# 1. Introduction to Network Protocols

Just as diplomats use diplomatic protocols in their meetings, computers use network protocols to communicate in computer networks. There are many network protocols in existence; TCP/IP is a family of network protocols that are used for the Internet.

A **network protocol** is a standard written down on a piece of paper (or, more precisely, with a text editor in a computer). The standards that are used for the Internet are called **Requests For Comment** (**RFC**). RFCs are numbered from 1 onwards. There are more than 4,500 RFCs today. Many of them have become out of date, so only a handful of the first thousand RFCs are still used today.

The **International Standardization Office** (**ISO**) has standardized a system of network protocols called as **ISO OSI**. Another organization that issues communication standards is the **International Telecommunication Union** (**ITU**) located in Geneva. The ITU was formerly known as the CCITT and, being founded in 1865, is one of the oldest worldwide organizations (for comparison, the Red Cross was founded in 1863). Some standards are also issued by the **Institute of Electrical and Electronics Engineers** (**IEEE**). RFC, standards released by **RIPE** (**Réseaux IP Européens**), and **PKCS** (**Public Key Cryptography Standard**) are freely available on the Internet and are easy to get hold of. Other organizations (ISO, ITU, and so on) do not provide their standards free of charge—you have to pay for them. If that presents a problem, then you have to spend some time doing some library research.

First of all, let's have a look at why network communication is divided into several protocols. The answer is simple although this is a very complex problem that reaches across many different professions. Most books concerning network protocols explain the problem using a metaphor of two foreigners (or philosophers, doctors, and so on) trying to communicate with each other. Each of the two can only communicate in his or her respective language. In order for them to be able to communicate with each other, they need a translator as shown in the following figure:

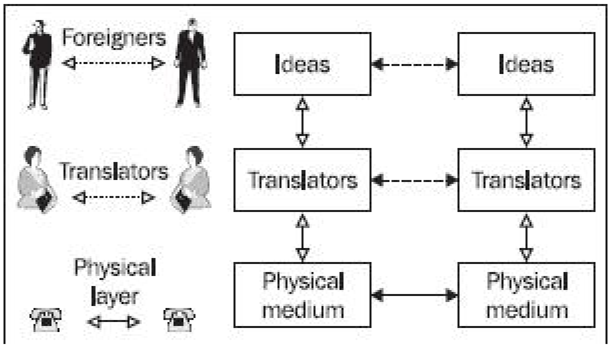


**Figure 1.1:** Three-layer communication architecture

The two foreigners exchange ideas, i.e., they communicate. But they only do so virtually. In reality, they are both handing over information to their interpreters, who then transmit this information by sending vibrations through the surrounding air with their vocal cords. Or if the parties are far away from each other, the interpreters communicate over the phone; thus the information is physically transmitted over phone lines. We can therefore talk about virtual communication in the horizontal direction (philosophical communication, the shared language between interpreters, and electronic signals transmitted via phone lines) and real communication in the vertical direction (foreigner-to-interpreter and interpreter-to-phone). We can thus distinguish three levels of communication:

1. Between two foreigners
2. Between interpreters
3. Physical transmission of information using media (phone lines, sound waves, etc.)

Communication between the two foreigners and between the two interpreters is only virtual. In fact, the only real communication happens between the foreigner and his or her interpreter.

Even more layers are used in computer networks. The number of layers depends on which system of network protocols you choose to use. The system of network protocols is sometimes referred to as the *network model*. You most commonly work with a system that uses the Internet, which is also referred to as the TCP/IP family. In addition to TCP/IP, we will also come across the ISO OSI model that was standardized by the ISO.

