Advanced Topics in Computer Architecture

Lecture 6
Interconnection Networks for Parallel Systems

Marenglen Biba
Department of Computer Science
University of New York Tirana

Outline

- Introduction
- Interconnecting Two Devices
- Connecting More than Two Devices
- Network Topology
- Switch Microarchitecture
- Practical Issues for Commercial Interconnection Networks
- Examples of Interconnection Networks
- Crosscutting Issues for Interconnection Networks

Interconnection networks

- Interconnection networks are also called *networks*, *communication subnets*, or *communication subsystems*.
- The interconnection of multiple networks is called *internetworking*.
- This relies on communication standards to convert information from one kind of network to another, such as with the Internet.

Interconnected devices

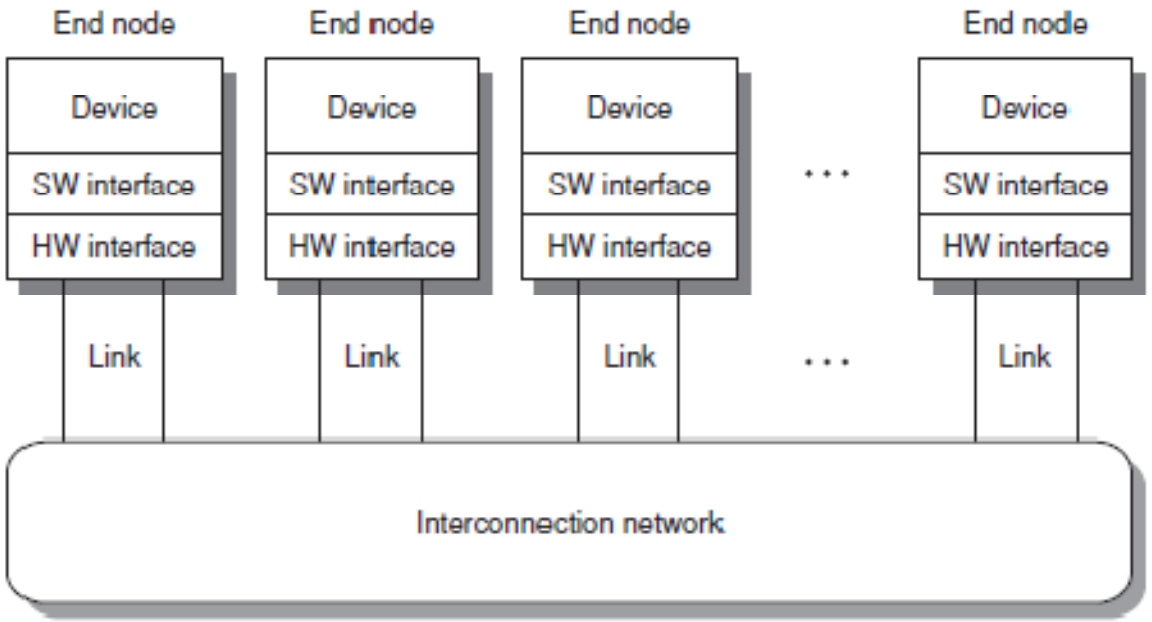


Figure E.1 A conceptual illustration of an interconnected community of devices.

Switched networks

- Computer architects should devote attention to interconnection networks.
- Switched networks are replacing buses as the normal means of communication between:
  - computers, between
  - I/O devices, between
  - boards, between
  - chips, and
  - even between modules inside chips.
- Computer architects must understand interconnect problems and solutions in order to more effectively design and evaluate computer systems.

Interconnection Network Domains

- Interconnection networks are designed for use at different levels within and across computer systems to meet the operational demands of various application areas:
  - high-performance computing,
  - storage I/O cluster/workgroup/enterprise systems,
  - internetworking, and so on.
- Depending on the number of devices to be connected and their proximity, we can group interconnection networks into four major networking domains.

On-chip networks

- OCNs—Also referred to as network-on-chip (NoC) are used for interconnecting microarchitecture functional units, register files, caches, compute tiles, and processor and IP cores within chips or multichip modules.
- Currently, OCNs support the connection of up to only a few tens of such devices with a maximum interconnection distance on the order of centimeters.
- An example custom OCN is the *Element Interconnect Bus* used in the Cell Broadband Engine processor chip.
  - This network peaks at ~2400 Gbps (for a 3.2 GHz processor clock) for 12 elements on the chip.

System/storage area networks

- SANs are used for interprocessor and processor-memory interconnections within multiprocessor and multi-computer systems, and also for the connection of storage and I/O components within server and data center environments.
- Typically, several hundreds of such devices can be connected, although some supercomputer SANs support the interconnection of many thousands of devices, like the IBM Blue Gene/L supercomputer.
- The maximum interconnection distance covers a relatively small area on the order of a few tens of meters usually but some SANs have distances spanning a few hundred meters.
- For example, *InfiniBand* , a popular SAN standard introduced in late 2000, supports system and storage I/O interconnects at up to 120 Gbps over a distance of 300 m.

Local area networks

- LANs are used for interconnecting autonomous computer systems distributed across a machine room or throughout a building or campus environment.
- Interconnecting PCs in a cluster is a prime example.
- Originally, LANs connected only up to a hundred devices, but with bridging,
- LANs can now connect up to a few thousand devices.
- The maximum interconnect distance covers an area of a few kilometers usually, but some have distance spans of a few tens of kilometers.
- For instance, the most popular and enduring LAN, *Ethernet* , has a 10 Gbps standard version that supports maximum performance over a distance of 40 km.

Wide area networks

- Also called *long-haul networks.*
- WANs connect computer systems distributed across the globe, which requires internetworking support.
- WANs connect many millions of computers over distance scales of many thousands of kilometers.

Networks

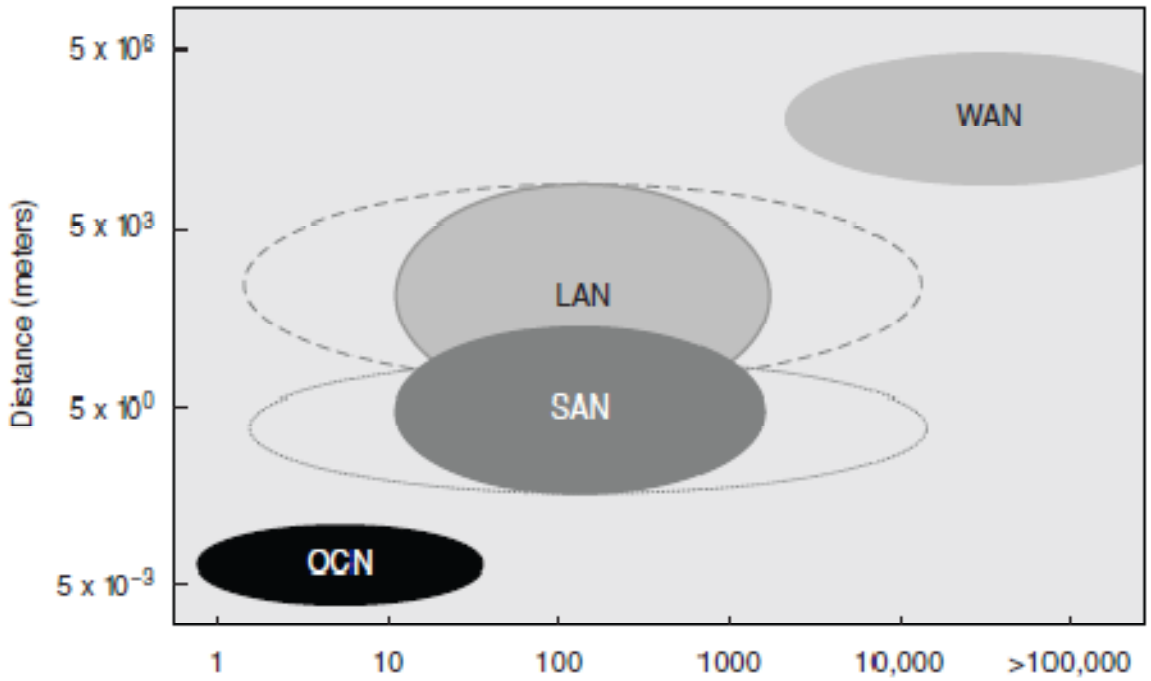


Figure E.2 Relationship of the four interconnection network domains in terms of number of devices connected and their distance scales: on-chip network (OCN), system/storage area network (SAN), local area network (LAN), and wide area network (WAN). Note that there are overlapping ranges where some of these networks compete. Some supercomputer systems use proprietary custom networks to interconnect several thousands of computers, while other systems, such as multicomputer clusters, use standard commercial networks.

Outline

- Introduction
- Interconnecting Two Devices
- Connecting More than Two Devices
- Network Topology
- Switch Microarchitecture
- Practical Issues for Commercial Interconnection Networks
- Examples of Interconnection Networks
- Crosscutting Issues for Interconnection Networks

Dedicated Link

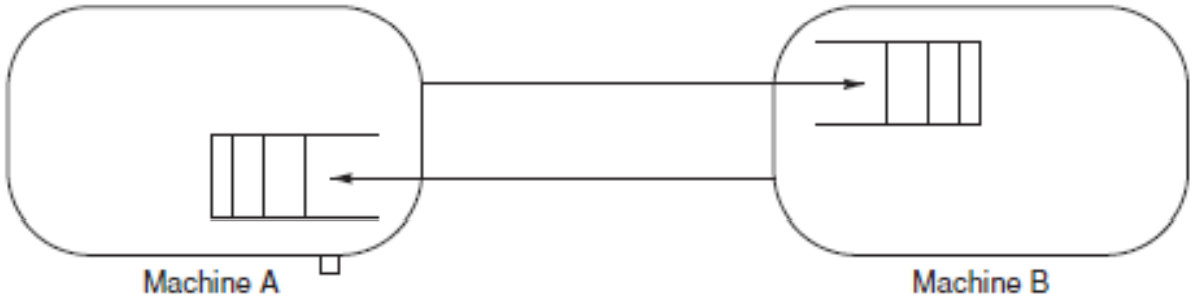


Figure E.3 A simple dedicated link network bidirectionally interconnecting two devices.

