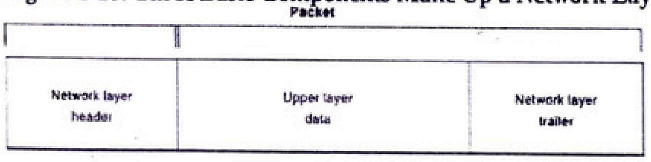
GATE CSE NOTES

by

UseMyNotes

A *packet* is an information unit whose source and destination are network layer entities. A packet is composed of the network layer header (and possibly a trailer) and upper-layer data. The header and trailer contain control information intended for the network layer entity in the destination system. Data from upper-layer entities is encapsulated in the network layer header and trailer. Figure 1-10 illustrates the basic components of a network layer packet.

Figure 1-10: Three Basic Components Make Up a Network Layer Packet



The term ***datagram*** usually refers to an information unit whose source and destination are network layer entities that use connectionless network service.

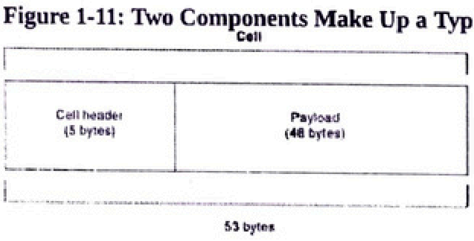
The term ***segment*** usually refers to an information unit whose source and destination are transport layer entities.

A ***message*** is an information unit whose source and destination entities exist above the network layer (often at the application layer).

A ***cell*** is an information unit of a fixed size whose source and destination are data link layer entities. Cells are used in switched environments, such as Asynchronous Transfer Mode (ATM) and Switched Multimegabit Data Service (SMDS) networks. A cell is composed of the header and payload. The header contains control information intended for the destination data link layer entity and is typically 5 bytes long. The payload contains upper-layer data that is encapsulated in the cell header and is typically 48 bytes long.

The length of the header and the payload fields always are the same for each cell.
Figure 1-11 depicts the components of a typical cell.

Figure 1-11: Two Components Make Up a Typical Cell



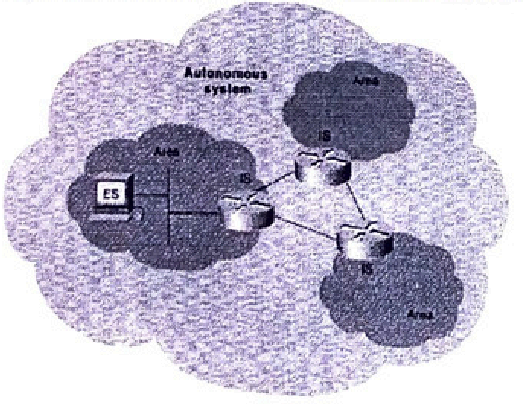
Data unit is a generic term that refers to a variety of information units. Some common data units are service data units (SDUs), protocol data units, and bridge protocol data units (BPDUs). SDUs are information units from upper-layer protocols that define a service request to a lower-layer protocol. PDU is OSI terminology for a packet. BPDUs are used by the spanning-tree algorithm as hello messages.

ISO Hierarchy of Networks

Large networks typically are organized as hierarchies. A hierarchical organization provides such advantages as ease of management, flexibility, and a reduction in unnecessary traffic. Thus, the International Organization for Standardization (ISO) has adopted a number of terminology conventions for addressing network entities. Key terms defined in this section include end system (ES), intermediate system (IS), area, and autonomous system (AS).

An ***ES*** is a network device that does not perform routing or other traffic forwarding functions. Typical ESs include such devices as terminals, personal computers, and printers. An ***IS*** is a network device that performs routing or other traffic-forwarding functions. Typical ISs include such devices as routers, switches, and bridges. Two types of IS networks exist: intradomain IS and interdomain IS. An intradomain IS communicates within a single autonomous system, while an interdomain IS communicates within and between autonomous systems. An ***area*** is a logical group of network segments and their attached devices. Areas are subdivisions of autonomous systems (AS's). An AS is a collection of networks under a common administration that share a common routing strategy. Autonomous systems are subdivided into areas, and an AS is sometimes called a domain. Figure 1-12 illustrates a hierarchical network and its components.