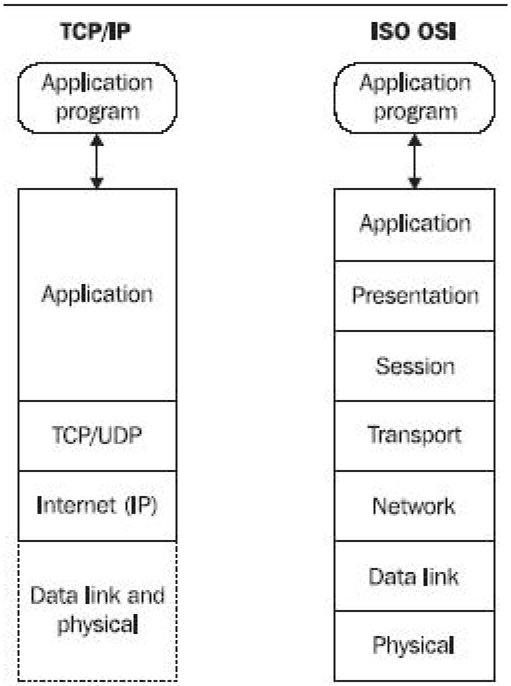


**Figure 1.2:** Comparison of TCP/IP and ISO OSI network models

The TCP/IP family uses four layers while ISO OSI uses seven layers as shown in the figure above. The TCP/IP and ISO OSI systems differ from each other significantly, although they are very similar on the network and transport layers.

Except for some exceptions like SLIP or PPP, the TCP/IP family does not deal with the link and physical layers. Therefore, even on the Internet, we use the link and physical protocols of the ISO OSI model.

# 1.1 ISO OSI

Communication between two computers is shown in the following figure:

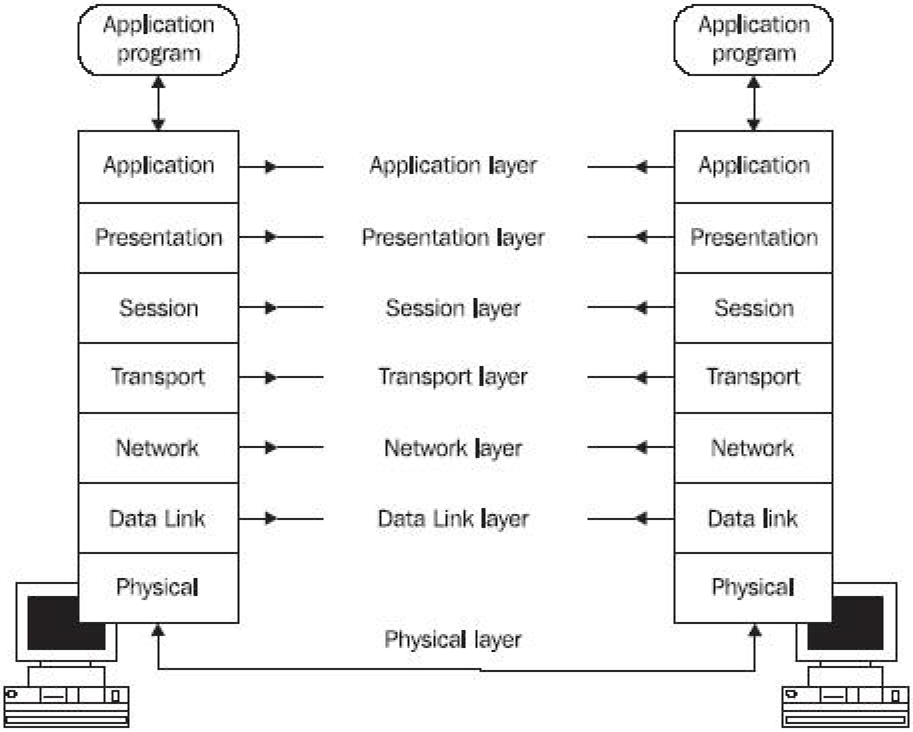


**Figure 1.3:** Seven-layer architecture of ISO OSI

## 1.1.1 Physical Layer

The physical layer is responsible for activating the physical circuit between the **Data Terminal Equipment** (**DTE**) and **Data Circuit-terminating Equipment** (**DCE**), communicating through it, and then deactivating it. Additionally, the physical layer is also responsible for the communication between DCEs (see Figure 1.3a). A computer or router can represent the DTE. The DCE, on the other hand, is usually represented by a modem or a multiplexer.

