network



Shared-media network

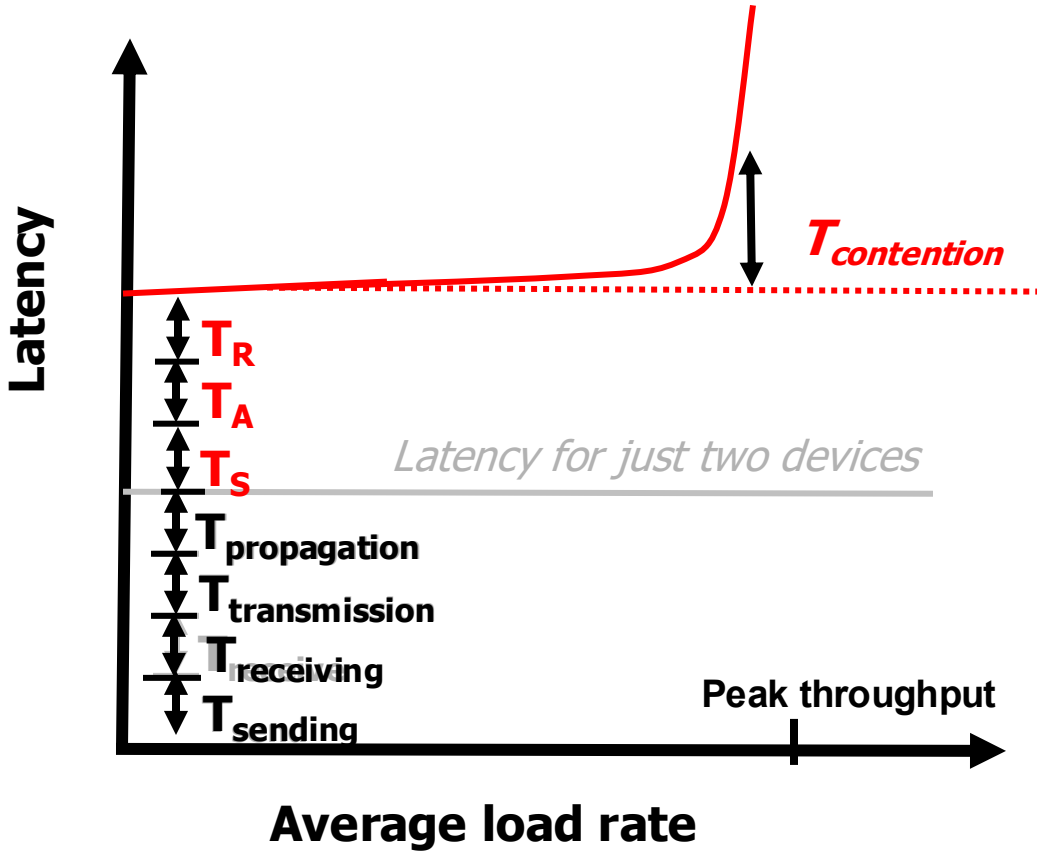


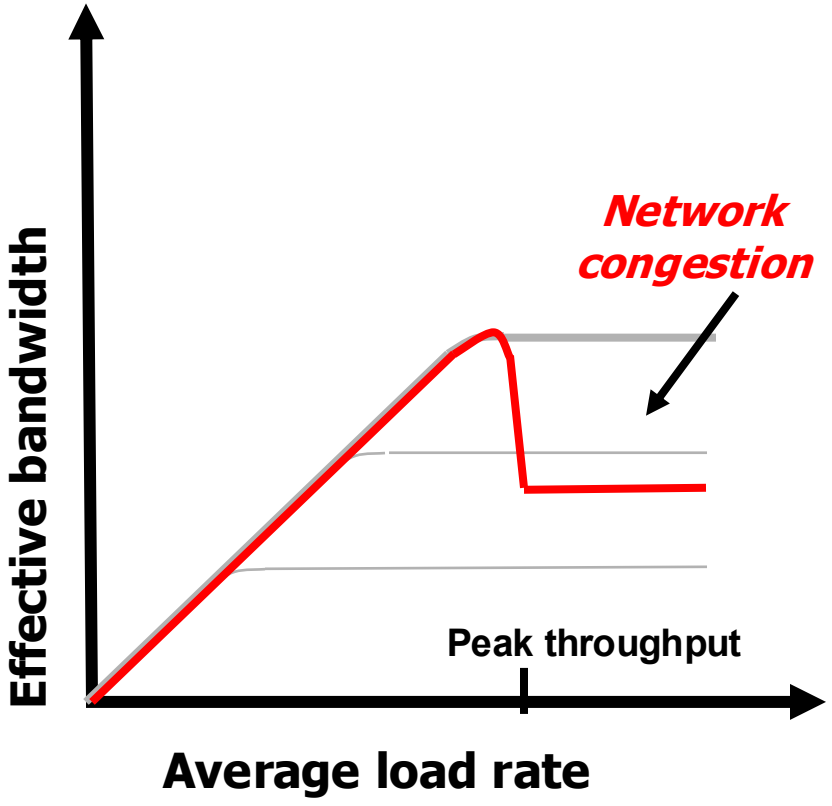
Switched-based network

Interconnecting Many Devices

Characterizing Performance: Latency & Effective Bandwidth

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





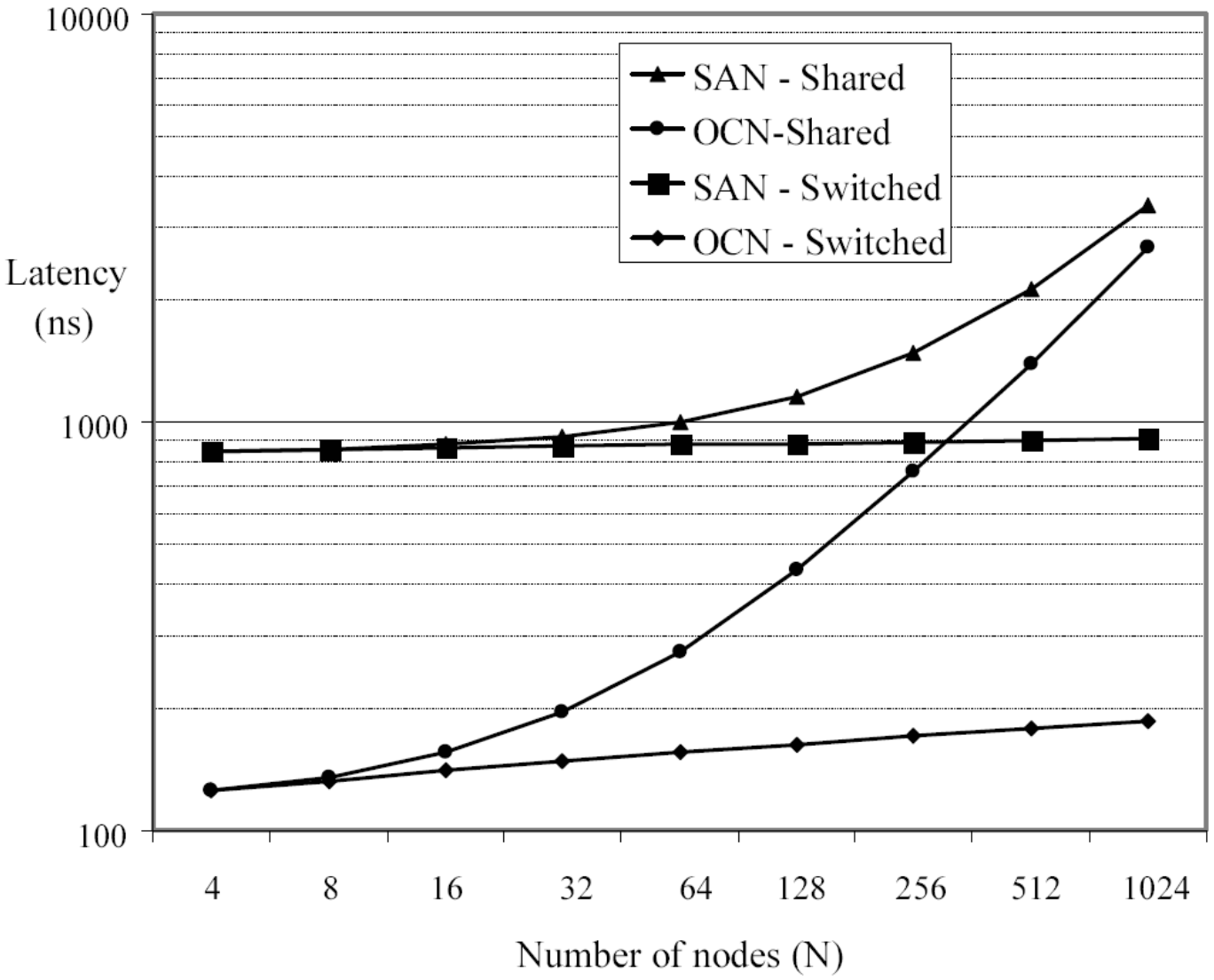
Interconnecting Many Devices

Characterizing Performance: Latency & Effective Bandwidth

- Example: plot the total packet latency
  - Assume *N* scales from 4 to 1024 end nodes and the following:
  - Shared media (one link) and switched-based (*N* links)
  - All network links have data bandwidth of 8 Gbps
  - Unicast transmission of packets of size 100 bytes
  - Overheads
    - sending: x + 0.05 ns/byte; receiving: 4/3(x) + 0.05 ns/byte
    - x is 0 for OCNs and 0.3 µs for SANs
    - Distances: 0.5 cm for OCNs and 5 m SANs
  - Routing, Arbitration, and Switching
    - Shared-media: T_R = 2.5 ns, T_A = 2.5(N) ns, T_S = 2.5 ns
    - Switched-based: $T_R = T_A = T_S$ = 2.5 ($\log_2 N$) ns
  - $\sigma = N^{-1}$ (shared), $\sigma = (\log_2 N)^{-1/4}$ (switched)

Interconnecting Many Devices

Characterizing Performance: Latency & Effective Bandwidth



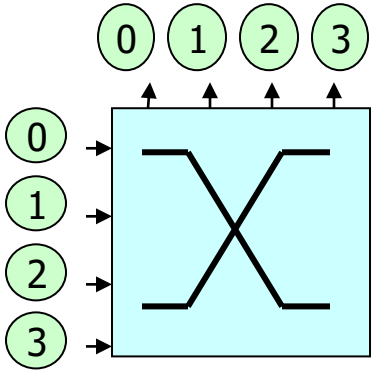
- For OCNs, T_R, T_A, and T_S combine to dominate time of flight delay and are >> than other latency components for all network sizes.
- For SANs, T_R, T_A, and T_S combine to dominate time of flight delay, but are less than other latency components for switched-based (but, not negligibly so)

Outline

- Introduction
- Interconnecting Two Devices
- Interconnecting Many Devices
- **Network Topology** (Now)
  - **Preliminaries and Evolution**
  - **Indirect Networks**
  - **Direct Networks**
  - **Comparison of Indirect and Direct Networks**
- Network Routing
- Concluding Remarks and References

Network Topology

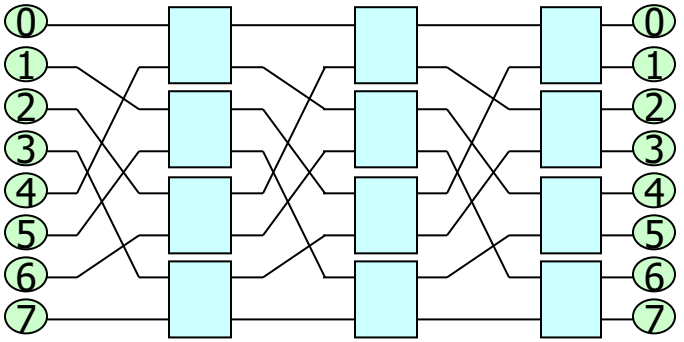
Preliminaries and Evolution



- One switch can only connect a small number of devices
  - Number of switch ports limited by VLSI technology, power consumption, packaging, and other *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*, and not least the routing function …