Figure 1-12: A Hierarchical Network Contains Numerous Components



Connection-Oriented and Connectionless Network Services

In general, transport protocols can be characterized as being either connection-oriented or connectionless. Connection-oriented services must first establish a connection with the desired service before passing any data. A connectionless service can send the data without any need to establish a connection first. In general, connection-oriented services provide some level of delivery guarantee, whereas connectionless services do not.

Connection-oriented service involves three phases: connection establishment, data transfer, and connection termination.

During connection establishment, the end nodes may reserve resources for the connection. The end nodes also may negotiate and establish certain criteria for the transfer, such as a window size used in TCP connections. This resource reservation is one of the things exploited in some denial of service (DOS) attacks. An attacking system will send many requests for establishing a connection but then will never complete the connection. The attacked computer is then left with resources allocated for many never-completed connections. Then, when an end node tries to complete an actual connection, there are not enough resources for the valid connection.

The data transfer phase occurs when the actual data is transmitted over the connection. During data transfer, most connection-oriented services will monitor for lost packets and handle resending them. The protocol is generally also responsible for putting the packets in the right sequence before passing the data up the protocol stack.

When the transfer of data is complete, the end nodes terminate the connection and release resources reserved for the connection.

Connection-oriented network services have more overhead than connectionless ones. Connection-oriented services must negotiate a connection, transfer data, and tear down the connection, whereas a connectionless transfer can simply send the data without the added overhead of creating and tearing down a connection. Each has its place in internetworks.

Internetwork Addressing

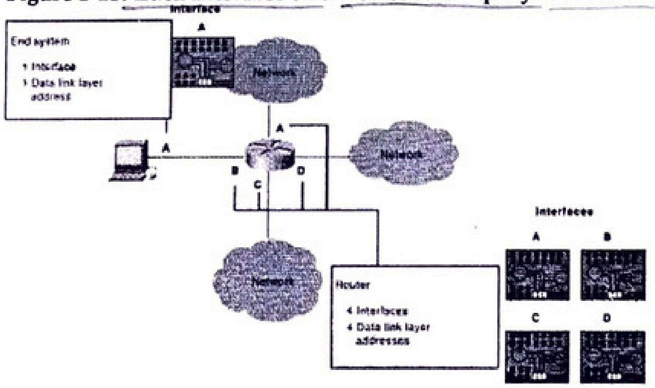
Internetwork addresses identify devices separately or as members of a group. Addressing schemes vary depending on the protocol family and the OSI layer. Three types of internetwork addresses are commonly used: data link layer addresses, Media Access Control (MAC) addresses, and network layer addresses.

Data Link Layer Addresses

A ***data link layer address*** uniquely identifies each physical network connection of a network device. Data-link addresses sometimes are referred to as ***physical*** or ***hardware addresses***. Data-link addresses usually exist within a flat address space and have a pre-established and typically fixed relationship to a specific device.

End systems generally have only one physical network connection and thus have only one data-link address. Routers and other internetworking devices typically have multiple physical network connections and therefore have multiple data-link addresses. Figure 1-13 illustrates how each interface on a device is uniquely identified by a data-link address.

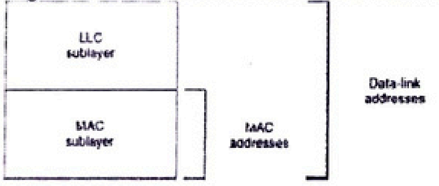
Figure 1-13: Each Interface on a Device Is Uniquely Identified by a Data-Link Address.



MAC Addresses

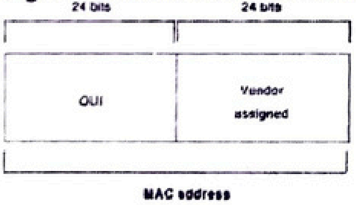
Media Access Control (MAC) addresses consist of a subset of data link layer addresses. MAC addresses identify network entities in LANs that implement the IEEE MAC addresses of the data link layer. As with most data-link addresses, MAC addresses are unique for each LAN interface. Figure 1-14 illustrates the relationship between MAC addresses, data-link addresses, and the IEEE sublayers of the data link layer.