Packet Format

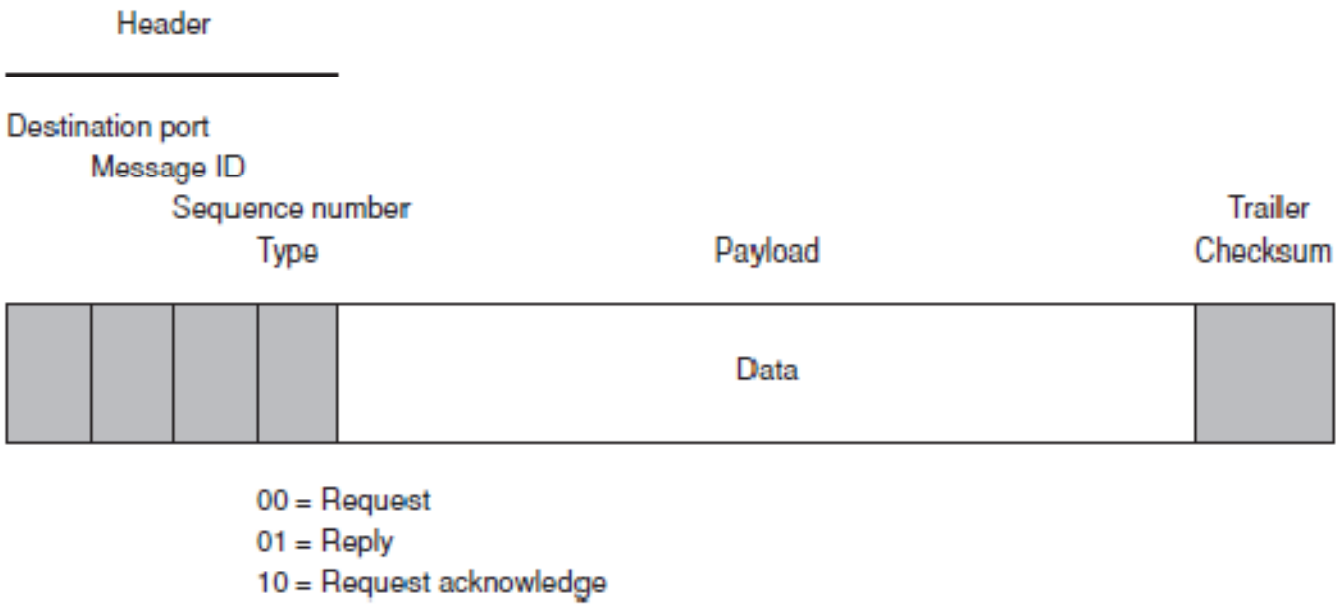


Figure E.4 An example packet format with header, payload, and checksum in the trailer.

Communication protocol

- The sequence of steps the end node follows to commence and complete communication over the network is called a *communication protocol*.
- Communication protocols are implemented by a combination of software and hardware to accelerate execution.
- For instance, many network interface cards implement hardware timers as well as hardware support to split messages into packets and reassemble them, compute the cyclic redundancy check (CRC) *checksum*, handle virtual memory addresses, and so on.

Characterizing Performance: Latency and Effective Bandwidth

- Let’s start by discussing the latency when transporting a single packet.

Definitions

- *Bandwidth* refers to the maximum rate at which information can be transferred, where information includes packet header, payload, and trailer. The units are traditionally bits per second although bytes per second is sometimes used.
- *Time of flight* — This is the time for the first bit of the packet to arrive at the receiver, including the propagation delay over the links and delays due to other hardware in the network such as link repeaters and network switches.
  - The unit of measure for time of flight can be in milliseconds for WANs, microseconds for LANs, nanoseconds for SANs, and picoseconds for OCNs.

Definitions

- *Transmission time* — This is the time for the packet to pass through the network, not including time of flight.
  - One way to measure it is the difference in time between when the first bit of the packet arrives at the receiver and when the last bit of that packet arrives at the receiver.
- *Transport latency* — This is the sum of time of flight and transmission time.
  - Transport latency is the time that the packet spends in the interconnection network.

Definitions

- *Sending overhead* — This is the time for the end node to prepare the packet (as opposed to the message) for injection into the network, including both hardware and software components.
  - Note that the end node is busy for the entire time, hence the use of the term *overhead*.
- *Receiving overhead* — This is the time for the end node to process an incoming packet, including both hardware and software components.

Packet Latency

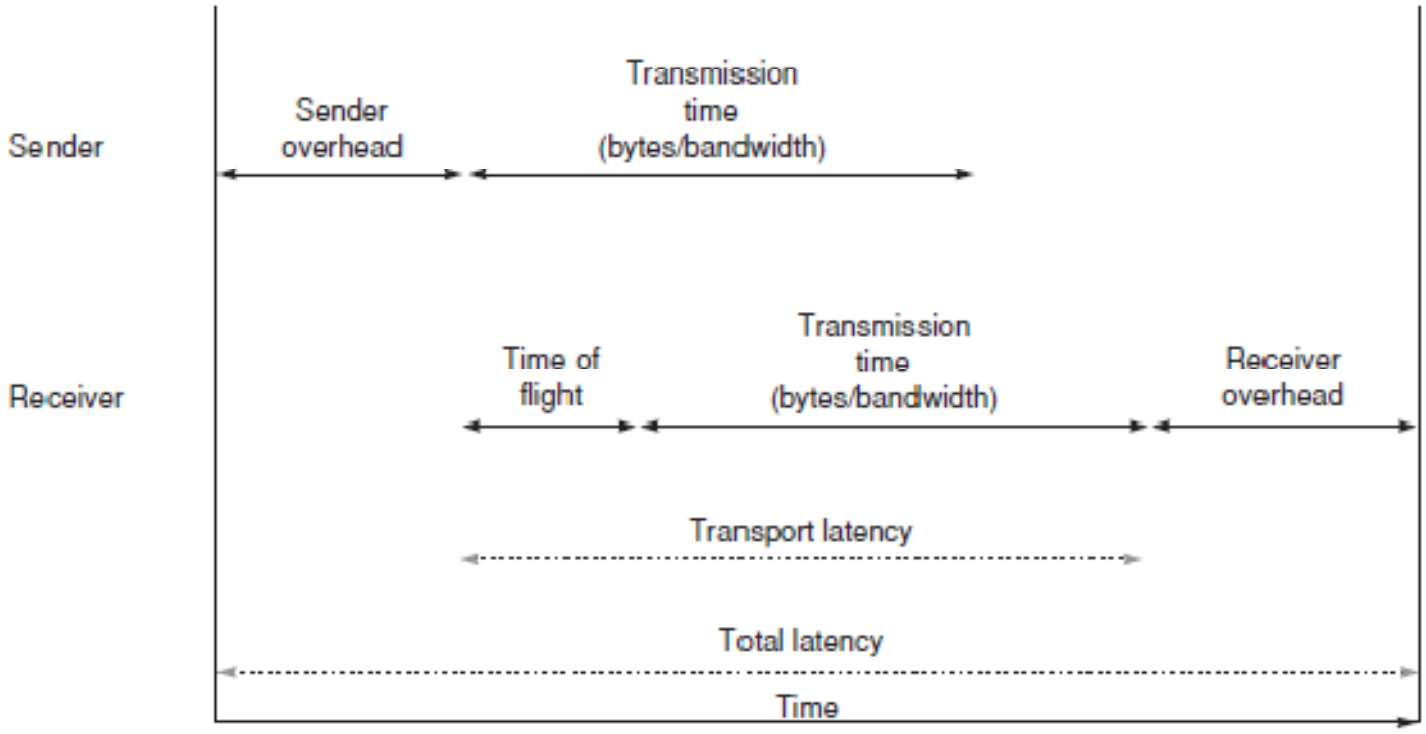


Figure E.5 Components of packet latency. Depending on whether it is an OCN, SAN, LAN, or WAN, the relative amounts of sending and receiving overhead, time of flight, and transmission time are usually quite different from those illustrated here.

Total Latency

- The total latency of a packet can be expressed algebraically by the following:

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

Link Pipelining

- We have considered the transport of only a single packet and computed the associated end-to-end total packet latency.
- In order to compute the effective bandwidth for two networked devices, we have to consider a continuous stream of packets transported between them.
- For applications that do not require a response before sending the next packet, the sender can overlap the sending overhead of later packets with the transport latency and receiver overhead of prior packets.
- This essentially pipelines the transmission of packets over the network, also known as *link pipelining*.

Link Pipelining

- Fortunately, as discussed in prior chapters of this book, there are many application areas where communication from either several applications or several threads from the same application can run concurrently (e.g., a Web server concurrently serving thousands of client requests or streaming media), thus allowing a device to send a stream of packets without having to wait for an acknowledgment or a reply.
- Also, as long messages are usually divided into packets of maximum size before transport, a number of packets are injected into the network in succession for such cases.
- If such overlap were not possible, packets would have to wait for prior packets to be acknowledged before being transmitted and, thus, suffer significant performance degradation.

Pipelining packet transport

- Pipelining packet transport over the network has many similarities with pipelining computation within a processor.
- But there is a difference: here information is simply propagated through network links as a sequence of signal waves.
- Thus, the network can be considered as a logical pipeline consisting of as many stages as are required so that the time of flight does not affect the effective bandwidth that can be achieved.

Link injection bandwidth

- For each link injecting a continuous stream of packets into a network, the resulting *link injection bandwidth*, $BW_{LinkInjection}$ is calculated with the following expression:

$$BW_{LinkInjection} = \frac{\text{Packet size}}{\max(\text{Sending overhead, Transmission time})}$$

Link injection bandwidth

- We must also consider what happens if the receiver is unable to consume packets at the same rate they arrive.
- This occurs if the receiving overhead is greater than the sending overhead and the receiver cannot process incoming packets fast enough.
- In this case, the *link reception bandwidth*, $BW_{LinkReception}$, for each reception link of the network is less than the link injection bandwidth and is obtained with this expression:

$$BW_{LinkReception} = \frac{\text{Packet size}}{\max(\text{Receiving overhead, Transmission time})}$$

Bottlenecks: Receiver buffer

- When communication takes place between two devices interconnected by dedicated links, all the packets sent by one device will be received by the other.
- If the receiver cannot process packets fast enough, the receiver buffer will become full, and flow control will throttle transmission at the sender.
- As this situation is produced by causes external to the network, we will not consider it further here.

Effective Bandwidth

- Assuming an ideal network that behaves like two dedicated links running in opposite directions at the full link bandwidth between the two devices, the resulting effective bandwidth is:

$$\text{Effective bandwidth} = \min(2 \times BW_{\text{LinkInjection}}, 2 \times BW_{\text{LinkReception}}) = \frac{2 \times \text{Packet size}}{\max(\text{Overhead}, \text{Transmission time})}$$

where Overhead = max(Sending overhead, Receiving overhead).