Network Topology

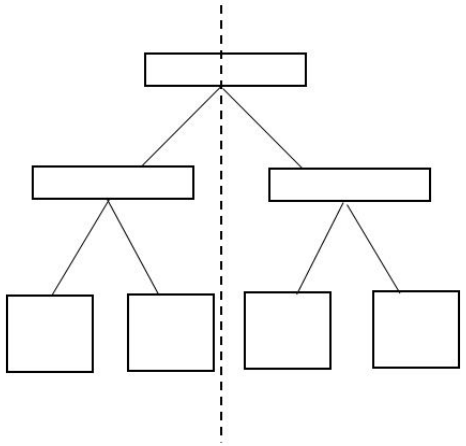
Preliminaries and Evolution



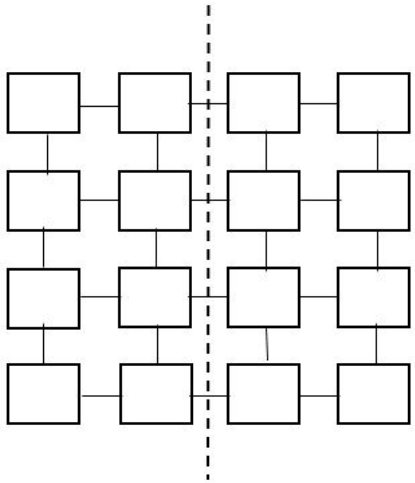
- One switch suffices to connect a small number of devices
  - Number of switch ports limited by VLSI technology, power consumption, packaging, and other such *cost* constraints
- A *fabric* of interconnected switches (i.e., *switch fabric* or *network fabric*) is needed when the number of devices is 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 defines the *performance potential*
- Topology is constrained primarily by local chip/board *pin-outs*, and not least the routing function …

Network Topology

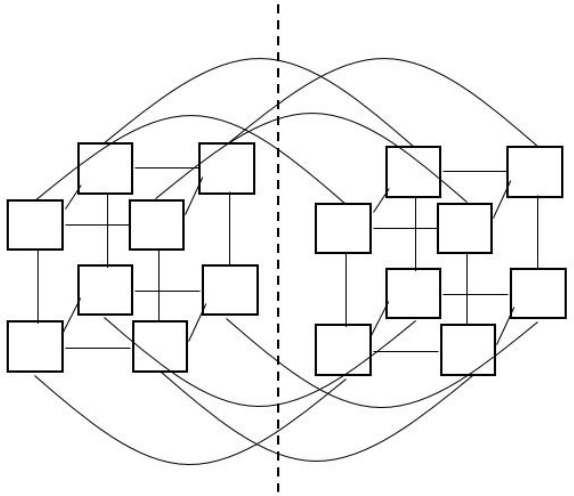
Bisection bandwidth



For tree topology with n nodes can be bisected at the root by «breaking» one link, so **bisection bandwidth is one link**.



For mesh topology with n nodes, $\sqrt{n}$ links should be «broken» to bisect the network, so **bisection bandwidth is $\sqrt{n}$ links**.



For Hyper-cube topology with n nodes, n/2 links should be «broken» to bisect the network, so **bisection bandwidth is bandwidth of n/2 links.**

Network Topology

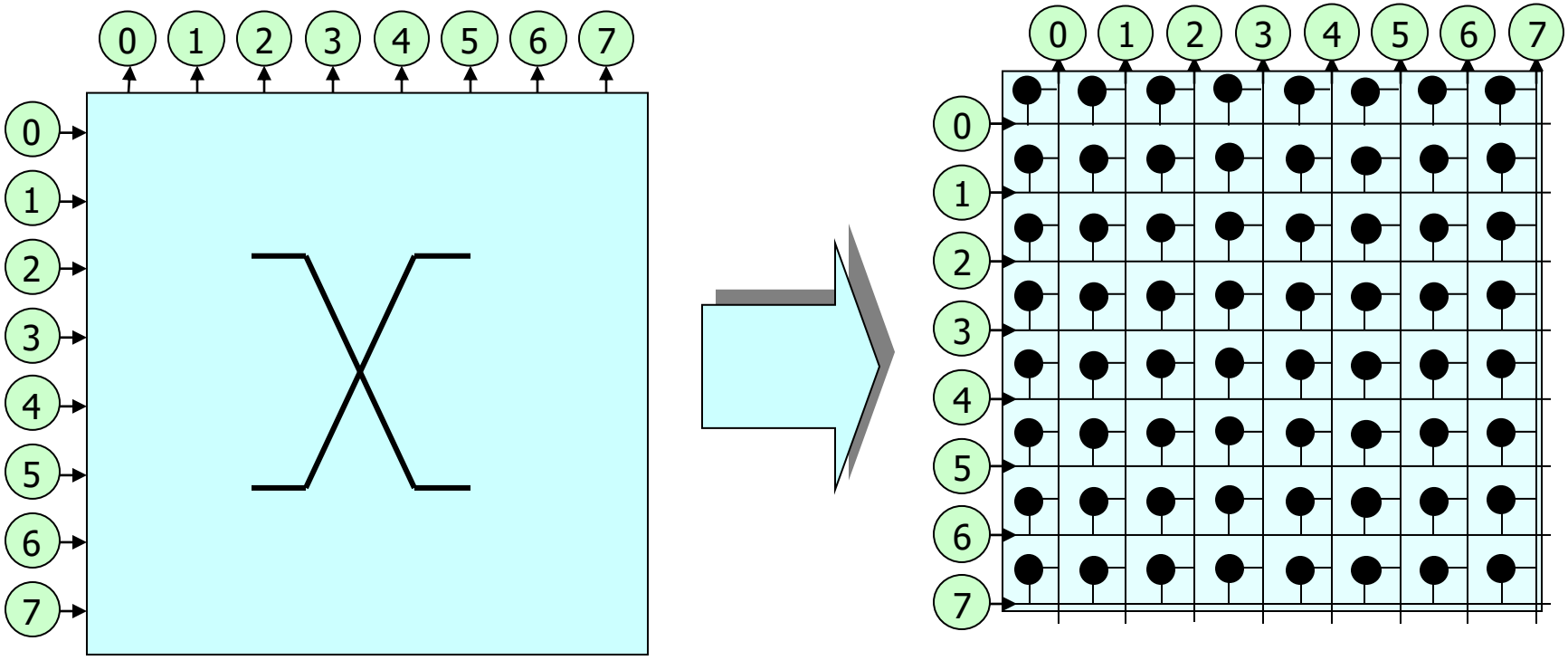
Preliminaries and Evolution

- 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 defines the *performance potential***
- **Topology is constrained primarily by local chip/board *pin-outs*, and not least the routing function …**

Network Topology

Single switch (crossbar) Networks

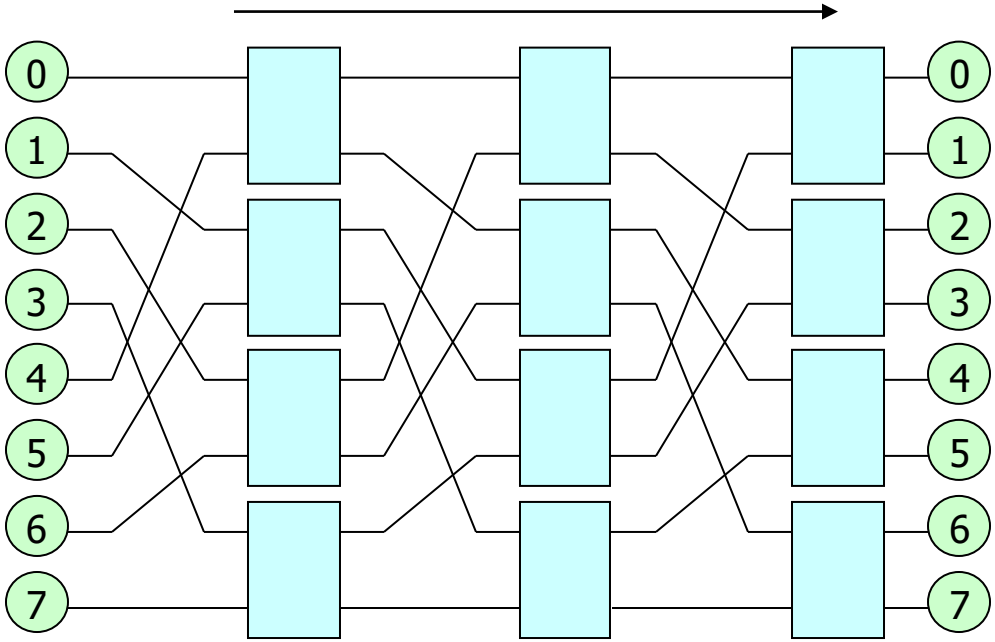
- Crossbar switch complexity increases quadratically with the number of crossbar input/output ports, *N*, i.e., grows as $O(N^2)$
- Has the property of being *non-blocking*



Network Topology

Indirect Networks, a.k.a.

- ***Multistage interconnection networks (MINs)***
  - Network split into several stages consisting of crossbars => *k* x *k* switches
  - Complexity grows as $O(N \times \log_k N)$, where *N* is # of end (compute) nodes, with $\log_k N$ switch stages, each with *N/k* switches, and the total number of switches is $N/k(\log_k N)$
  - Inter-stage connections represented by a set of permutation functions

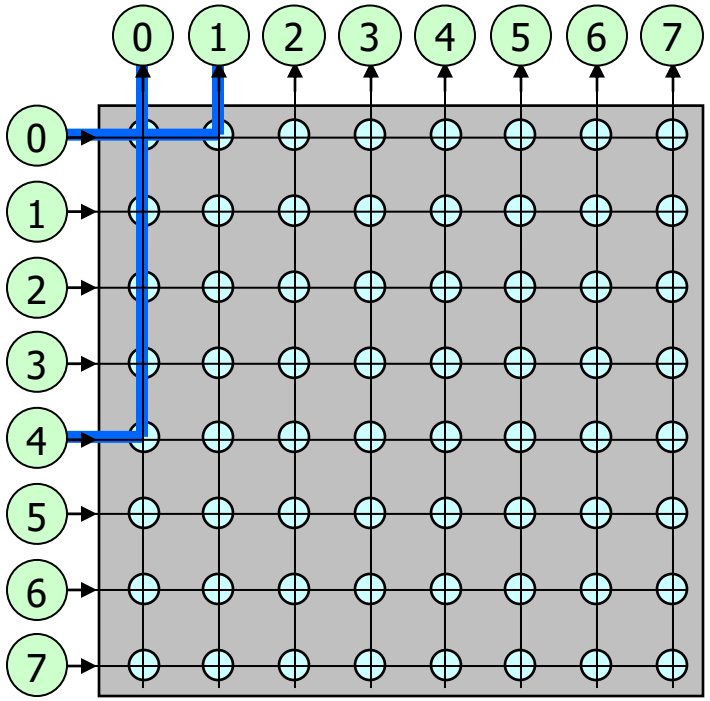


Omega topology, perfect-shuffle exchange

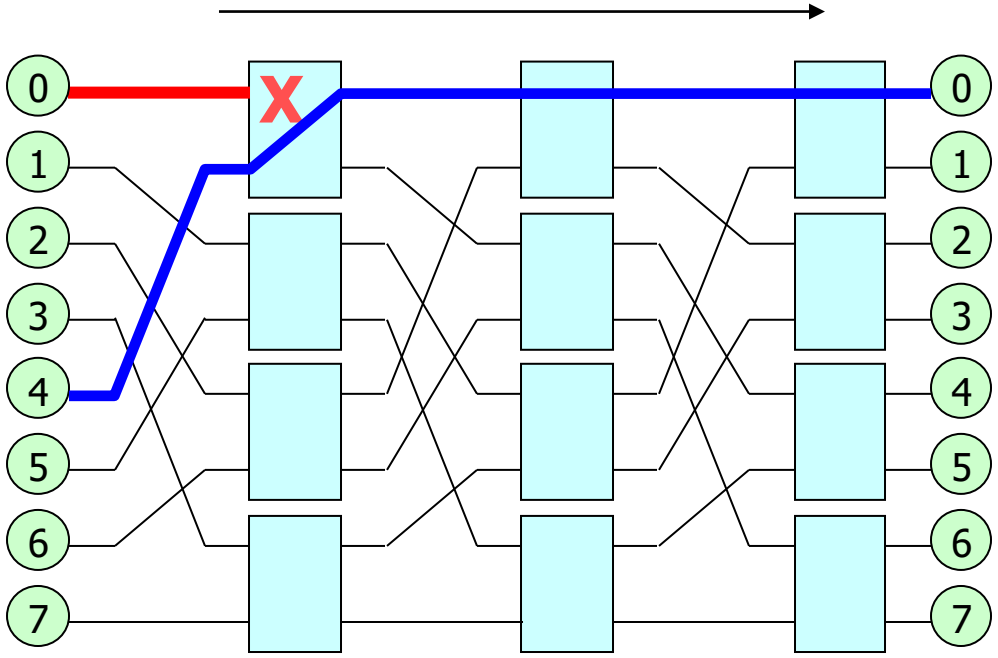
Network Topology

Indirect Networks

- Going from a single switch network to a MIN comes at the price of *performance*
  - MINs have 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