Figure 1-14: MAC Addresses, Data-Link Addresses, and the IEEE Sublayers of the Data Link Layer Are All Related



MAC addresses are 48 bits in length and are expressed as 12 hexadecimal digits. The first 6 hexadecimal digits, which are administered by the IEEE, identify the manufacturer or vendor and thus comprise the Organizationally Unique Identifier (OUI). The last 6 hexadecimal digits comprise the interface serial number, or another value administered by the specific vendor. MAC addresses sometimes are called ***burned-in addresses (BIAs)*** because they are burned into read-only memory (ROM) and are copied into random-access memory (RAM) when the interface card initializes. Figure 1-15 illustrates the MAC address format.

Figure 1-15: The MAC Address Contains a Unique Format of Hexadecimal Digits



Mapping Addresses

Because internetworks generally use network addresses to route traffic around the network, there is a need to map network addresses to MAC addresses. When the network layer has determined the destination station's network address, it must forward the information over a physical network using a MAC address. Different protocol suites use different methods to perform this mapping, but the most popular is Address Resolution Protocol (ARP).

Different protocol suites use different methods for determining the MAC address of a device. The following three methods are used most often. Address Resolution Protocol (ARP) maps network addresses to MAC addresses. The Hello protocol enables network devices to learn the MAC addresses of other network devices. MAC addresses either are embedded in the network layer address or are generated by an algorithm.

Address Resolution Protocol (ARP) is the method used in the TCP/IP suite. When a network device needs to send data to another device on the same network, it knows the source and destination network addresses for the data transfer. It must somehow map the destination address to a MAC address before forwarding the data. First, the sending station will check its ARP table to see if it has already discovered this destination station's MAC address. If it has not, it will send a broadcast on the network with the destination station's IP address contained in the broadcast. Every station on the network receives the broadcast and compares the embedded IP address to its own. Only the station with the matching IP address replies to the sending station with a packet containing the MAC address for the station. The first station then adds this information to its ARP table for future reference and proceeds to transfer the data.

When the destination device lies on a remote network, one beyond a router, the process is the same except that the sending station sends the ARP request for the MAC address of its default gateway. It then forwards the information to that device. The default gateway will then forward the information over whatever networks necessary to deliver the packet to the network on which the destination device resides. The router on the destination device's network then uses ARP to obtain the MAC of the actual destination device and delivers the packet.

The Hello protocol is a network layer protocol that enables network devices to identify one another and indicate that they are still functional. When a new end system powers up, for example, it broadcasts hello messages onto the network. Devices on the network then return hello replies, and hello messages are also sent at specific intervals to indicate that they are still functional. Network devices can learn the MAC addresses of other devices by examining Hello protocol packets.

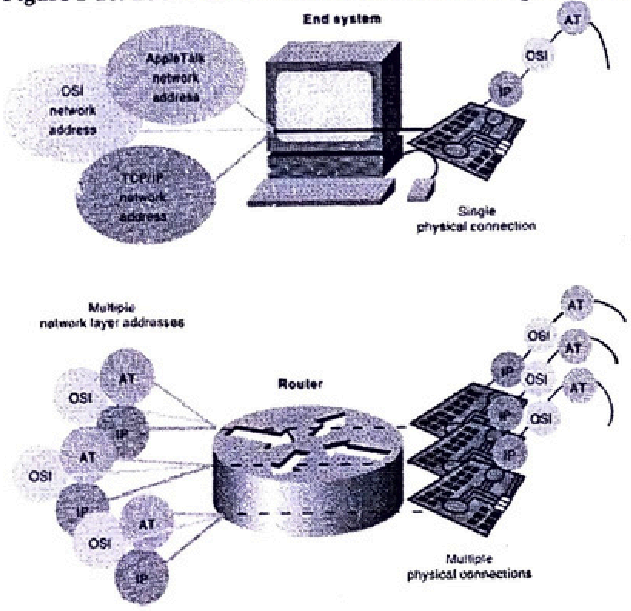
Three protocols use predictable MAC addresses. In these protocol suites, MAC addresses are predictable because the network layer either embeds the MAC address in the network layer address or uses an algorithm to determine the MAC address. The three protocols are Xerox Network Systems (XNS), Novell Internetwork Packet Exchange (IPX), and DECnet Phase IV.

