

Departament d'Arquitectura de Computadors

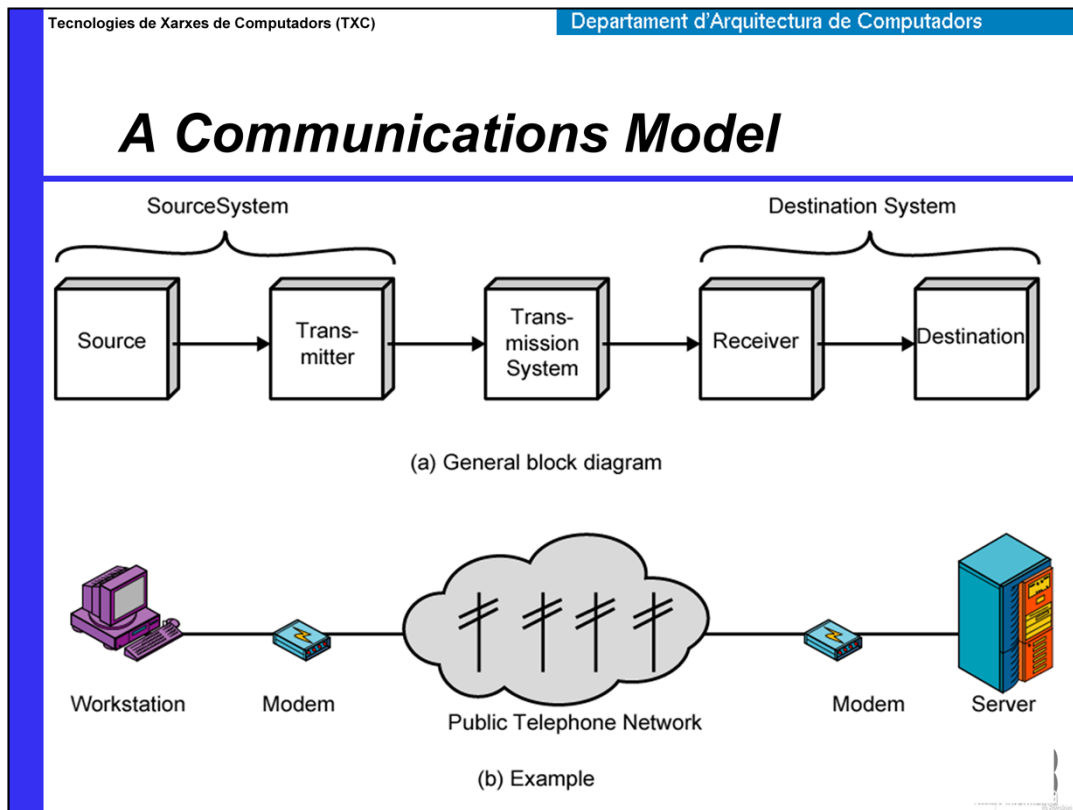
Part 1. Introducció

El paquet IP a les Xarxes de Computadors

Capítol 1 Stallings

¹ Source: Data and Computer Communications. W. Stallings.

Consultar llibre Stallings capítol 1.



The fundamental purpose of a communications system is the exchange of data between two parties. Figure 1.3b presents one particular example, which is communication between a workstation and a server over a public telephone network. Another example is the exchange of voice signals between two telephones over the same network. The following are key elements of the model:

- **Source:** This device generates the data to be transmitted; examples are telephones and personal computers.
- **Transmitter:** Usually, the data generated by a source system are not transmitted directly in the form in which they were generated. Rather, a transmitter transforms and encodes the information in such a way as to produce electromagnetic signals that can be transmitted across some sort of transmission system. For example, a modem takes a digital bit stream from an attached device such as a personal computer and transforms that bit stream into an analog signal that can be handled by the telephone network.
- **Receiver:** The receiver accepts the signal from the transmission system and converts it into a form that can be handled by the destination device. For example, a modem will accept an analog signal coming from a network or transmission line and convert it into a digital bit stream.
- **Destination:** Takes the incoming data from the receiver and
- **Transmission system:** This can be a single transmission line or a complex network connecting source and destination.