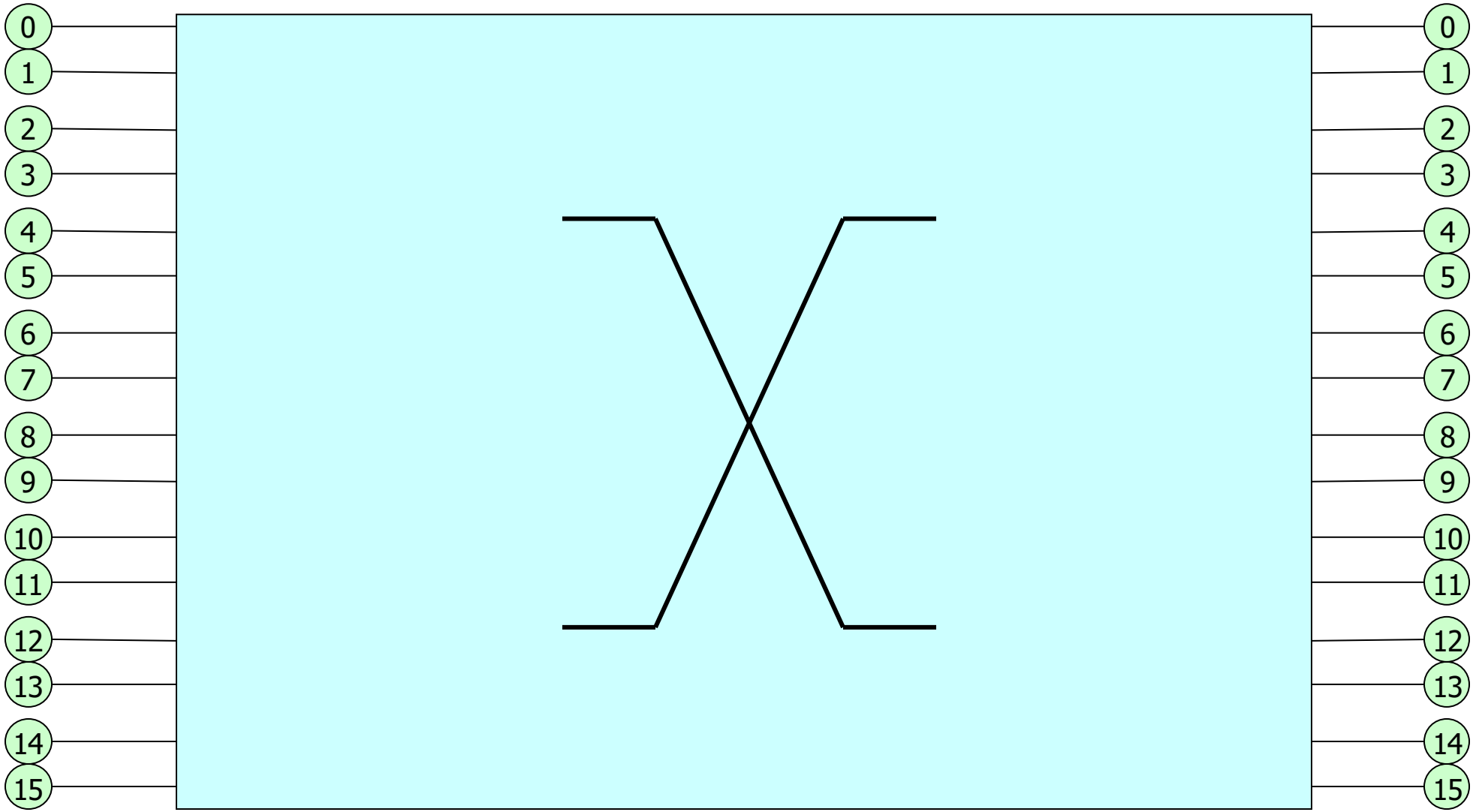
blocking topology

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

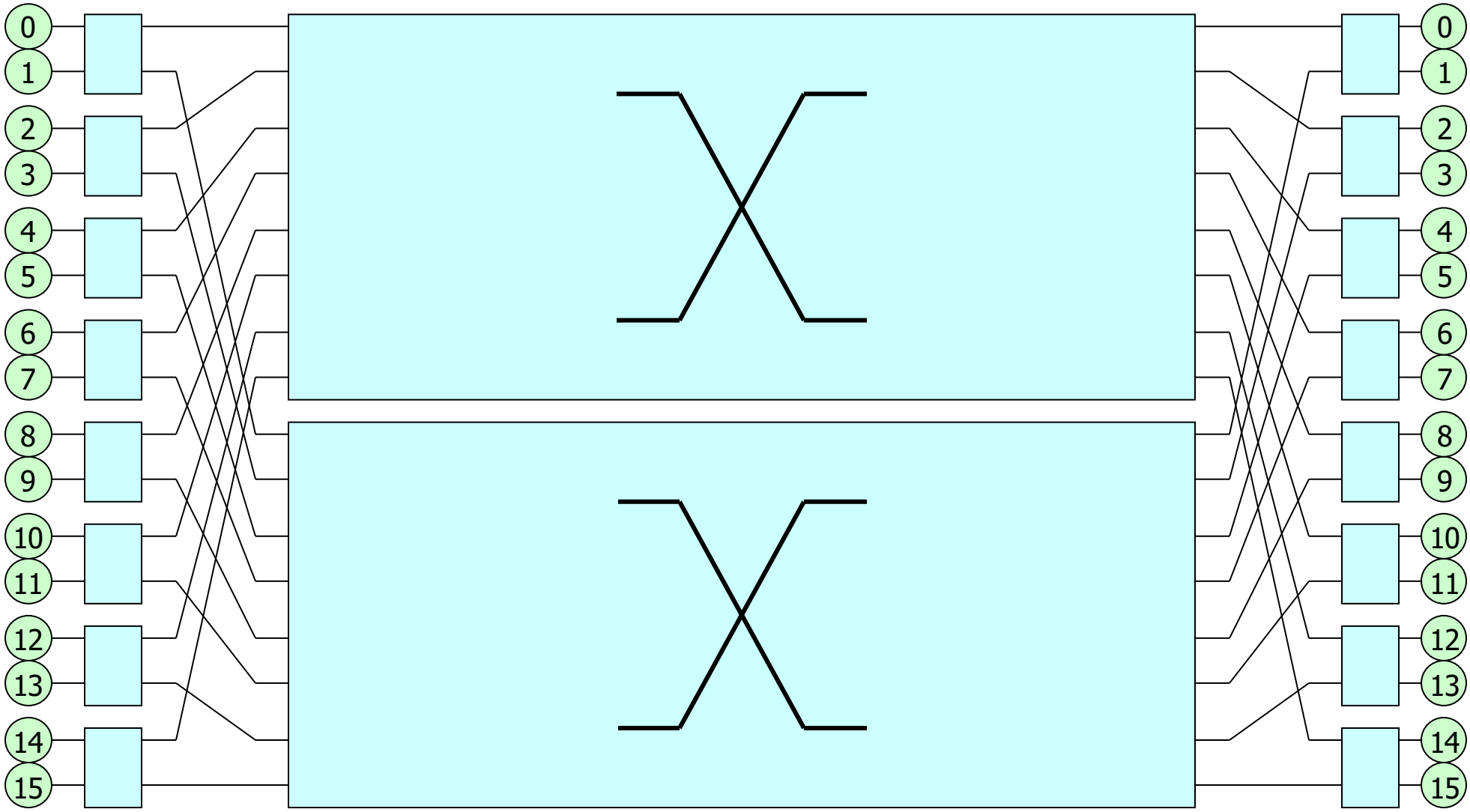
Indirect Networks / Multistage Interconnection Networks



16 port ***Crossbar*** network

Network Topology

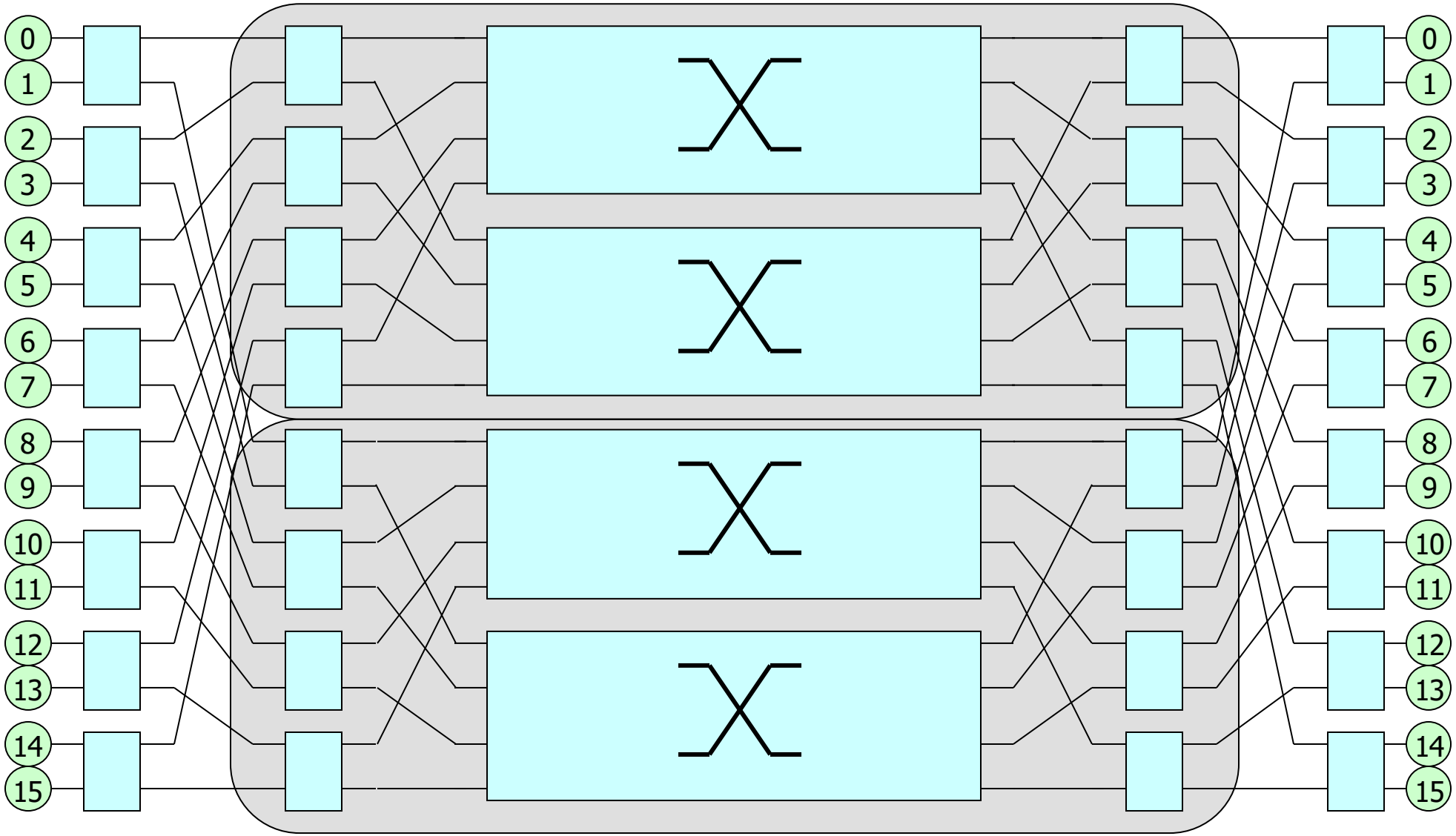
Indirect Networks / Multistage Interconnection Networks