Network Layer Addresses

A *network layer address* identifies an entity at the network layer of the OSI layers. Network addresses usually exist within a hierarchical address space and sometimes are called *virtual* or *logical addresses*.

The relationship between a network address and a device is logical and unfixed; it typically is based either on physical network characteristics (the device is on a particular network segment) or on groupings that have no physical basis (the device is part of an AppleTalk zone). End systems require one network layer address for each network layer protocol that they support. (This assumes that the device has only one physical network connection.) Routers and other internetworking devices require one network layer address per physical network connection for each network layer protocol supported. For example, a router with three interfaces each running AppleTalk, TCP/IP, and OSI must have three network layer addresses for each interface. The router therefore has nine network layer addresses. Figure 1-16 illustrates how each network interface must be assigned a network address for each protocol supported.

Figure 1-16: Each Network Interface Must Be Assigned a Network Address for Each Protocol Supported


End system
AppleTalk network address
OSI network address
TCP/IP network address
AT
OSI
IP
Single physical connection
Multiple network layer addresses
Router
AT
OSI
IP
AT
OSI
IP
AT
OSI
IP
Multiple physical connections

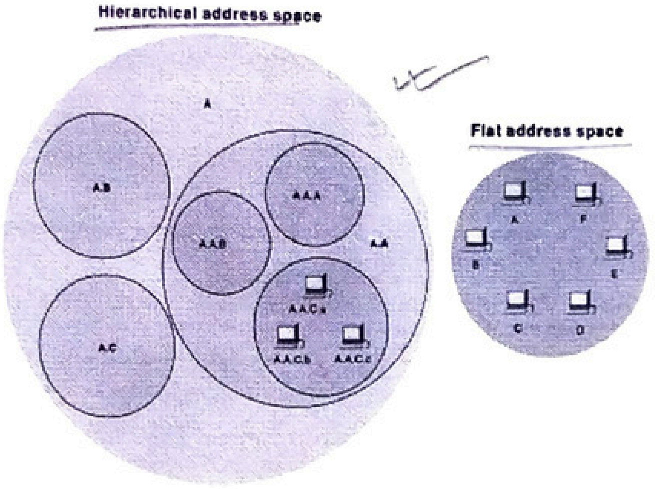
Data LL addr. → physical
N/W L addr → logical/
virtual

Hierarchical Versus Flat Address Space

Internetwork address space typically takes one of two forms: hierarchical address space or flat address space. A ***hierarchical address space*** is organized into numerous subgroups, each successively narrowing an address until it points to a single device (in a manner similar to street addresses). A ***flat address space*** is organized into a single group (in a manner similar to U.S. Social Security numbers).

Hierarchical addressing offers certain advantages over flat-addressing schemes. Address sorting and recall is simplified using comparison operations. For example, "Ireland" in a street address eliminates any other country as a possible location. Figure 1-17 illustrates the difference between hierarchical and flat address spaces.

Figure 1-17: Hierarchical and Flat Address Spaces Differ in Comparison Operations



Address Assignments

Addresses are assigned to devices as one of two types: static and dynamic. ***Static addresses*** are assigned by a network administrator according to a preconceived internetwork addressing plan. A static address does not change until the network administrator manually changes it. ***Dynamic addresses*** are obtained by devices when they attach to a network, by means of some protocol-specific process. A device using a dynamic address often has a different address each time that it connects to the network. Some networks use a server to assign addresses. Server-assigned addresses are recycled for reuse as devices disconnect.
A device is therefore likely to have a different address each time that it connects to the network.