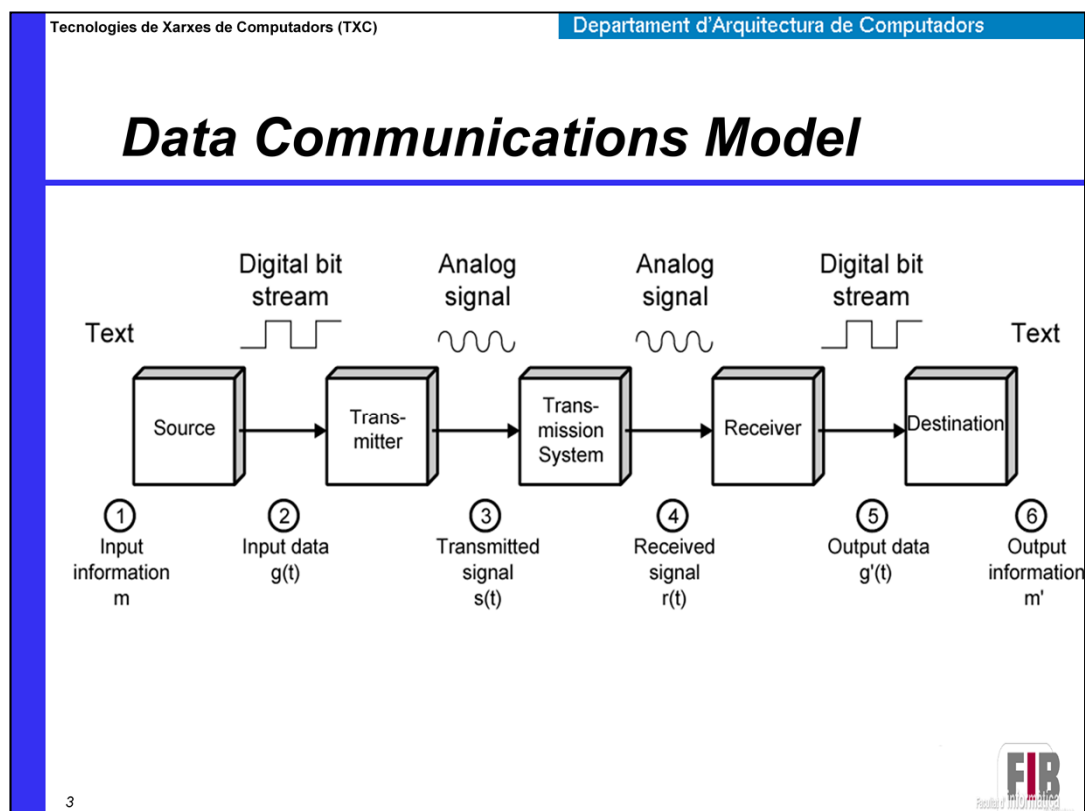
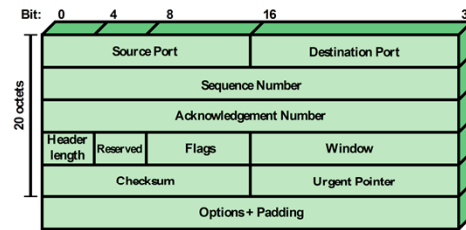


Figure 1.4 provides a new perspective on the communications model of Figure 1.3a. We trace the details of this figure using electronic mail as an example.

TCP/UDP



(a) TCP Header



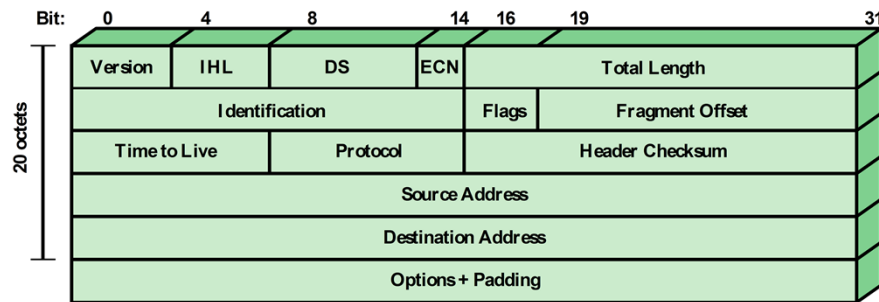
(b) UDP Header

Figure 2.6 TCP and UDP Headers



Figure 2.6a shows the header format for TCP, which is a minimum of 20 octets, or 160 bits. The Source Port and Destination Port fields identify the applications at the source and destination systems that are using this connection. The Sequence Number, Acknowledgment Number, and Window fields provide flow control and error control. The checksum is a 16-bit frame check sequence used to detect errors in the TCP segment. Chapter 15 provides more details.