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

Network Topology

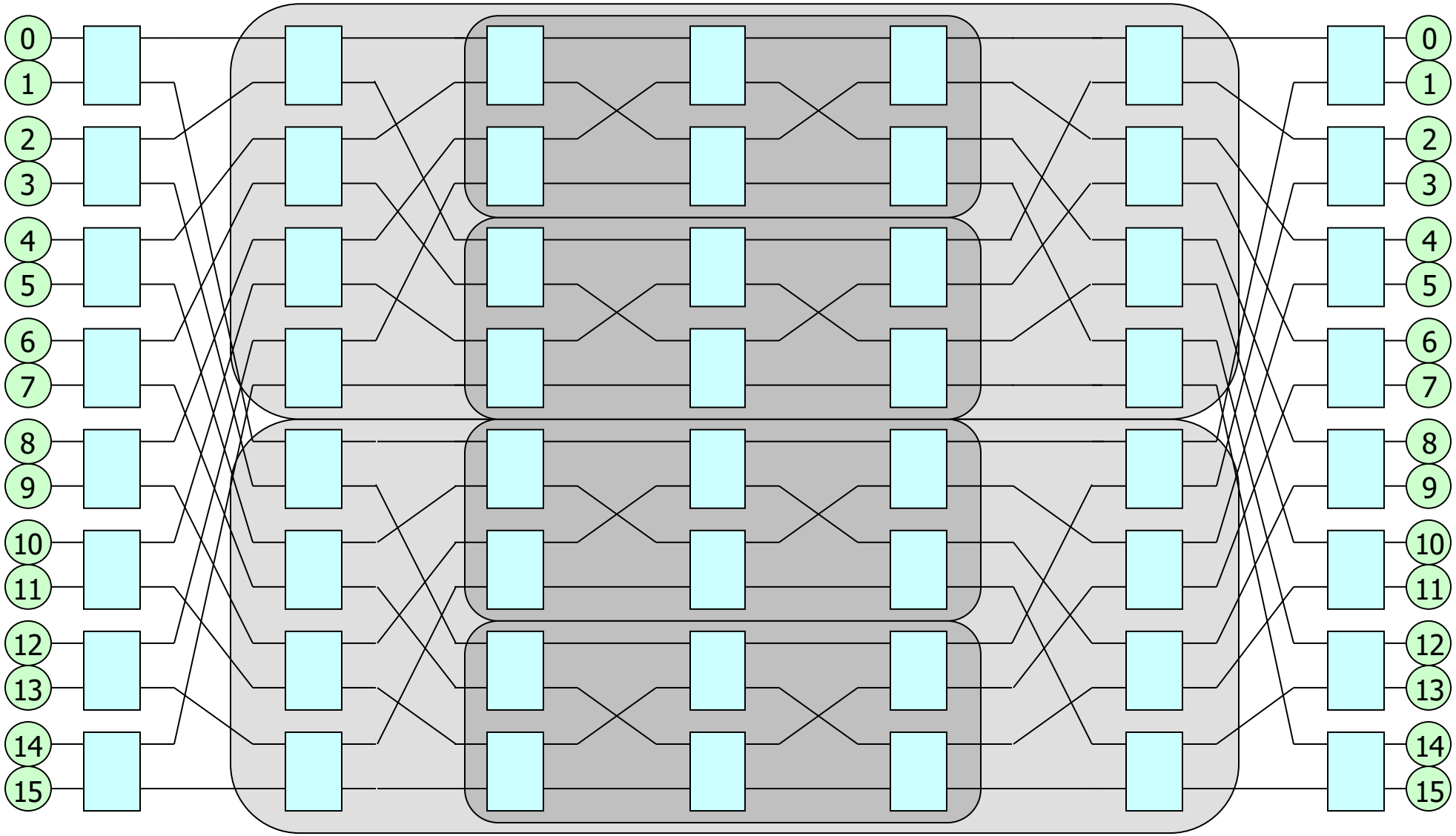
Indirect Networks / Multistage Interconnection Networks



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

Network Topology

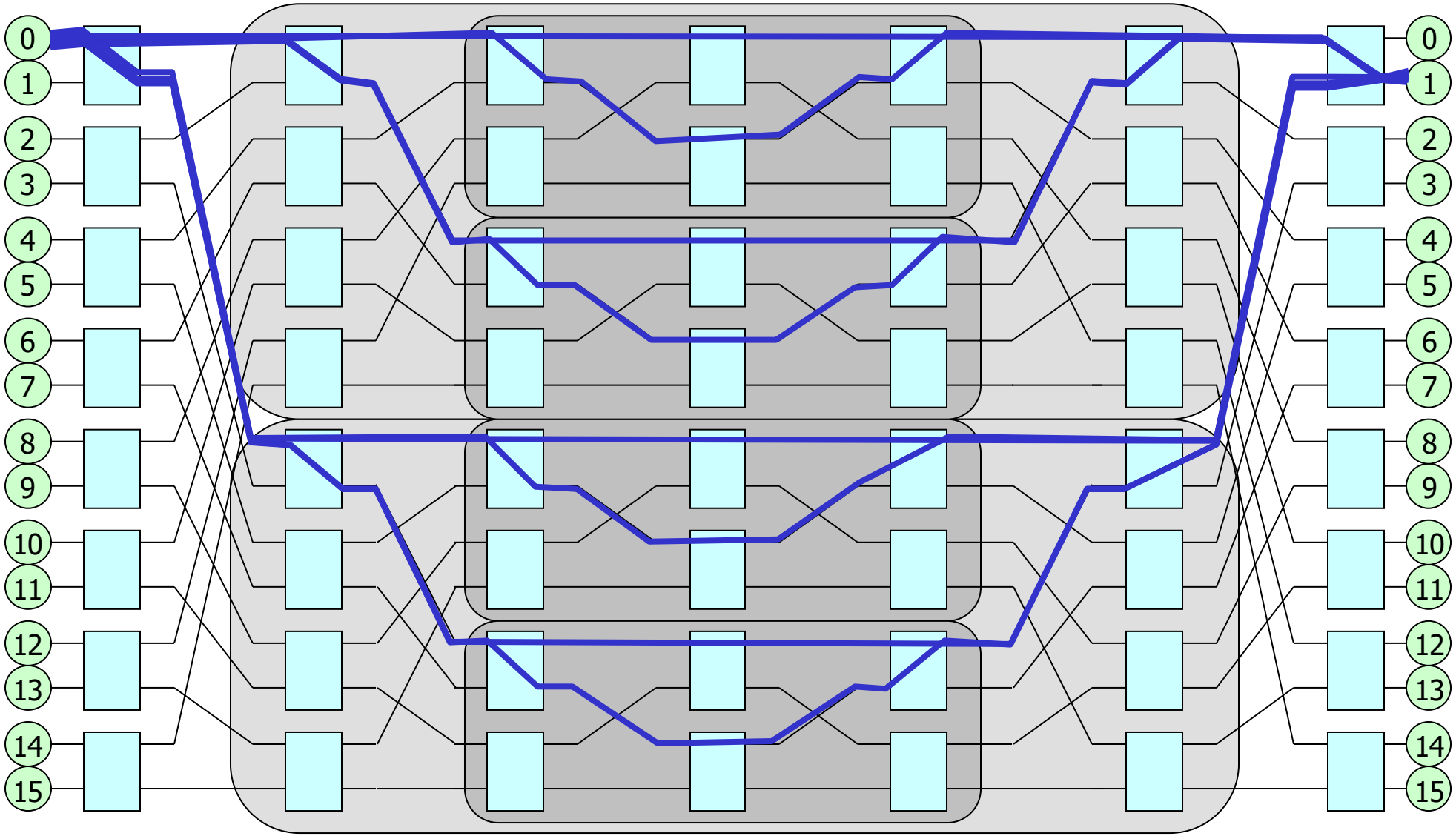
Indirect Networks / Multistage Interconnection Networks



16 port, ***7 stage Clos*** network = ***Benes topology***

Network Topology

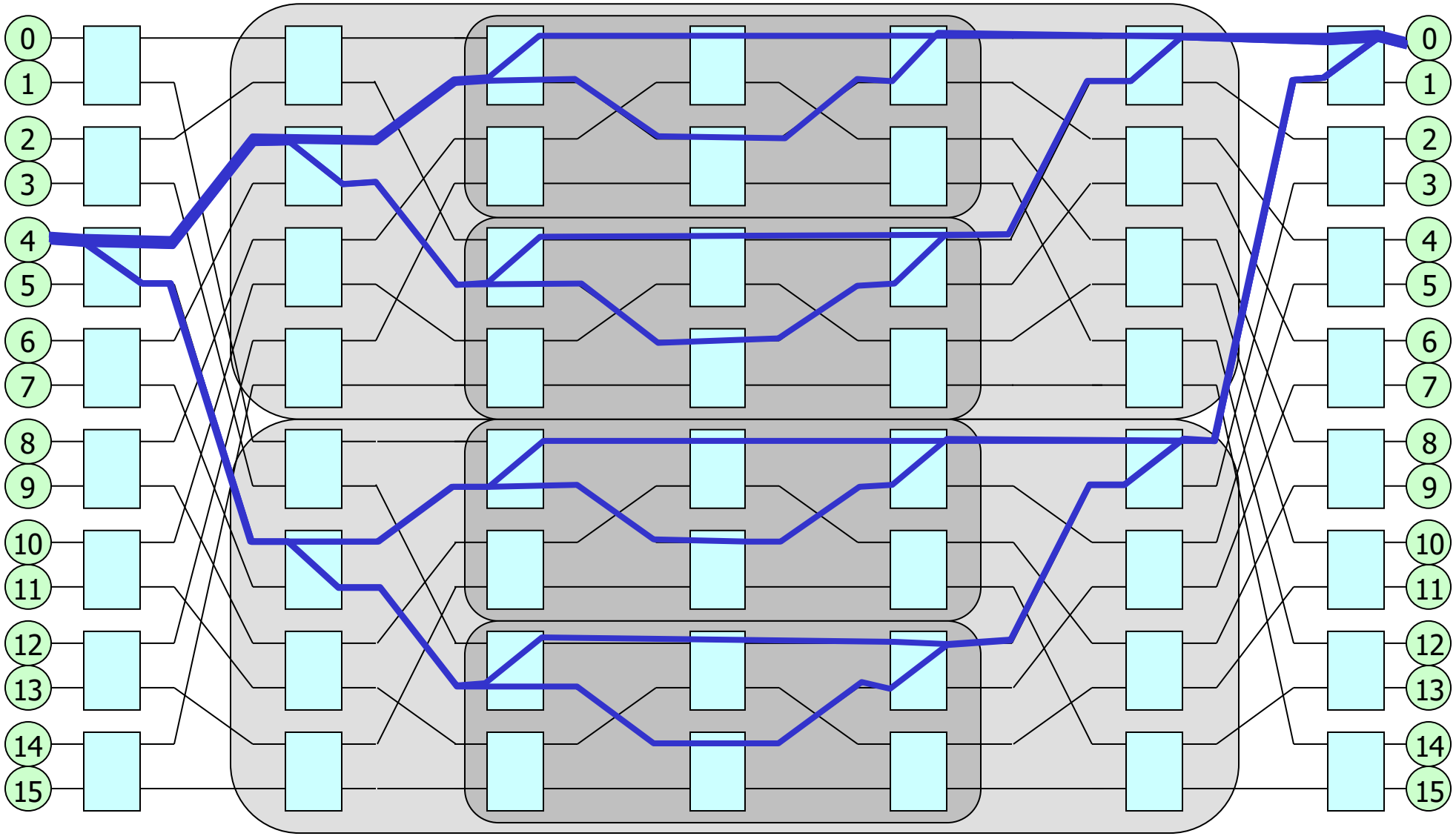
Indirect Networks / Multistage Interconnection Networks