Packet size and effective bandwidth

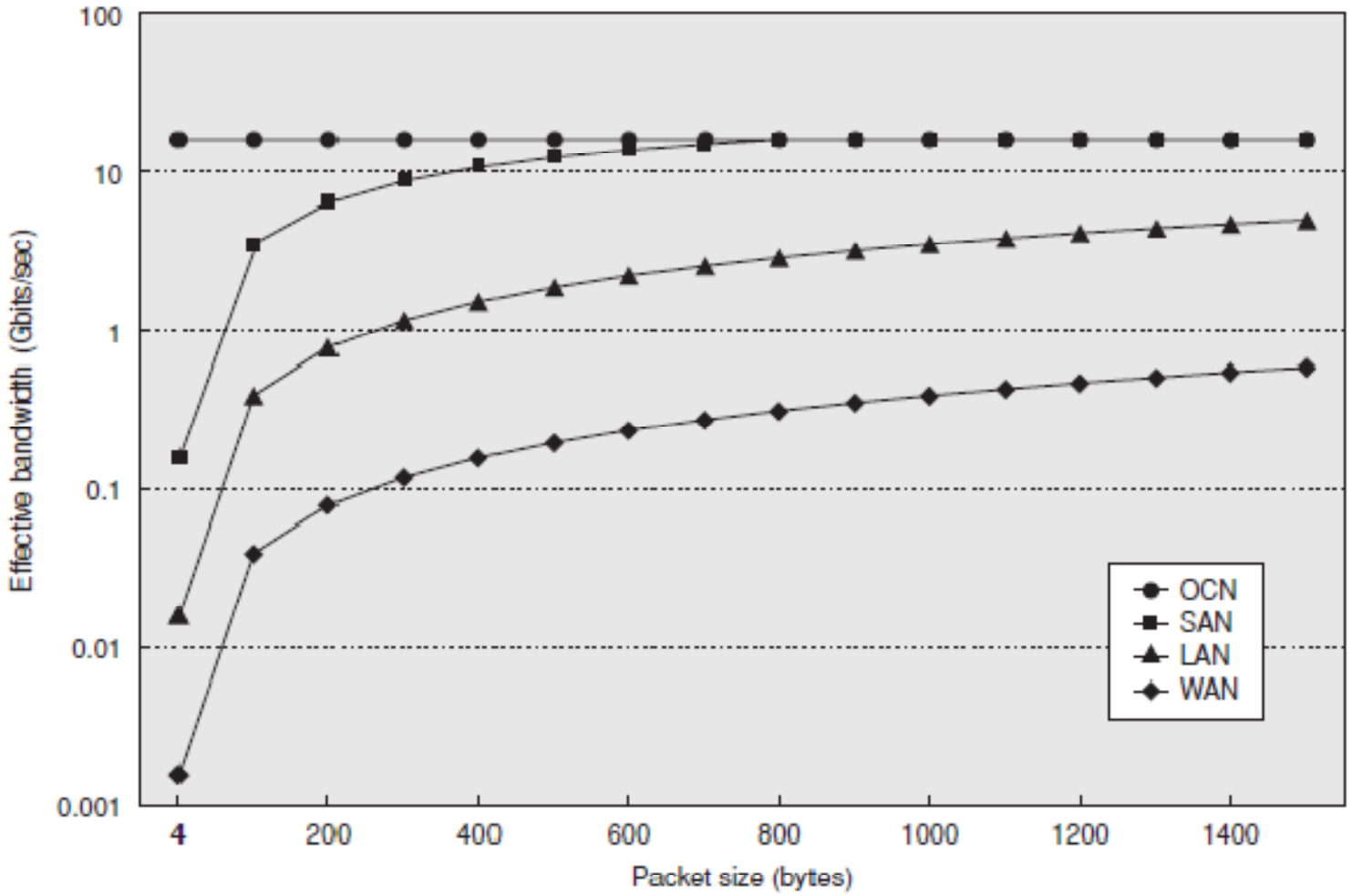


Figure E.6 Effective bandwidth versus packet size plotted in semi-log form for the four network domains. Overhead can be amortized by increasing the packet size, but for too large of an overhead (e.g., for WANs and some LANs) scaling the packet size is of little help. Other considerations come into play that limit the maximum packet size.

Comparing Systems

Company	System [network] name	Intro year	Max. number of compute nodes [× # CPUs]	System footprint for max. configuration	Packet [header] max size (bytes)	Injection [reception] node BW in MB/sec	Minimum send/ receive overhead	Maximum copper link length; flow control; error
Intel	ASCI Red Paragon	2001	4510 [× 2]	2500 sq. feet	1984 [4]	400 [400]	few µs	handshaking; CRC + parity
IBM	ASCI White SP Power3 [Colony]	2001	512 [× 16]	10,000 sq. feet	1024 [6]	500 [500]	~ 3 µs	25 m; credit-based; CRC
Intel	Thunder Itanium2 Tiger4 [QsNet^{II}]	2004	1024 [× 4]	120 m^2	2048 [14]	928 [928]	0.240 µs	13 m; credit-based; CRC for link, dest.
Cray	XT3 [SeaStar]	2004	30,508 [× 1]	263.8 m^2	80 [16]	3200 [3200]	few µs	7 m; credit-based; CRC
Cray	X1E	2004	1024 [× 1]	27 m^2	32 [16]	1600 [1600]	0 (direct LD ST accesses)	5 m; credit-based; CRC
IBM	ASC Purple pSeries 575 [Federation]	2005	>1280 [× 8]	6720 sq. feet	2048 [7]	2000 [2000]	~ 1 µs with up to 4 packets processed in ||	25 m; credit-based; CRC
IBM	Blue Gene/L eServer Sol. [Torus Net.]	2005	65,536 [× 2]	2500 sq. feet (.9 × .9 × 1.9 m^3/1K node rack)	256 [8]	612.5 [1050]	~ 3 µs (2300 cycles)	8.6 m; credit-based; CRC (header/pkt)

Figure E.7 Basic characteristics of interconnection networks in commercial high-performance computer systems.

Outline

- Introduction
- Interconnecting Two Devices
- Connecting More than Two Devices
- Network Topology
- Switch Microarchitecture
- Practical Issues for Commercial Interconnection Networks
- Examples of Interconnection Networks
- Crosscutting Issues for Interconnection Networks

Connecting More than Two Devices

- To this point, we have considered the connection of only two devices communicating over a network viewed as a black box, but what makes interconnection networks interesting is the ability to connect hundreds or even many thousands of devices together.
- Consequently, what makes them interesting also makes them more challenging to build.
- In order to connect more than two devices, a suitable structure and more functionality must be supported by the network.
- We classify networks into two broad categories based on their connection structure — *shared-media* versus *switched-media* networks.

Routing

- Every network that interconnects more than two devices also requires some mechanism to deliver each packet to the correct destination.
- The associated function is referred to as *routing*, which can be defined as the set of operations that need to be performed to compute a valid path from the packet source to its destination.
- Routing addresses the important issue of: *"Which of the possible paths are allowable (valid) for packets?"*

Arbitration

- In general, as networks usually contain shared paths or parts thereof among different pairs of devices, packets may request some shared resources.
- When several packets request the same resources at the same time, an *arbitration* function is required to resolve the conflict.
- Arbitration, along with flow control, addresses the important issue of *"When are paths available for packets?"*

Switching

- Every time arbitration is performed, there is a winner and, possibly, several losers.
- The losers are not granted access to the requested resources and are typically buffered.
- The winner proceeds toward its destination once the granted resources are switched in, providing a path for the packet to advance.
- This function is referred to as *switching*.
- Switching addresses the important issue of *"How are paths allocated to packets?"*

Network Topology

- When multiple devices are interconnected by a network, the connections between them oftentimes cannot be permanently established with dedicated links.
- This could either be too restrictive or prohibitively expensive as a dedicated link would be needed from every source to every destination.
- Therefore, networks usually share paths among different pairs of devices, but how those paths are shared is determined by the network connection structure, commonly referred to as the network *topology*.
- Topology addresses the important issue of *"What paths are possible for packets?"* in order for packets to reach their intended destination.

Shared-Media Networks

- The simplest way to connect multiple devices is to have them share the network media.
- This has been the traditional way of interconnecting devices.
- The shared media can operate in:
  - *half-duplex* mode, where data can be carried in either direction over the media but simultaneous transmission and reception of data by the same device is not allowed, or in
  - *Full-duplex*,where the data can be carried in both directions and simultaneously transmitted and received by the same device.

Switched-Media Networks

- The alternative to sharing the entire network media at once across all attached nodes is to switch between disjoint portions of it shared by the nodes.
- Those portions consist of passive *point-to-point links* between active *switch* components that dynamically establish communication between sets of source-destination pairs.
- These passive and active components make up what is referred to as the network *switch fabric* or *network fabric*, to which end nodes are connected.

Shared-media Vs. Switched-media

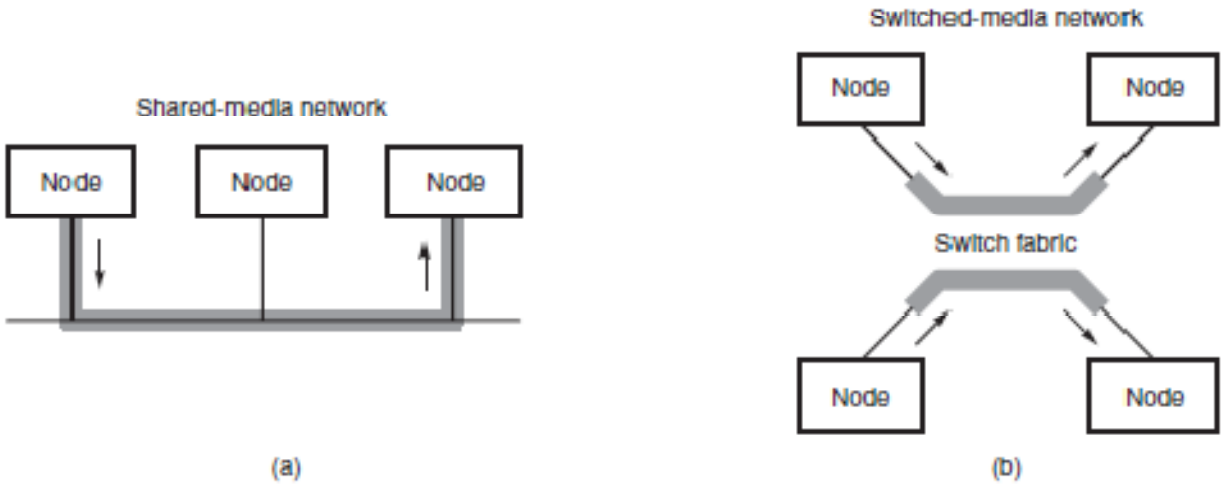


Figure E.8 (a) A shared-media network versus (b) a switched-media network. Ethernet was originally a shared media network, but switched Ethernet is now available. All nodes on the shared-media must dynamically share the raw bandwidth of one link, but switched-media networks can support multiple links, providing higher raw aggregate bandwidth.