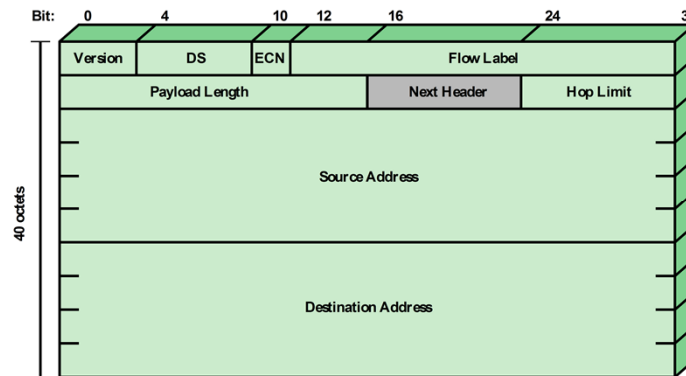
IPv4



(a) IPv4 Header

For decades, the keystone of the TCP/IP architecture has been IPv4, generally referred to as IP. Figure 2.7a shows the IP header format, which is a minimum of 20 octets, or 160 bits. The header, together with the segment from the transport layer, forms an IP-level PDU referred to as an IP datagram or an IP packet. The header includes 32-bit source and destination addresses. The Header Checksum field is used to detect errors in the header to avoid misdelivery. The Protocol field indicates which higher-layer protocol is using IP. The ID, Flags, and Fragment Offset fields are used in the fragmentation and reassembly process. Chapter 14 provides more details.

IPv6



(b) IPv6 Header

DS = Differentiated services field
ECN = Explicit congestion notification field

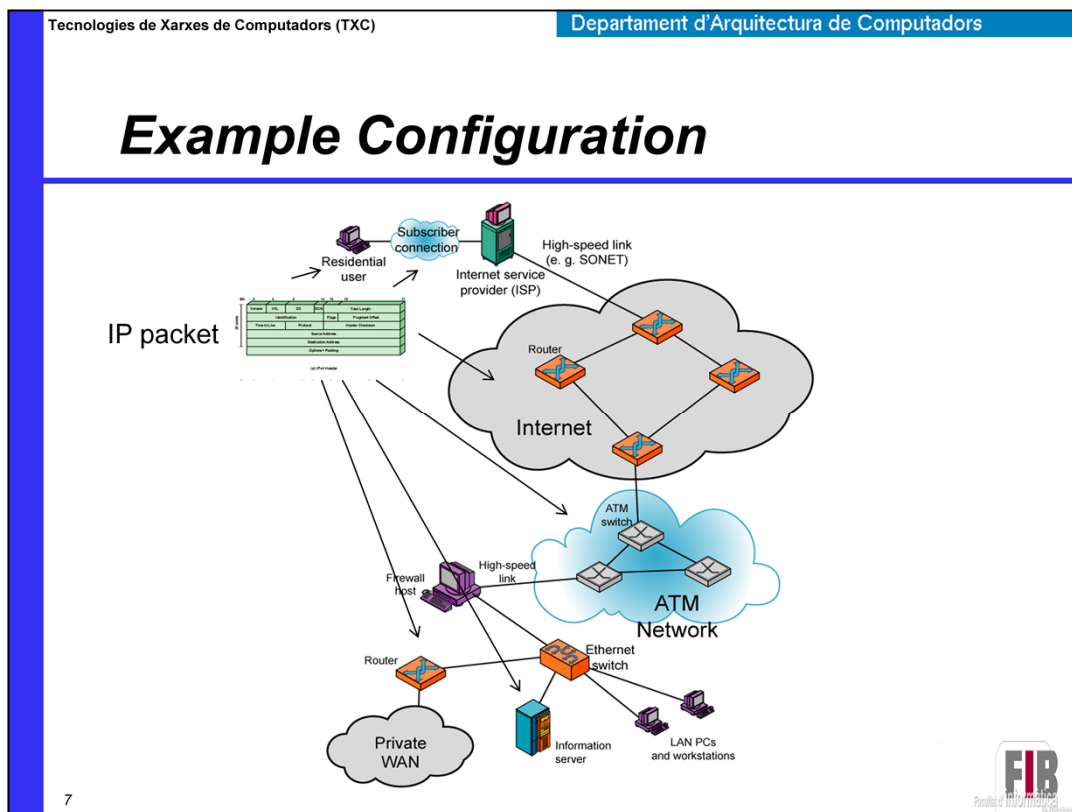
Note: The 8-bit DS/ECN fields were formerly known as the Type of Service field in the IPv4 header and the Traffic Class field in the IPv6 header.