Alternative paths from 0 to 1. 16 port, *7 stage Clos* network = *Benes topology*

Network Topology

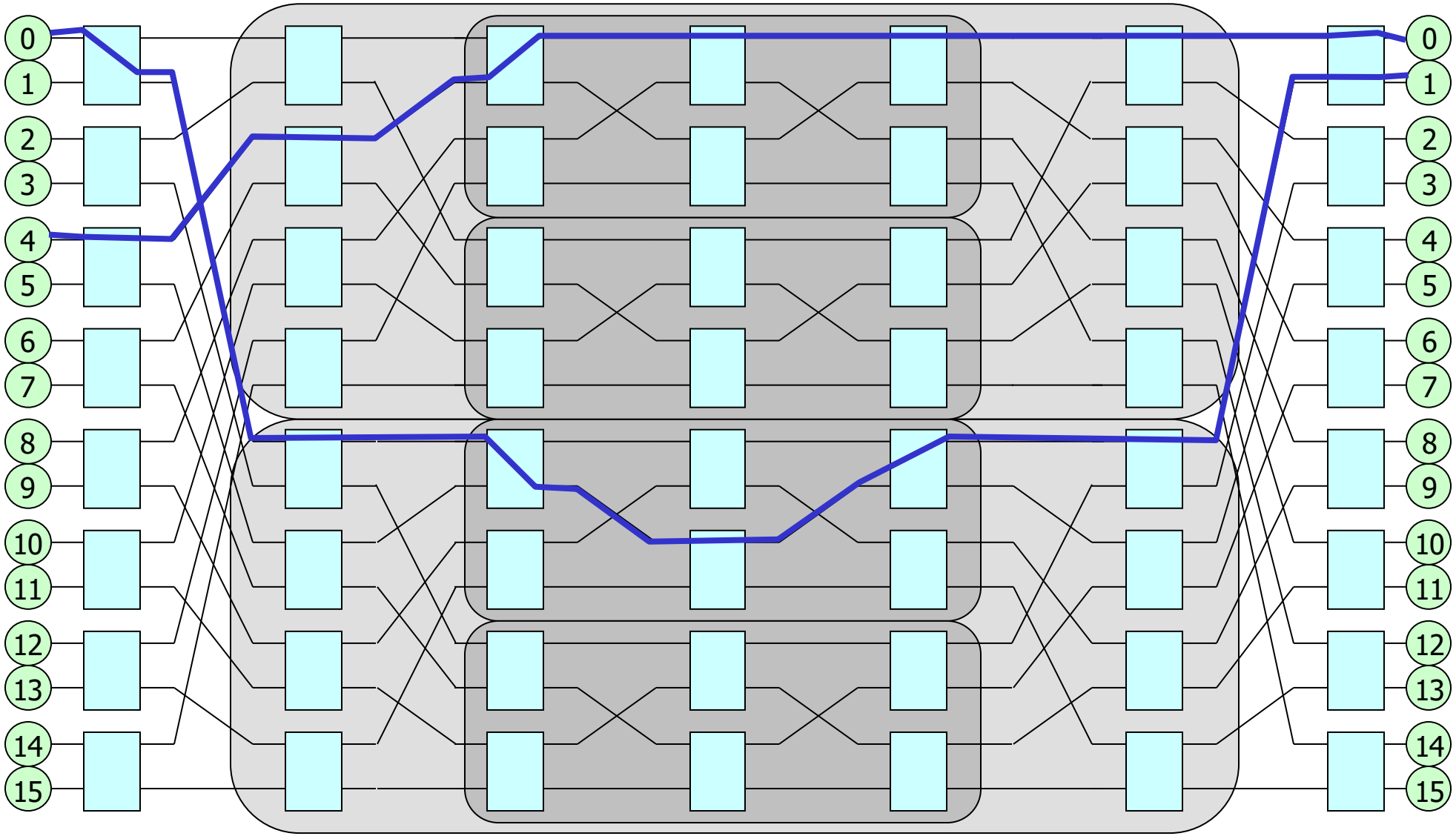
Indirect Networks / Multistage Interconnection Networks



Alternative paths from 4 to 0. 16 port, *7 stage Clos* network = *Benes topology*

Network Topology

Indirect Networks / Multistage Interconnection Networks

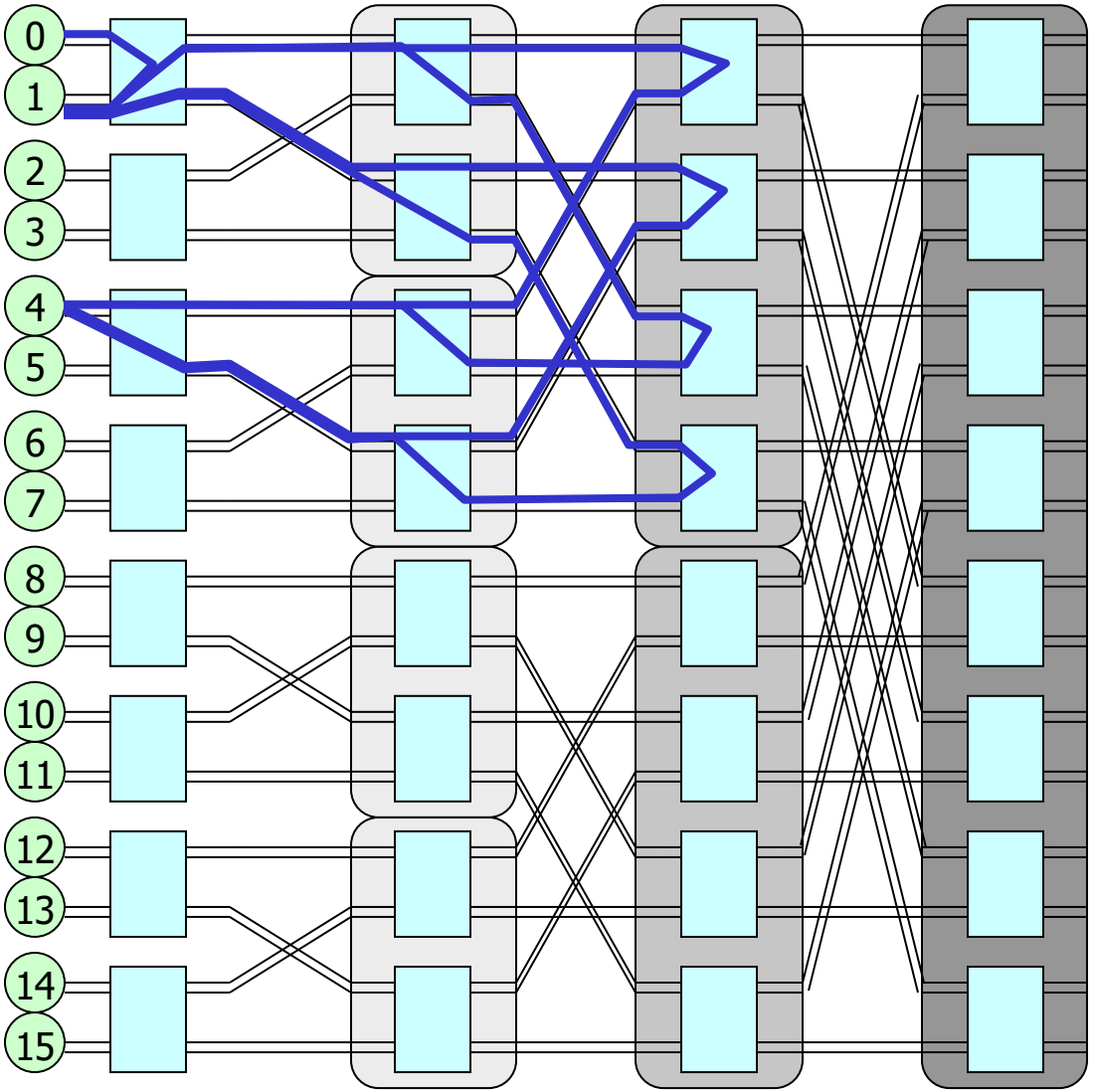


Contention free, paths 0 to 1 and 4 to 0. 16 port, *7 stage Clos* network = *Benes topology*

Network Topology

Indirect Networks / Multistage Interconnection Networks

- *Bidirectional MINs*
- Increase modularity
- Reduce hop count, *d*
- ***Fat tree network***
  - Nodes as tree leaves
  - Switches as tree vertices
  - Total link bandwidth is constant across all tree levels, with ***full bisection bandwidth***
  - Equivalent to folded Benes topology
  - Preferred topology in many SANs

