Appendix F

Authors: John Hennessy & David Patterson

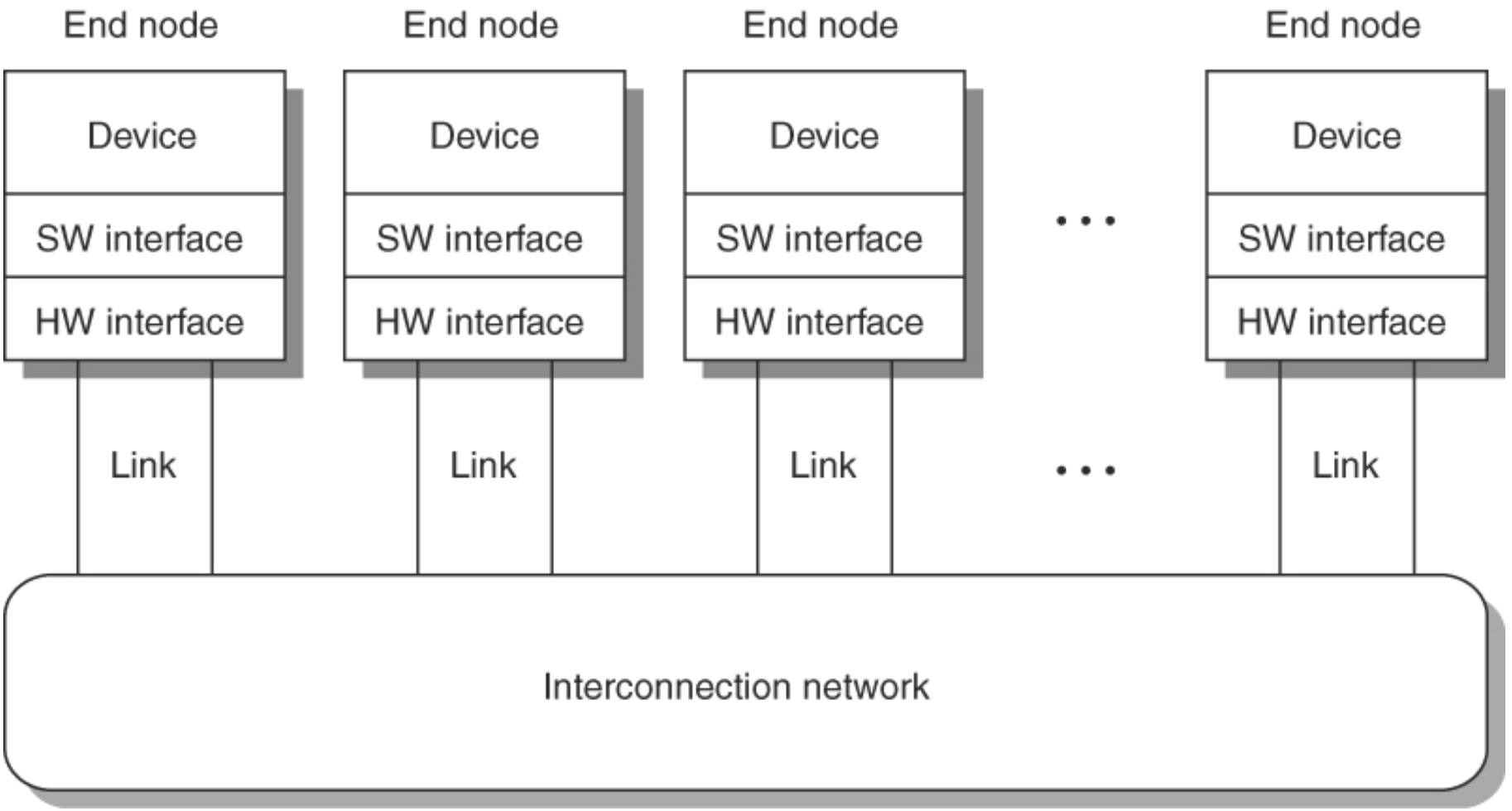


Figure F.1 A conceptual illustration of an interconnected community of devices.

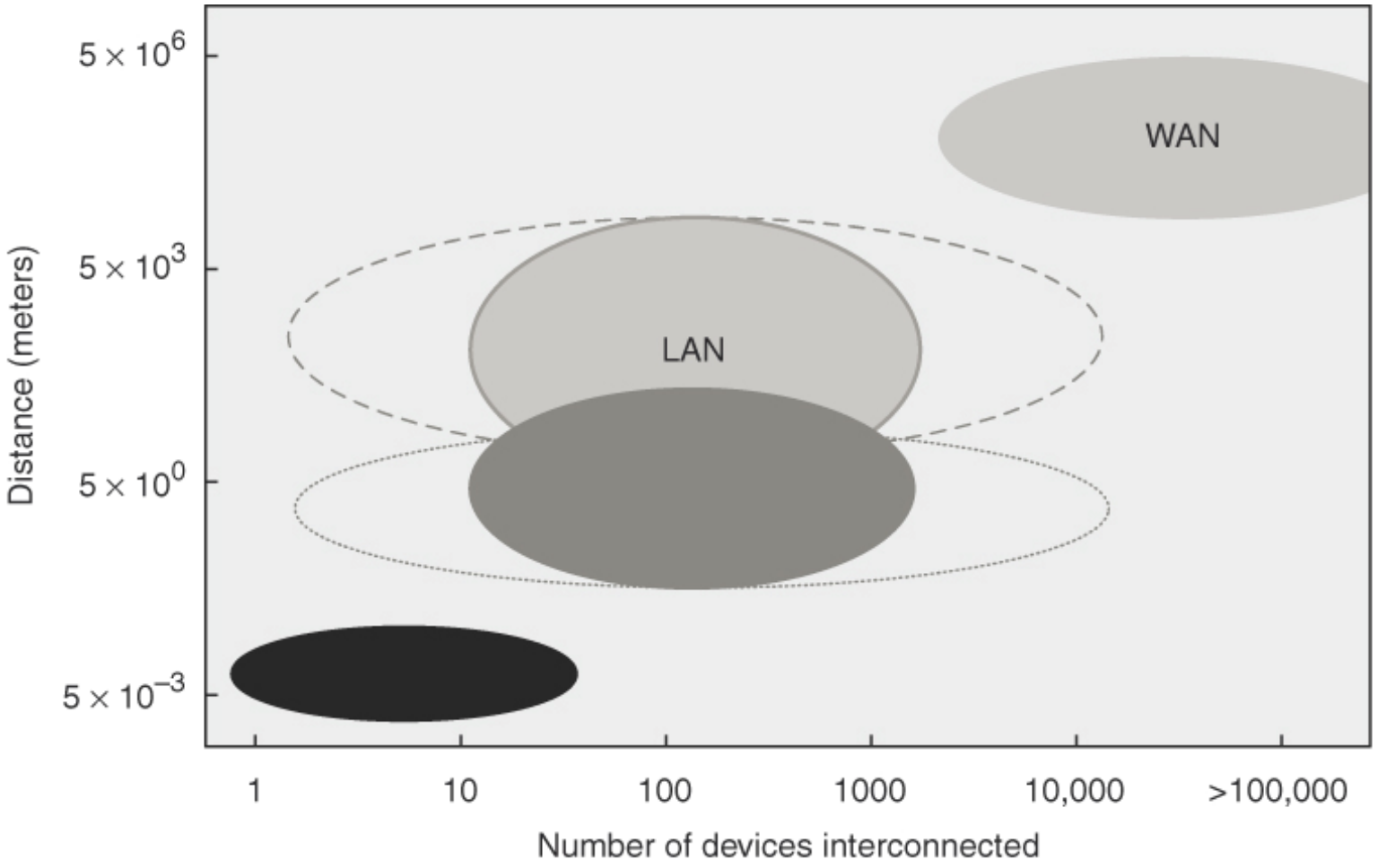


Figure F.2 Relationship of the four interconnection network domains in terms of number of devices connected and their distance scales: on-chip network (OCN), system/storage area network (SAN), local area network (LAN), and wide area network (WAN). Note that there are overlapping ranges where some of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

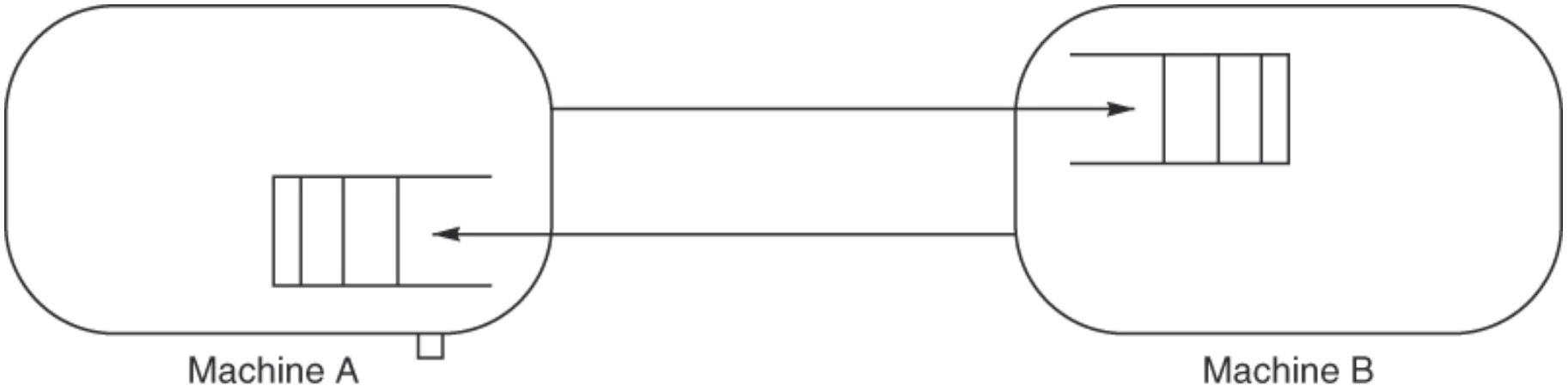


Figure F.3 A simple dedicated link network bidirectionally interconnecting two devices.

