Appendix F
Interconnection Networks

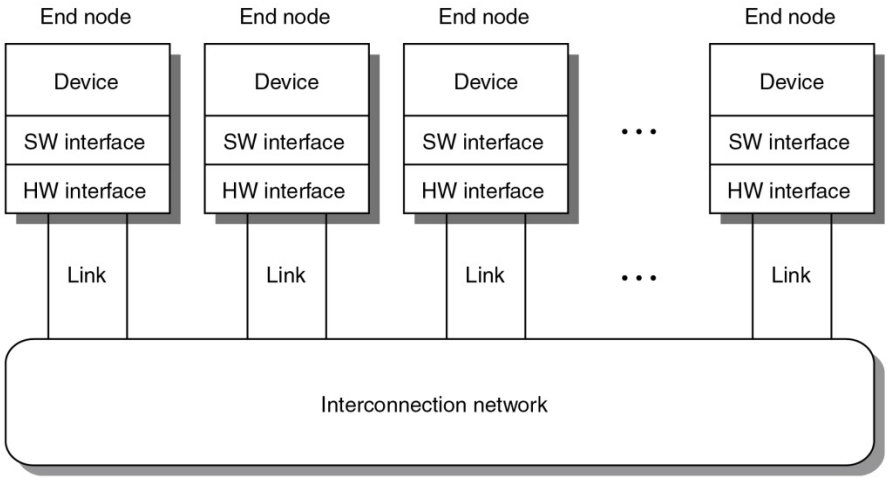


Figure F.1 A conceptual illustration of an interconnected community of devices.

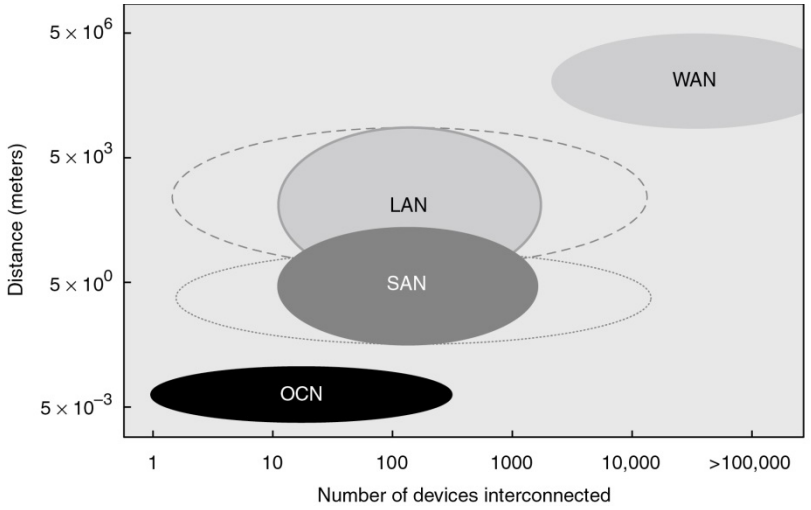


Figure F.2 Relationship of the four interconnection network domains in terms of number of devices connected and their distance scales: on-chip network (OCN), system/storage area network (SAN), local area network (LAN), and wide area network (WAN). Note that there are overlapping ranges where some of 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.

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

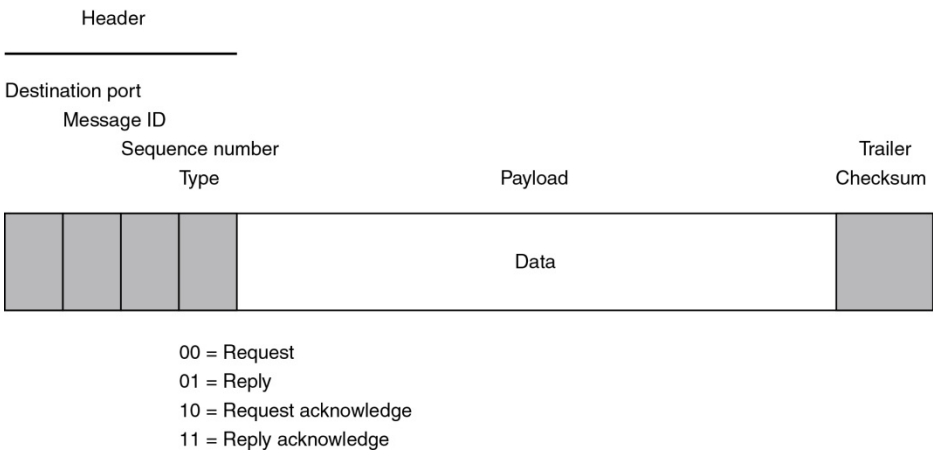


Figure F.4 An example packet format with header, payload, and checksum in the trailer.

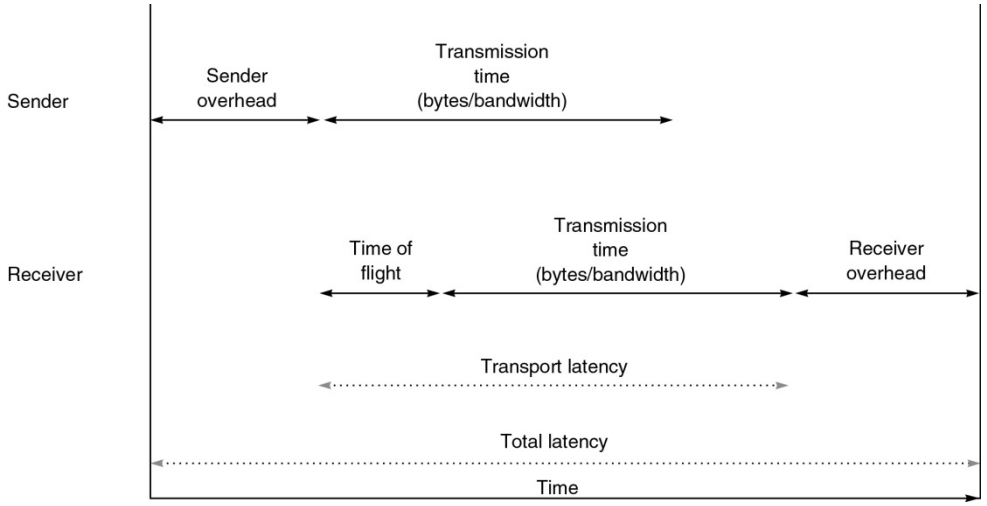


Figure F.5 Components of packet latency. Depending on whether it is an OCN, SAN, LAN, or WAN, the relative amounts of sending and receiving overhead, time of flight, and transmission time are usually quite different from those illustrated here.

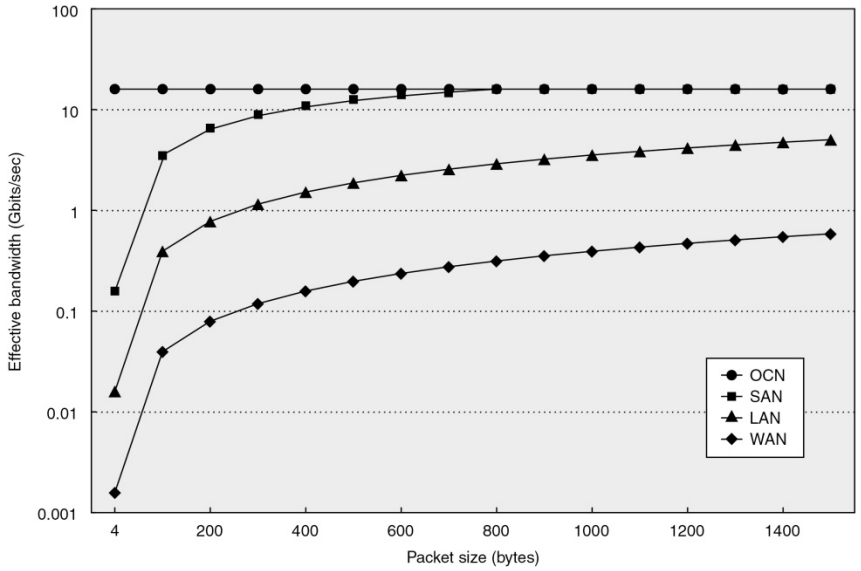


Figure F.6 Effective bandwidth versus packet size plotted in semi-log form for the four network domains. Overhead can be amortized by increasing the packet size, but for too large of an overhead (e.g., for WANs and some LANs) scaling the packet size is of little help. Other considerations come into play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

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 ft^2	1984 [4]	400 [400]	Few µs	Handshaking; CRC + parity
IBM	ASCI White SP Power3 [Colony]	2001	512 [×16]	10,000 ft^2	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 ft^2	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 ft^2 (.9 × .9 × 1.9 m^3/ 1 K node rack)	256 [8]	612.5 [1050]	~3 µs (2300 cycles)	8.6 m; credit-based; CRC (header/pkt)

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

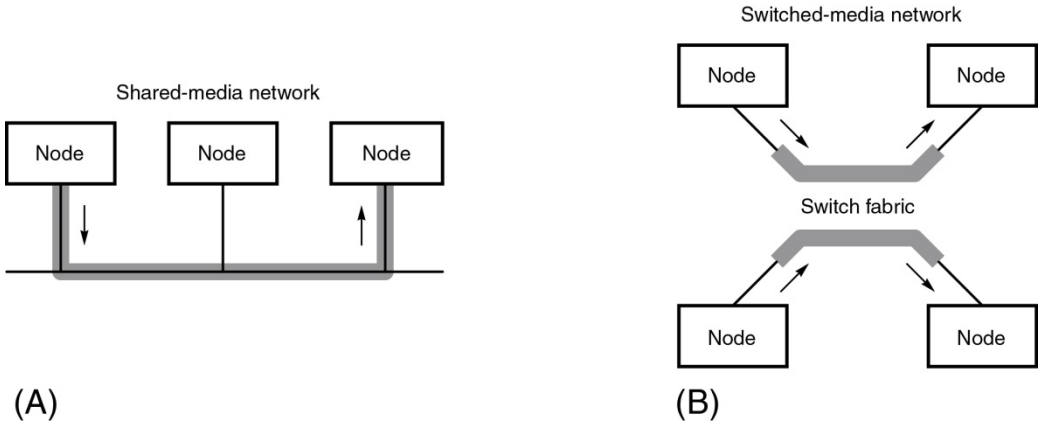


Figure F.8 (a) A shared-media network versus (b) a switched-media network. Ethernet was originally a shared media network, but switched Ethernet is now available. All nodes on the shared-media networks must dynamically share the raw bandwidth of one link, but switched-media networks can support multiple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