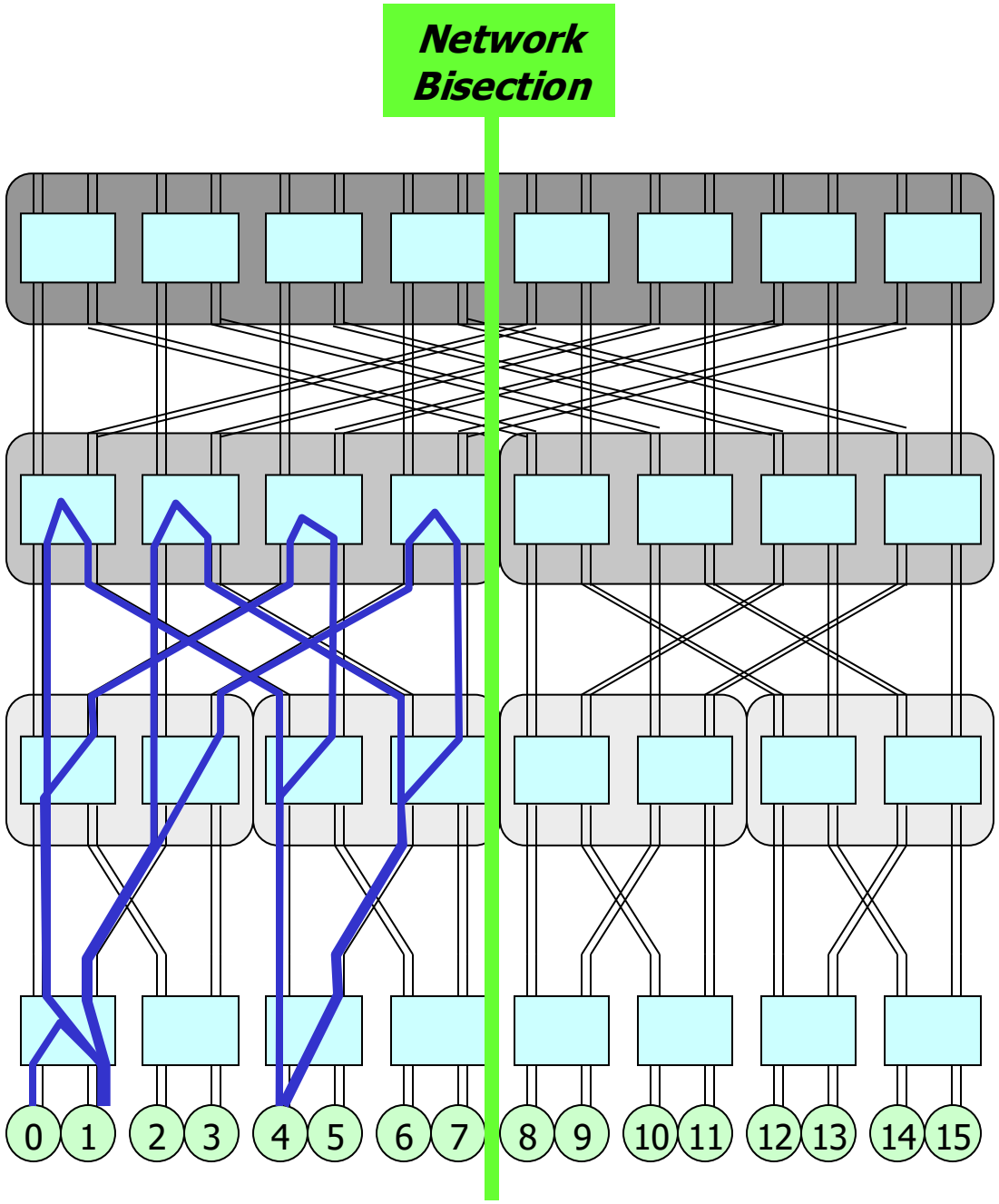


Folded Clos = ***Folded Benes*** = ***Fat tree*** network

Network Topology

Indirect Networks

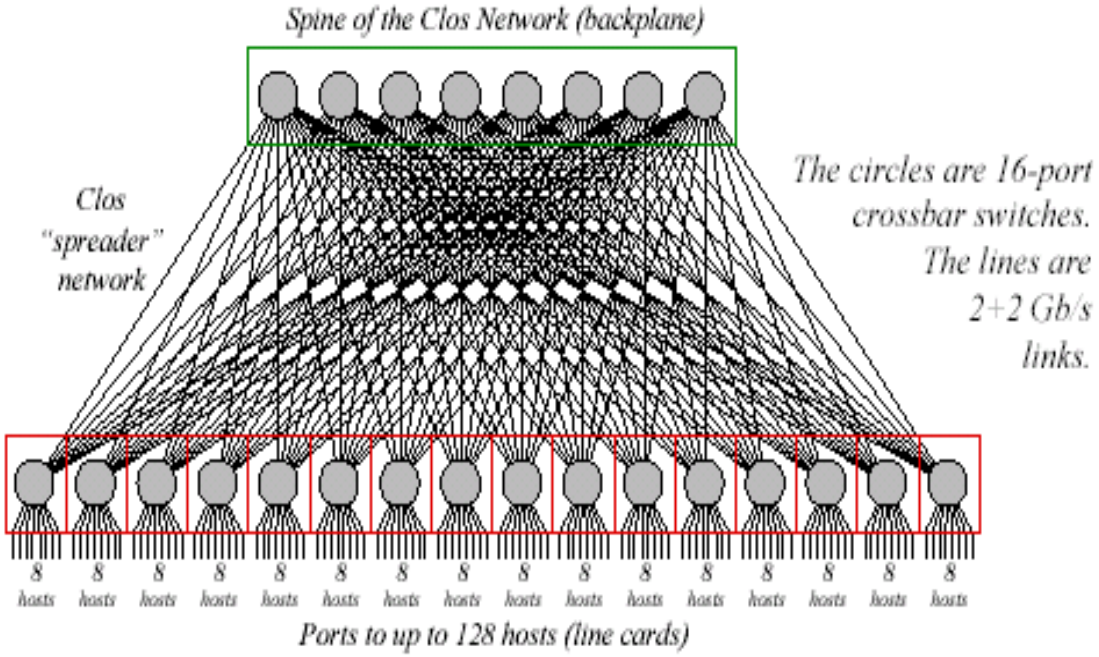
- *Bidirectional MINs*
- Increase modularity
- Reduce hop count, *d*
- ***Fat tree network***
  - Nodes as tree leaves
  - Switches as tree vertices
  - Total link bandwidth is constant across all tree levels, with ***full bisection bandwidth***
  - Equivalent to folded Benes topology
  - Preferred topology in many SANs

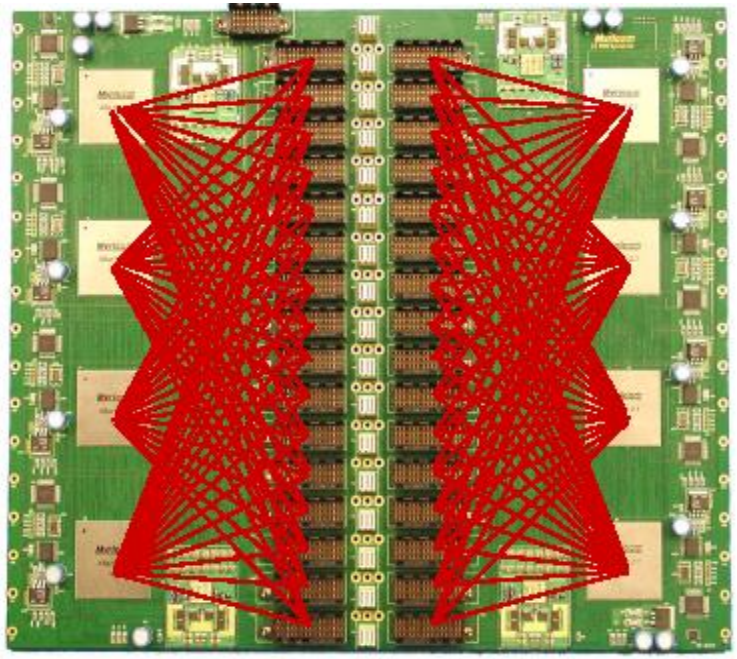


Folded Clos = ***Folded Benes*** = ***Fat tree*** network

Network Topology

Myrinet-2000, two-level Fat-tee Network for 128 Hosts







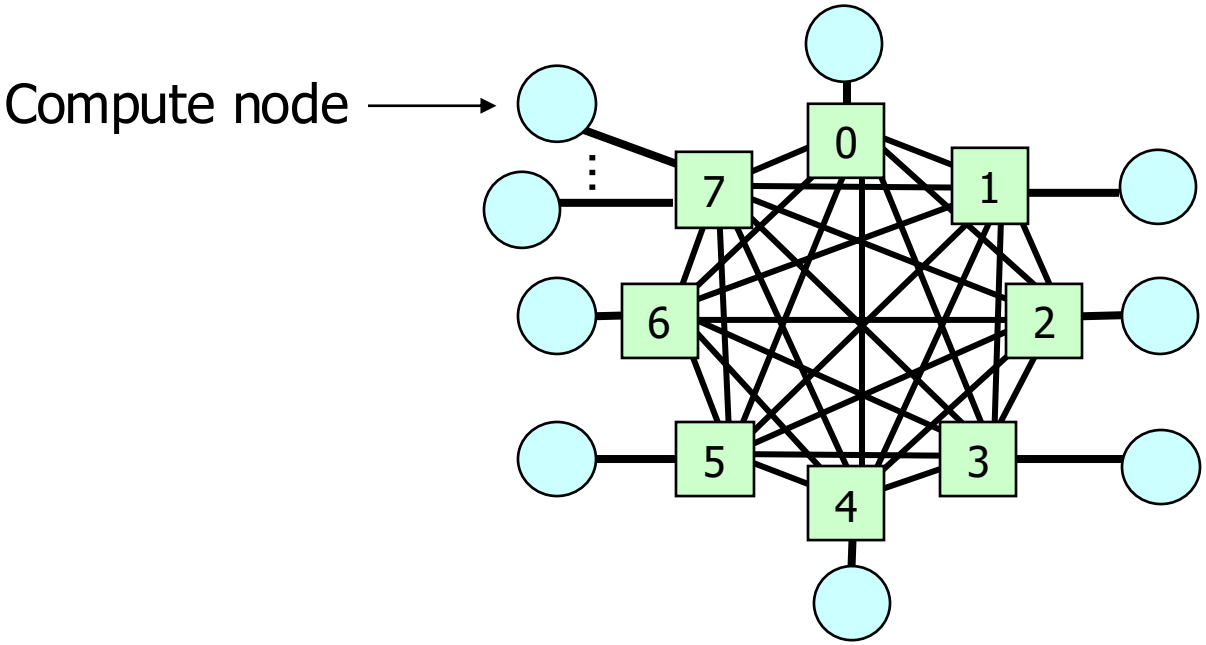
- Backplane of the M3-E128 Switch
- M3-SW16-8F fiber line card (8 ports)

http://myri.com

Network Topology

Direct Networks

- All the switches will have one or more compute directly connected to them, i.e. tight integration of the compute nodes with the network resources
  - Compute nodes distributed among the switches within the network
  - The packets are routed / forwarded through one or more switches
- ***Fully-connected network***: all nodes (switches) are directly connected to all other nodes using bidirectional dedicated links



Network Topology

Direct Networks

- *Example:* Compute the switch and link costs of interconnecting *N* nodes using a fully connected topology relative to a crossbar, assuming
  - Cost of a *k* x *k* switch grows quadratically with the number of unidirectional ports
  - Cost of a 1 x *k* switch on the other hand grows linearly

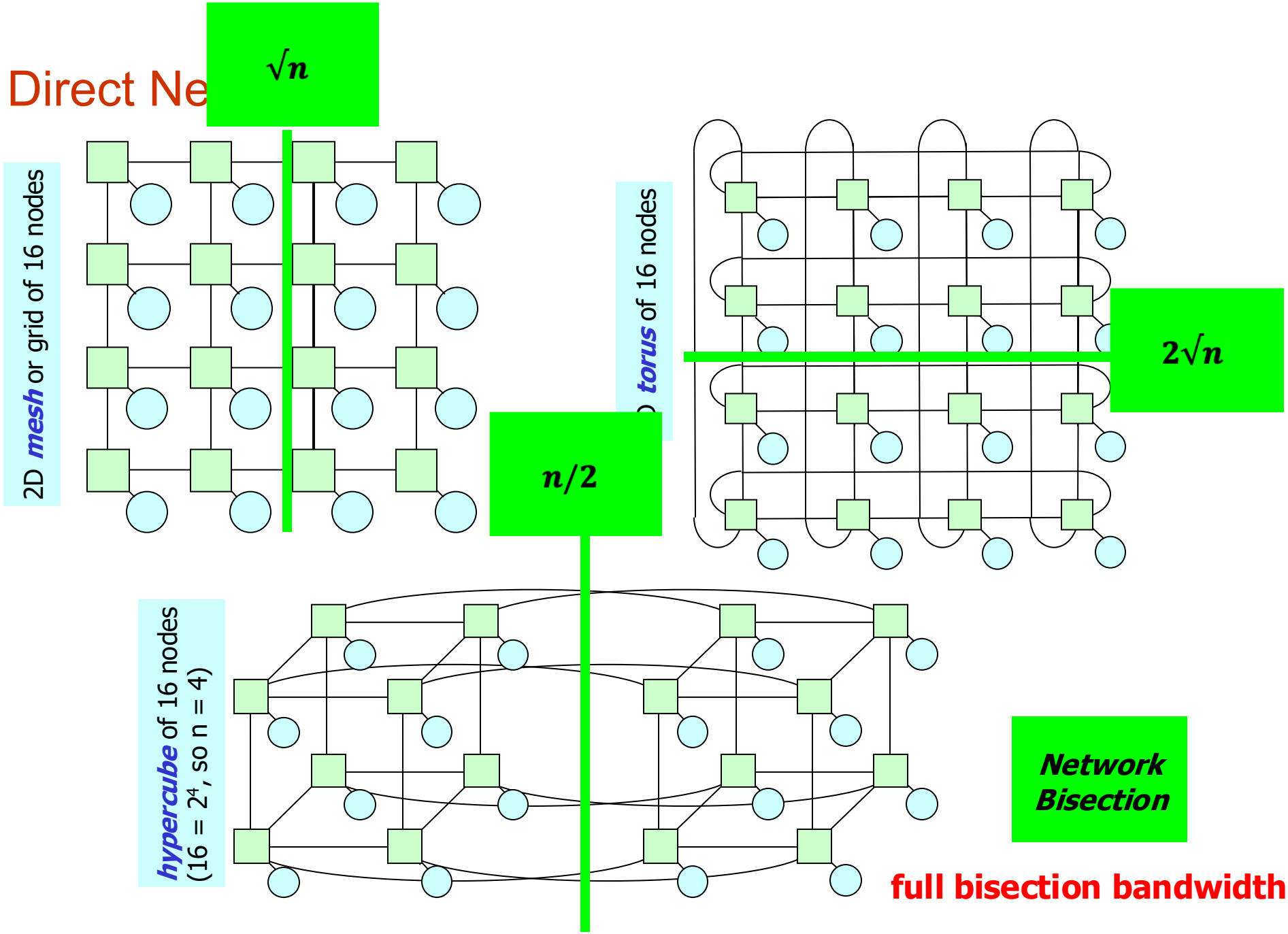
$$\text{Relative cost}_{\text{switches}} = \frac{2N(N-1)}{N^2} = 2\left(1 - \frac{1}{N}\right)$$

$$\text{Relative cost}_{\text{links}} = \frac{N(N+1)}{2N} = \frac{N+1}{2}$$

- As *N* increases, the switch cost nearly doubles the crossbar's
- Link cost is always higher than a crossbar's
- ***No extra benefits*** *of a fully connected network over a crossbar!*