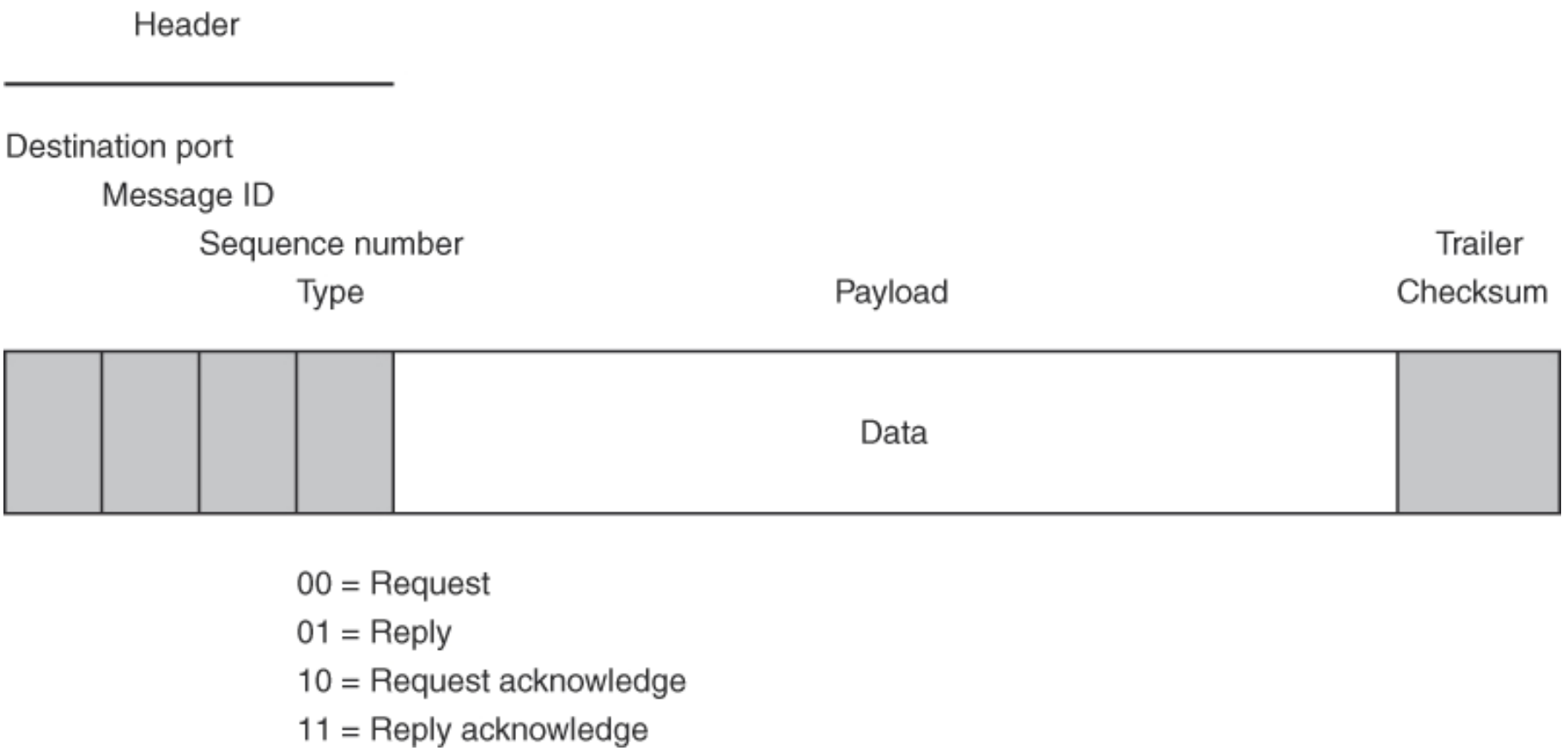


Figure F.4 An example packet format with header, payload, and checksum in the trailer.

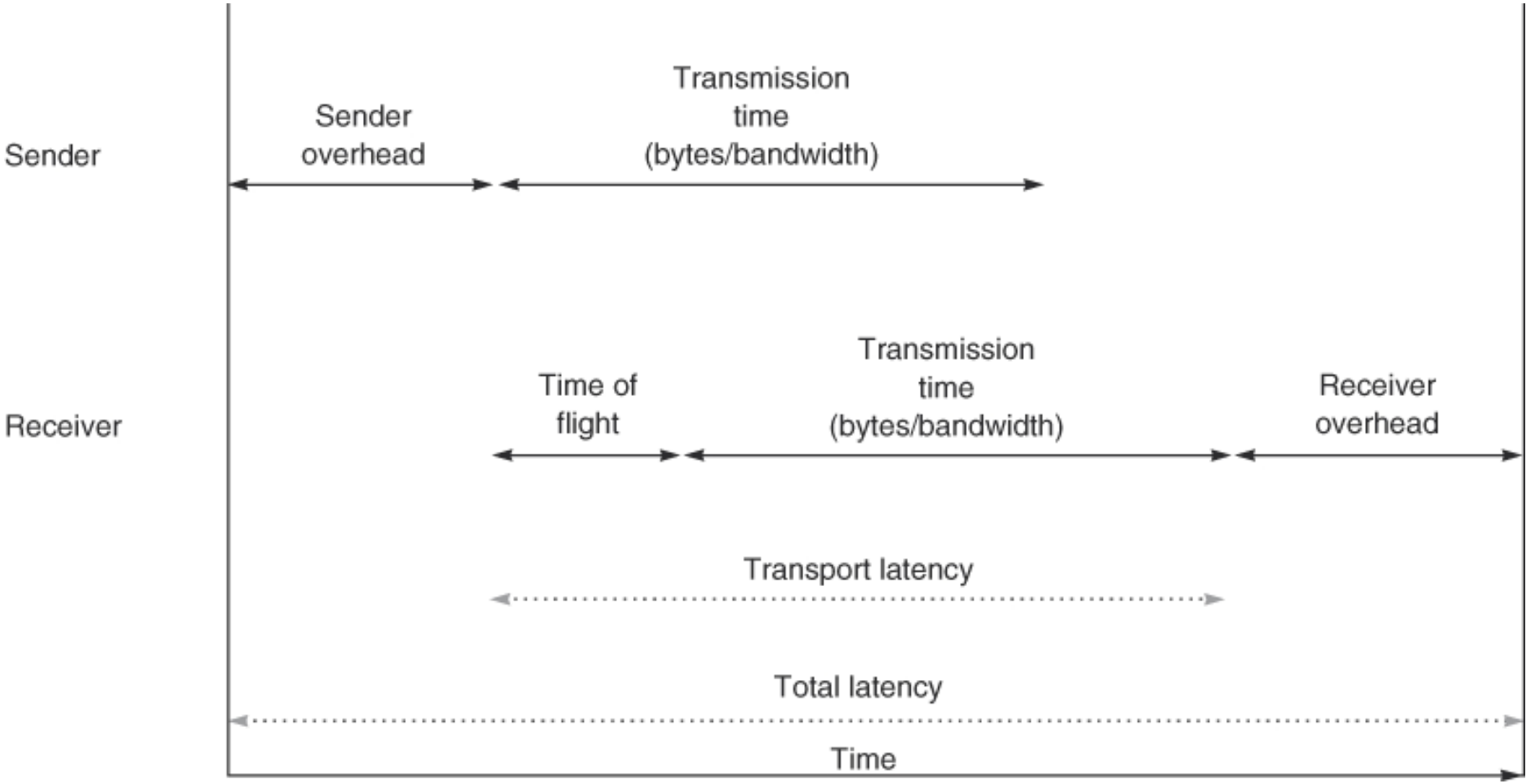


Figure F.5 Components of packet latency. Depending on whether it is an OCN, SAN, LAN, or WAN, the relative amounts of sending and receiving overhead, time of flight, and transmission time are usually quite different from those illustrated here.

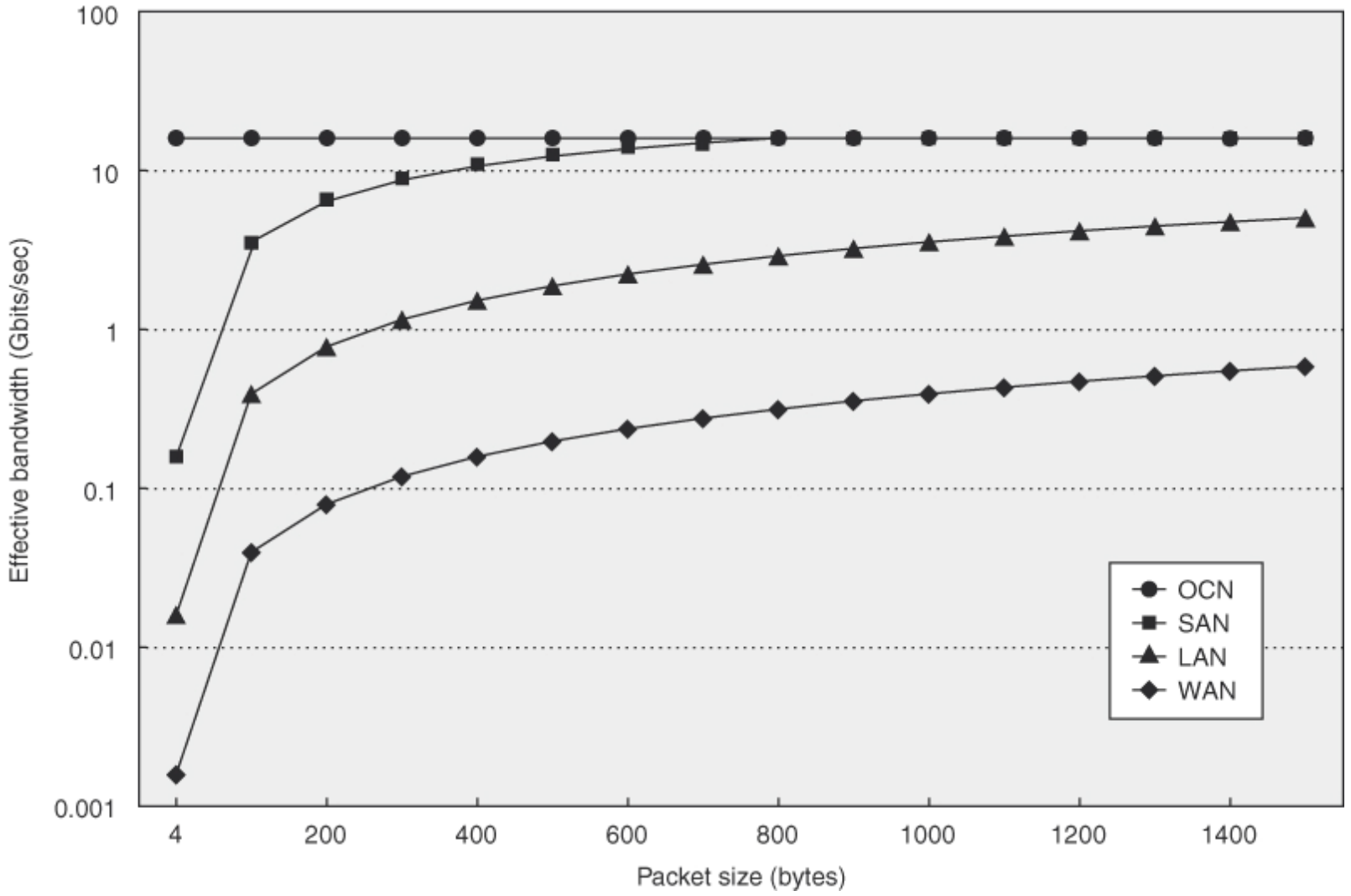


Figure F.6 Effective bandwidth versus packet size plotted in semi-log form for the four network domains. Overhead can be amortized by increasing the packet size, but for too large of an overhead (e.g., for WANs and some LANs) scaling the packet size is of little help. Other considerations come into play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