Shared-media Vs. Switched-media

- There is a potential bandwidth improvement of switched-media networks over shared-media networks:
  - aggregate bandwidth can be many times higher than that of shared-media networks, allowing the possibility of greater effective bandwidth to be achieved.
- At best, only one node at a time can transmit packets over the shared media, whereas it is possible for all attached nodes to do so over the switched-media network.

Shared-media Vs. Switched-media

- In general, the advantage of shared-media networks is their low cost, but, consequently, their aggregate network bandwidth does not scale at all with the number of interconnected devices.
- The main advantage of switched-media networks is that the amount of network resources implemented scales with the number of connected devices, increasing the aggregate network bandwidth.
- Also, switched-media networks allow the system to scale to very large numbers of nodes, which is not feasible when using shared media.

Characterizing Performance: Latency and Effective Bandwidth

- The routing, switching, and arbitration functionality described above introduces some additional components of packet transport latency that must be taken into account in the expression for total packet latency.
- The total packet latency is given by the following:

$$\text{Latency} = \text{Sending overhead} + (T_{\text{TotalProp}} + T_{\text{R}} + T_{\text{A}} + T_{\text{S}}) + \frac{\text{Packet size}}{\text{Bandwidth}} + \text{Receiving overhead}$$

where T_{R}, T_{A}, and T_{S} are the total routing time, arbitration time, and switching time experienced by the packet, respectively, and are either measured quantities or calculated quantities derived from more detailed analyses.

- These components are added to the total propagation delay through the network links, $T_{\text{TotalProp}}$, to give the overall time of flight of the packet.

Latency Vs. Number of Nodes

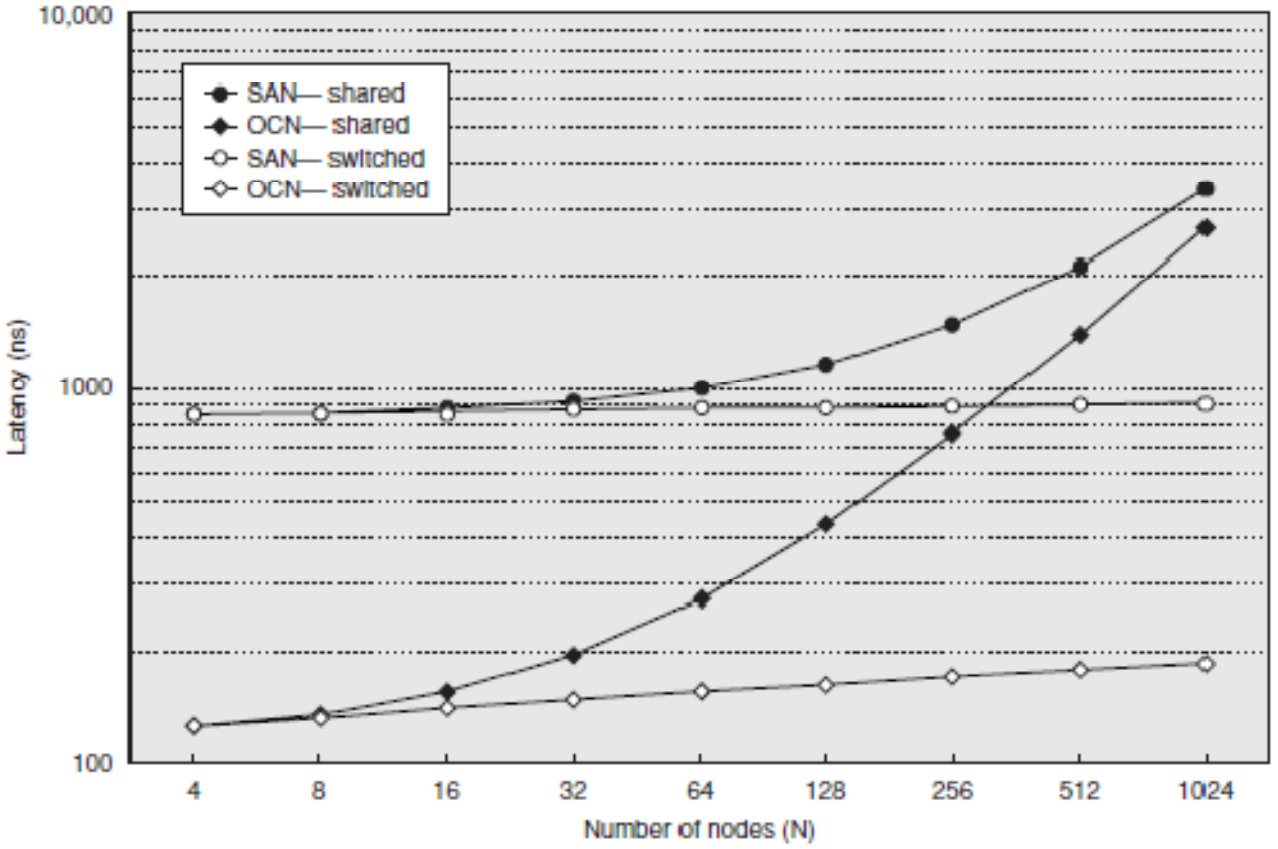


Figure E.9 Latency versus number of interconnected nodes plotted in semi-log form for OCNs and SANs. Routing, arbitration, and switching have more of an impact on latency for networks in these two domains, particularly for networks with a large number of nodes, given the low sending and receiving overheads and low propagation delay.

Effective bandwidth versus number of interconnected nodes

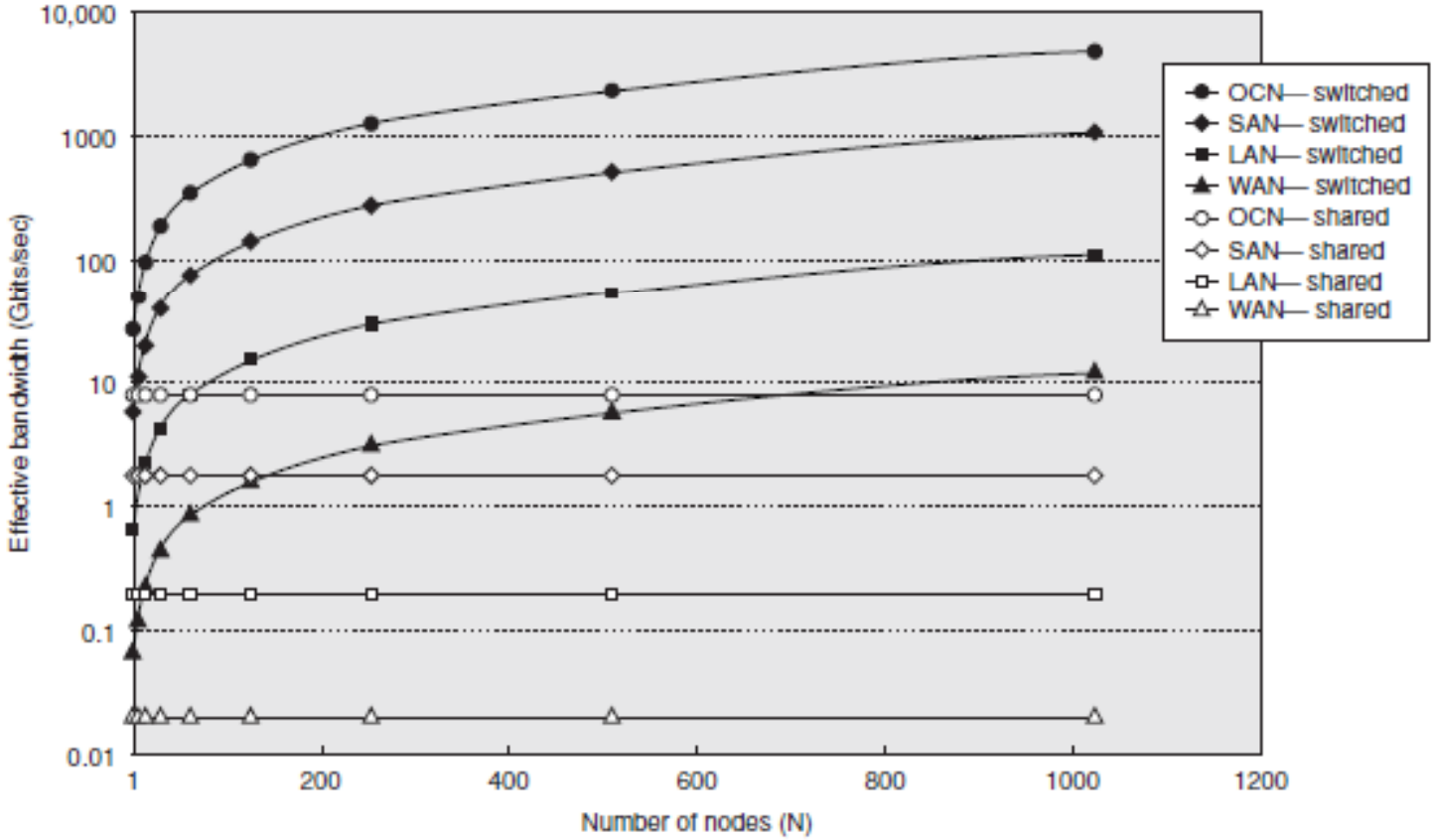


Figure E.10 Effective bandwidth versus number of interconnected nodes plotted in semi-log form for the four network domains. The disparity in effective bandwidth between shared- and switched-media networks for all interconnect domains widens significantly as the number of nodes in the network increases. Only the switched on-chip network is able to achieve an effective bandwidth equal to the aggregate bandwidth for the parameters given in this example.