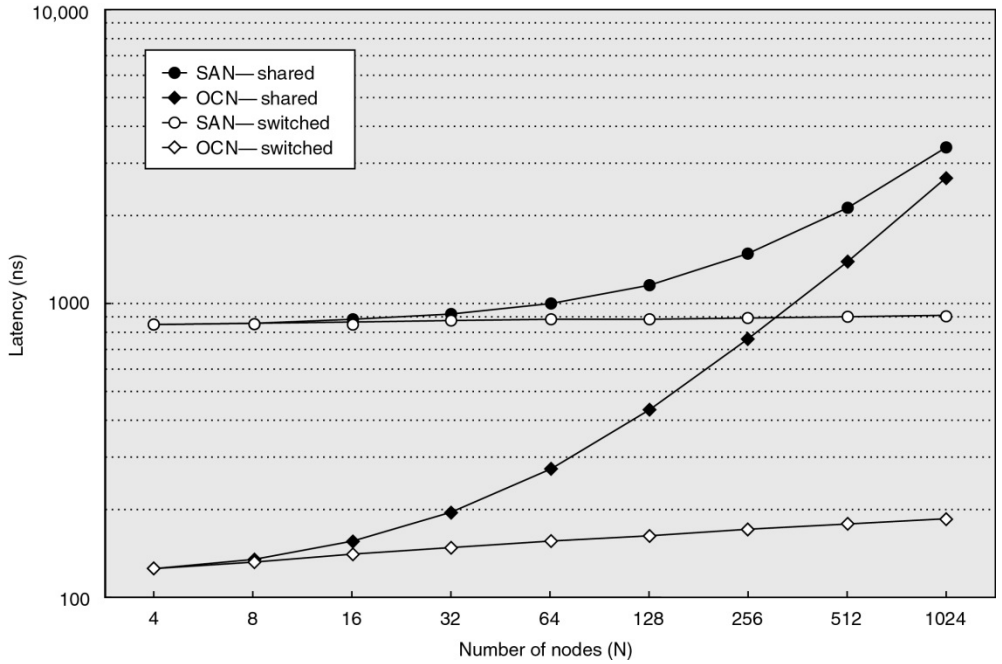


Figure F.9 Latency versus number of interconnected nodes plotted in semi-log form for OCNs and SANs. Routing, arbitration, and switching have more of an impact on latency for networks in these two domains, particularly for networks with a large number of nodes, given the low sending and receiving overheads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

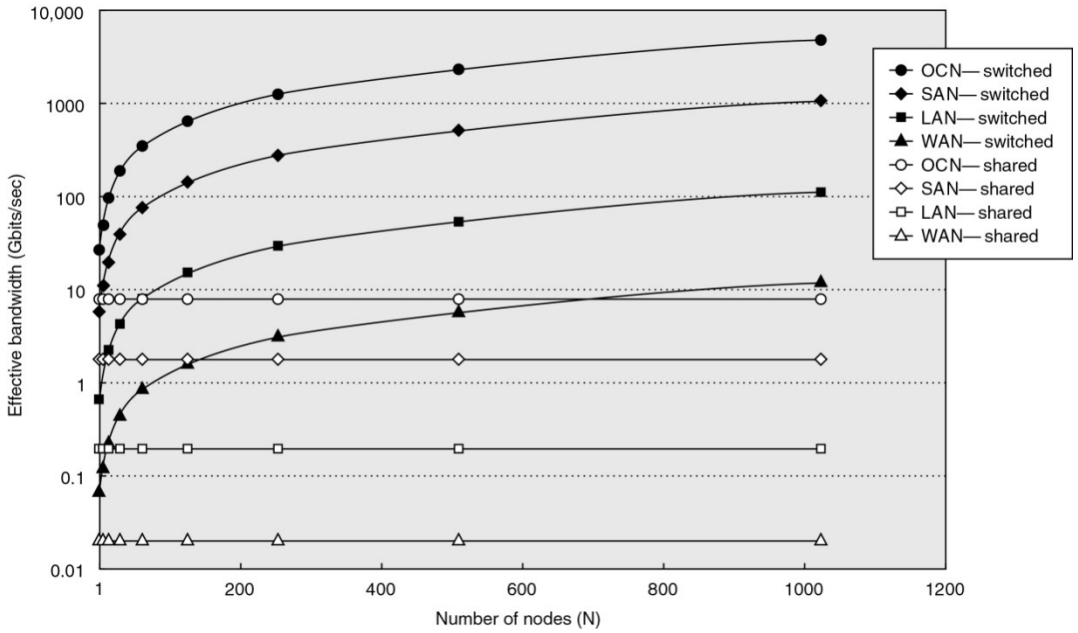


Figure F.10 Effective bandwidth versus number of interconnected nodes plotted in semi-log form for the four network domains. The disparity in effective bandwidth between shared- and switched-media networks for all interconnect domains widens significantly as the number of nodes in the network increases. 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.

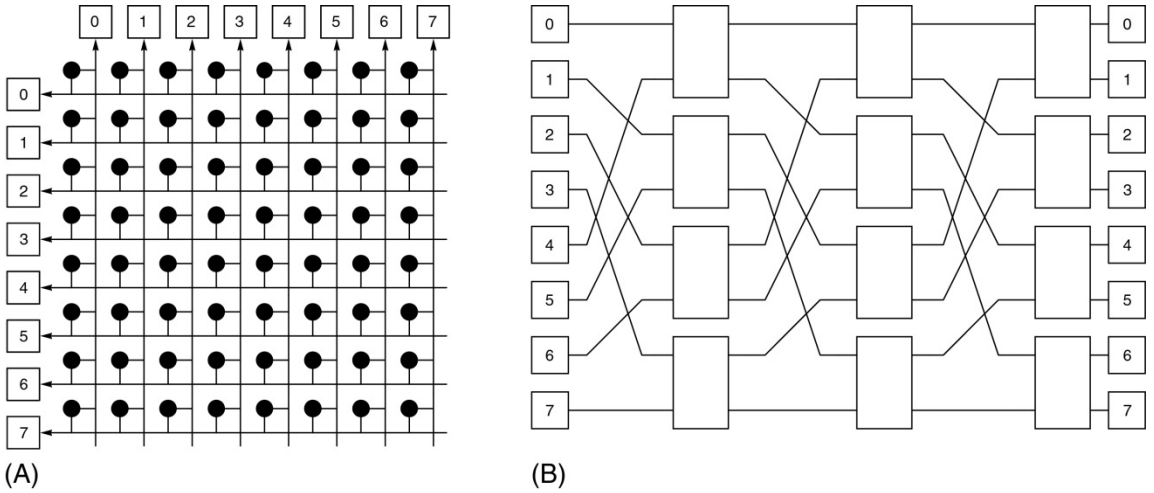


Figure F.11 Popular centralized switched networks: (a) the crossbar network requires N^2 crosspoint switches, shown as black dots; (b) the Omega, a MIN, requires $N/2 \log_2 N$ switches, shown as vertical rectangles. End node devices are shown as numbered squares (total of eight). Links are unidirectional—data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

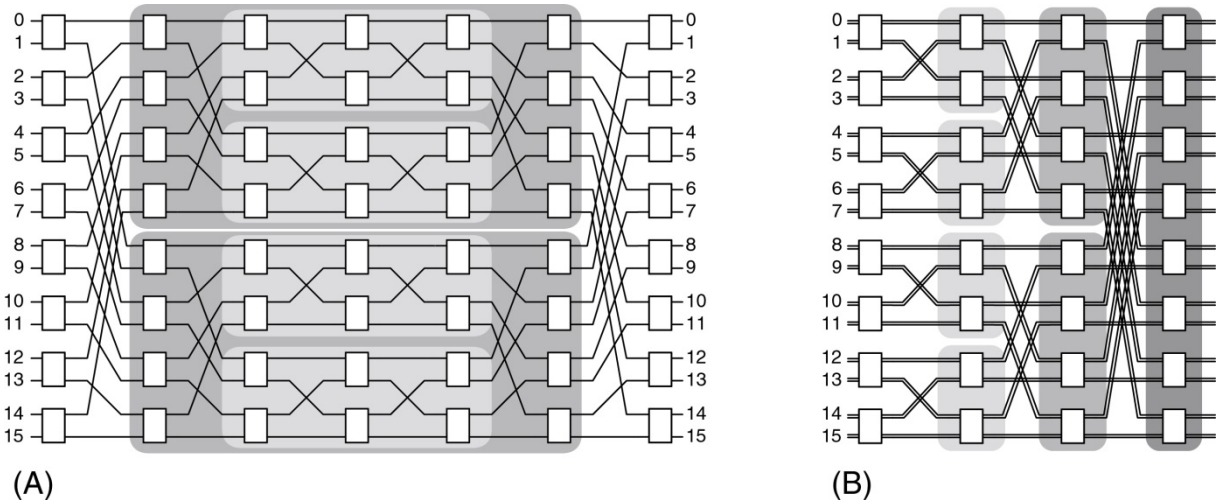


Figure F.12 Two Beneŝ networks. (a) A 16-port Clos topology, where the middle-stage switches shown in the darker shading are implemented with another Clos network whose middle-stage switches shown in the lighter shading are implemented with yet another Clos network, and so on, until a Beneŝ network is produced that uses only 2 × 2 switches everywhere. (b) A folded Beneŝ network (bidirectional) in which 4 × 4 switches are used; end nodes attach to the innermost set of the Beneŝ network (unidirectional) switches. This topology is equivalent to a fat tree, where tree vertices are shown in shades.