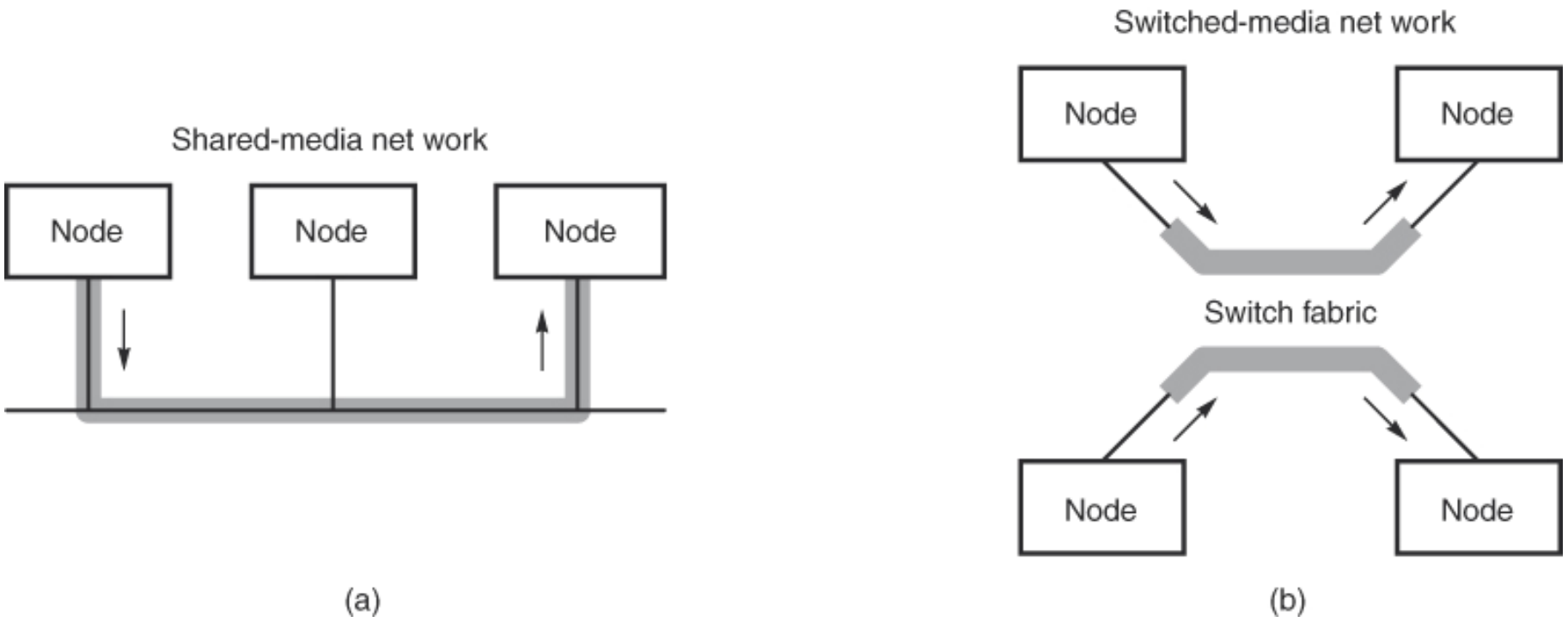


Figure F.8 (a) A shared-media network versus (b) a switched-media network. Ethernet was originally a shared media network, but switched Ethernet is now available. All nodes on the shared-media must dynamically share the raw bandwidth of one link, but switched-media networks can support multiple links,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

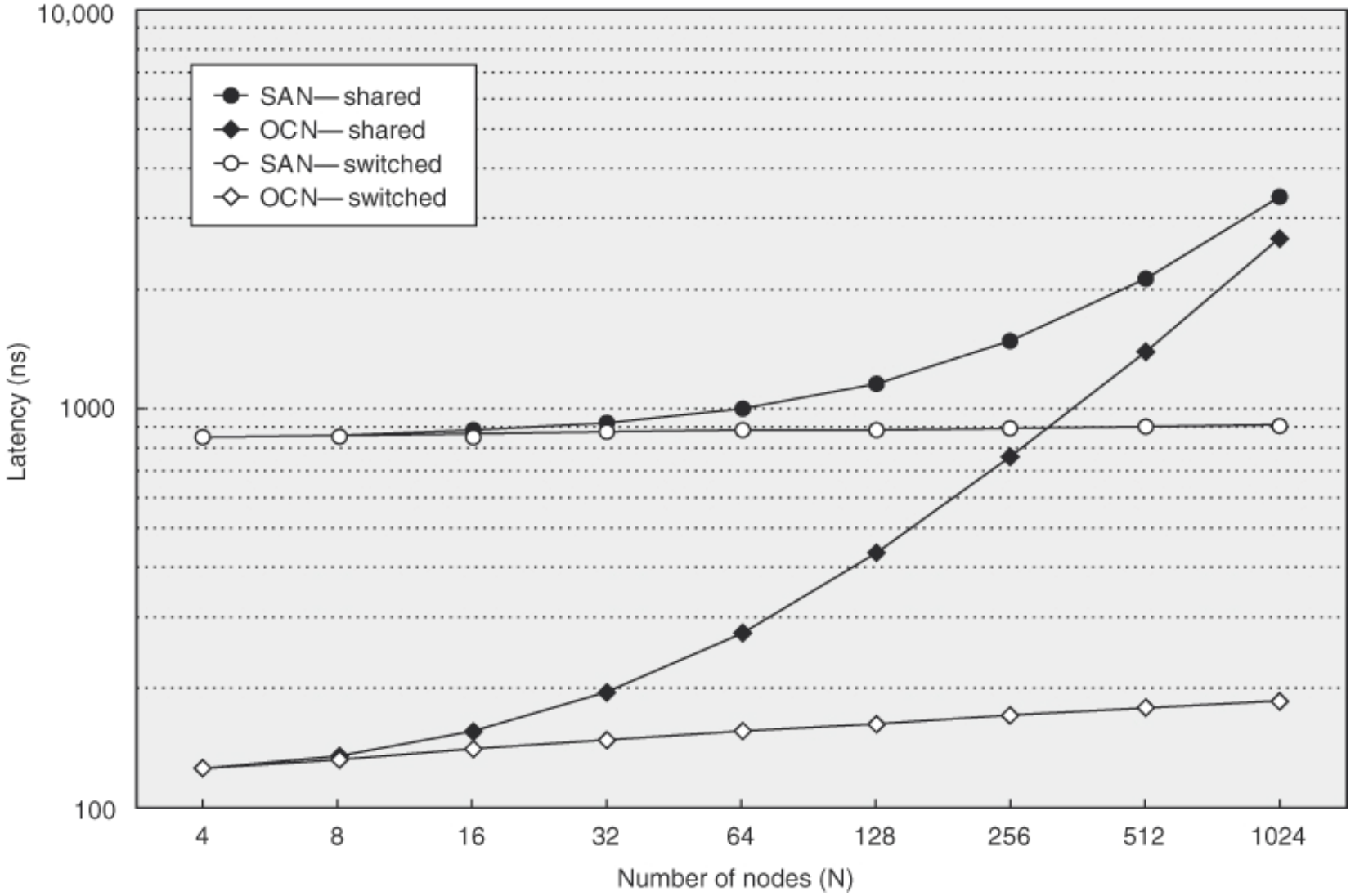


Figure F.9 Latency versus number of interconnected nodes plotted in semi-log form for OCNs and SANs. Routing, arbitration, and switching have more of an impact on latency for networks in these two domains, particularly for networks with a large number of nodes, given the low sending and receiving overheads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

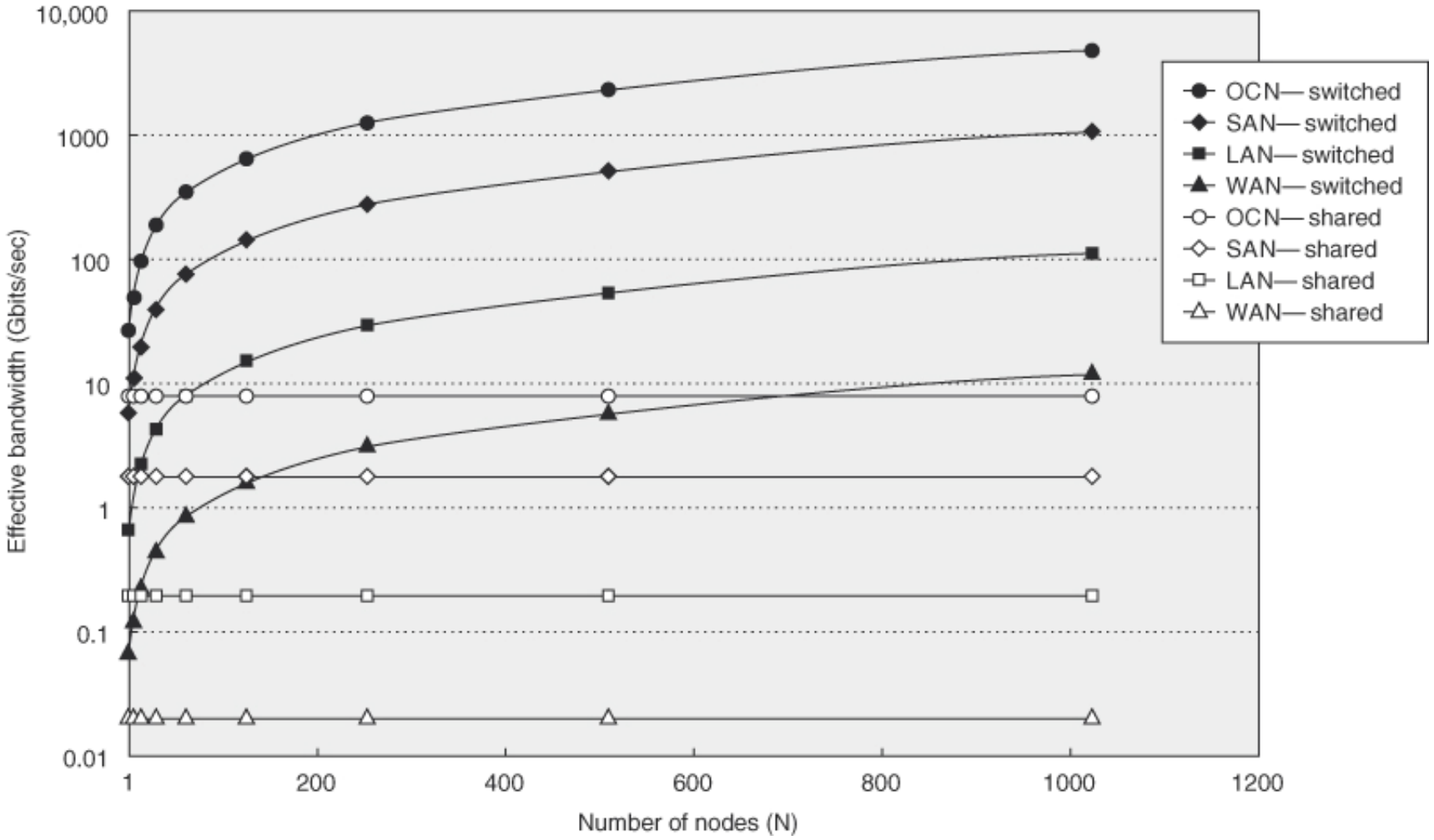


Figure F.10 Effective bandwidth versus number of interconnected nodes plotted in semi-log form for the four network domains. The disparity in effective bandwidth between shared- and switched-media networks for all interconnect domains widens significantly as the number of nodes in the network increases.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