Figure 2.7 IP Headers

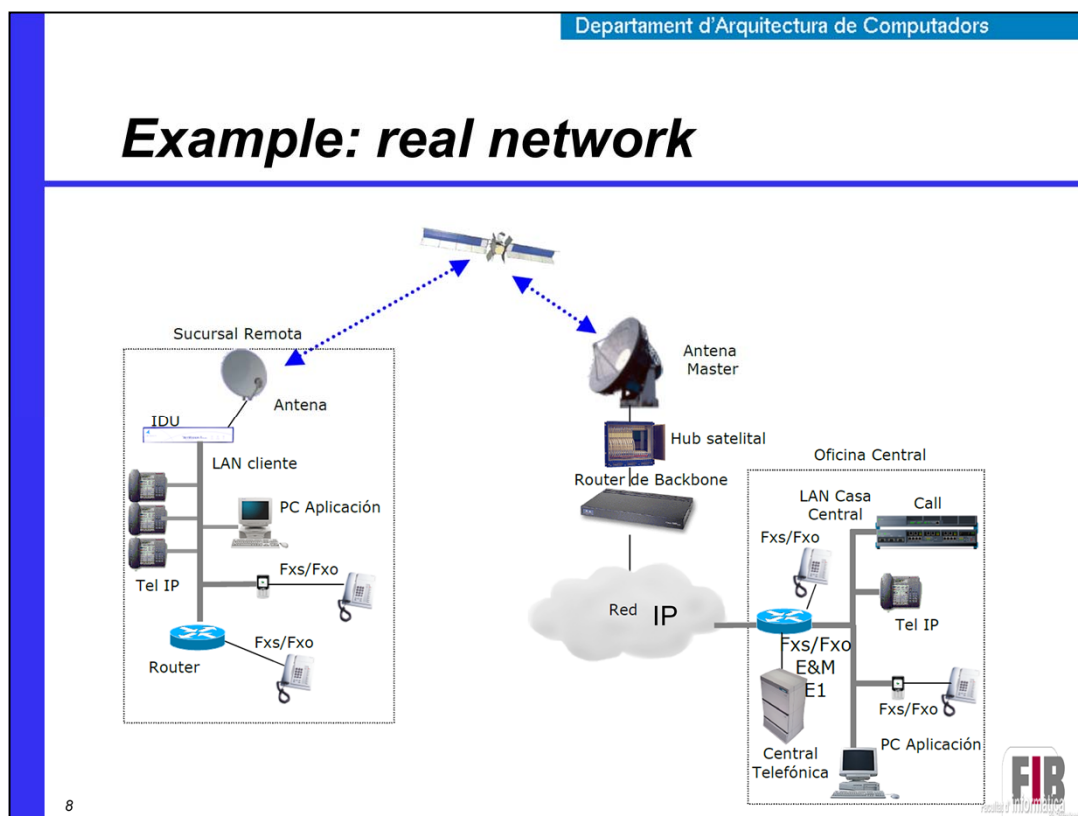


In 1995, the Internet Engineering Task Force (IETF), which develops protocol standards for the Internet, issued a specification for a next-generation IP, known then as IPng. This specification was turned into a standard in 1996 known as IPv6. IPv6 provides a number of functional enhancements over the existing IP, designed to accommodate the higher speeds of today's networks and the mix of data streams, including graphic and video, that are becoming more prevalent. But the driving force behind the development of the new protocol was the need for more addresses. IPv4 uses a 32-bit address to specify a source or destination. With the explosive growth


of the Internet and of private networks attached to the Internet, this address length became insufficient to accommodate all systems needing addresses. As Figure 2.7b shows, IPv6 includes 128-bit source and destination address fields. Ultimately, all installations using TCP/IP are expected to migrate from the current IP to IPv6, but this process will take many years, if not decades.