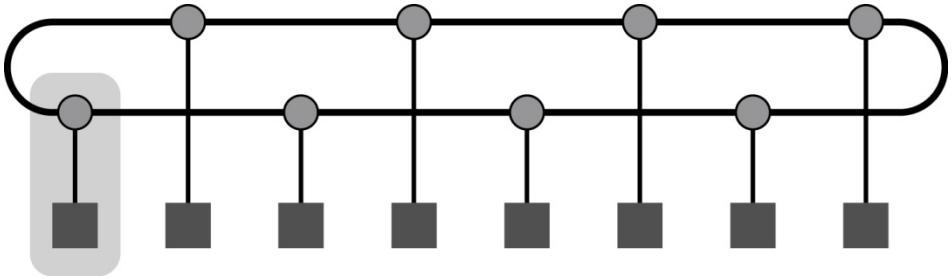


Figure F.13 A ring network topology, folded to reduce the length of the longest link. Shaded circles represent switches, and black squares represent end node devices. The gray rectangle signifies a network node consisting of a switch, a device, and its connecting link.

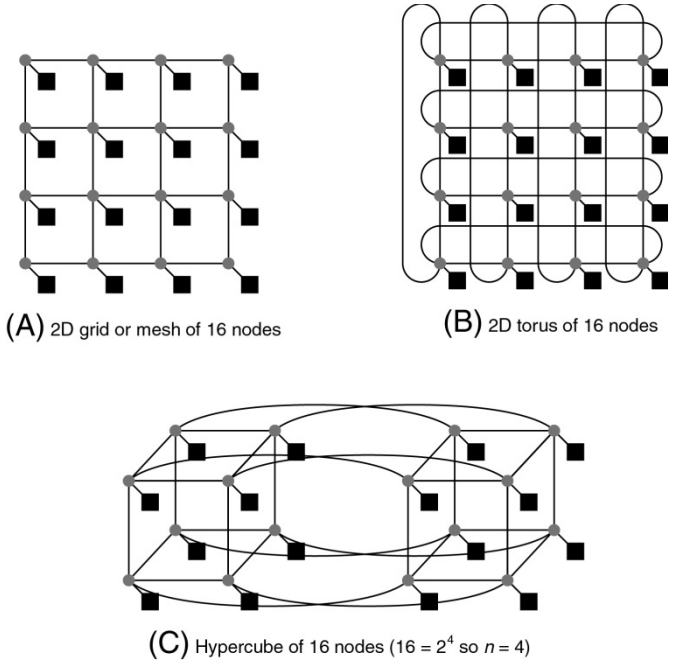


Figure F.14 Direct network topologies that have appeared in commercial systems, mostly supercomputers. The shaded circles represent switches, and the black squares represent end node devices. Switches have many bidirectional network links, but at least one link goes to the end node device. These basic topologies can be supplemented with extra links to improve performance and reliability. For example, connecting the switches on the periphery of the 2D mesh, shown in (a), using the unused ports on each switch forms a 2D torus, shown in (b). The hypercube topology, shown in (c) 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.

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 F.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.

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 $40 \times 32 \times 24$	3200 [3200]	12 bits	3800	5836.8
Cray	X1E	1024 [×1]	4-way bristled 2D torus ($\sim 23 \times 11$) 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 $32 \times 32 \times 64$	612.5 [1050]	1 bit (bit serial)	175	358.4

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

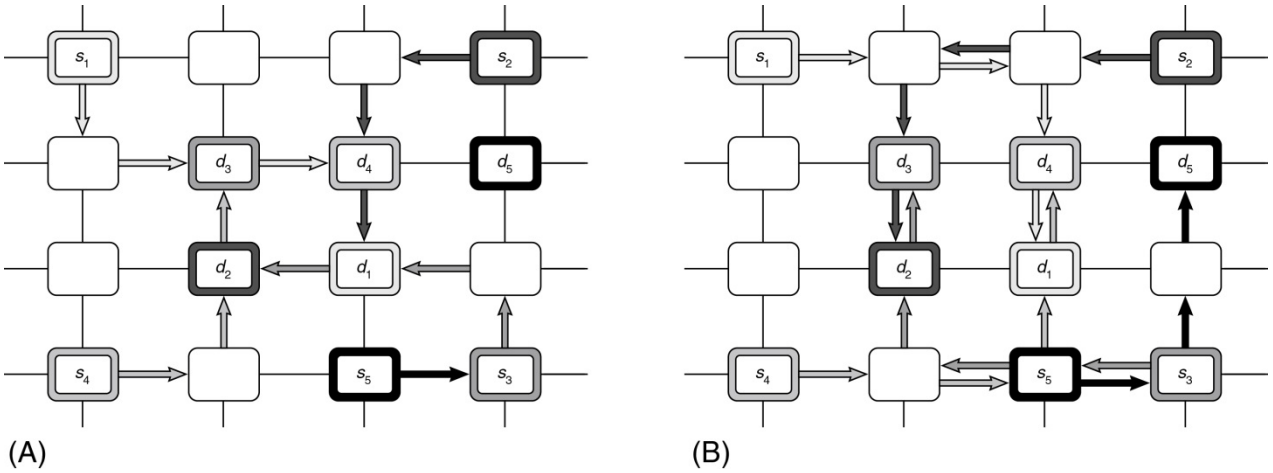


Figure F.17 A mesh network with packets routing from sources, s_i, to destinations, d_i. (a) Deadlock forms from packets destined to d_1 through d_4 blocking on others in the same set that fully occupy their requested buffer resources one hop away from their destinations. This deadlock cycle causes other packets needing those resources also to block, like packets from s_5 destined to d_5 that have reached node s_3. (b) Deadlock is avoided using dimension-order routing. In this case, packets exhaust their routes in the *X* dimension before turning into the *Y* dimension in order to complete their routing.

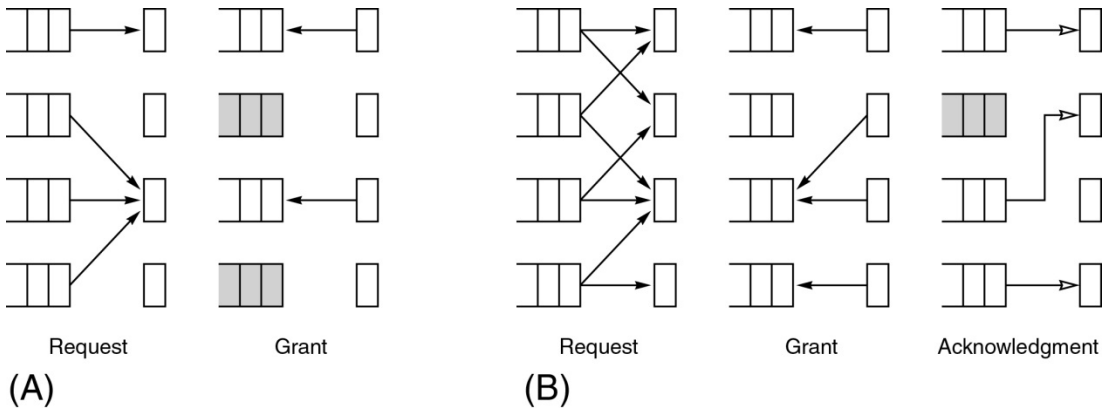


Figure F.18 Two arbitration techniques. (a) Two-phased arbitration in which two of the four input ports are granted requested output ports. (b) Three-phased arbitration in which three of the four input ports are successful in gaining the requested output ports, resulting in higher switch utilization.

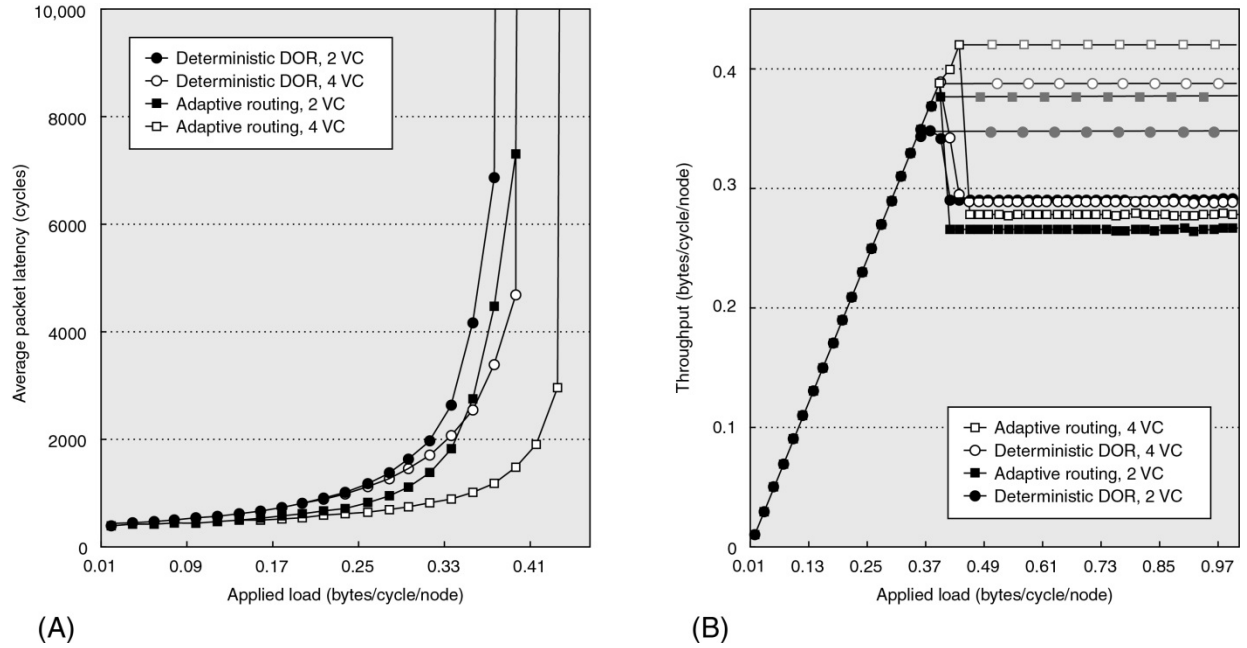


Figure F.19 Deterministic routing is compared against adaptive routing, both with either two or four virtual channels, assuming uniformly distributed traffic on a 4 K node 3D torus network with virtual cut-through switching and bubble flow control to avoid deadlock. (a) Average latency is plotted versus applied load, and (b) throughput is plotted versus applied load (the upper grayish plots show peak throughput, and the lower black plots show sustained throughput). Simulation data were collected by P. Gilabert and J. Flich at the Universidad Politècnica de València, Spain (2006).