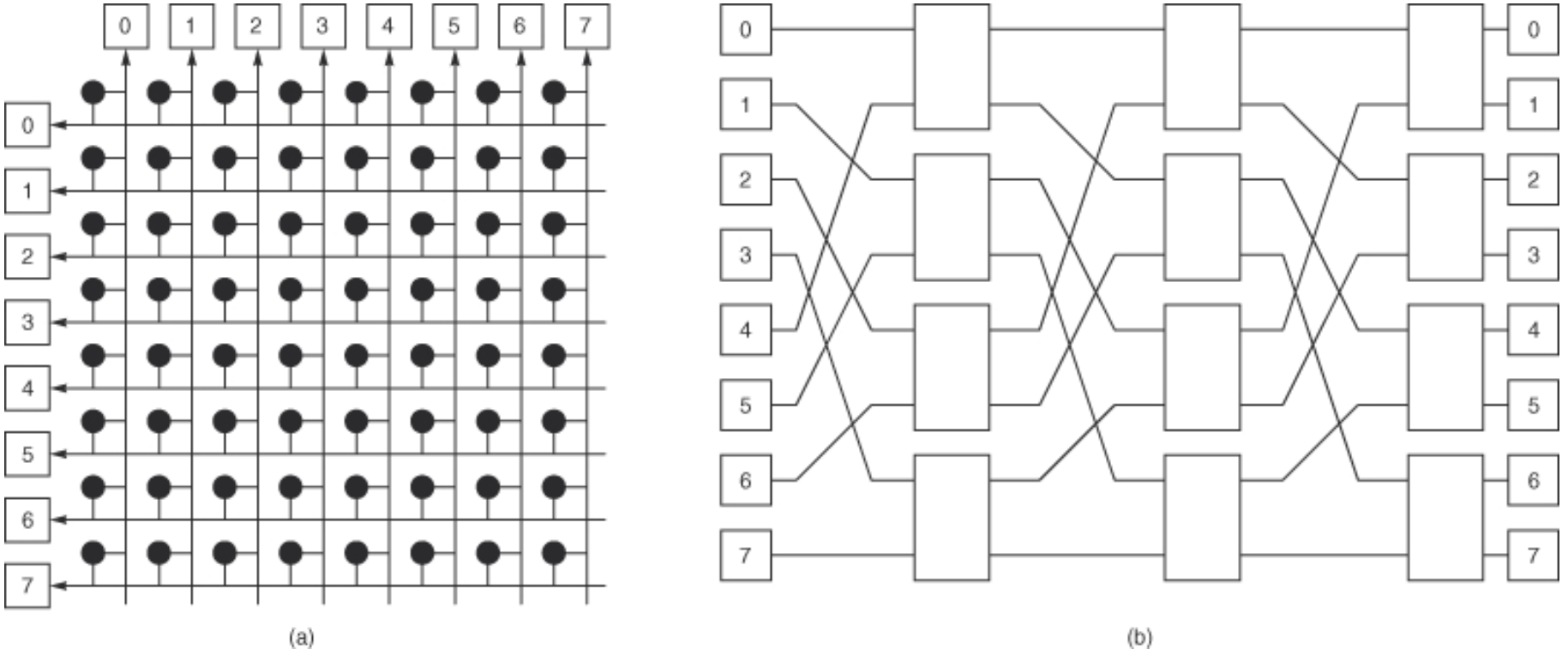


Figure F.11 Popular centralized switched networks: (a) the crossbar network requires *N*2 crosspoint switches, shown as black dots; (b) the Omega, a MIN, requires *N*/2 log2 *N* switches, shown as vertical rectangles. End node devices are shown as numbered squares (total of eight). Links are unidirectional—data enter at the left and exit out the top or right.

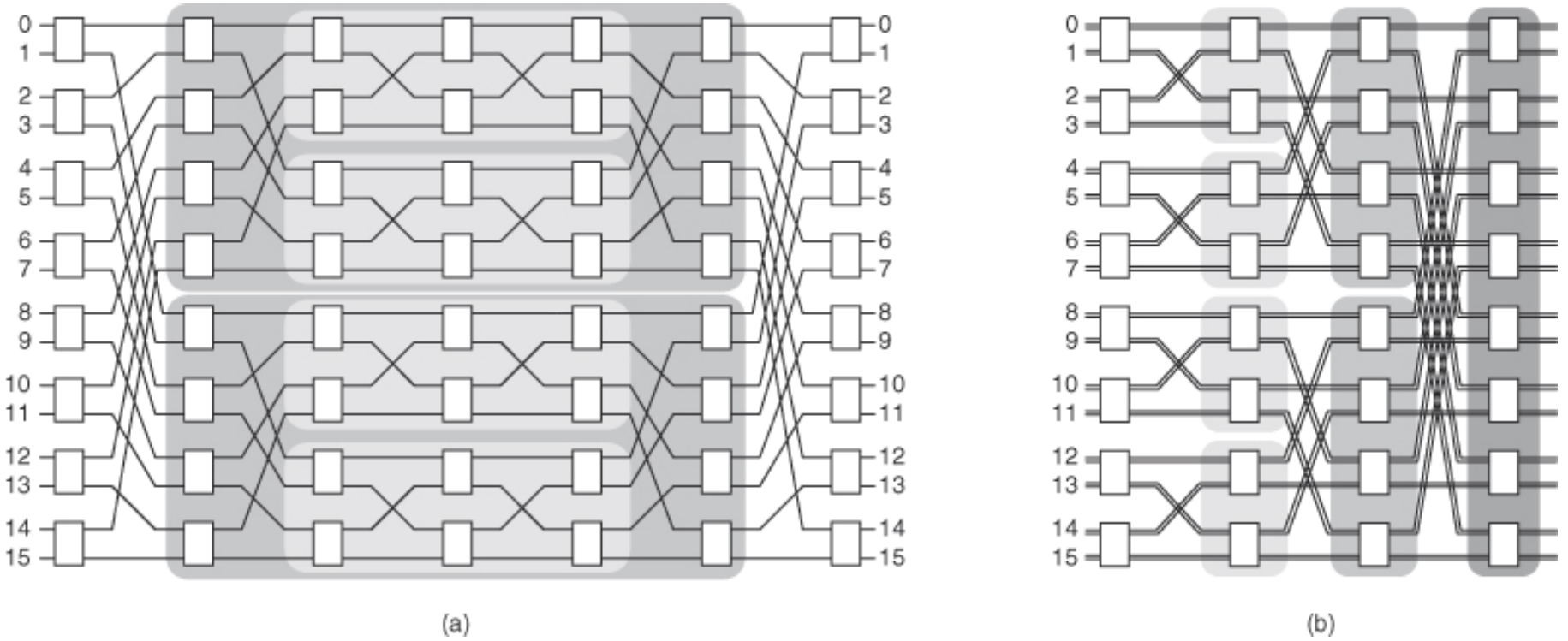


Figure F.12 Two Beneŝ networks. (a) A 16-port Clos topology, where the middle-stage switches shown in the darker shading are implemented with another Clos network whose middle-stage switches shown in the lighter shading are implemented with yet another Clos network, and so on, until a Beneŝ network is produced that uses only 2 x 2 switches everywhere. (b) A folded Beneŝ network (bidirectional) in which 4 x 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.

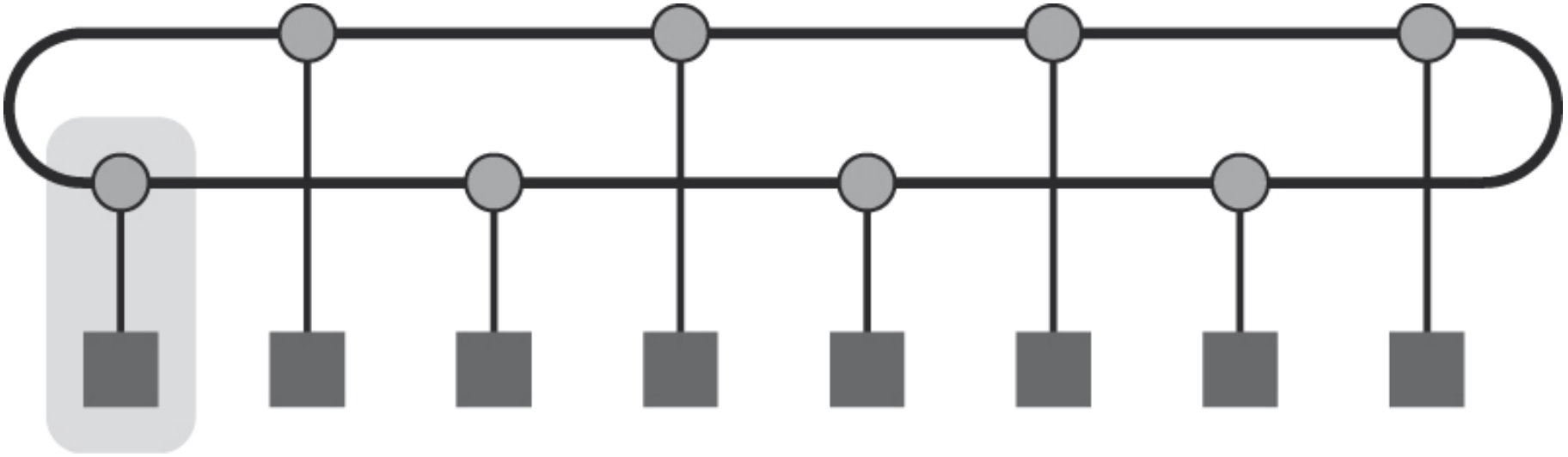


Figure F.13 A ring network topology, folded to reduce the length of the longest link. Shaded circles represent switches, and black squares represent end node devices. The gray rectangle signifies a network node consisting of a switch, a device, and its connecting link.

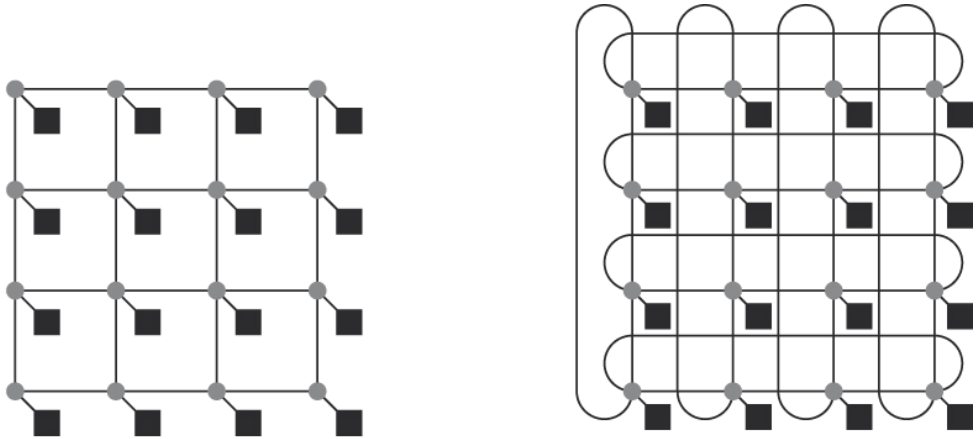


Figure F.14 Direct network topologies that have appeared in commercial systems, mostly supercomputers. The shaded circles represent switches, and the black squares represent end node devices. Switches have many bidirectional network links, but at least one link goes to the end node device. These basic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

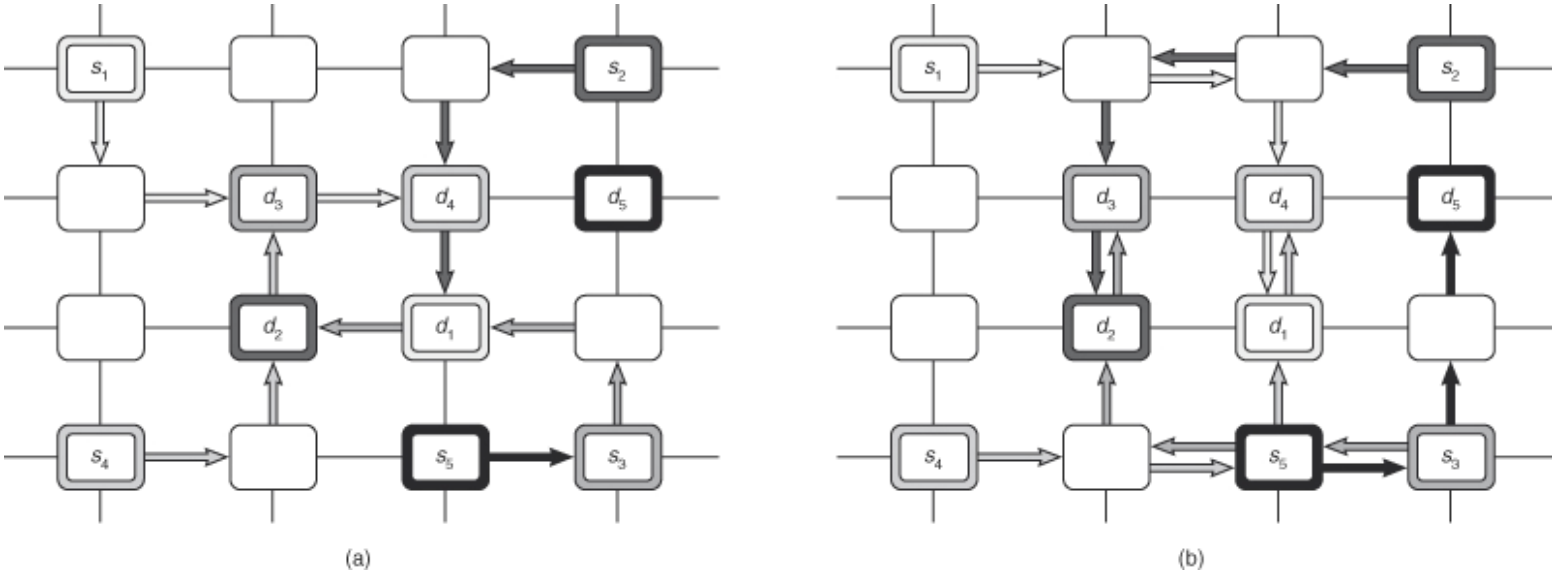


Figure F.17 A mesh network with packets routing from sources, *si*, to destinations, *di*. (a) Deadlock forms from packets destined to *d*1 through *d*4 blocking on others in the same set that fully occupy their requested buffer resources one hop away from their destinations. This deadlock cycle causes 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.