La figura il·lustra algunes de les comunicacions típics i elements de xarxa en ús avui en dia a Internet. A la part superior esquerra de la figura, veiem monousuari residencial connectat a un proveïdor de serveis d'Internet (ISP) a través d'algun tipus de connexió d'abonat. La Internet es compon d'un nombre d'encaminadors interconnectats que abasten tot el món. Els routers transmeten paquets de dades des de l'origen a la destinació a través d'Internet. La part inferior mostra una LAN implementat usant un únic commutador Ethernet. Això, que és comú en una petita empresa o una organització petita.



Exemple de xarxa amb interconnexió via satèl·lit entre un usuari ubicat en una sucursal i la seu central d'una empresa.



Departament d'Arquitectura de Computadors

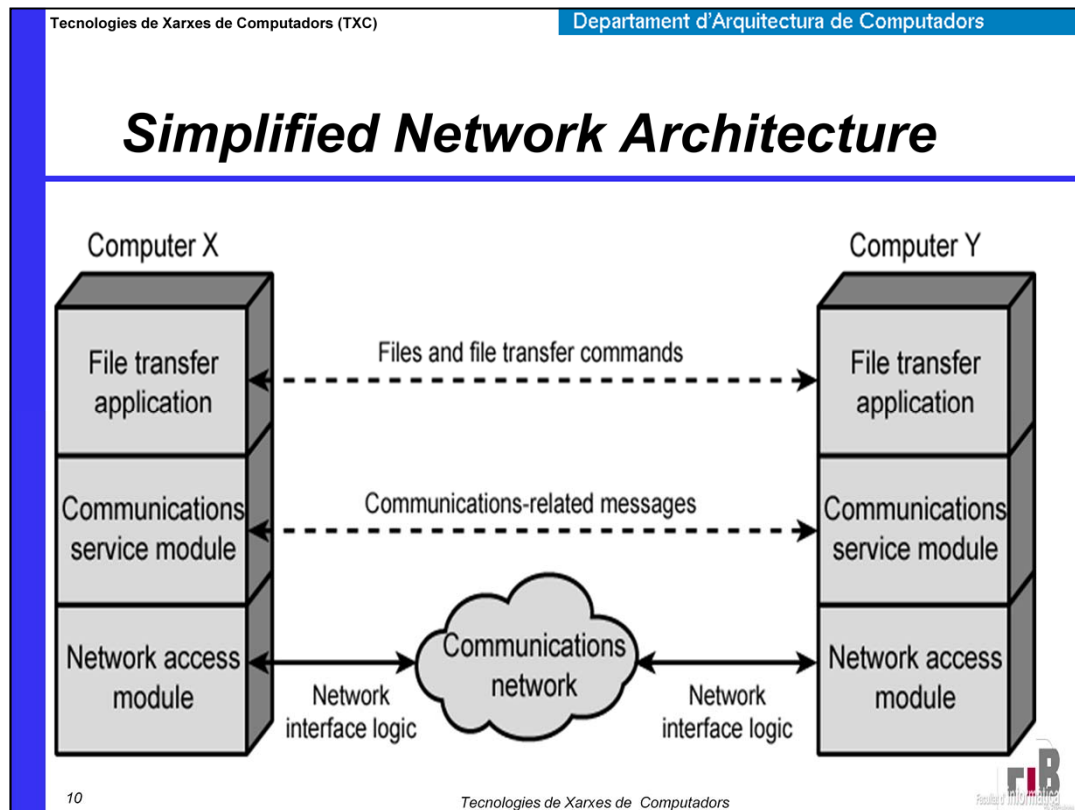
Model arquitectònic d'Internet.