Addresses Versus Names

Internetwork devices usually have both a name and an address associated with them. Internetwork names typically are location-independent and remain associated with a device wherever that device moves (for example, from one building to another). Internetwork addresses usually are location-dependent and change when a device is moved (although MAC addresses are an exception to this rule). As with network addresses being mapped to MAC addresses, names are usually mapped to network addresses through some protocol. The Internet uses Domain Name System (DNS) to map the name of a device to its IP address. For example, it's easier for you to remember www.cisco.com instead of some IP address. Therefore, you type www.cisco.com into your browser when you want to access Cisco's web site. Your computer performs a DNS lookup of the IP address for Cisco's web server and then communicates with it using the network address.

Flow Control Basics

Flow control is a function that prevents network congestion by ensuring that transmitting devices do not overwhelm receiving devices with data. A high-speed computer, for example, may generate traffic faster than the network can transfer it, or faster than the destination device can receive and process it. The three commonly used methods for handling network congestion are buffering, transmitting source-quench messages, and windowing.

Buffering is used by network devices to temporarily store bursts of excess data in memory until they can be processed. Occasional data bursts are easily handled by buffering. Excess data bursts can exhaust memory, however, forcing the device to discard any additional datagrams that arrive.

Source-quench messages are used by receiving devices to help prevent their buffers from overflowing. The receiving device sends source-quench messages to request that the source reduce its current rate of data transmission. First, the receiving device begins discarding received data due to overflowing buffers. Second, the receiving device begins sending source-quench messages to the transmitting device at the rate of one message for each packet dropped. The source device receives the source-quench messages and lowers the data rate until it stops receiving the messages. Finally, the source device then gradually increases the data rate as long as no further source-quench requests are received.

Windowing is a flow-control scheme in which the source device requires an acknowledgment from the destination after a certain number of packets have been transmitted. With a window size of 3, the source requires an acknowledgment after sending three packets, as follows. First, the source device sends three packets to the destination device. Then, after receiving the three packets, the destination device sends an acknowledgment to the source. The source receives the acknowledgment and sends three more packets. If the destination does not receive one or more of the packets for some reason, such as overflowing buffers, it does not receive enough packets to send an acknowledgment. The source then retransmits the packets at a reduced transmission rate.

Error-Checking Basics

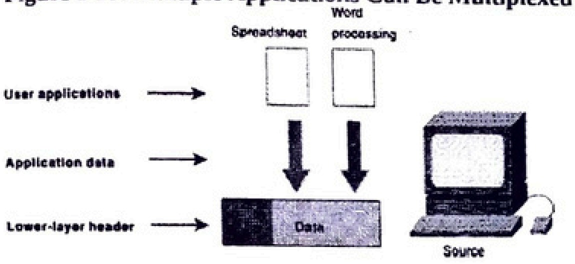
Error-checking schemes determine whether transmitted data has become corrupt or otherwise damaged while traveling from the source to the destination. Error checking is implemented at several of the OSI layers.

One common error-checking scheme is the cyclic redundancy check (CRC), which detects and discards corrupted data. Error-correction functions (such as data retransmission) are left to higher-layer protocols. A CRC value is generated by a calculation that is performed at the source device. The destination device compares this value to its own calculation to determine whether errors occurred during transmission. First, the source device performs a predetermined set of calculations over the contents of the packet to be sent. Then, the source places the calculated value in the packet and sends the packet to the destination. The destination performs the same predetermined set of calculations over the contents of the packet and then compares its computed value with that contained in the packet. If the values are equal, the packet is considered valid. If the values are unequal, the packet contains errors and is discarded.

Multiplexing Basics

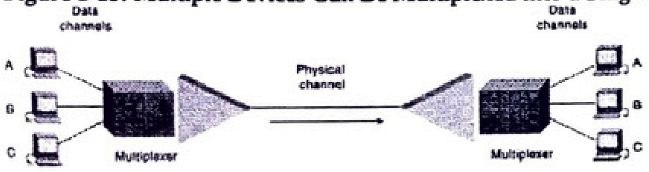
Multiplexing is a process in which multiple data channels are combined into a single data or physical channel at the source. Multiplexing can be implemented at any of the OSI layers. Conversely, ***demultiplexing*** is the process of separating multiplexed data channels at the destination. One example of multiplexing is when data from multiple applications is multiplexed into a single lower-layer data packet. Figure 1-18 illustrates this example.