Company	System [network] name	Max. number of nodes [× # CPUs]	Basic network topology	Switch queuing (buffers)	Network routing algorithm	Switch arbitration technique	Network switching technique
Intel	ASCI Red Paragon	4510 [×2]	2D mesh (64 × 64)	Input buffered (1 flit)	Distributed dimension-order routing	2-phased RR, distributed across switch	Wormhole with no virtual channels
IBM	ASCI White SP Power3 [Colony]	512 [×16]	Bidirectional MIN with 8-port bidirectional switches (typically a fat tree or Omega)	Input and central buffer with output queuing (8-way speedup)	Source-based LCA adaptive, shortest-path routing, and table-based multicast routing	2-phased RR, centralized and distributed at outputs for bypass paths	Buffered wormhole and virtual cut-through for multicasting, no virtual channels
Intel	Thunder Itanium2 Tiger4 [QsNet^{II}]	1024 [×4]	Fat tree with 8-port bidirectional switches	Input buffered	Source-based LCA adaptive, shortest-path routing	2-phased RR, priority, aging, distributed at output ports	Wormhole with 2 virtual channels
Cray	XT3 [SeaStar]	30,508 [×1]	3D torus (40 × 32 × 24)	Input with staging output	Distributed table-based dimension-order routing	2-phased RR, distributed at output ports	Virtual cut-through with 4 virtual channels
Cray	X1E	1024 [×1]	4-way bristled 2D torus (∼23 × 11) with express links	Input with virtual output queuing	Distributed table-based dimension-order routing	2-phased wavefront (pipelined) global arbiter	Virtual cut-through with 4 virtual channels
IBM	ASC Purple pSeries 575 [Federation]	>1280 [×8]	Bidirectional MIN with 8-port bidirectional switches (typically a fat tree or Omega)	Input and central buffer with output queuing (8-way speedup)	Source and distributed table-based LCA adaptive, shortest-path routing, and multicast	2-phased RR, centralized and distributed at outputs for bypass paths	Buffered wormhole and virtual cut-through for multicasting with 8 virtual channels
IBM	Blue Gene/L eServer Solution [Torus Net.]	65,536 [×2]	3D torus (32 × 32 × 64)	Input-output buffered	Distributed, adaptive with bubble escape virtual channel	2-phased SLQ, distributed at input and output	Virtual cut-through with 4 virtual channels

Figure F.20 Routing, arbitration, and switching characteristics of interconnections networks in commercial machines.

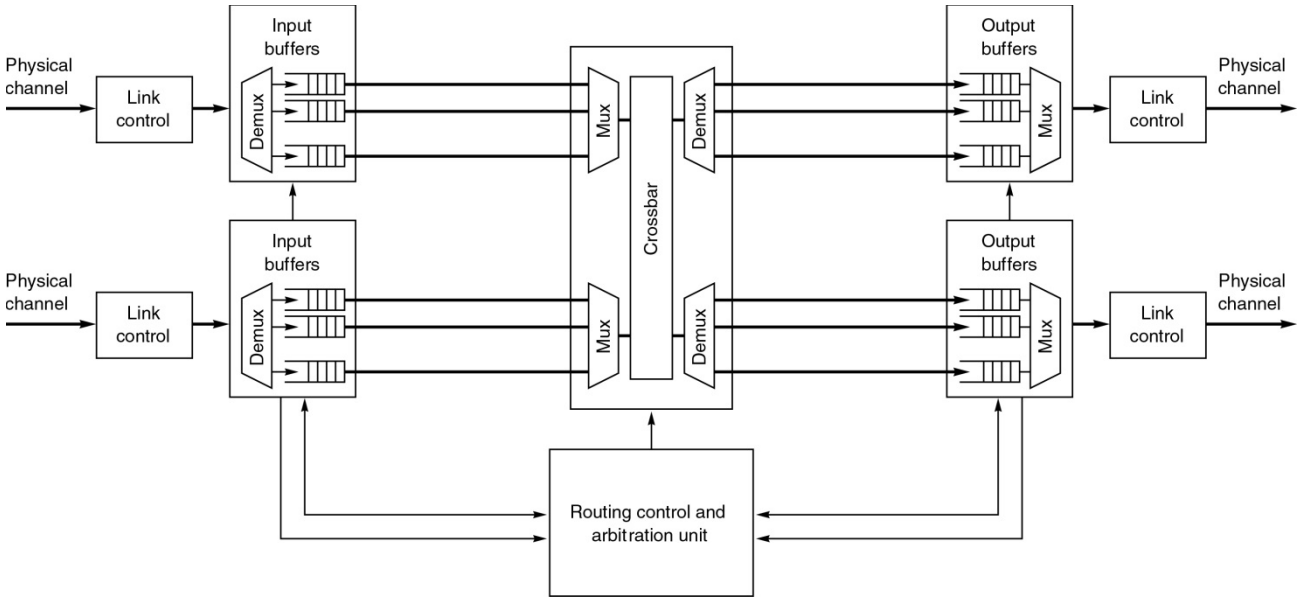


Figure F.21 Basic microarchitectural components of an input-output-buffered switch.

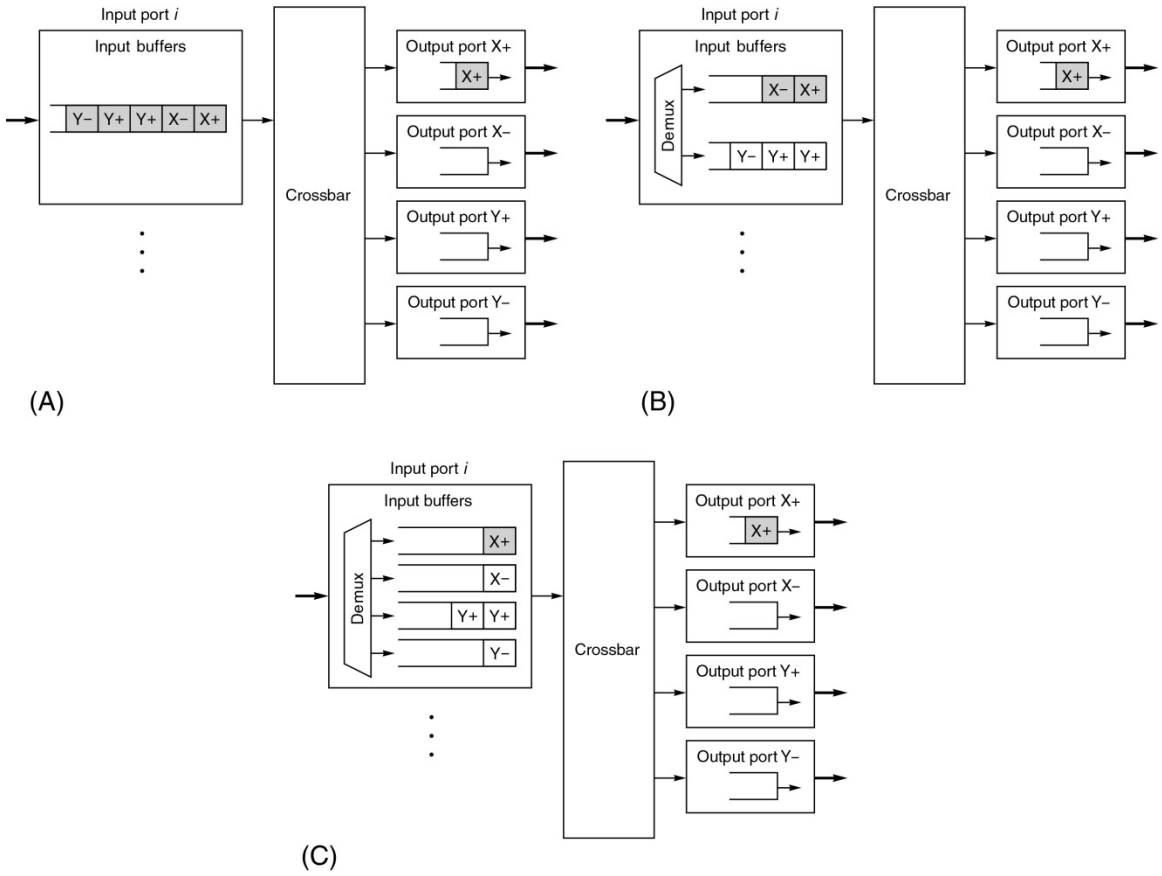
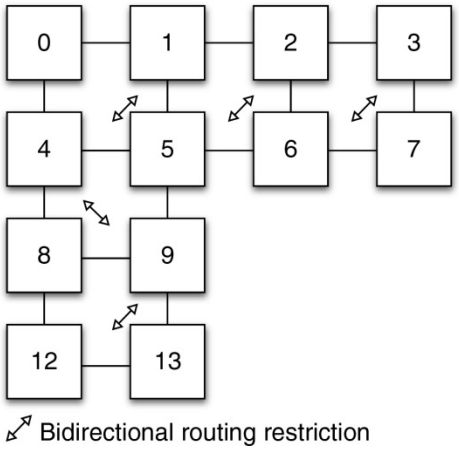


Figure F.22 (a) Head-of-line blocking in an input buffer, (b) the use of two virtual channels to reduce HOL blocking, and (c) the use of virtual output queuing to eliminate HOL blocking within a switch. The shaded input buffer is the one to which the crossbar is currently allocated. This assumes each input port has only one access port to the switch's internal crossbar.