Network Topology

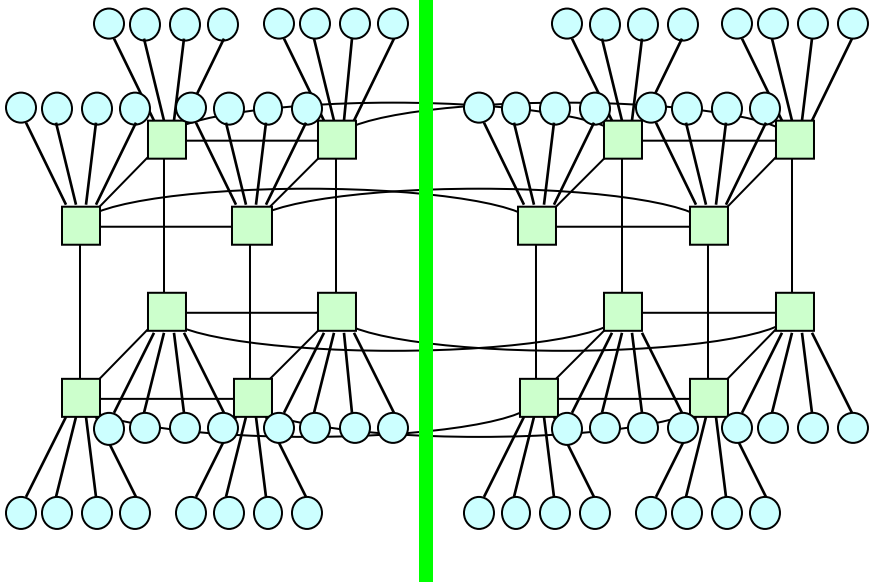
Direct Ne



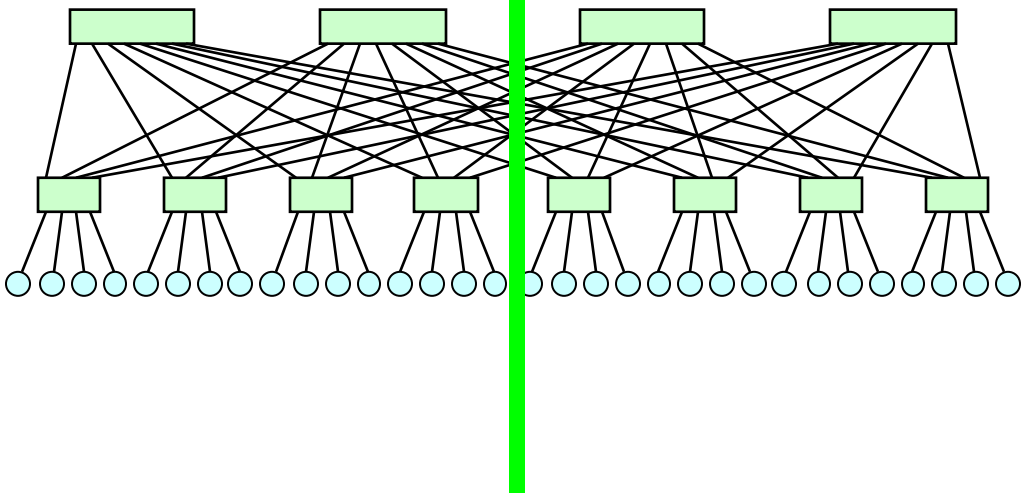
Network Topology

Comparison of Indirect and Direct Networks

- *Bristling* can be used to reduce direct network switch & link costs
  - *"b"* compute nodes connect to each switch, where *b* is bristling factor
  - Allows larger systems to be built from fewer switches and links
  - Requires lower *switch degree*
  - For *N* = $\frac{n}{8} = 8$ *k* = 8, requiring fewer switches $\frac{n}{2} = 16$ ks than a fat tree



64-node system with 8-port switches, b = 4



32-node system with 8-port switches

Network Topology

Comparison of Indirect and Direct Networks

- Blocking reduced by maximizing dimensions (switch degree)
  - Can increase bisection bandwidth, but
    - Additional dimensions may increase wire length (must observe 3D packaging constraints)
    - Flow control issues (buffer size increases with link length)
    - Pin-out constraints (limit the number of dimensions achievable)

	Evaluation category	Bus	Ring	2D mesh	2D torus	Hypercube	Fat tree	Fully connected
Perf.	$BW_{Bisection}$ in # links	1	2	8	16	32	32	1024
	Max (ave.) hop count	1 (1)	32 (16)	14 (7)	8 (4)	6 (3)	11 (9)	1 (1)
Cost	I/O ports per switch	NA	3	5	5	7	4	64
	Number of switches	NA	64	64	64	64	192	64
	Number of net. links	1	64	112	128	192	320	2016
	Total number of links	1	128	176	192	256	384	2080

Performance and cost of several network topologies for 64 nodes. Values are given in terms of bidirectional links & ports. Hop count includes a switch and its output link (in the above, end node links are not counted for the bus topology).

Outline

- Introduction
- Interconnecting Two Devices
- Interconnecting Many Devices
- Network Topology
- **Network Routing** *(Now)*
- Concluding Remarks and References

Routing in Interconnection Networks

Routing

- Performed at each switch, regardless of topology
- Defines the "allowed" path(s) for each packet (*Which paths?*)
- Need 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 to never 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

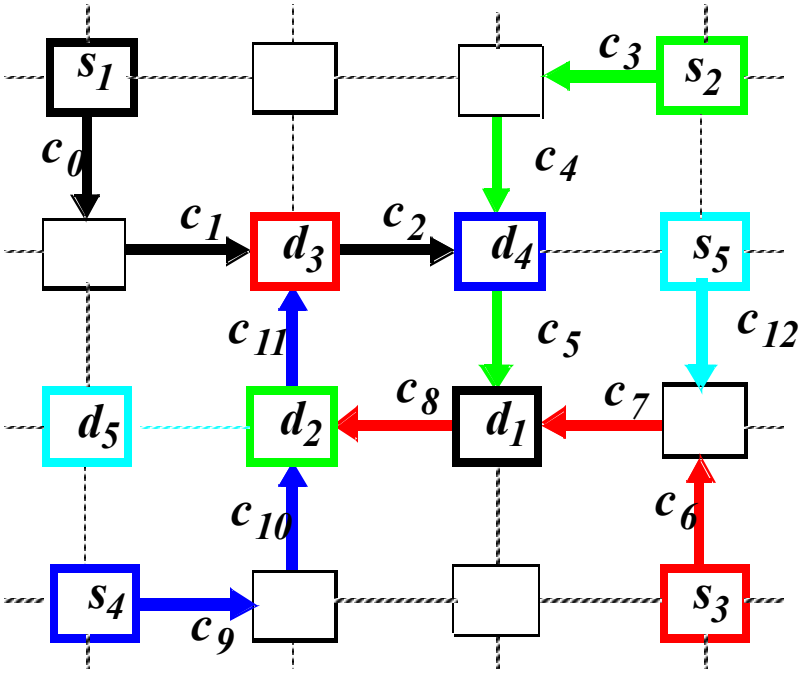
Routing in Interconnection Networks

Routing

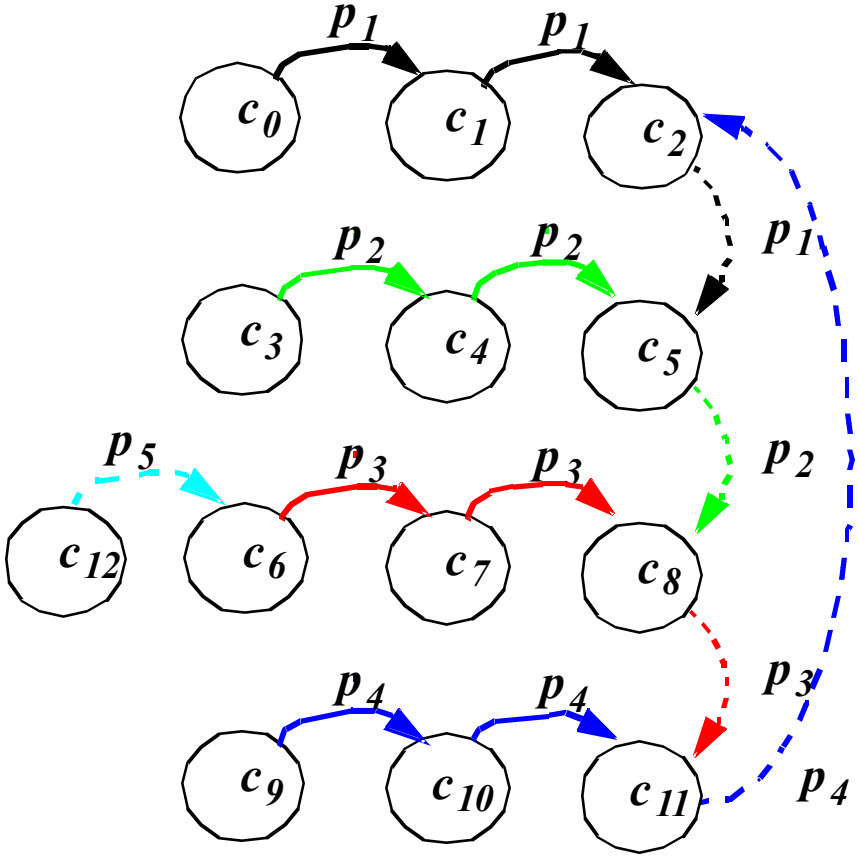
- Common forms of deadlock:
  - ***Routing-induced deadlock***

c_i = channel ***i***
s_i = source node ***i***
d_i = destination node ***i***
p_i = routing (packet) dependency ***i***

Routing of packets in a 2D mesh



Channel dependency graph



Routing in Interconnection Networks

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, i.e. the channel dependency graph cycle free
    - › *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)*