Outline

- Introduction
- Interconnecting Two Devices
- Connecting More than Two Devices
- Network Topology
- Switch Microarchitecture
- Practical Issues for Commercial Interconnection Networks
- Examples of Interconnection Networks
- Crosscutting Issues for Interconnection Networks

Number of Switches

- When the number of devices is small enough, a single switch is sufficient to interconnect them within a switched-media network.
- However, the number of *switch ports* is limited by existing VLSI technology, cost considerations, power consumption, and so on.
- When the number of required *network ports* exceeds the number of ports supported by a single switch, a fabric of interconnected switches is needed.
- To embody the necessary property of *full access* (i.e., *connectedness*), the network switch fabric must provide a path from every end node device to every other device.

Interconnection structure

- All the connections to the network fabric and between switches within the fabric use point-to-point links as opposed to shared links — that is, links with only one switch or end node device on either end.
- The interconnection structure across all the components — including switches, links, and end node devices — is referred to as the *network topology*.

Centralized Switched Networks

- A single switch suffices to interconnect a set of devices when the number of switch ports is equal to or larger than the number of devices.
- This simple network is usually referred to as a *crossbar* or *crossbar switch*.
- Within the crossbar, crosspoint switch complexity increases quadratically with the number of ports.

Centralized Switched Networks

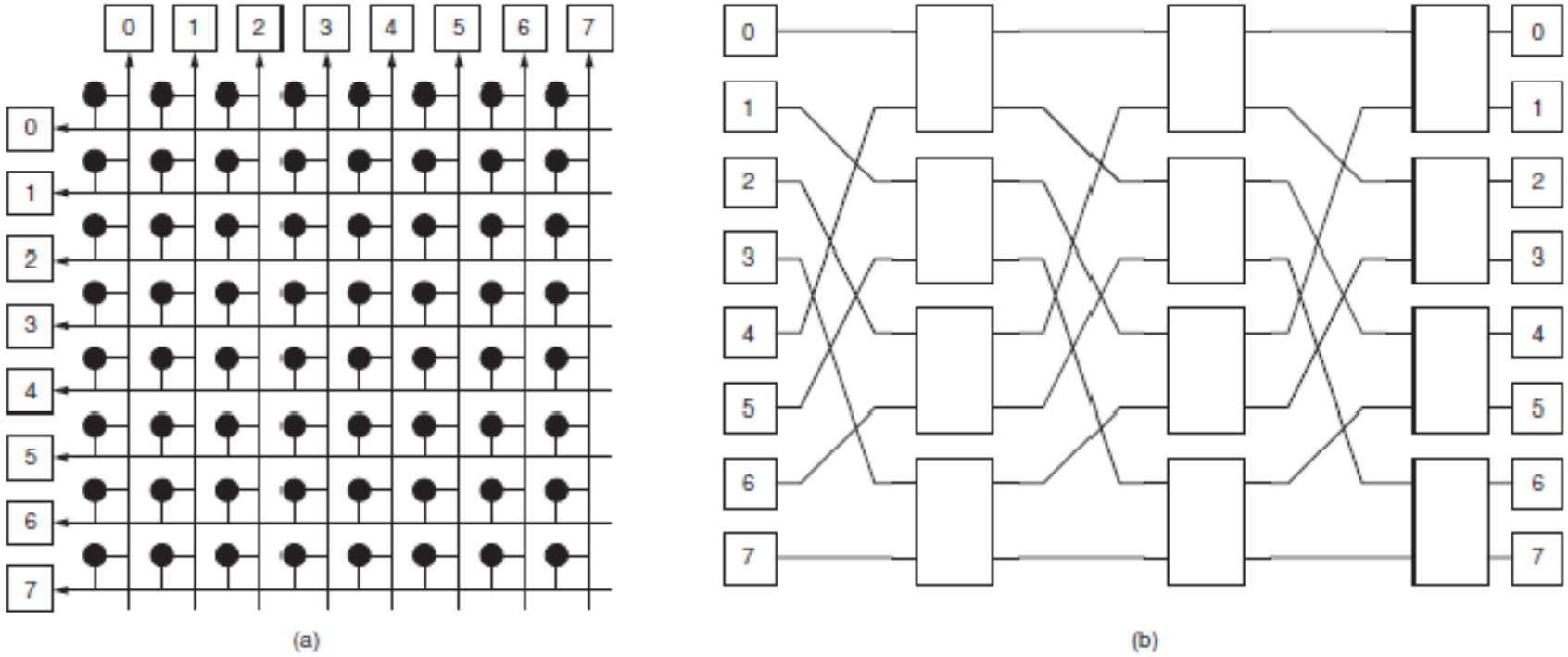


Figure E.11 Popular centralized switched networks: (a) the crossbar network requires N^2 crosspoint switches, shown as black dots; (b) the Omega, a MIN, requires $N/2 \log_2 N$ switches, shown as vertical rectangles. End node devices are shown as numbered squares (total of eight). Links are unidirectional—data enter at the left and exit out the top or right.

Distributed Switched Networks

- Switched-media networks provide a very flexible framework to design communication subsystems external to the devices that need to communicate.
- However, there are cases where it is convenient to more tightly integrate the end node devices with the network resources used to enable them to communicate.
- Instead of centralizing the switch fabric in an external subsystem, an alternative approach is to distribute the network switches among the end nodes, which then become *network nodes* or simply *nodes*, yielding a *distributed switched network*.

Fully connected topology

- A quite obvious way of interconnecting nodes consists of connecting a dedicated link between each node and every other node in the network.
- This *fully connected* topology provides the best connectivity (full connectivity in fact), but it is more costly than a crossbar network.
- A lower-cost alternative to fully connecting all nodes in the network is to directly connect nodes in sequence along a *ring* topology.

Rings

- Rings can allow many simultaneous transfers:
  - the first node can send to the second while the second sends to the third, and so on.
- However, as dedicated links do not exist between logically nonadjacent node pairs, packets must hop across intermediate nodes before arriving at their destination, increasing their transport latency.
- For bidirectional rings, packets can be transported in either direction, with the shortest path to the destination usually being the one selected.

Ring network topology,

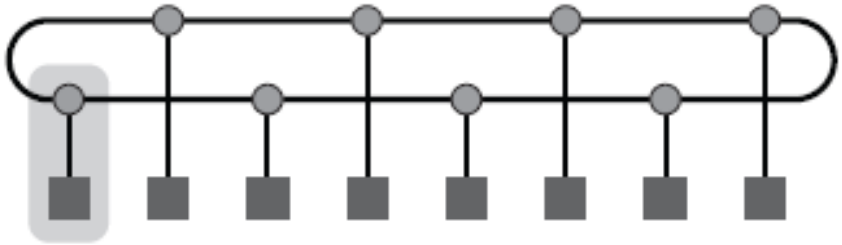


Figure E.13 A ring network topology, folded to reduce the length of the longest link. Shaded circles represent switches, and black squares represent end node devices. The gray rectangle signifies a network node consisting of a switch, a device, and its connecting link.

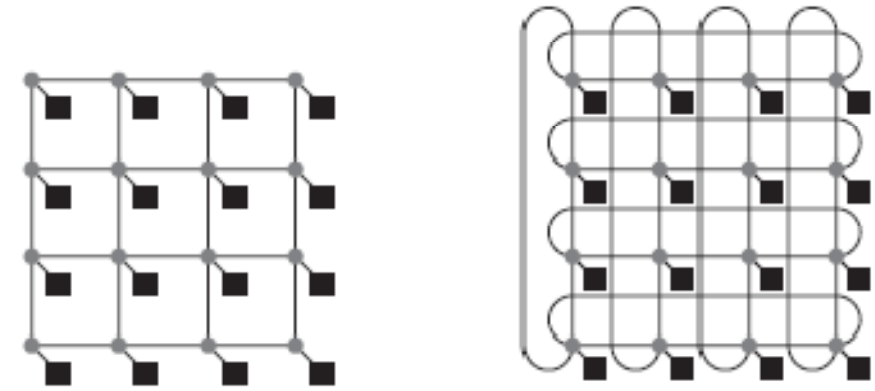
Ideal switched-media topology

- Fully connected and ring connected networks delimit the two extremes of distributed switched topologies, but there are many points of interest in between for a given set of cost-performance requirements.
- Generally speaking, the ideal switched-media topology has cost approaching that of a ring but performance approaching that of a fully connected topology.

Mesh and Torus Topology

- In the *mesh* or *grid* topology, all the nodes in each dimension form a linear array.
- In the *torus* topology, all the nodes in each dimension form a ring.
- Both of these topologies provide direct communication to neighboring nodes with the aim of reducing the number of hops suffered by packets in the network with respect to the ring.

Popular direct network topologies



(a) 2D grid or mesh of 16 nodes

(b) 2D torus of 16 nodes

(c) Hypercube of 16 nodes ($16 = 2^4$ so $n = 4$)

Figure E.14 Direct network topologies that have appeared in commercial systems, mostly supercomputers. The shaded circles represent switches, and the black squares represent end node devices. Switches have many bidirectional network links, but at least one link goes to the end node device. These basic topologies can be supplemented with extra links to improve performance and reliability. For example, connecting the switches on the periphery of the 2-D mesh using the unused ports on each switch forms a 2-D torus. The hypercube topology is an n-dimensional interconnect for 2^n nodes, requiring $n + 1$ ports per switch: one for the n nearest neighbor nodes and one for the end node device.

Performance and Cost

Evaluation category	Bus	Ring	2D mesh	2D torus	Hypercube	Fat tree	Fully connected
Performance							
$BW_{Bisection}$ 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

Figure E.15 Performance and cost of several network topologies for 64 nodes. The bus is the standard reference at unit network link cost and bisection bandwidth. Values are given in terms of bidirectional links and ports. Hop count includes a switch and its output link, but not the injection link at end nodes. Except for the bus, values are given for the number of network links and total number of links, including injection/reception links between end node devices and the network.

Packet latency

- If the average packet has to traverse *d* hops to its destination, then $T_R + T_A + T_S = (T_r + T_a + T_s) \times d$, where T_r, T_a, and T_s are the routing, arbitration, and switching delays, respectively, of a switch.