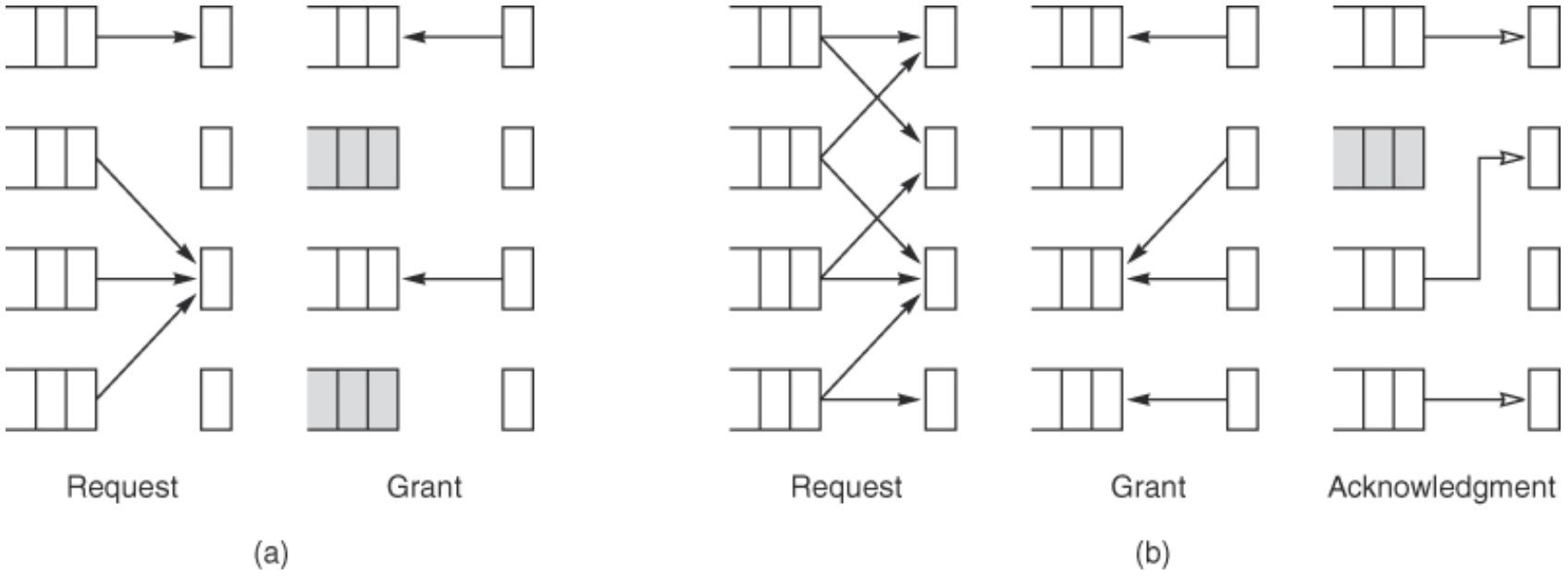


Figure F.18 Two arbitration techniques. (a) Two-phased arbitration in which two of the four input ports are granted requested output ports. (b) Three-phased arbitration in which three of the four input ports are successful in gaining the requested output ports, resulting in higher switch utilization.

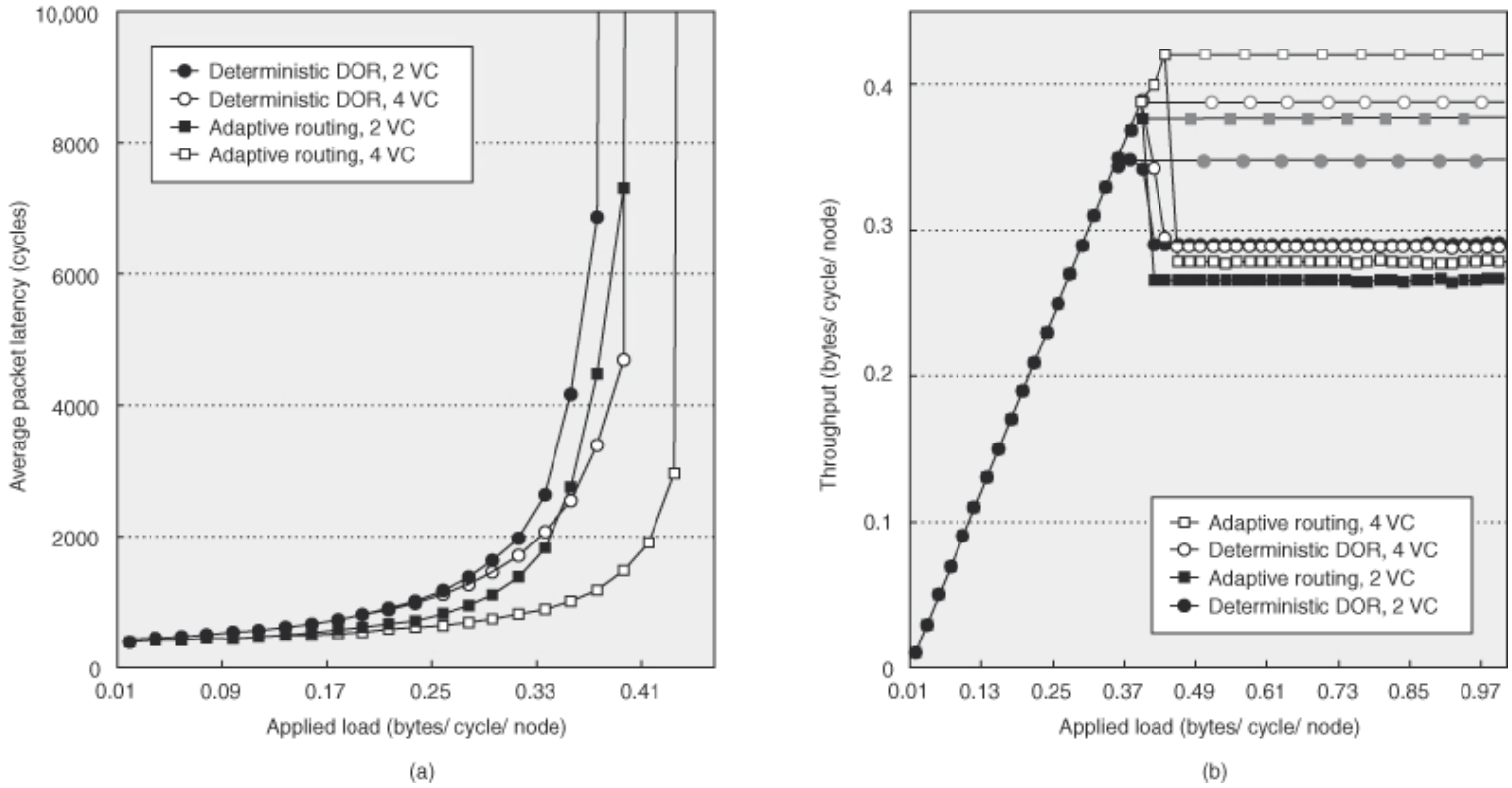


Figure F.19 Deterministic routing is compared against adaptive routing, both with either two or four virtual channels, assuming uniformly distributed traffic on a 4K node 3D torus network with virtual cut-through switching and bubble flow control to avoid deadlock. (a) Average latency is plotted versus 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).

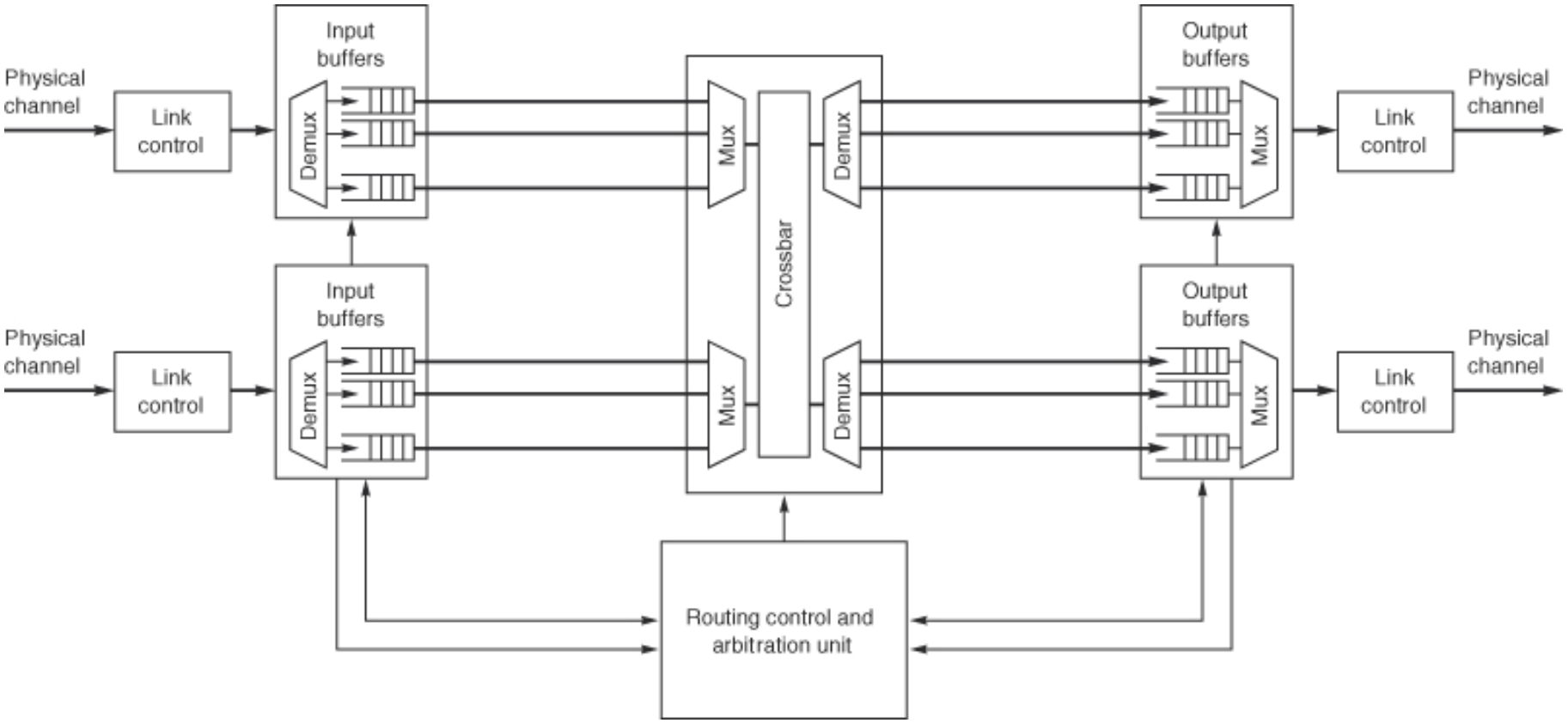


Figure F.21 Basic microarchitectural components of an input-output-buffered switch.