Router	Cn	Ce	Cw	Cs
0	0	1	0	1
1	0	1	1	1
2	0	1	1	1
3	0	0	1	1
4	1	1	0	1
5	1	1	1	1
6	1	1	1	0
7	1	0	1	0
8	1	1	0	1
9	1	0	1	1
10	-	-	-	-
11	-	-	-	-
12	1	1	0	0
13	1	0	1	0
14	-	-	-	-
15	-	-	-	-

Rne	Rnw	Ren	Res
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	0	1
1	1	0	1
1	1	0	1
1	1	1	1
1	1	1	1
1	1	1	1
-	-	-	-
-	-	-	-
1	1	0	1
1	1	1	1
-	-	-	-
-	-	-	-

Rwn	Rws	Rse	Rsw
1	1	1	1
1	1	1	0
1	1	1	0
1	1	1	0
1	1	0	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
0	1	1	0
-	-	-	-
-	-	-	-
1	1	1	1
1	1	1	1
-	-	-	-
-	-	-	-

Figure F.23 Shown is an example of an irregular network that uses LBDR to implement the routing algorithm. For each router, connectivity and routing bits are defined.

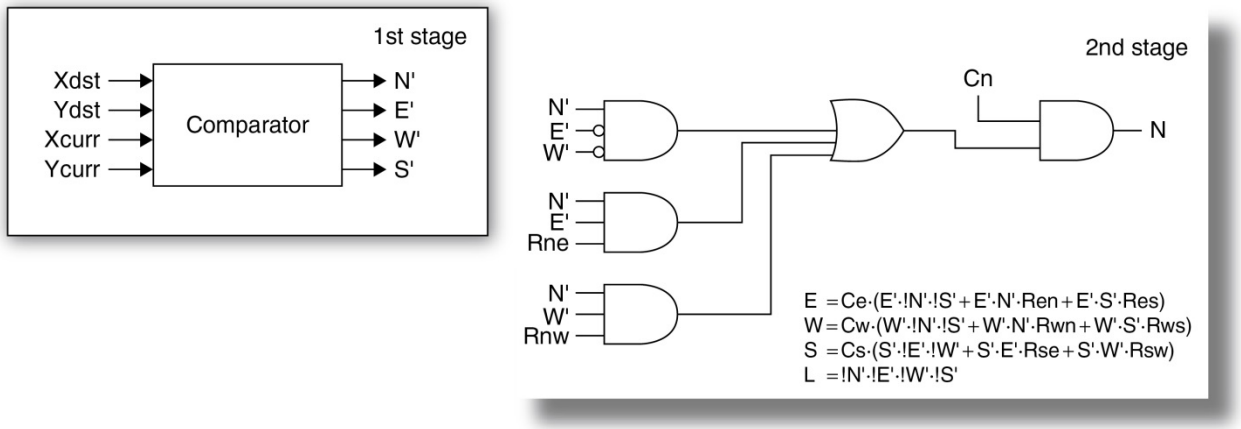


Figure F.24 LBDR logic at each input port of the router.

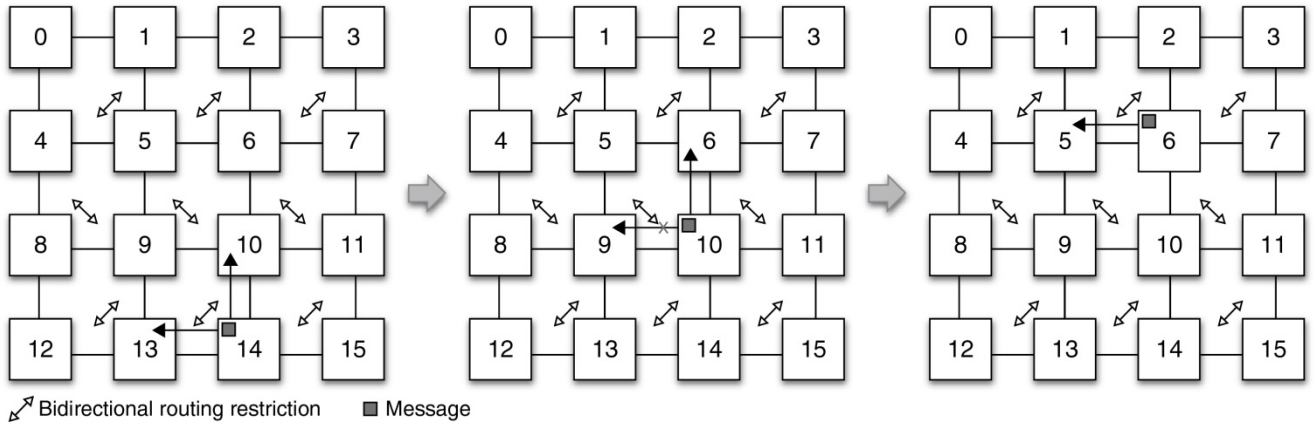


Figure F.25 Example of routing a message from Router 14 to Router 5 using LBDR at each router.

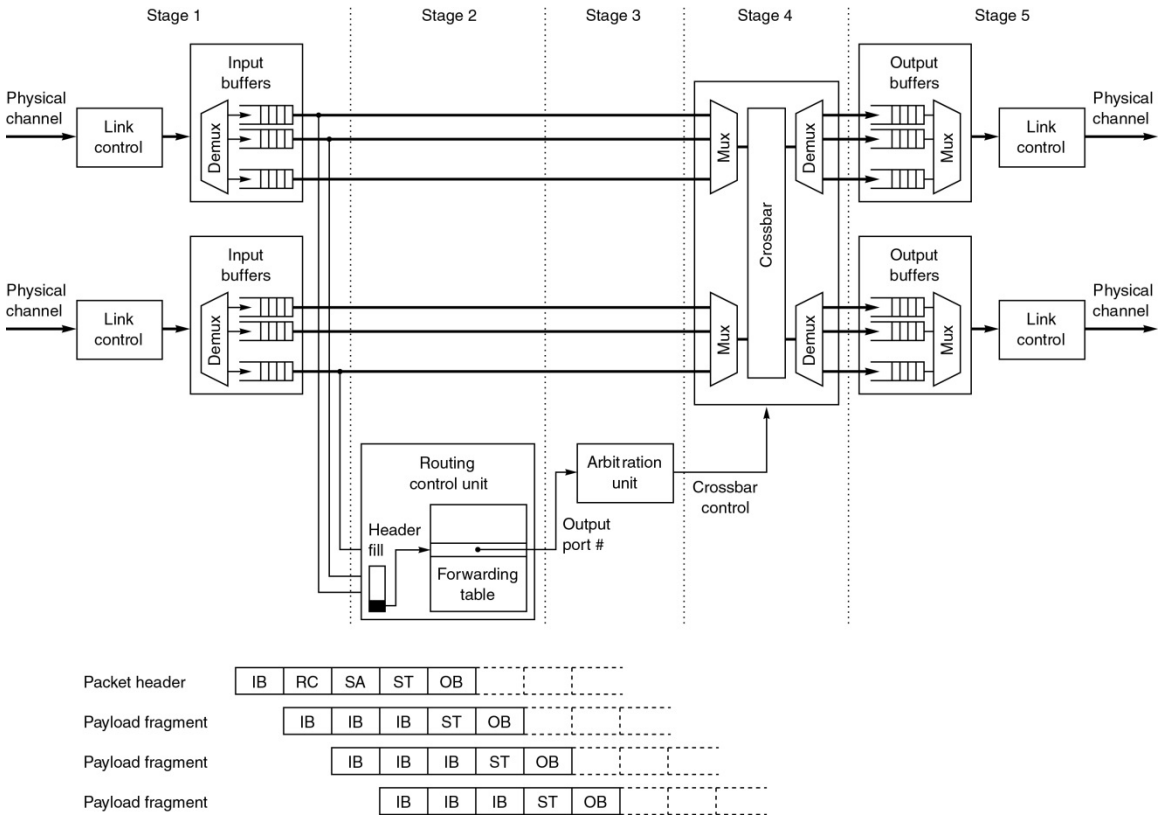


Figure F.26 Pipelined version of the basic input-output-buffered switch. The notation in the figure is as follows: IB is the input link control and buffer stage, RC is the route computation stage, SA is the crossbar switch arbitration stage, ST is the crossbar switch traversal stage, and OB is the output 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.