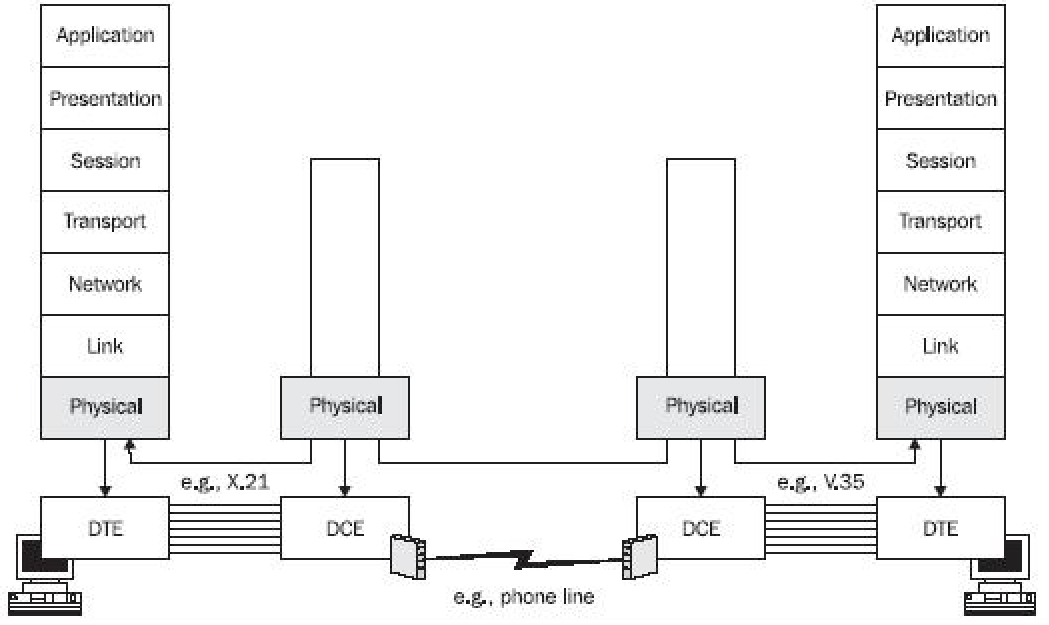


**Figure 1.3a:** DTE and DCE

To put it differently, the physical layer describes the electric or optical signals used for communicating between two computers. Physical circuits are created on the physical layer. Other appliances such as modems modulating a signal for a phone line are often put in the physical circuits created between two computers.

Physical layer protocols specify the following:

- Electrical signals (for example, +1V)
- Connector shapes (for example, V.35)
- Media type (twisted pair, coaxial cable, optical fiber, etc.)
- Modulation (for example, FM, PM, etc.)
- Coding (for example, RZ, NRZ, etc.)
- Synchronization (synchronous and asynchronous communication, time source, and so on)

## 1.1.2 Data Link Layer

As for serial links, the link layer provides data exchange between neighboring computers as well as data exchange between computers within a local network.

For the link layer, the basic unit of data transfer is the data link packet frame (see Figure 1.4). A data frame is composed of a header, payload, and trailer.

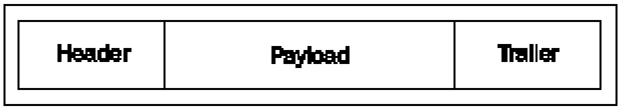


**Figure 1.4:** Data link packet or frame

A frame carries the destination link address, source link address, and other control information in the header. The trailer usually contains the checksum of the transported data. By using the checksum, we can find out whether the payload has been damaged during transfer. The network-layer packet is usually included in the payload.