- The packet latency in the network is:

$$\text{Latency} = \text{Sending overhead} + T_{\text{LinkProp}} \times (d+1) + (T_r + T_a + T_s) \times \text{d} + \frac{\text{Packet size}}{\text{Bandwidth}} \times (d+1) + \text{Receiving overhead}$$

Topology in high performance commercial machines

Company	System [network] name	Max. number of nodes [× # CPUs]	Basic network topology	Injection [reception] node BW in MB/sec	# of data bits per link per direction	Raw network link BW per direction in MB/sec	Raw network bisection BW (bidirectional) in GB/sec
Intel	ASCI Red Paragon	4816 [× 2]	2D mesh 64 × 64	400 [400]	16 bits	400	51.2
IBM	ASCI White SP Power3 [Colony]	512 [× 16]	bidirectional MIN with 8-port bidirectional switches (typically a fat tree or Omega)	500 [500]	8 bits (+ 1 bit of control)	500	256
Intel	Thunder Itanium2 Tiger4 [QsNet^{II}]	1024 [× 4]	fat tree with 8-port bidirectional switches	928 [928]	8 bits (+ 2 of control for 4b/5b encoding)	1333	1365
Cray	XT3 [SeaStar]	30,508 [× 1]	3D torus 40 × 32 × 24	3200 [3200]	12 bits	3800	5836.8
Cray	X1E	1024 [× 1]	4-way bristled 2D torus (~ 23 × 11) with express links	1600 [1600]	16 bits	1600	51.2
IBM	ASC Purple pSeries 575 [Federation]	>1280 [× 8]	bidirectional MIN with 8-port bidirectional switches (typically a fat tree or Omega)	2000 [2000]	8 bits (+ 2 bits of control for novel 5b/6b encoding scheme)	2000	2560
IBM	Blue Gene/L eServer Sol. [Torus Net.]	65,536 [× 2]	3D torus 32 × 32 × 64	612.5 [1050]	1 bit (bit serial)	175	358.4

Figure E.16 Topological characteristics of interconnection networks used in commercial high-performance machines.

Outline

- Introduction
- Interconnecting Two Devices
- Connecting More than Two Devices
- Network Topology
- Switch Microarchitecture
- Practical Issues for Commercial Interconnection Networks
- Examples of Interconnection Networks
- Crosscutting Issues for Interconnection Networks

Basic Switch Microarchitecture

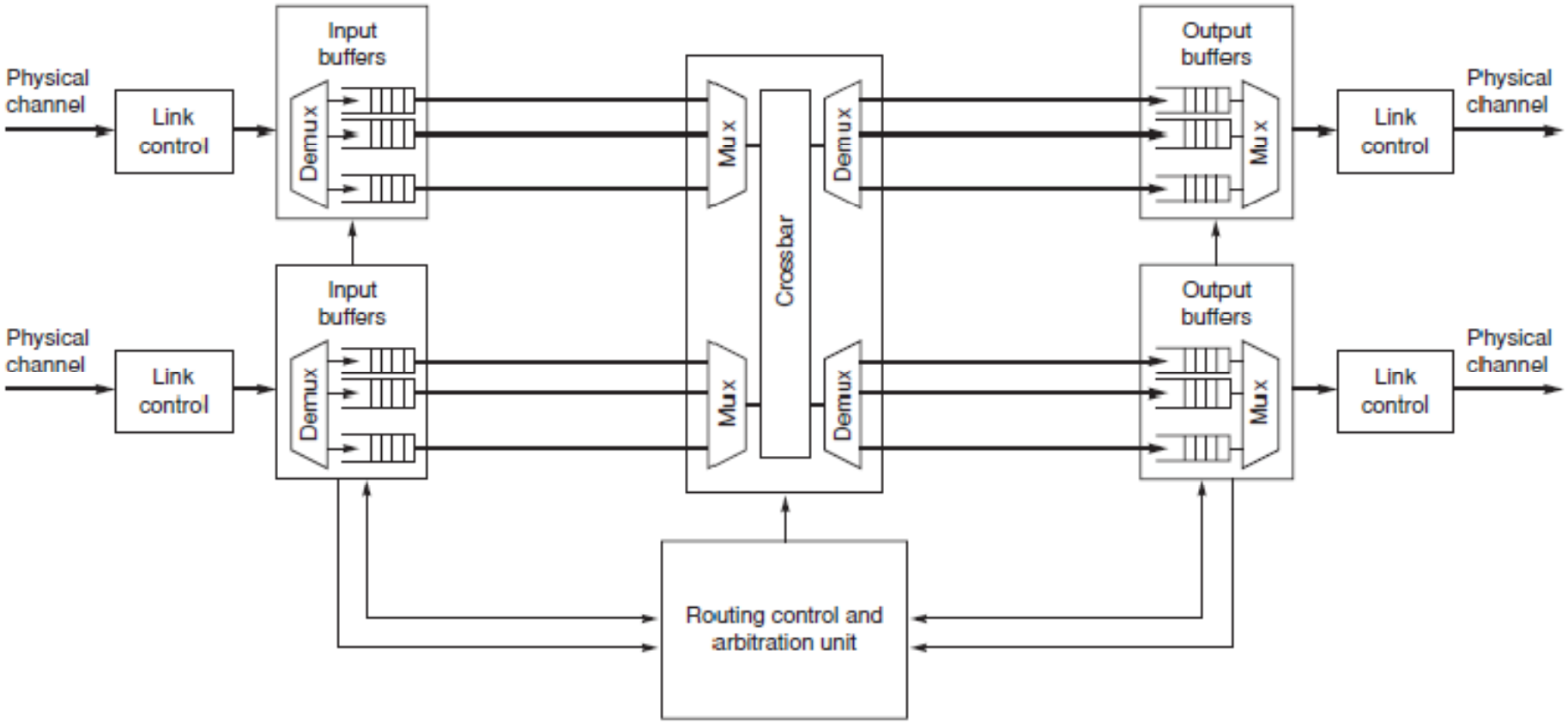


Figure E.21 Basic microarchitectural components of an input-output-buffered switch.

Pipelining the Switch Microarchitecture

- Performance can be enhanced by pipelining the switch microarchitecture.
- Pipelined processing of packets in a switch has similarities with pipelined execution of instructions in a SIMD.
- In a SIMD pipeline, a single instruction indicates what operation to apply to all the vector elements executed in a pipelined way.
- Similarly, in a switch pipeline, a single packet header indicates how to process all of the internal data path physical transfer units (or *phits*) of a packet, which are processed in a pipelined fashion.
- Also, as packets at different input ports are independent of each other, they can be processed in parallel similar to the way multiple independent instructions or threads of pipelined instructions can be executed in parallel.

Pipelined version of the switch

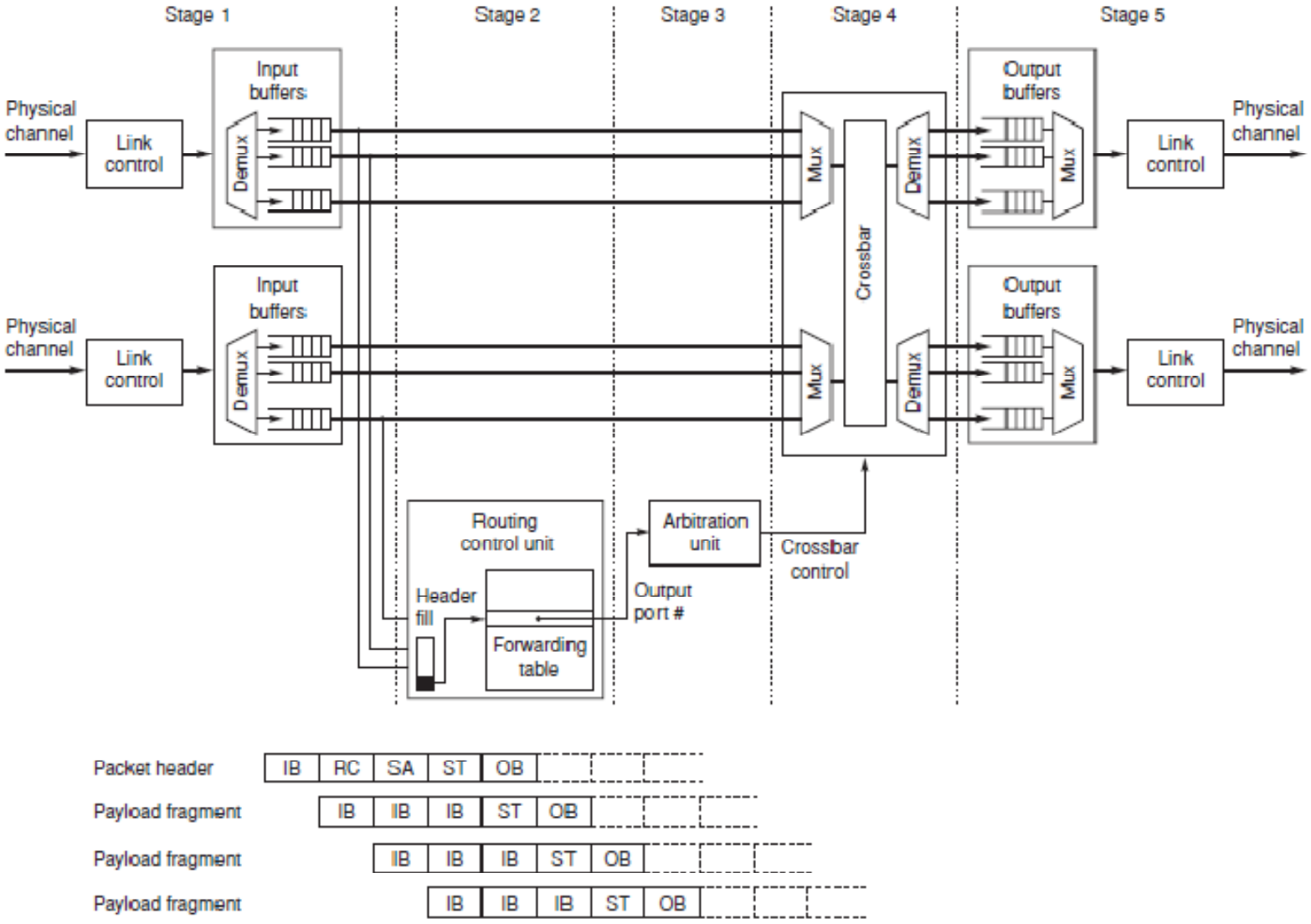


Figure E.23 Pipelined version of the basic input-output-buffered switch. The notation in the figure is as follows: IB is the input link control and buffer stage, RC is the route computation stage, SA is the crossbar switch arbitration stage, ST is the crossbar switch traversal stage, and OB is the output buffer and link control stage. Packet fragments (flits) coming after the header remain in the IB stage until the header is processed and the crossbar switch resources are provided.