Failed machines per time interval	One-hour intervals with number of failed machines in first column	Total failures per one-hour interval	One-day intervals with number of failed machines in first column	Total failures per one-day interval
0	8605	0	184	0
1	264	264	105	105
2	50	100	35	70
3	25	75	11	33
4	10	40	6	24
5	7	35	9	45
6	3	18	6	36
7	1	7	4	28
8	1	8	4	32
9	2	18	2	18
10	2	20		
11	1	11	2	22
12			1	12
17	1	17		
20	1	20		
21	1	21	1	21
31			1	31
38			1	38
58			1	58
Total	**8974**	**654**	**373**	**573**

Figure F.27 Measurement of reboots of 58 DECstation 5000 s running Ultrix over a 373-day period. These reboots are distributed into time intervals of one hour and one day. The first column sorts the intervals according to the number of machines that failed in that interval. The next two columns concern one-hour intervals, and the last two columns concern one-day intervals. The second and fourth columns show the number of intervals for each number of failed machines. The third and fifth columns are just the product of the number of failed machines and the number of intervals. For example, there were 50 occurrences of one-hour intervals with 2 failed machines, for a total of 100 failed machines, and there were 35 days with 2 failed machines, for a total of 70 failures. As we would expect, the number of failures per interval changes with the size of the interval. For example, the day with 31 failures might include one hour with 11 failures and one hour with 20 failures. The last row shows the total number of each column; the number of failures doesn't agree because multiple reboots of the same machine in the same interval do not result in separate entries. (Randy Wang of the University of California–Berkeley collected these data.)

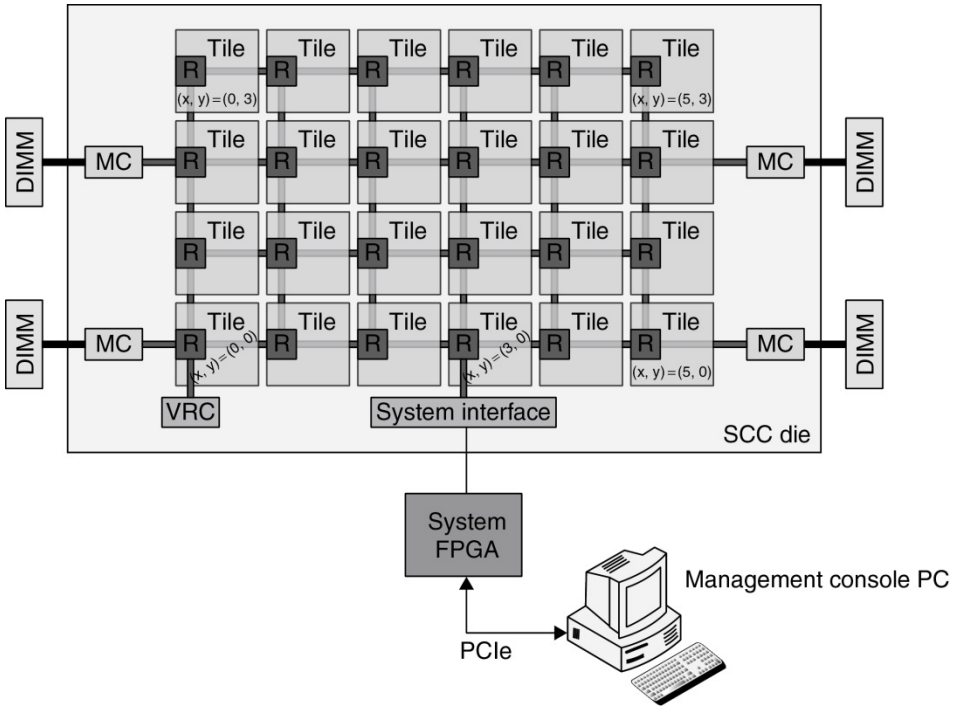


Figure F.28 SCC Top-level architecture. From Howard, J. et al., *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 58–59.

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 F.29 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.

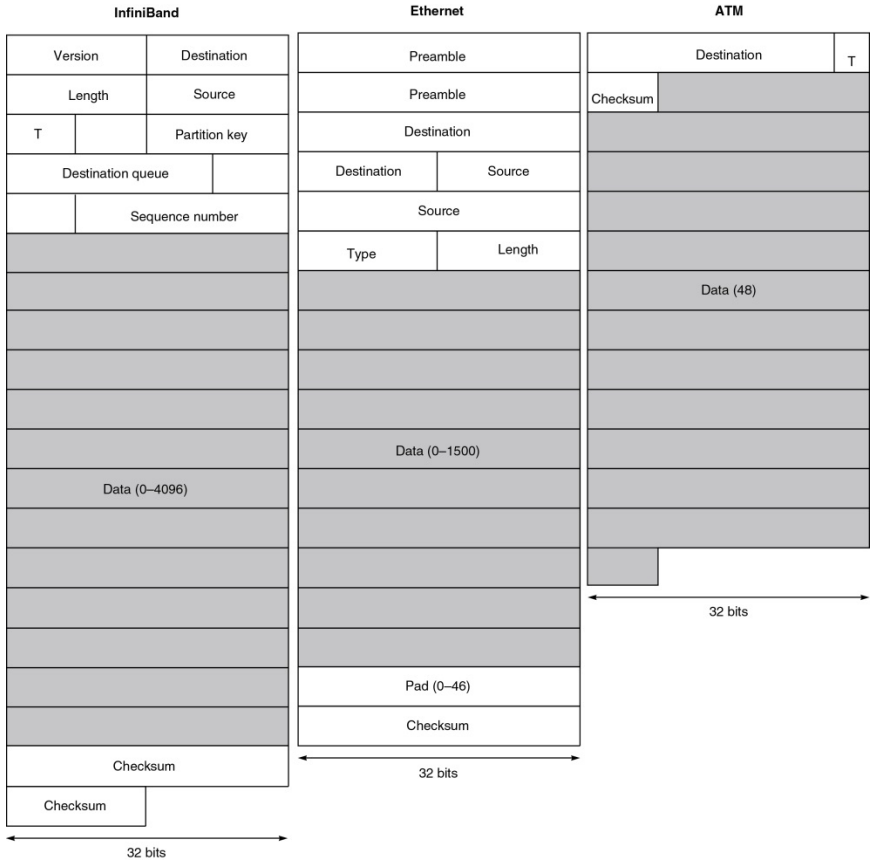


Figure F.30 Packet format for InfiniBand, Ethernet, and ATM. ATM calls their messages “cells” instead of packets, so the proper name is ATM cell format. The width of each drawing is 32 bits. All three formats have destination addressing fields, encoded differently for each situation. All three also have a checksum field to catch transmission errors, although the ATM checksum field is calculated only over the header; ATM relies on higher-level protocols to catch errors in the data. Both InfiniBand and Ethernet have a length field, since the packets hold a variable amount of data, with the former counted in 32-bit words and the latter in bytes. InfiniBand and ATM headers have a type field (T) that gives the type of packet. The remaining Ethernet fields are a preamble to allow the receiver to recover the clock from the self-clocking code used on the Ethernet, the source address, and a pad field to make sure the smallest packet is 64 bytes (including the header). InfiniBand includes a version field for protocol version, a sequence number to allow in-order delivery, a field to select the destination queue, and a partition key field. Infiniband has many more small fields not shown and many other packet formats; above is a simplified view. ATM’s short, fixed packet is a good match to real-time demand of digital voice.

Network name [vendors]	Used in top 10 supercomputer clusters (2005)	Number of nodes	Basic network topology	Raw link bidirectional 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 F.31 Characteristics of system area networks implemented in various top 10 supercomputer clusters in 2005.

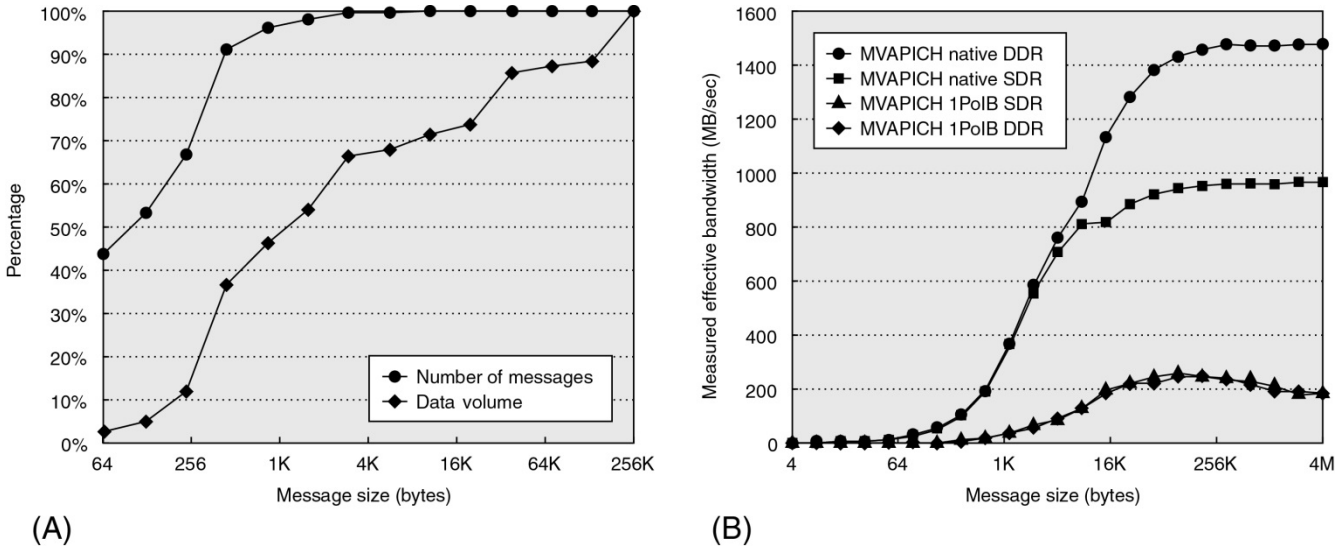


Figure F.32 Data collected by D.K. Panda, S. Sur, and L. Chai (2005) in the Network-Based Computing Laboratory at The Ohio State University. (a) Cumulative percentage of messages and volume of data transferred as message size varies for the Fluent application (*www.fluent.com*). Each *x*-axis entry includes all bytes up to the next one; for example, 128 represents 1 byte to 128 bytes. About 90% of the messages are less than 512 bytes, which represents about 40% of the total bytes transferred. (b) Effective bandwidth versus message size measured on SDR and DDR InfiniBand networks running MVAPICH (*http://nowlab.cse.ohio-state.edu/projects/mpi-iba*) with OS bypass (native) and without (IPoIB).

