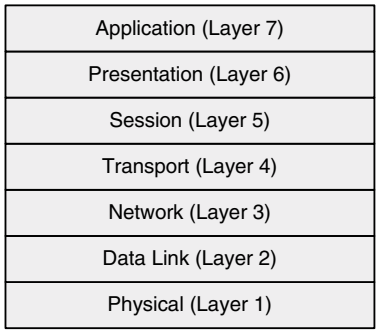

FIGURE 9.1 OSI seven-layer model.

may segment an OSI layer into multiple sublayers. The consistency of definitions decreases as one moves up the stack, because of functional protocol variations.

Layer one, the *physical layer*, comprises the electromechanical characteristics of the medium used to convey bits of information. The use of twisted pair cable, the amplitude of 1s and 0s, and associated connectors and transducers are examples of that which is specified in the physical layer. Channel coding, how the bits are represented on the physical medium, is usually classified as part of the physical layer.

The *data link layer*, layer two, encompasses the control logic and frame formatting that enables data to be injected into the network's physical layer and retrieved at the destination node. Layer-two functions are usually handled by a *media access controller* (MAC), a hardware device that contains all of the logic necessary to gain access to the network medium, properly format and transmit a frame, and properly detect and process an incoming frame. Network frame formats specify data link layer characteristics. Link level error detection mechanisms such as checksums and CRCs (more on these later) are generated and verified by the MAC. Node addresses, called *MAC addresses* in Ethernet networks, are layer-two constructs that uniquely identify individual nodes. Layer-two functions are usually handled in hardware, because they are repetitive, high-frequency, and time-critical operations. The data link layer is closely tied to the topology of the network because of its handling of access control functions and unique node addresses. Network *switches* operate at layer two by knowing which node address is connected to which port and then directing traffic to the relevant port. If port 20 of a switch is connected to node 87, all frames that enter the switch destined for node 87 will be sent out port 20. Because it is necessary to maintain unique layer-two addresses, they are generally not under the control of the user but rather are configured by the manufacturer. In the case of Ethernet, each manufacturer of equipment licenses an arbitrary range of MAC addresses from the IEEE and then assigns them one at a time as products roll off the assembly lines.

9.2 PROTOCOL LAYERS THREE AND FOUR

More flexible communications are possible when a protocol is not tied too closely to network topology or even the type of network accomplishing the exchange of information. The *network layer*, layer three, enables nodes to establish end-to-end connections without strict knowledge of the network topology. Layer-three packets are encapsulated within the payload of a layer-two frame. The packets typically contain their own header, payload, and sometimes a trailer as well. Perhaps the most common example of a layer-three protocol is *Internet Protocol* (IP). IP packets consist of a

header and payload. Included within the header are 32-bit layer-three destination and source *IP addresses*. A separate set of network addresses can be implemented at layer three that is orthogonal to layer-two addresses. This gives network nodes two different addresses: one at layer three and one at layer two. For a simple network, this may appear to be redundant and inefficient. Yet modern networking protocols must support complex topologies that span buildings and continents, often with a mix of data links connecting many smaller subnetworks that may cover a single office or floor of a building. The benefit of layer-three addressing and communication is that traffic can be carried on a variety of underlying communications interfaces and not require the end points to know the exact characteristics of each interface.

Network *routers* operate at layer three by separating the many subnetworks that make up a larger network and only passing traffic that must travel between the subnetworks. Network addresses are typically broken into *subnets* that correlate to physically distinct portions of the network. A router has multiple ports, each of which is connected to a different subnetwork that is represented by a range of network addresses. A frame entering a router port will not be sent to another particular port on that router unless its network address matches a subnet configuration on that particular port. Strictly speaking, this separation could be performed by layer-two addressing, but the practical reality is that layer-two addresses are often not under the user's control (e.g., Ethernet) and therefore cannot be organized in a meaningful way. In contrast, layer-three addresses are soft properties of each network installation and are not tied to a particular type of network medium.

Layer-three functions are performed by both hardware and software according to the specific implementation and context. Layer-three packets are usually first generated by software but then manipulated by hardware as they flow through the network. A typical router processes layer-three packets in hardware so that it does not fall behind the flow of traffic and cause a bottleneck.

The bottom three layers cumulatively move data from one place to another but sometimes do not have the ability to actually guarantee that the data arrived intact. Layers one and two are collectively responsible for moving properly formatted frames onto the network medium and then recovering those in transit. The network layer adds some addressing flexibility on top of this basic function. A true end-to-end guarantee of data delivery is missing from certain lower-level protocols (e.g., Ethernet and IP) because of the complexity that this guarantee adds.

The *transport layer*, layer four, is responsible for ensuring end-to-end communication between software services running on each node. Transport layer complexity varies according to the demands of the application. Many applications are written with the simplifying assumption that once data is passed to the transport layer for transmission, it is guaranteed to arrive at the destination. *Transmission control protocol* (TCP) is one of the most common layer-four protocols, because it is used to guarantee the delivery of data across an unreliable IP network. When communicating via TCP, an application can simply transfer the desired information and then move on to new tasks. TCP is termed a *stateful* protocol, because it retains information about packets after they are sent until their successful arrival has been acknowledged. TCP operates using a sliding data transmission window shown in Fig. 9.2 and overlays a 32-bit range of indices onto the data that is being sent. Pointers are referenced into this 32-bit range to track data as it is transmitted and received.

The basic idea behind TCP is that the transmitter retains a copy of data that has already been sent until it receives an acknowledgement that the data was properly received at the other end. If an ac-

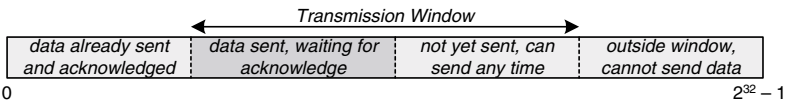

FIGURE 9.2 TCP transmission window.