Consultar Capítol 2. Stallings

9

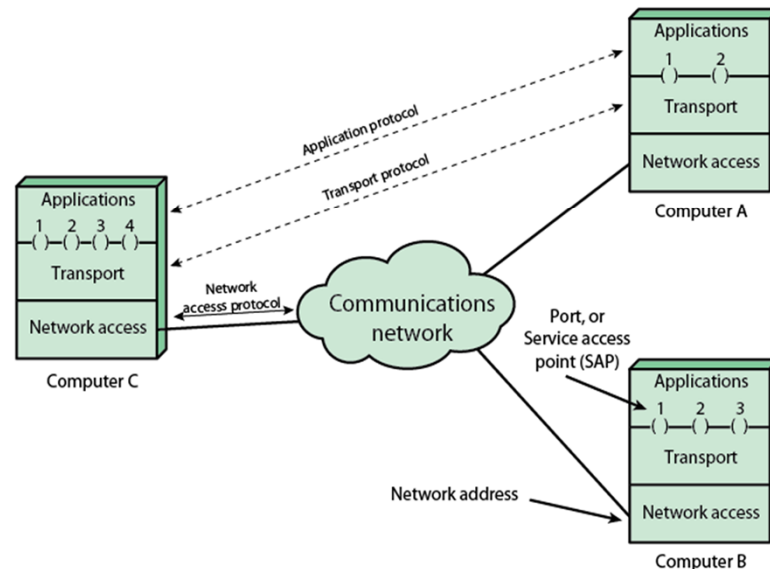
Tecnologies de Xarxes de Computadors

Model OSI d'interconnexió de sistemes oberts. Consultar el llibre Stallings en el capítol 2.



In general terms, communications can be said to involve three agents: applications (eg. file transfer), computers (eg. PCs & servers), and networks. These applications, and others, execute on computers that can often support multiple simultaneous applications. Computers are connected to networks, and the data to be exchanged are transferred by the network from one computer to another. Thus, data transfer involves first getting the data to the computer in which the application resides and then getting the data to the intended application within the computer. Can think of partitioning these tasks into 3 layers as shown.

Protocol Architecture and Networks



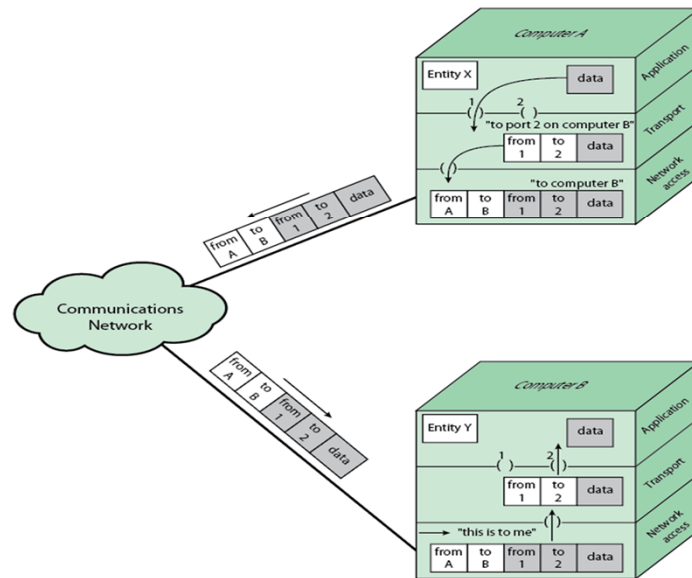
11

Tecnologies de Xarxes de Computadors



Figure 2.1 indicates that modules at the same level (peers) on different computers communicate with each other by means of a protocol. An application entity (e.g., a file transfer application) in one computer communicates with an application in another computer via an application-level protocol (e.g., the File Transfer Protocol). The interchange is not direct (indicated by the dashed line) but is mediated by a transport protocol that handles many of the details of transferring data between two computers. The transport protocol is also not direct, but relies on a network-level protocol to achieve network access and to route data through the network to the destination system. At each level, the cooperating peer entities focus on what they need to communicate to each other.

Protocols in a Simplified Architecture



12

Tecnologies de Xarxes de Computadors



Let us trace a simple operation. Suppose that an application, associated with port 1 at computer A, wishes to send a message to another application, associated with port 2 at computer B. The application at A hands the message over to its transport layer with instructions to send it to port 2 on computer B. The transport layer hands the message over to the network access layer, which instructs the network to send the message to computer B. Note that the network need not be told the identity of the destination port. All that it needs to know is that the data are intended for computer B. To control this operation, control information, as well as user data, must be transmitted, as suggested in Figure 2.2. Let us say that the sending application generates a block of data and passes this to the transport layer. The transport layer may break this block into two smaller pieces for convenience, as discussed subsequently. To each of these pieces the transport layer appends a transport **header**, containing protocol control information. The addition of control information to data is referred to as **encapsulation**.