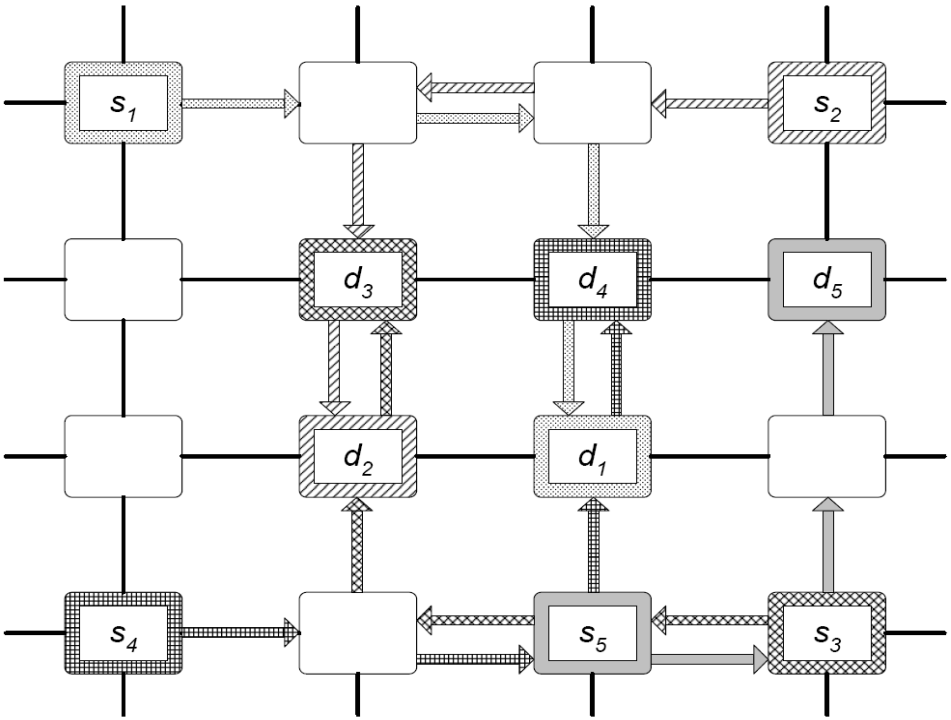
Routing in Interconnection Networks

Routing

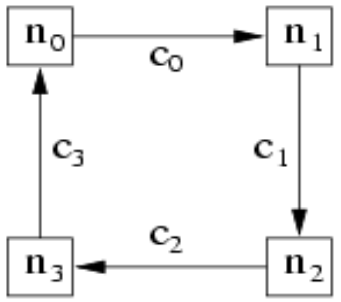
- Common strategies to deal with deadlock
  - *Deadlock avoidance:*

n_i = node i
c_i = physical channel i
c_{1i} = high-ordered VC i
c_{0i} = low-ordered VC i

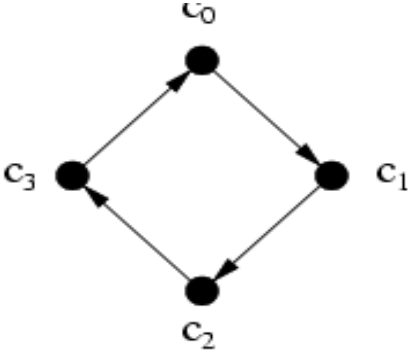
Deadlock avoidance in 2D mesh using *DOR*



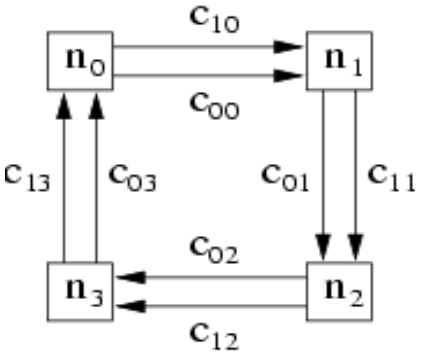
Deadlock avoidance in ring using *VCs*

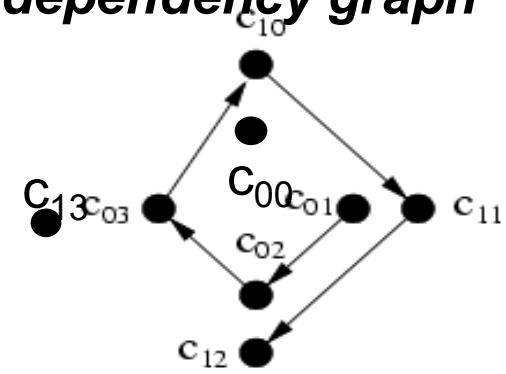


Network graph



Channel dependency graph





2 virtual channels per physical channel

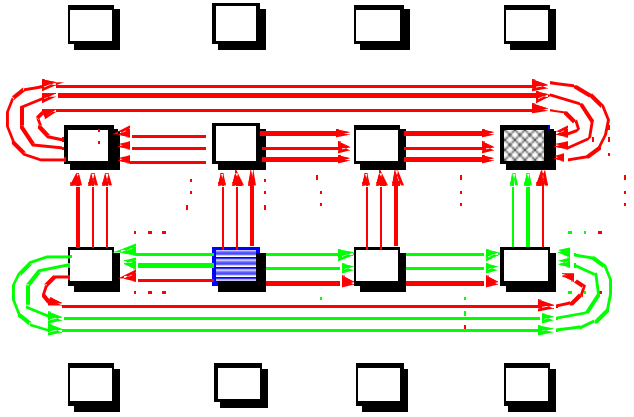
Routing in Interconnection Networks

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
  - Increases routing freedom to improve network efficiency, ρ

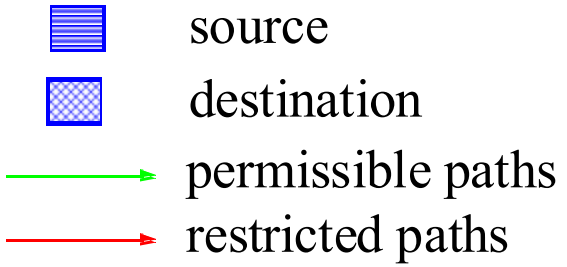
Routing in Interconnection Networks

Routing

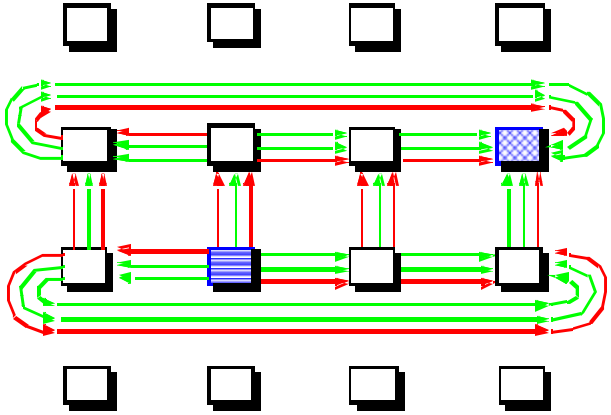


Dimension-order routing (deterministic)

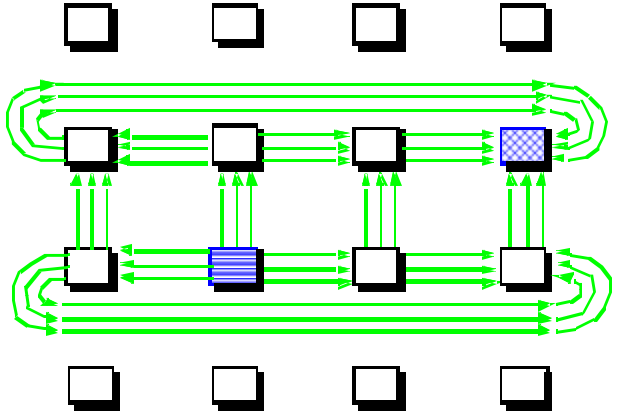
Comparison of Routing Freedom



Routing freedom can increase ρ (i.e., ρ_R)



Partial adaptive routing (Duato´s algorithm)



Recovery routing (true fully adaptive)

Concluding Remarks and References

Concluding Remarks

- Interconnection network design is an important area of computer architecture, in the era of HPC-AI
  - on-chip networks between cores within a chip
  - off-chip networks between chips and boards within a HPC-AI system
- 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 general throughput model can be used to guide design*
- Improving the performance of interconnection networks is critical to advancing our AI- and communication-centric world

Interconnecting Two Devices

Characterizing Performance: Latency & Effective Bandwidth

Terms and Definitions:

- *Bandwidth:*
  - Maximum rate at which information can be transferred (including packet header, payload and trailer)
  - Unit: bits per second (bps) or bytes per second (Bps)
  - *Aggregate bandwidth*: Total data bandwidth supplied by network
  - *Effective bandwidth* (*throughput*): fraction of aggregate bandwidth that gets delivered to the application
- *Time of flight:* Time for first bit of a packet to arrive at the receiver
  - Includes the time for a packet to pass through the network, not including the transmission time (defined next)
  - Picoseconds (OCNs), nanoseconds (SANs), microseconds (LANs), milliseconds (WANs)

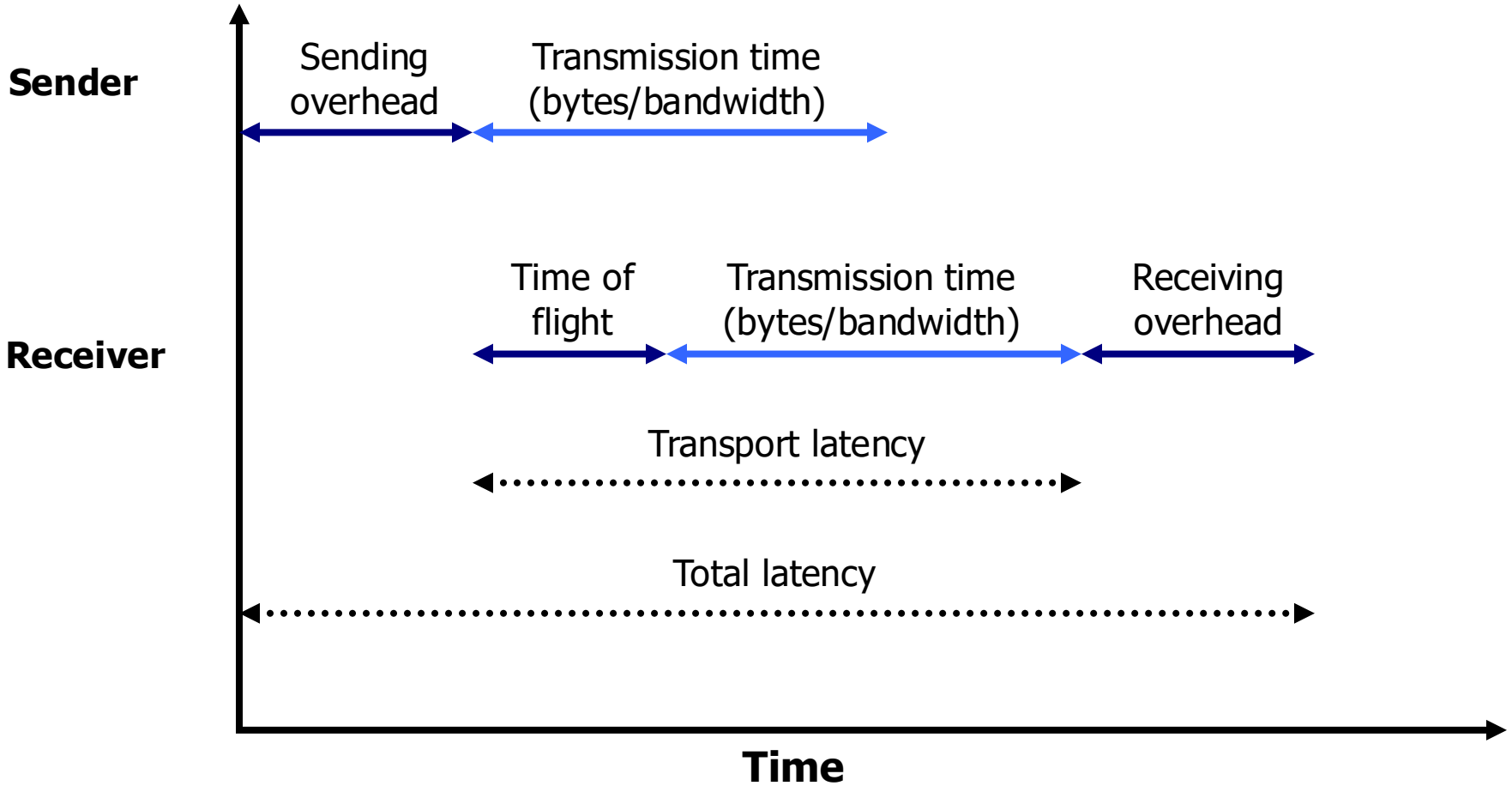
Interconnecting Two Devices

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



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