Outline

- Introduction
- Interconnecting Two Devices
- Connecting More than Two Devices
- Network Topology
- Switch Microarchitecture
- Practical Issues for Commercial Interconnection Networks
- Examples of Interconnection Networks
- Crosscutting Issues for Interconnection Networks

Practical Issues for Commercial Interconnection Networks

- There are practical issues in addition to the technical issues described thus far that are important considerations for interconnection networks within certain domains.

Connectivity

- Among some of the issues are the following:
  - How lightweight should the network interface hardware/software be?
  - Should it attach to the memory network or the I/O network?
  - Should it support cache coherence?
  - If the operating system must get involved for every network transaction, the sending and receiving overhead becomes quite large.
  - If the network interface attaches to the I/O network (PCI-Express or HyperTransport interconnect), the injection and reception bandwidth will be limited to that of the I/O network.
- This is the case for the Cray XT3 SeaStar, Intel Thunder Tiger 4 QsNetII, and many other supercomputer and cluster networks.

Connectivity

- Computer systems typically have a multiplicity of interconnects with different functions and cost-performance objectives:
- The personal computer in 2006 had a processor-memory interconnect and an I/O interconnect (e.g., PCI-X 2.0, PCIe or Hyper- Transport) designed to connect both fast and slow devices (e.g., USB 2.0, Gigabit Ethernet LAN, Firewire 800, etc.).
- The Blue Gene/L supercomputer uses five interconnection networks, only one of which is the 3D torus used for most of the interprocessor application traffic.

On-chip networks

- The University of Texas at Austin's TRIPS Edge processor has eight specialized on-chip networks — some with bidirectional channels as wide as 128 bits and some with 168 bits in each direction — to interconnect the 106 heterogeneous tiles composing the two processor cores with L2 on-chip cache.
- It also has a chip-to-chip switched network to interconnect multiple chips in a multiprocessor configuration.
- Two of the on-chip networks are switched networks: one is used for operand transport and the other is used for on-chip memory communication.
- The portion of chip area allocated to the interconnect is substantial, with five of the seven metal layers used for global network wiring.

Standardization: Cross-Company Interoperability

- Standards are useful in many places in computer design, including interconnection networks.
- Advantages of successful standards include low cost and stability.
- One drawback of standards is the time it takes for committees and special interest groups to agree on the definition of standards, which is a problem when technology is changing rapidly.
- Another problem is *when* to standardize: on the one hand, designers would like to have a standard before anything is built; on the other hand, it would be better if something were built before standardization to avoid legislating useless features or omitting important ones.

Standardization

- LANs and WANs use standards and interoperate effectively.
- WANs involve many types of companies and must connect to many brands of computers, so it is difficult to imagine a proprietary WAN ever being successful.
- The ubiquitous nature of the Ethernet shows the popularity of standards for LANs as well as WANs,
- Some SANs are standardized such as Fibre Channel, but most are proprietary.
- OCNs for the most part are proprietary designs, with a few gaining widespread commercial use in system-on-chip (SoC) applications, such as IBM's CoreConnect and ARM's AMBA.

Outline

- Introduction
- Interconnecting Two Devices
- Connecting More than Two Devices
- Network Topology
- Switch Microarchitecture
- Practical Issues for Commercial Interconnection Networks
- Examples of Interconnection Networks
- Crosscutting Issues for Interconnection Networks

On-Chip Network

- With continued increases in transistor integration processor designers are under the gun to find ways of combating chip-crossing wire delay and other problems associated with deep submicron technology scaling.
- *Multicore* microarchitectures are gaining popularity, given their advantages of simplicity, modularity, and ability to exploit parallelism beyond that which can be achieved through aggressive pipelining and multiple instruction/data issuing on a single core.
- No matter whether the processor consists of a single core or multiple cores, higher and higher demands are being placed on intrachip communication bandwidth to keep pace — not to mention interchip bandwidth.

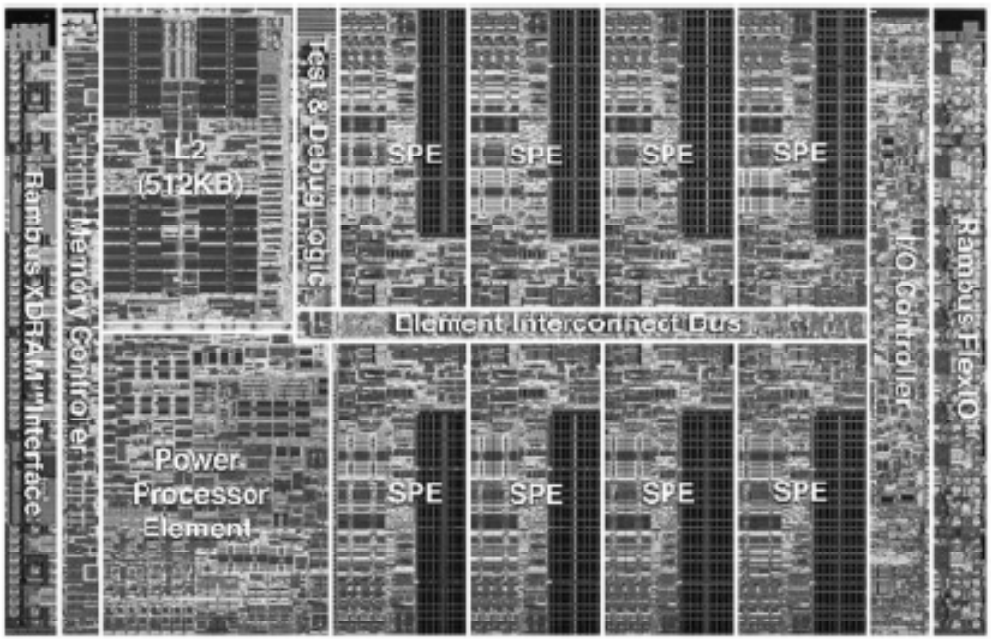
Cell Broadband Engine's Element Interconnect Bus

- On-chip networking has spurred a great amount of interest in OCN designs that efficiently support communication of instructions, register operands, memory, and I/O data within and between processor cores both on and off the chip.
- One of such onchip networks is: the Cell Broadband Engine's Element Interconnect Bus.

Cell Broadband Engine (Cell BE)

- The Cell Broadband Engine (Cell BE) is a heterogeneous multicore processor designed for high performance on multimedia and game applications requiring real-time responsiveness to users.
- Development of the processor started in 2000 by Sony, IBM, and Toshiba, with the first products shipped in 2005.
- The 200 GFLOPS (peak) Cell BE chip incorporates a 64-bit Power processor element (PPE), eight 128-bit SIMD synergistic processor elements (SPEs) with local store, a memory interface controller (MIC) element, and two configurable I/O interface elements—one of which supports a coherent protocol.
- Using the coherent I/O interface configurable to 20 GB/sec bandwidth, up to two Cell BEs can be directly connected or up to four Cell BEs can be assembled into a four-way symmetric multiprocessor system via an external switch.

Cell Broadband Engine



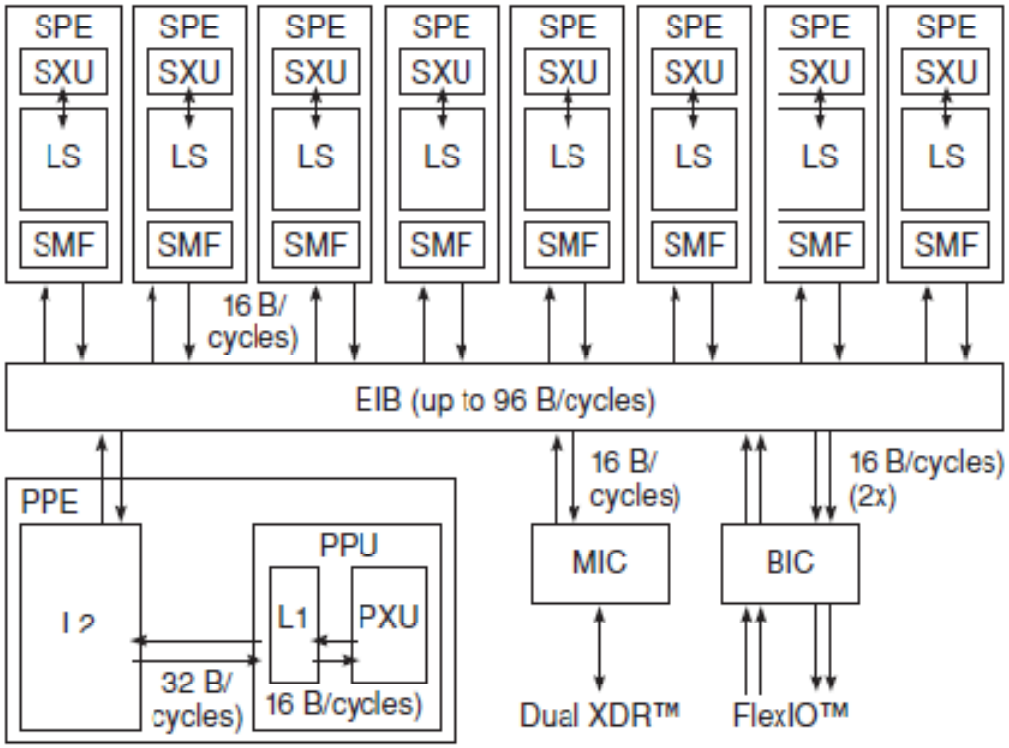


Figure E.25 Cell Broadband Engine (a) die photo and (b) high-level block diagram illustrating the function of the EIB.