TCP/IP

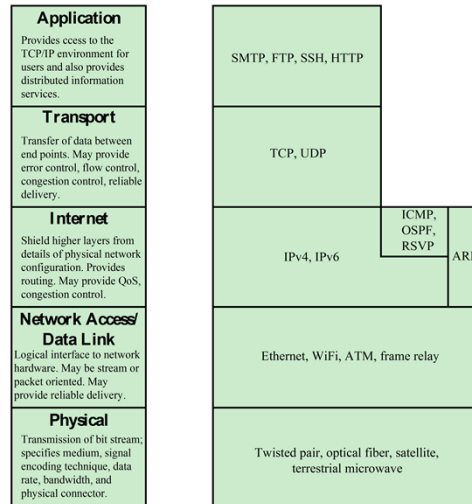
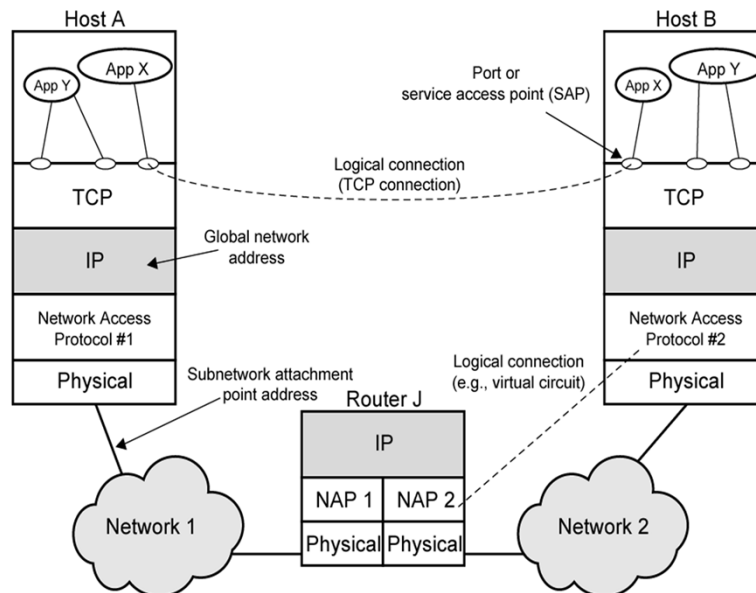


Figure 2.3 The TCP/IP Layers and Example Protocols



In general terms, computer communications can be said to involve three agents: applications, computers, and networks. Examples of applications include file transfer and electronic mail. The applications that we are concerned with here are distributed applications that involve the exchange of data between two computer systems. These applications, and others, execute on computers that can often support multiple simultaneous applications. Computers are connected to networks, and the data to be exchanged are transferred by the network from one computer to another. Thus, the transfer of data from one application to another involves first getting the data to the computer in which the application resides and then getting the data to the intended application within the computer. With these concepts in mind, we can organize the communication task into five relatively independent layers: • Physical layer • Network access/data link layer • Internet layer • Host-to-host, or transport layer • Application layer