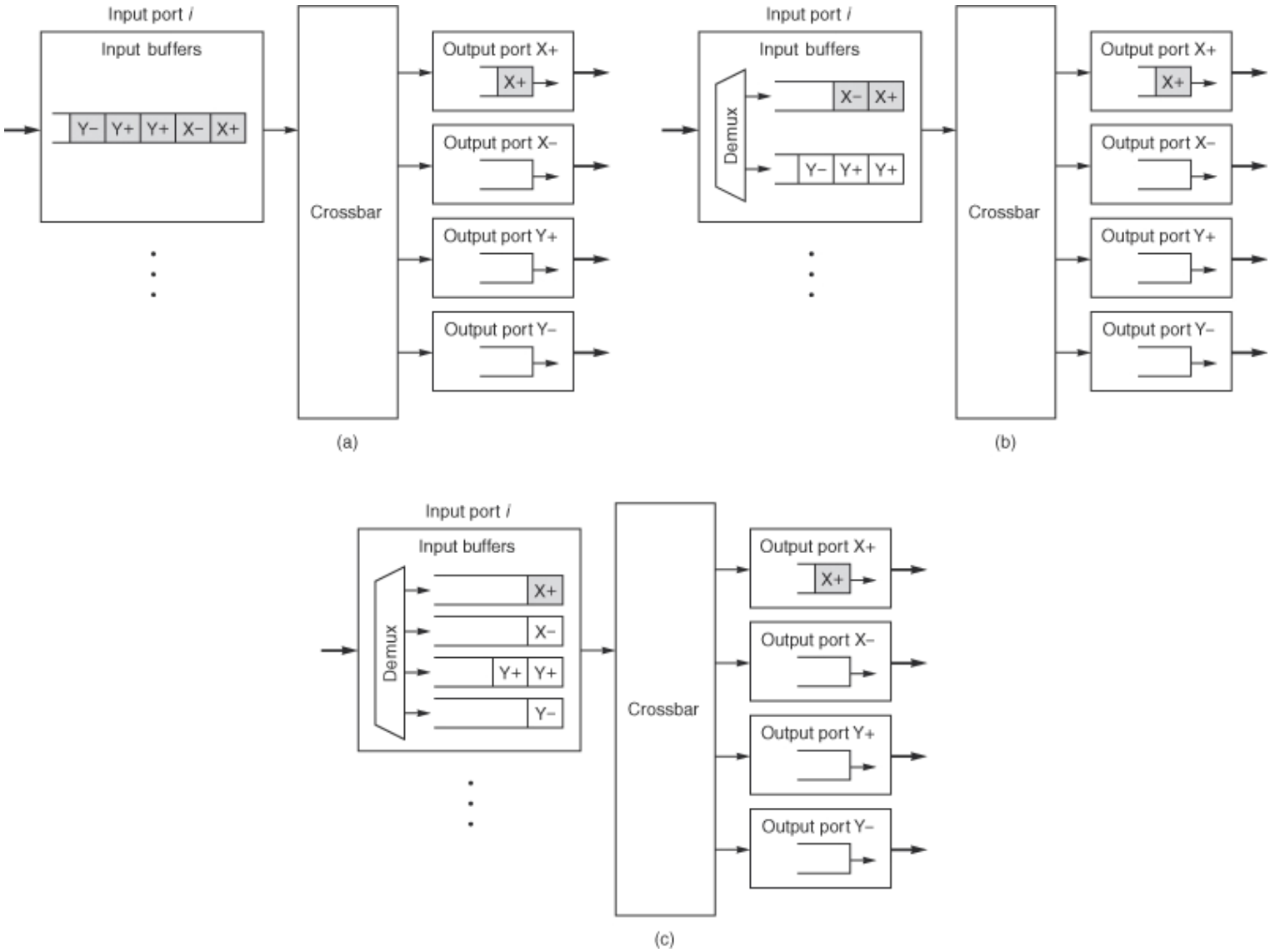
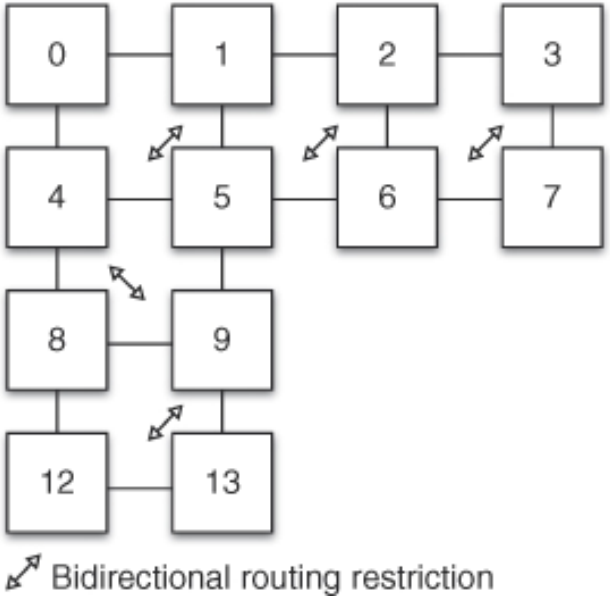


Figure F.22 (a) Head-of-line blocking in an input buffer, (b) the use of two virtual channels to reduce HOL blocking, and (c) the use of virtual output queuing to eliminate HOL blocking with a switch. The shaded input buffer is the one to which the crossbar is currently allocated. This assumes each input 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
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	1
-	-
-	-
1	1
1	1
-	-
-	-

Ren	Res
1	1
1	1
1	1
1	1
0	1
0	1
0	1
1	1
1	1
1	1
-	-
-	-
0	1
1	1
-	-
-	-

Rwn	Rws
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	1
0	1
-	-
-	-
1	1
1	1
-	-
-	-

Rse	Rsw
1	1
1	0
1	0
1	0
0	1
1	1
1	1
1	1
1	1
1	0
-	-
-	-
1	1
1	1
-	-
-	-

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.

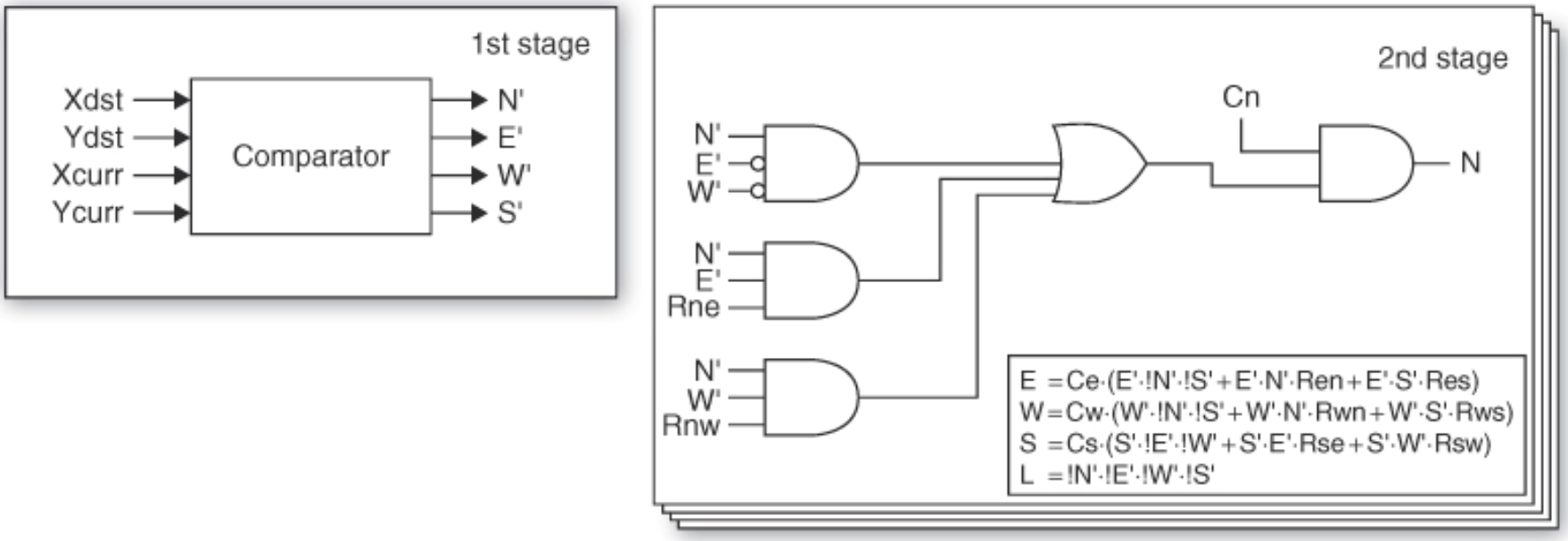


Figure F.28 SCC Top-level architecture.

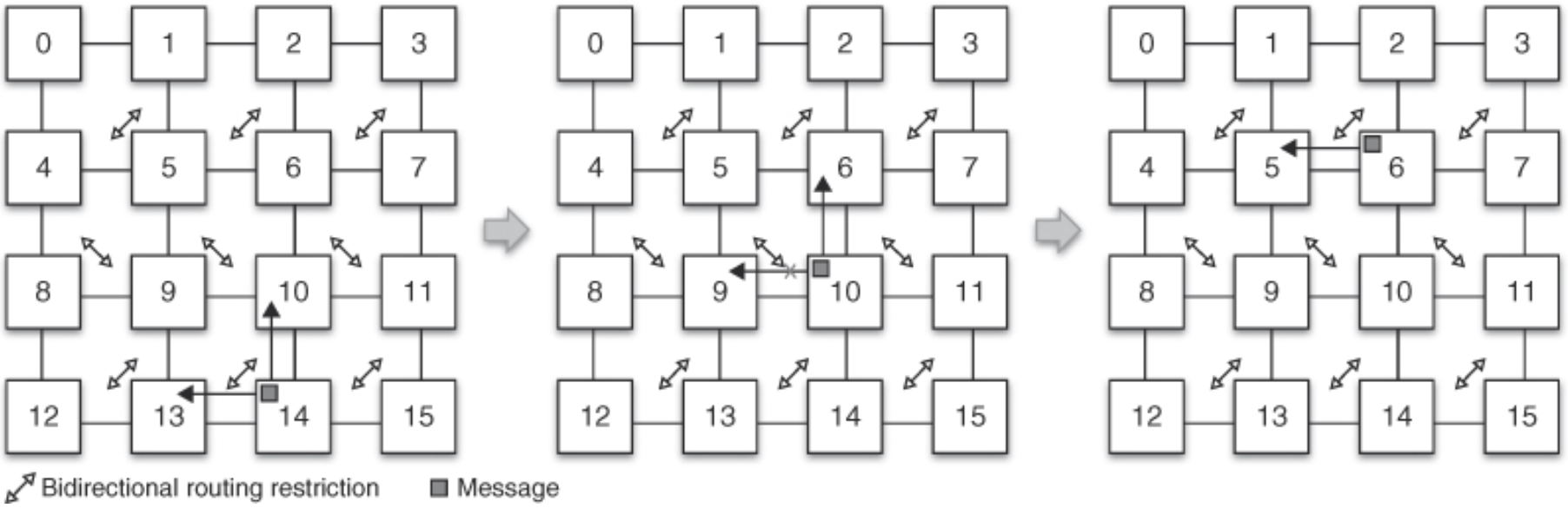


Figure F.30 Packet format for InfiniBand, Ethernet, and ATM. ATM calls their messages "cells" instead of packets, so the proper name is ATM cell format. The width of each drawing is 32 bits. All three formats have destination addressing fields, encoded differently for each situation. All three also have 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.