On-chip networks

Institution and 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; # virtual channels
MIT Raw [General Dynamic Network]	2002	16 ports [16 tiles]	2D mesh 4 × 4	32 bits	0.9 GB/sec [225 MHz, clocked at proc speed]	XY DOR with request-reply deadlock recovery; RR arbitration; wormhole	6 layers; credit-based; no virtual channels
IBM Power5	2004	7 ports [2 PE cores + 5 other ports]	crossbar	256 bits Inst fetch; 64 bits for stores; 256 bits LDs	[1.9 GHz, clocked at proc speed]	shortest-path; nonblocking; circuit switch	7 layers; handshaking; no virtual channels
U.T. Austin TRIP Edge [Operand Network]	2005	25 ports [25 execution unit tiles]	2D mesh 5 × 5	110 bits	5.86 GB/sec [533 MHz clock scaled by 80%]	YX DOR; distributed RR arbitration; wormhole	7 layers; on/off flow control; no virtual channels
U.T. Austin TRIP Edge [On-Chip Network]	2005	40 ports [16 L2 tiles + 24 network interface tile]	2D mesh 10 × 4	128 bits	6.8 GB/sec [533 MHz clock scaled by 80%]	YX DOR; distributed RR arbitration; VCT switched	7 layers; credit-based flow control; 4 virtual channels
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 GB/sec [1.6 GHz, clocked at half the proc speed]	shortest-path; tree-based RR arbitration (centralized); pipelined circuit switch	8 layers; credit-based flow control; no virtual channels
Sun UltraSPARC T1 processor	2005	up to 13 ports [8 PE cores + 4 L2 banks + 1 shared I/O]	crossbar	128 bits both for the 8 cores and the 4 L2 banks	19.2 GB/sec [1.2 GHz, clocked at proc speed]	shortest-path; age-based arbitration; VCT switched	9 layers; handshaking; no virtual channels

Figure E.26 Characteristics of on-chip networks implemented in recent research and commercial processors. Some processors implement multiple on-chip networks (not all shown)—for example, two in the MIT Raw and eight in the TRIP Edge.

System Area Network: IBM Blue Gene/L 3D Torus Network

- The IBM BlueGene/L was the largest-scaled, highest-performing computer system in the world in 2005, according to *www.top500.org*.
- With 65,536 dual-processor compute nodes and 1024 I/O nodes, this 360 TFLOPS (peak) supercomputer has a system footprint of approximately 2500 square feet.
- Both processors at each node can be used for computation and can handle their own communication protocol processing in virtual mode or, alternatively, one of the processors can be used for computation and the other for network interface processing.
- Packets range in size from 32 bytes to a maximum of 256 bytes, and 8 bytes are used for the header.
- The header includes routing, virtual channel, link-level flow control, packet size, and other such information, along with 1 byte for CRC to protect the header.

System Area Network: IBM Blue Gene/L 3D Torus Network

- The main interconnection network is a proprietary 32 × 32 × 64 3D torus SAN that interconnects all 64K nodes.
- Each node switch has six 350 MB/sec bidirectional links to neighboring torus nodes, an injection bandwidth of 612.5 MB/sec from the two node processors, and a reception bandwidth of 1050 MB/ sec to the two node processors.

System/Storage Area Network: InfiniBand

- InfiniBand is an industry-wide de facto networking standard developed by a consortium of companies belonging to the InfiniBand Trade Association in October 2000.
- InfiniBand can be used as a system area network for interprocessor communication or as a storage area network for server I/O.
- It is a switch-based interconnect technology that provides flexibility in the topology, routing algorithm, and arbitration technique implemented by vendors and users.
- InfiniBand supports data transmission rates of 2–120 Gbp/link per direction across distances of 300 m.

Svstem Area Networks

Network name [vendors]	Used in top 10 supercom-puter clusters (2005)	Number of nodes	Basic network topology	Raw link bidirec-tional BW	Routing algorithm	Arbitration technique	Switching technique; flow control
InfiniBand [Mellanox, Voltair]	SGI Altrix and Dell Poweredge Thunderbird	> millions (2^{128} GUID addresses, like IPv6)	completely configurable (arbitrary)	4–240 Gbps	arbitrary (table-driven), typically up*/down*	weighted RR fair scheduling (2-level priority)	cut-through, 16 virtual channels (15 for data); credit-based
Myrinet-2000 [Myricom]	Barcelona Supercomputer Center in Spain	8192 nodes	bidirectional MIN with 16-port bidirectional switches (Clos net.)	4 Gbps	source-based dispersive (adaptive) minimal routing	round-robin arbitration	cut-through switching with no virtual channels; Xon/Xoff flow control
$QsNet^{II}$ [Quadrics]	Intel Thunder Itanium2 Tiger4	> tens of thousands	fat tree with 8-port bidirectional switches	21.3 Gbps	source-based LCA adaptive shortest-path routing	2-phased RR, priority, aging, distributed at output ports	wormhole with 2 virtual channels; credit-based

Figure E.28 Characteristics of system area networks implemented in various top 10 supercomputer clusters in 2005.

Ethernet: The Local Area Network

- Ethernet has been extraordinarily successful as a LAN—from the 10 Mbit/sec standard proposed in 1978 used practically everywhere today to the more recent 10 Gbit/sec standard that will likely be widely used.
- *Bridges* — These devices connect LANs together, passing traffic from one side to another depending on the addresses in the packet.
- Routers or *gateways* — These devices connect LANs to WANs, or WANs to WANs, and resolve incompatible addressing.

Bridge

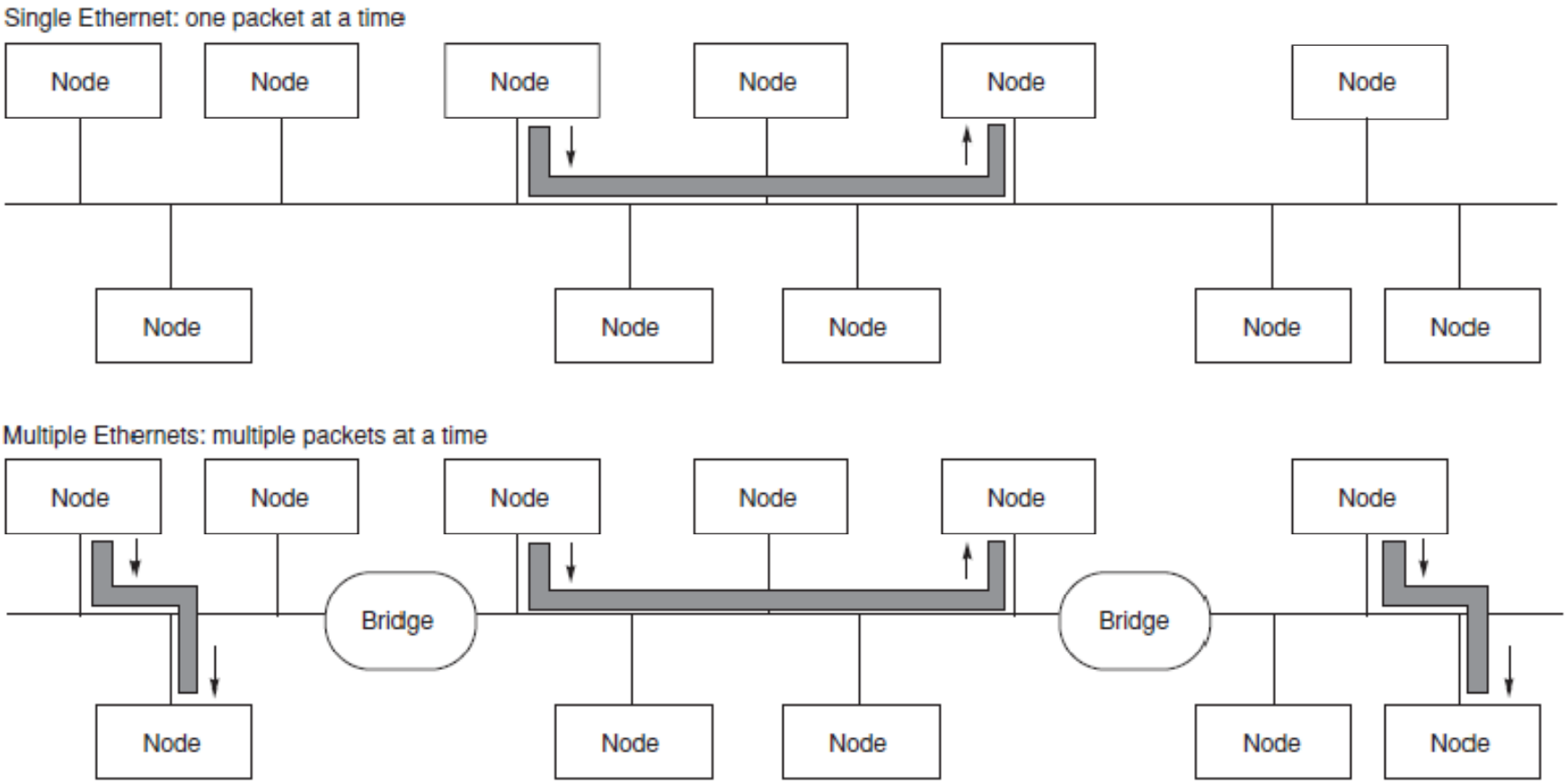


Figure E.30 The potential increased bandwidth of using many Ethernets and bridges.

Wide Area Network: ATM

- *Asynchronous Transfer Mode* (ATM) is a wide area networking standard set by the telecommunications industry.
- Although it flirted as competition to Ethernet as a LAN in the 1990s, ATM has since retreated to its WAN stronghold.
- The telecommunications standard has scalable bandwidth built in.
- It starts at 155 Mbits/sec, and scales by factors of 4 to 620 Mbits/sec, 2480 Mbits/sec, and so on. Since it is a WAN, ATM's medium is fiber, both single mode and multimode.

Outline

- Introduction
- Interconnecting Two Devices
- Connecting More than Two Devices
- Network Topology
- Switch Microarchitecture
- Practical Issues for Commercial Interconnection Networks
- Examples of Interconnection Networks
- Crosscutting Issues for Interconnection Networks

Migration from fewer “hotter” to many “cooler”

- The Google cluster is a prime example of this migration to many “cooler” processors versus fewer “hotter” processors.
- It uses racks of up to 80 Intel Pentium III 1 GHz processors instead of more power-hungry high-end processors.
- Other examples include blade servers consisting of 1-inch-wide by 7-inch-high rack unit blades designed based on mobile processors.
- The HP ProLiant BL10eG2 blade server supports up to 20 1 GHz ultra-low-voltage Intel Pentium M processors with a 400 MHz front-side bus, 1 MB L2 cache, and up to 1 GB memory.
- The Fujitsu Primergy BX300 blade server supports up to 20 1.4 or 1.6 GHz Intel Pentium M processors, each with 512 MB of memory expandable to 4 GB.

Efficient Interface to the Memory Hierarchy versus the Network

- Traditional evaluations of processor performance, such as SPECint and SPECfp, encourage integration of the memory hierarchy with the processor as the efficiency of the memory hierarchy translates directly into processor performance.
- Hence, microprocessors have multiple levels of caches on chip along with buffers for writes.
- Benchmarks such as SPECint and SPECfp do not reward good interfaces to interconnection networks, and hence, many machines make the access time to the network delayed by the full memory hierarchy.

End of Lecture

- Readings
  - Book: Appendix E.