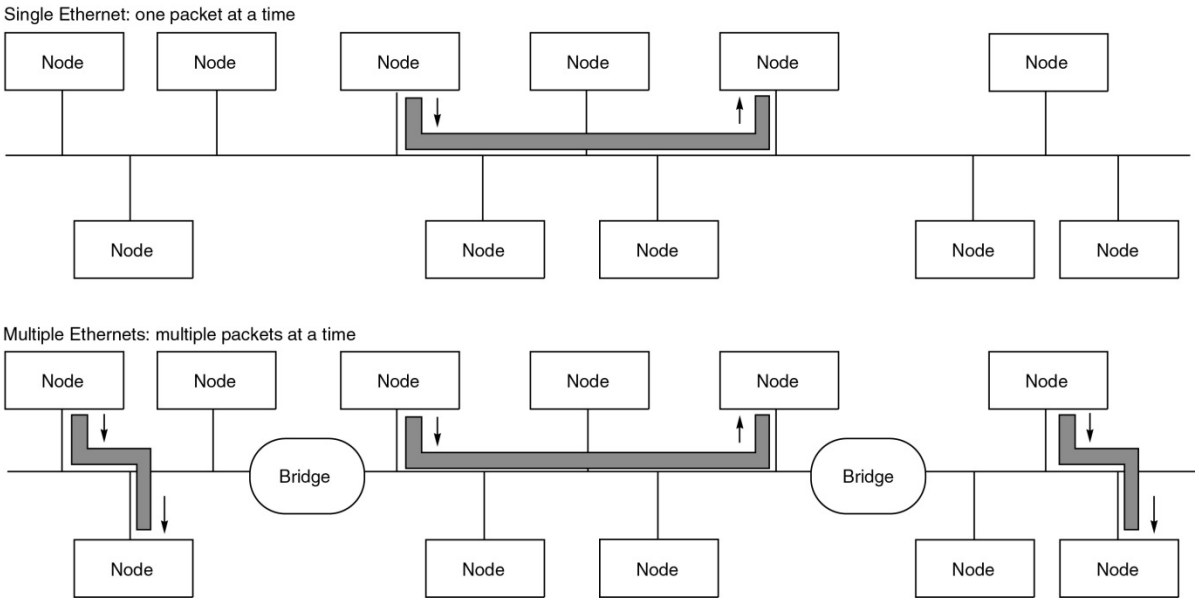


Figure F.33 The potential increased bandwidth of using many Ethernets and bridges.

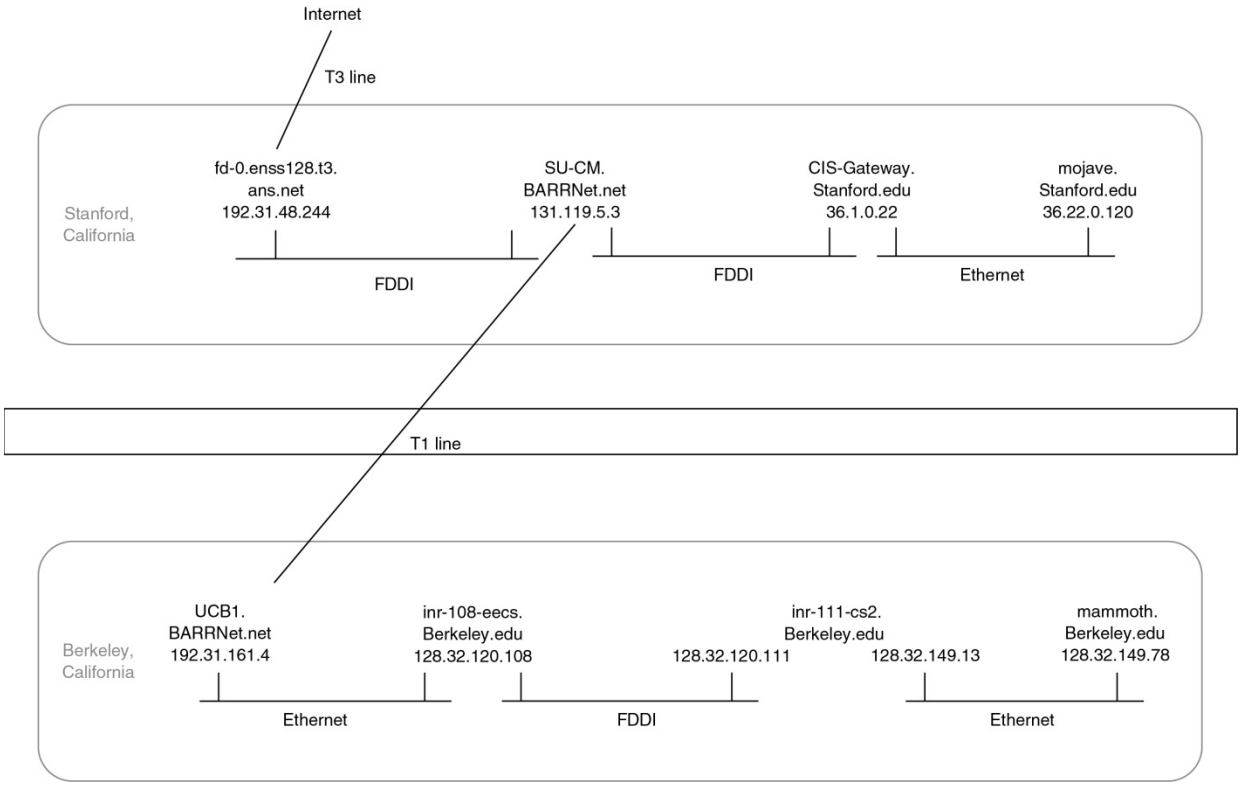


Figure F.34 The connection established between mojave.stanford.edu and mammoth.berkeley.edu (1995). FDDI is a 100 Mbit/sec LAN, while a T1 line is a 1.5 Mbit/sec telecommunications line and a T3 is a 45 Mbit/sec telecommunications line. BARRNet stands for Bay Area Research Network. Note that inr-111-cs2.Berkeley.edu is a router with two Internet addresses, one for each port.

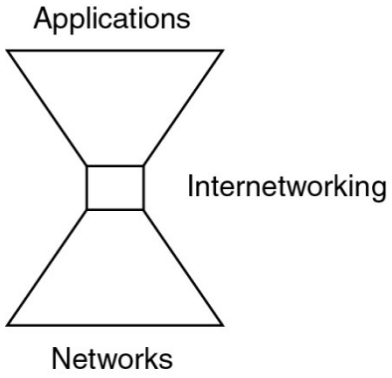


Figure F.35 The role of internetworking. The width indicates the relative number of items at each level.

Layer number	Layer name	Main function	Example protocol	Network component
7	Application	Used for applications specifically written to run over the network	FTP, DNS, NFS, http	Gateway, smart switch
6	Presentation	Translates from application to network format, and *vice versa*		Gateway
5	Session	Establishes, maintains, and ends sessions across the network	Named pipes, RPC	Gateway
4	Transport	Additional connection below the session layer	TCP	Gateway
3	Network	Translates logical network address and names to their physical address (e.g., computer name to MAC address)	IP	Router, ATM switch
2	Data Link	Turns packets into raw bits and at the receiving end turns bits into packets	Ethernet	Bridge, network interface card
1	Physical	Transmits raw bit stream over physical cable	IEEE 802	Hub

Figure F.36 The OSI model layers. Based on *www.geocities.com/SiliconValley/Monitor/3131/ne/osimodel.html*.

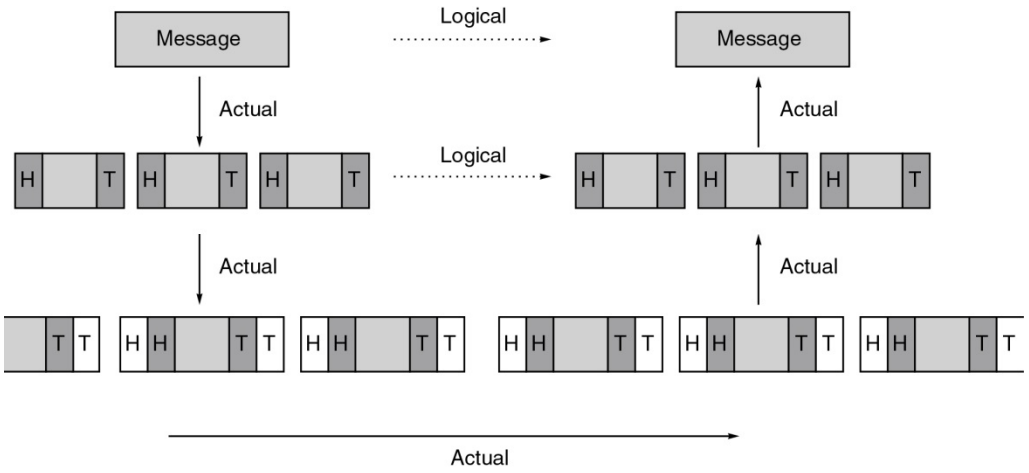


Figure F.37 A generic protocol stack with two layers. Note that communication is peer-to-peer, with headers and trailers for the peer added at each sending layer and removed by each receiving layer. Each layer offers services to the one above to shield it from unnecessary details.