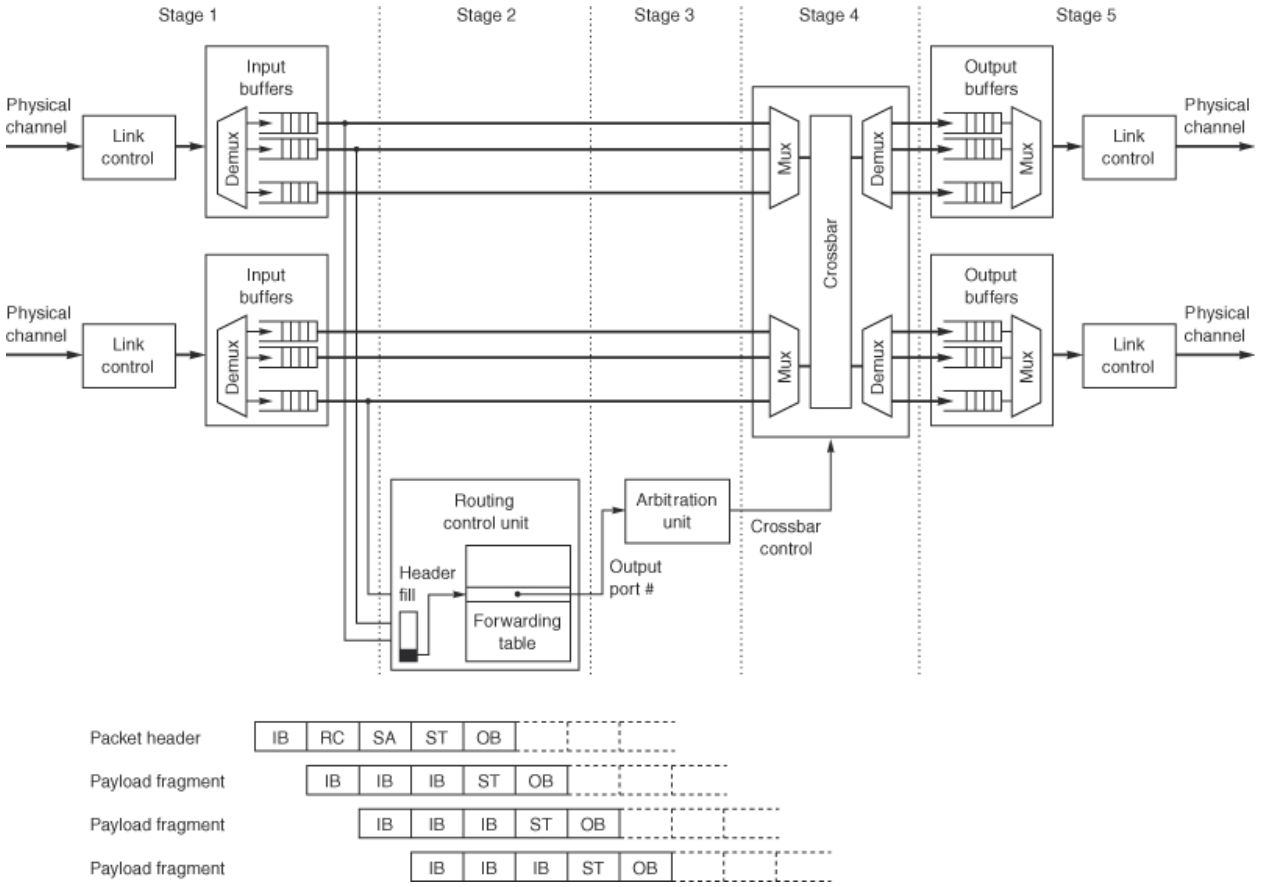


Figure F.32 Data collected by D.K. Panda, S. Sur, and L. Chai (2005) in the Network-Based Computing Laboratory at The Ohio State University. (a) Cumulative percentage of messages and volume of data transferred as message size varies for the Fluent application (www.fluent.com). Each *x*-axis entry includes 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).

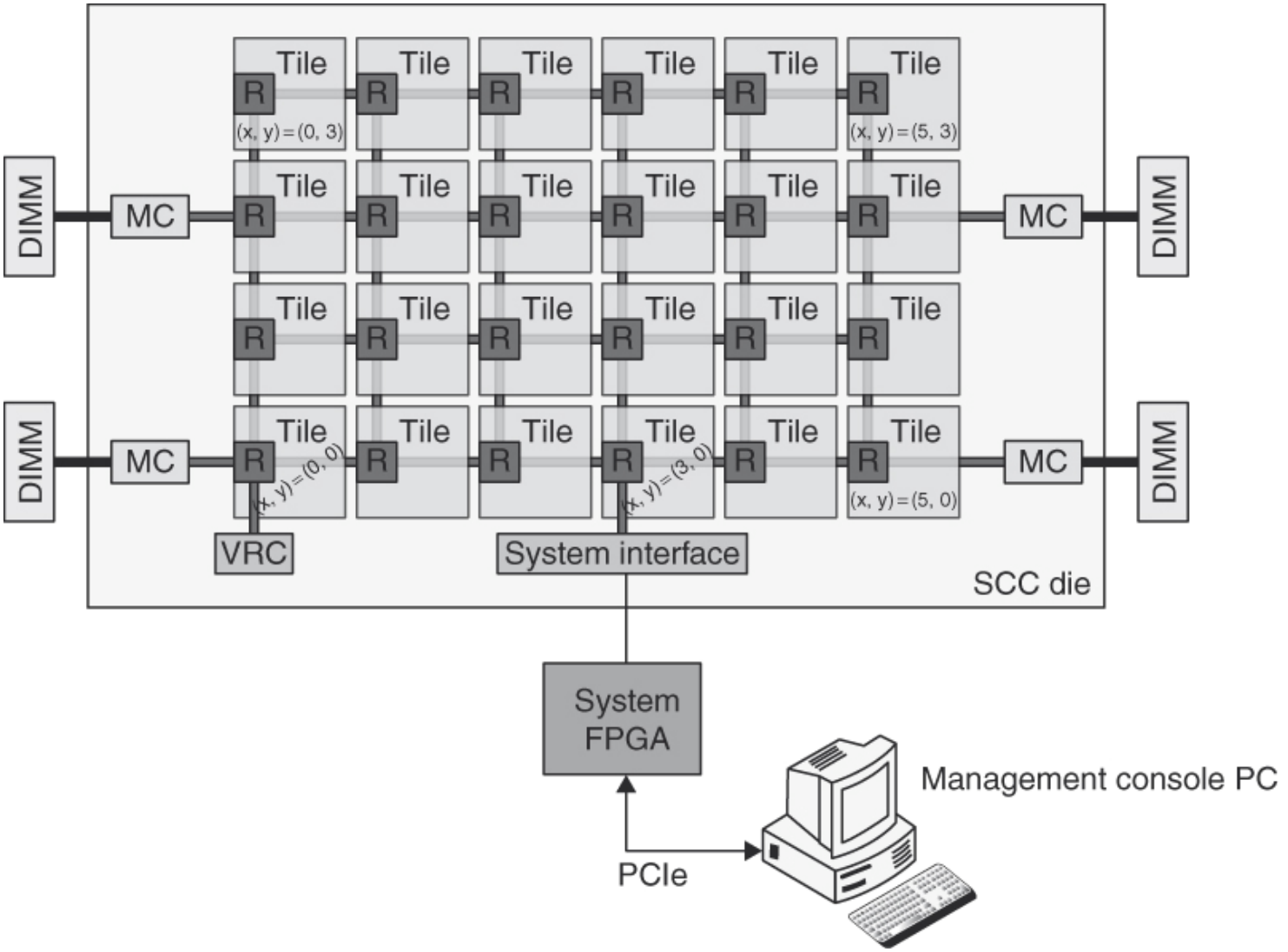


Figure F.33 The potential increased bandwidth of using many Ethernets and bridges.

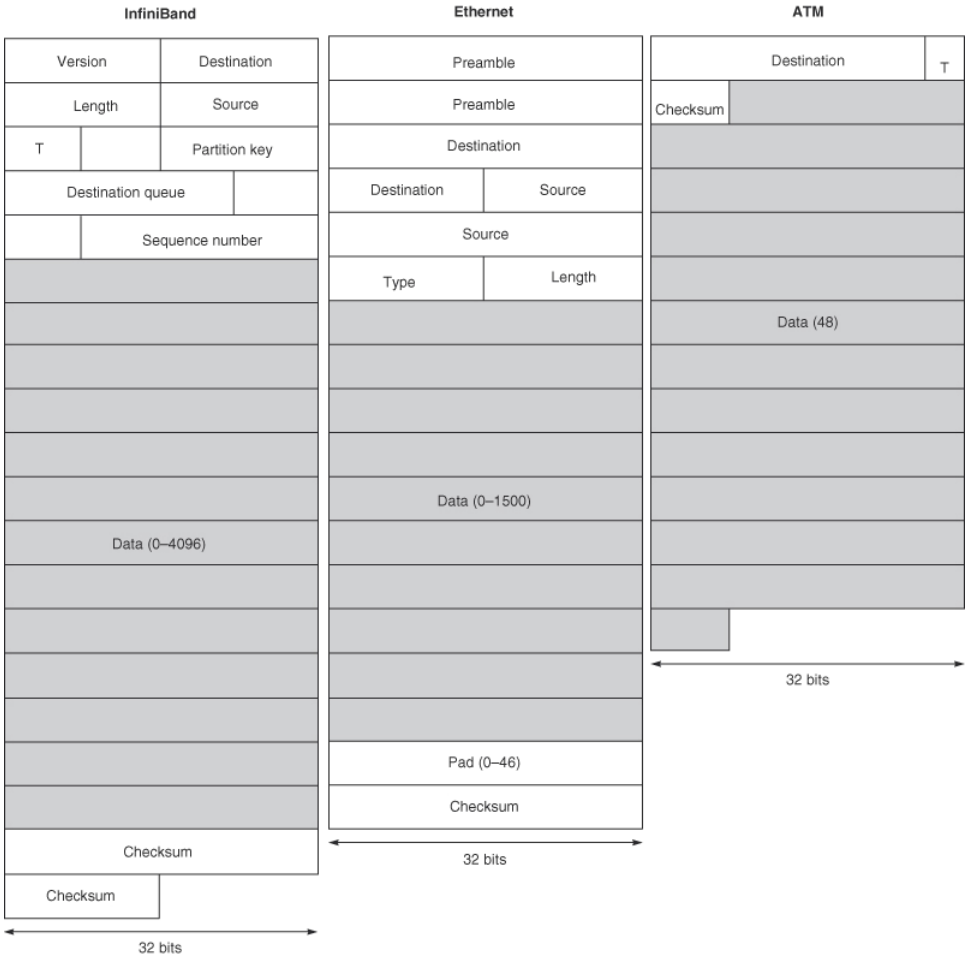


Figure F.34 The connection established between mojave.stanford.edu and mammoth.berkeley.edu (1995). FDDI is a 100 Mbit/sec LAN, while a T1 line is a 1.5 Mbit/sec telecommunications line and a T3 is a 45 Mbit/sec telecommunications line. BARRNet stands for Bay Area Research Network. Note that inr-111-cs2.Berkeley.edu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