Figure 1-18: Multiple Applications Can Be Multiplexed into a Single Lower-Layer Data Packet



Another example of multiplexing is when data from multiple devices is combined into a single physical channel (using a device called a multiplexer). Figure 1-19 illustrates this example.

Figure 1-19: Multiple Devices Can Be Multiplexed into a Single Physical Channel



A *multiplexer* is a physical layer device that combines multiple data streams into one or more output channels at the source. Multiplexers demultiplex the channels into multiple data streams at the remote end and thus maximize the use of the bandwidth of the physical medium by enabling it to be shared by multiple traffic sources.

Some methods used for multiplexing data are time-division multiplexing (TDM), asynchronous time-division multiplexing (ATDM), frequency-division multiplexing (FDM), and statistical multiplexing.

In TDM, information from each data channel is allocated bandwidth based on preassigned time slots, regardless of whether there is data to transmit. In ATDM, information from data channels is allocated bandwidth as needed by using dynamically assigned time slots. In FDM, information from each data channel is allocated bandwidth based on the signal frequency of the traffic. In statistical multiplexing, bandwidth is dynamically allocated to any data channels that have information to transmit.

Standards Organizations

A wide variety of organizations contribute to internetworking standards by providing forums for discussion, turning informal discussion into formal specifications, and proliferating specifications after they are standardized.

Most standards organizations create formal standards by using specific processes: organizing ideas, discussing the approach, developing draft standards, voting on all or certain aspects of the standards, and then formally releasing the completed standard to the public.

Some of the best-known standards organizations that contribute to internetworking standards include these:

- **International Organization for Standardization (ISO)**—ISO is an international standards organization responsible for a wide range of standards, including many that are relevant to networking. Its best-known contribution is the development of the OSI reference model and the OSI protocol suite.
- **American National Standards Institute (ANSI)**—ANSI, which is also a member of the ISO, is the coordinating body for voluntary standards groups within the United States. ANSI developed the Fiber Distributed Data Interface (FDDI) and other communications standards.
- **Electronic Industries Association (EIA)**—EIA specifies electrical transmission standards, including those used in networking. The EIA developed the widely used EIA/TIA-232 standard (formerly known as RS-232).
- **Institute of Electrical and Electronic Engineers (IEEE)**—IEEE is a professional organization that defines networking and other standards. The IEEE developed the widely used LAN standards IEEE 802.3 and IEEE 802.5.
- **International Telecommunication Union Telecommunication Standardization Sector (ITU-T)**—Formerly called the Committee for International Telegraph and Telephone (CCITT), ITU-T is now an international organization that develops communication standards. The ITU-T developed X.25 and other communications standards.
- **Internet Activities Board (IAB)**—IAB is a group of internetwork researchers who discuss issues pertinent to the Internet and set Internet policies through decisions and task forces. The IAB designates some Request For Comments (RFC) documents as Internet standards, including Transmission Control Protocol/Internet Protocol (TCP/IP) and the Simple Network Management Protocol (SNMP).

Summary

This chapter introduced the building blocks on which internetworks are built. Under-standing where complex pieces of internetworks fit into the OSI model will help you understand the concepts better. Internetworks are complex systems that, when viewed as a whole, are too much to understand. Only by breaking the network down into the conceptual pieces can it be easily understood. As you read and experience internetworks, try to think of them in terms of OSI layers and conceptual pieces.

Understanding the interaction between various layers and protocols makes designing, configuring, and diagnosing internetworks possible. Without understanding of the building blocks, you cannot understand the interaction between them.

Review Questions

Q—*What are the layers of the OSI model?*

A—Application, presentation, session, transport, network, data link, physical. Remember the sentence "All people seem to need data processing."

Q—*Which layer determines path selection in an internetwork?*

A—Layer 3, the network layer.