Operation of TCP/IP



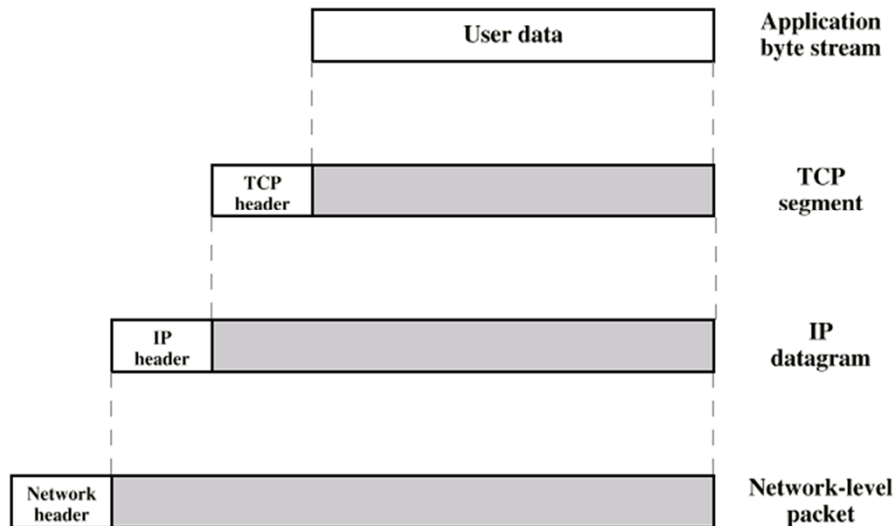
14

Tecnologies de Xarxes de Computadors



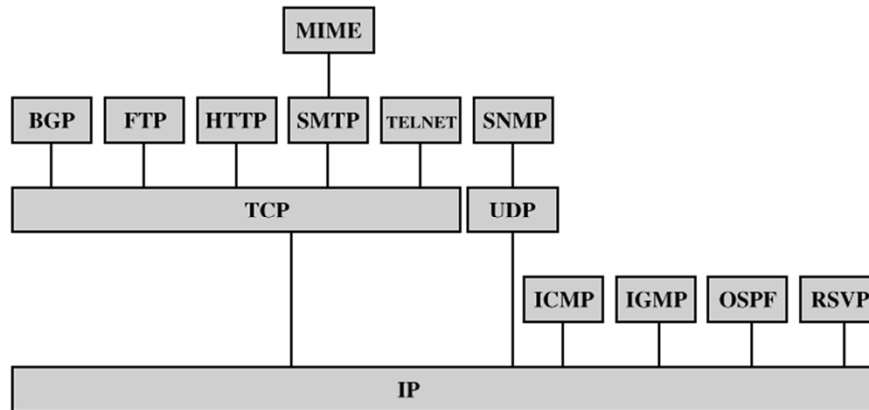
Figure 2.4 indicates how these protocols are configured for communications. To make clear that the total communications facility may consist of multiple networks, the constituent networks are usually referred to as **subnetworks**. Some sort of network access protocol, such as the Ethernet or WiFi logic, is used to connect a computer to a subnetwork. This protocol enables the host to send data across the subnetwork to another host or, if the target host is on another subnetwork, to a router that will forward the data. IP is implemented in all of the end systems and the routers. It acts as a relay to move a block of data from one host, through one or more routers, to another host. TCP is implemented only in the end systems; it keeps track of the blocks of data to assure that all are delivered reliably to the appropriate application.

Operation of TCP/IP



Let us trace a simple operation. Suppose that a process, associated with port 3 at host A, wishes to send a message to another process, associated with port 2 at host B. The process at A hands the message down to TCP with instructions to send it to host B, port 2. TCP hands the message down to IP with instructions to send it to host B. Note that IP need not be told the identity of the destination port. All it needs to know is that the data are intended for host B. Next, IP hands the message down to the network access layer (e.g., Ethernet logic) with instructions to send it to router J (the first hop on the way to B). To control this operation, control information as well as user data must be transmitted, as suggested in Figure 2.5. Let us say that the sending process generates a block of data and passes this to TCP. TCP may break this block into smaller pieces to make it more manageable. To each of these pieces, TCP appends control information known as the TCP header, forming a **TCP segment**. The control information is to be used by the peer TCP protocol entity at host B.

TCP/IP Protocols



BGP = Border Gateway Protocol	OSPF = Open Shortest Path First
FTP = File Transfer Protocol	RSVP = Resource ReSerVation Protocol
HTTP = Hypertext Transfer Protocol	SMTP = Simple Mail Transfer Protocol
ICMP = Internet Control Message Protocol	SNMP = Simple Network Management Protocol
IGMP = Internet Group Management Protocol	TCP = Transmission Control Protocol
IP = Internet Protocol	UDP = User Datagram Protocol
MIME = Multi-Purpose Internet Mail Extension	

16

Tecnologies de Xarxes de Computadors



Each layer in the TCP/IP protocol suite interacts with its immediate adjacent layers. At the source, the application layer makes use of the services of the end-to-end layer and provides data down to that layer. A similar relationship exists at the interface between the transport and internet layers and at the interface of the internet and network access layers. At the destination, each layer delivers data up to the next higher layer. This use of each individual layer is not required by the architecture. As Figure suggests, it is possible to develop applications that directly invoke the services of any one of the layers. Most applications require a reliable end-to-end protocol and thus make use of TCP. Some special-purpose applications do not need the services of TCP. Some of these applications, such as the Simple Network Management Protocol (SNMP), use an alternative end-to-end protocol known as the User Datagram Protocol (UDP); others may make use of IP directly. Applications that do not involve internetworking and that do not need TCP have been developed to invoke the network access layer directly.