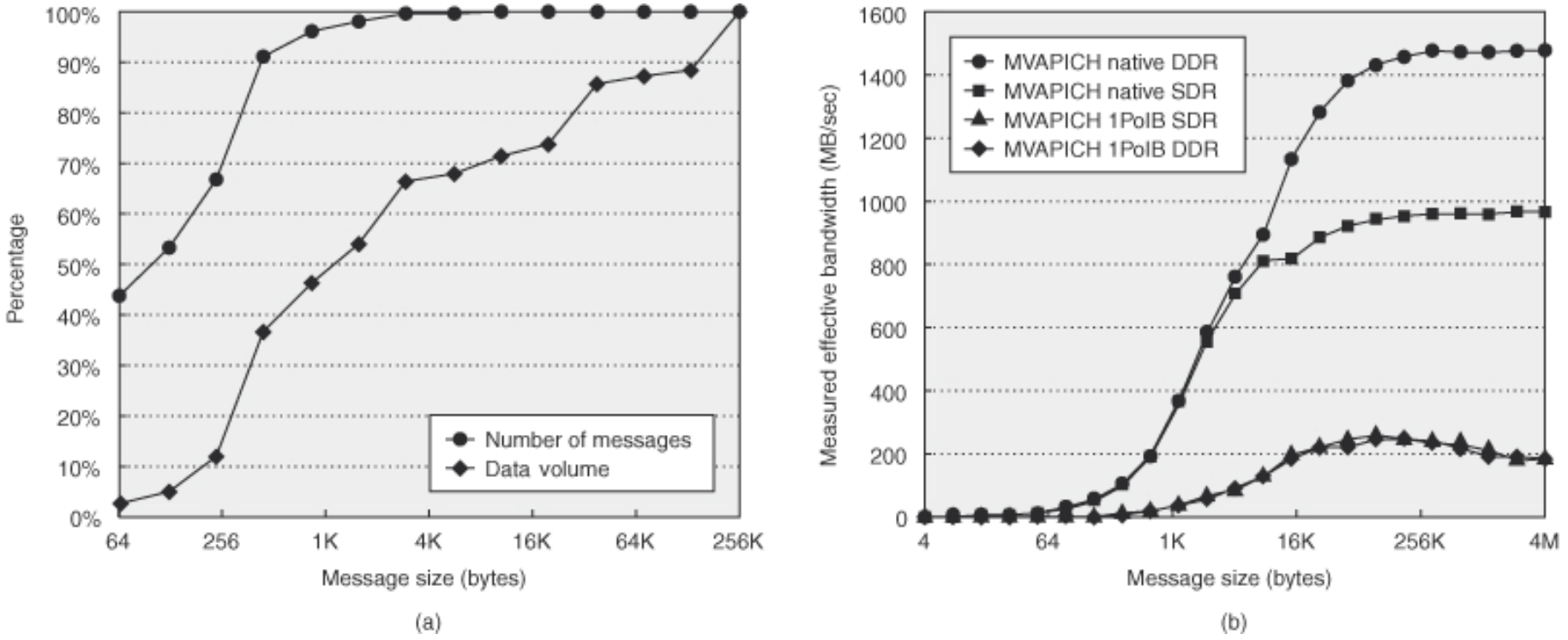


Figure F.35 The role of internetworking. The width indicates the relative number of items at each level.

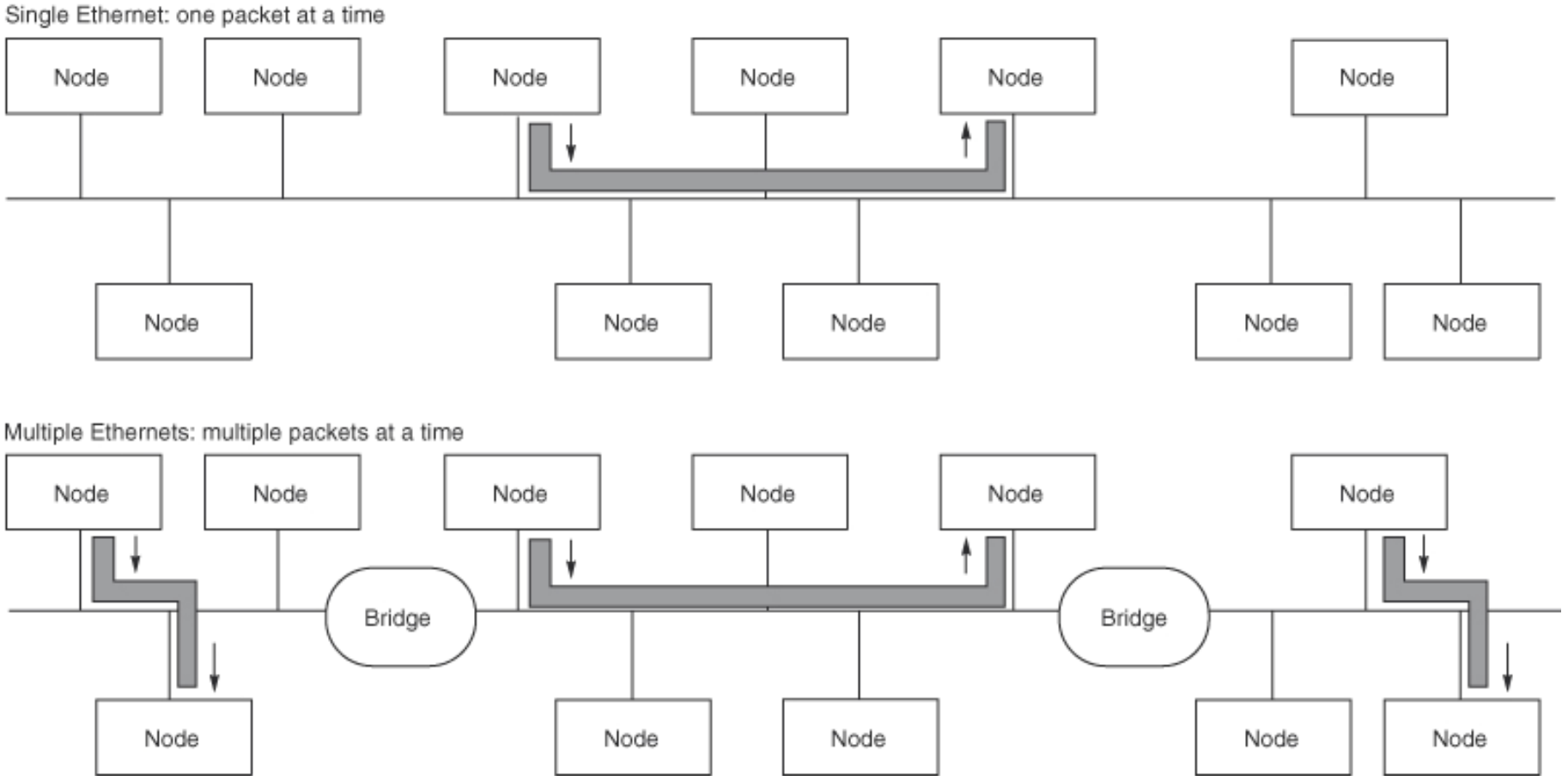


Figure F.37 A generic protocol stack with two layers. Note that communication is peer-to-peer, with headers and trailers for the peer added at each sending layer and removed by each receiving layer. Each layer offers services to the one above to shield it from unnecessary details.

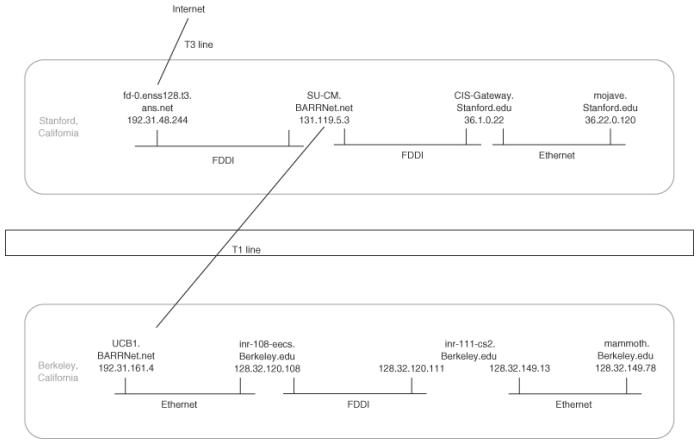


Figure F.38 The headers for IP and TCP. This drawing is 32 bits wide. The standard headers for both are 20 bytes, but both allow the headers to optionally lengthen for rarely transmitted information. Both headers have a length of header field (L) to accommodate the optional fields, as well as source 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. In 2006, version six of the Internet protocol (IPv6) was widely used.

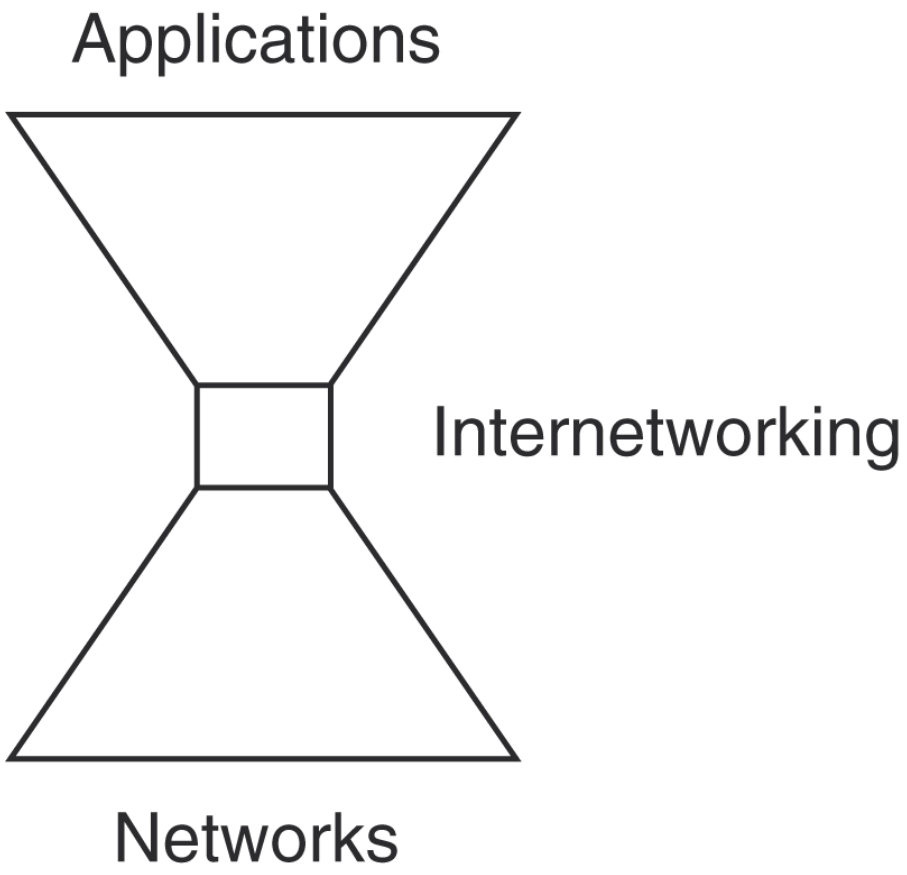


Figure F.39 Intelligence in a network: switch versus network interface card. Note that Ethernet switches come in two styles, depending on the size of the network, and that InfiniBand network interfaces come in two styles, depending on whether they are attached to a computer or to a storage device. Myrinet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