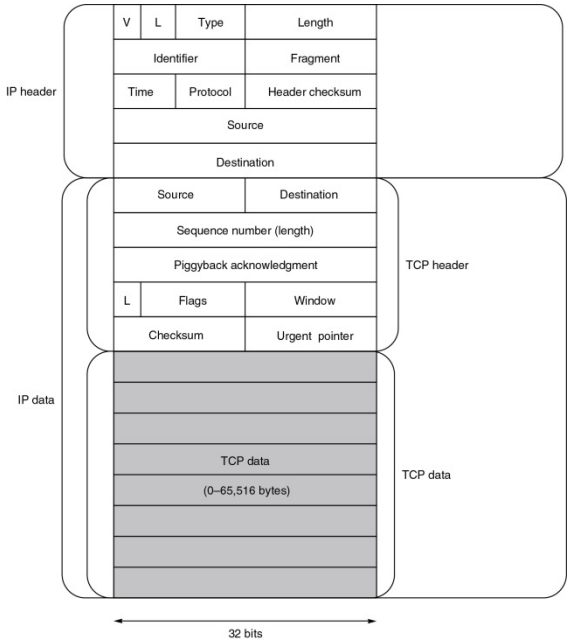


Figure F.38 The headers for IP and TCP. This drawing is 32 bits wide. The standard headers for both are 20 bytes, but both allow the headers to optionally lengthen for rarely transmitted information. Both headers have a length of header field (L) to accommodate the optional fields, as well as source and destination fields. The length field of the whole datagram is in a separate length field in IP, while TCP combines the length of the datagram with the sequence number of the datagram by giving the sequence number in bytes. TCP uses the checksum field to be sure that the datagram is not corrupted, and the sequence number field to be sure the datagrams are assembled into the proper order when they arrive. IP provides checksum error detection only for the header, since TCP has protected the rest of the packet. One optimization is that TCP can send a sequence of datagrams before waiting for permission to send more. The number of datagrams that can be sent without waiting for approval is called the *window*, and the window field tells how many bytes may be sent beyond the byte being acknowledged by this datagram. TCP will adjust the size of the window depending on the success of the IP layer in sending datagrams; the more reliable and faster it is, the larger TCP makes the window. Since the window slides forward as the data arrive and are acknowledged, this technique is called a *sliding window protocol.* The piggyback acknowledgment field of TCP is another optimization. Since some applications send data back and forth over the same connection, it seems wasteful to send a datagram containing only an acknowledgment. This piggyback field allows a datagram carrying data to also carry the acknowledgment for a previous transmission, “piggybacking” on top of a data transmission. The urgent pointer field of TCP gives the address within the datagram of an important byte, such as a break character. This pointer allows the application software to skip over data so that the user doesn’t have to wait for all prior data to be processed before seeing a character that tells the software to stop. The identifier field and fragment field of IP allow intermediary machines to break the original datagram into many smaller datagrams. A unique identifier is associated with the original datagram and placed in every fragment, with the fragment field saying which piece is which. The time-to-live field allows a datagram to be killed off after going through a maximum number of intermediate switches no matter where it is in the network. Knowing the maximum number of hops that it will take for a datagram to arrive—if it ever arrives—simplifies the protocol software. The protocol field identifies which possible upper layer protocol sent the IP datagram; in our case, it is TCP. The V (for version) and type fields allow different versions of the IP protocol software for the network. Explicit version numbering is included so that software can be upgraded gracefully machine by machine, without shutting down the entire network. Nowadays, version six of the Internet protocol (IPv6) was widely used.

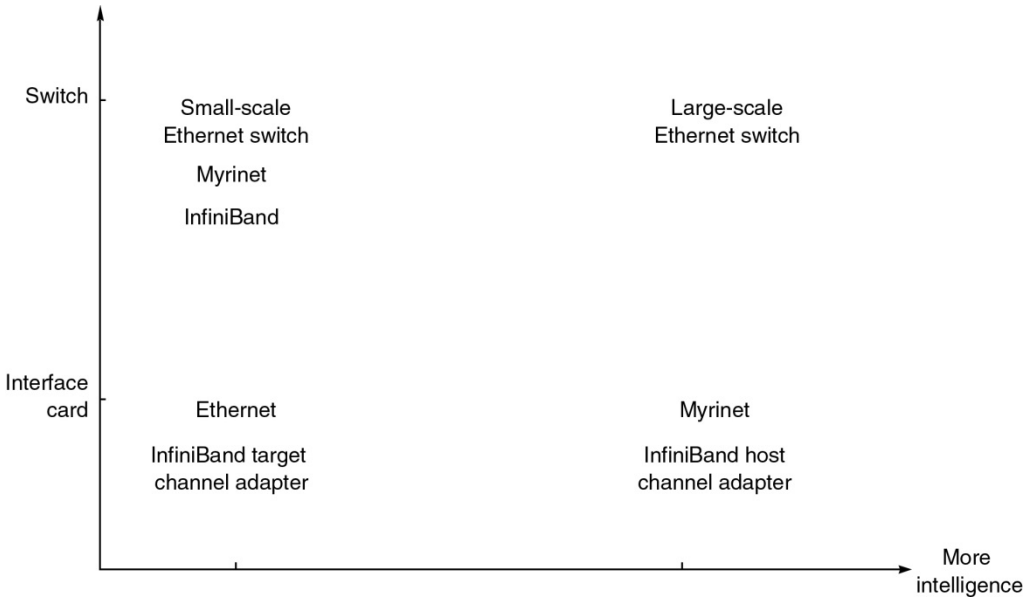


Figure F.39 Intelligence in a network: switch versus network interface card. Note that Ethernet switches come in two styles, depending on the size of the network, and that InfiniBand network interfaces come in two styles, depending on whether they are attached to a computer or to a storage device. Myrinet is a proprietary system area network.

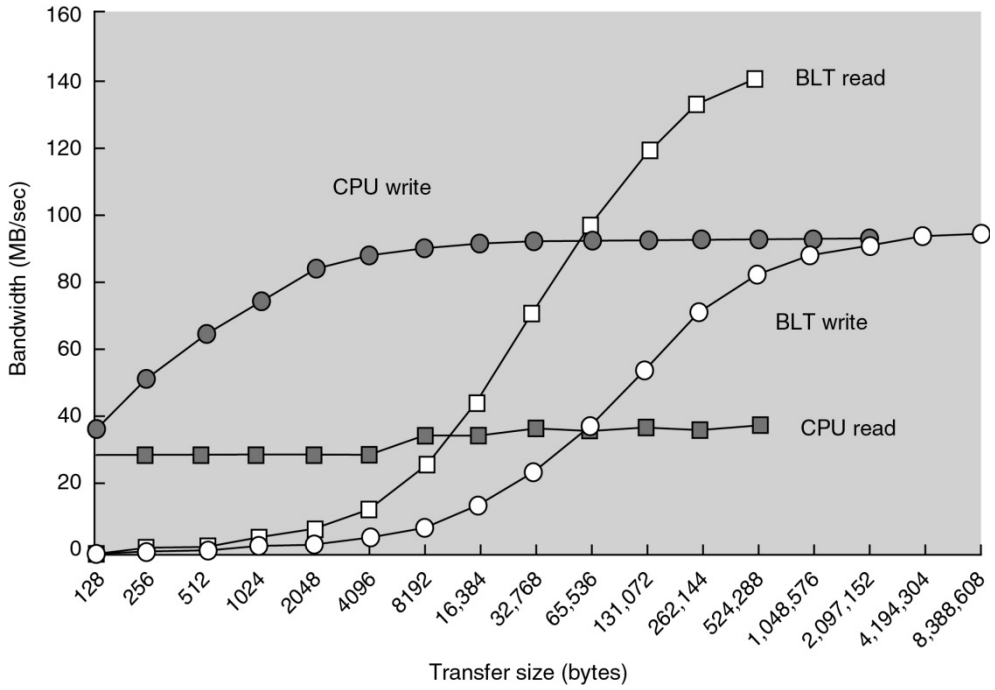


Figure F.40 Bandwidth versus transfer size for simple memory access instructions versus a block transfer device on the Cray Research T3D. (From Arpaci et al. [1995].)

Size	Number of messages	Overhead (sec)		Number of data bytes	Transmission (sec)		Total time (sec)	
		ATM	Ethernet		ATM	Ethernet	ATM	Ethernet
32	771,060	532	389	33,817,052	4	48	536	436
64	56,923	39	29	4,101,088	0	5	40	34
96	4,082,014	2817	2057	428,346,316	46	475	2863	2532
128	5,574,092	3846	2809	779,600,736	83	822	3929	3631
160	328,439	227	166	54,860,484	6	56	232	222
192	16,313	11	8	3,316,416	0	3	12	12
224	4820	3	2	1,135,380	0	1	3	4
256	24,766	17	12	9,150,720	1	9	18	21
512	32,159	22	16	25,494,920	3	23	25	40
1024	69,834	48	35	70,578,564	8	72	56	108
1536	8842	6	4	15,762,180	2	14	8	19
2048	9170	6	5	20,621,760	2	19	8	23
2560	20,206	14	10	56,319,740	6	51	20	61
3072	13,549	9	7	43,184,992	4	39	14	46
3584	4200	3	2	16,152,228	2	14	5	17
4096	67,808	47	34	285,606,596	29	255	76	290
5120	6143	4	3	35,434,680	4	32	8	35
6144	5858	4	3	37,934,684	4	34	8	37
7168	4140	3	2	31,769,300	3	28	6	30
8192	287,577	198	145	2,390,688,480	245	2132	444	2277
Total	11,387,913	7858	5740	4,352,876,316	452	4132	8310	9872

Figure F.41 Total time on a 10-Mbit Ethernet and a 155-Mbit ATM, calculating the total overhead and transmission time separately. Note that the size of the headers needs to be added to the data bytes to calculate transmission time. The higher overhead of the software driver for ATM offsets the higher bandwidth of the network. These measurements were performed in 1994 using SPARCstation 10s, the ForeSystems SBA-200 ATM interface card, and the Fore Systems ASX-200 switch. (NFS measurements taken by Mike Dahlin of the University of California–Berkeley.)