In Figure 1.3a, the link layer does not engage in a conversation between DTE and DCE (the link layer *does not see* the DCE). It is engaged, however, in the frame exchange between DTEs. (It relies on the physical layer to handle the DCE issue.)

The following figure illustrates that different protocols can be used for each end of the connection on the physical layer. In our case, one of the ends uses the X.21 protocol while the other end uses the V.35 protocol. This rule is valid not only for serial links, but also for local networks. In local networks, you are more likely to encounter more complicated setups in which a switch that converts the link frames of one link protocol into link frames of a second one (for example, Ethernet into FDDI) is inserted between the two ends of the connection. This obviously results in different protocols being used on the physical layer.

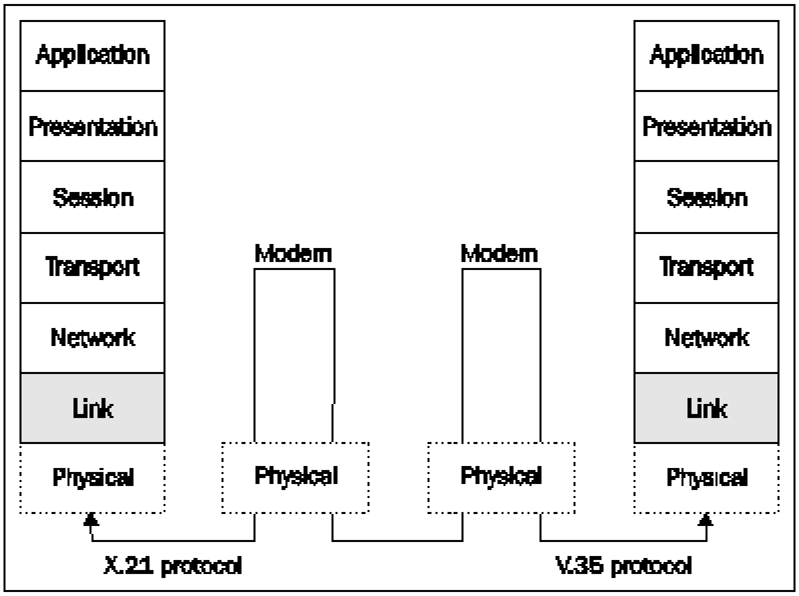


**Figure 1.5:** Link layer communication

A serial port or an Ethernet card can serve as a link interface. A link interface has a link address that is unique within a particular **Local Area Network** (**LAN**).

## 1.1.3 Network Layer

The network layer ensures the data transfer between two remote computers within a particular **Wide Area Network** (**WAN**). The basic unit of transfer is a datagram that is wrapped (encapsulated) in a frame. The datagram is also composed of a header and data field. Trailers are not very common in network protocols.

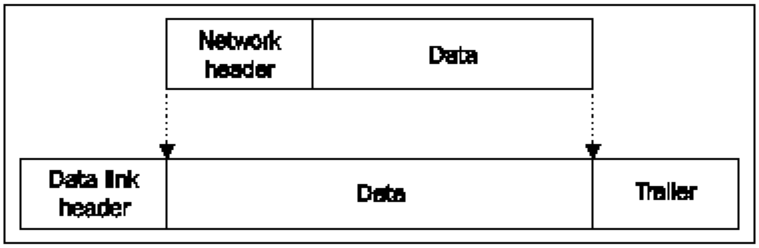


**Figure 1.6:** Network packet and its insertion in the link frame

As shown in the figure above, the datagram header, together with data (network-layer payload), creates the payload or data field of the frame.

There is usually at least one router on WANs between two computers. The connection between two neighboring routers on the link layer is always direct. The router unpacks the datagram from a frame, only to wrap it again into a different frame (or, more generally, in a frame of different link protocol) before sending it to a different line. The network layer does not see the appliances on the physical and link layers (modems, repeaters, switches, etc.).

The network layer does not care about what kind of link protocols are used on route between the source and the destination.

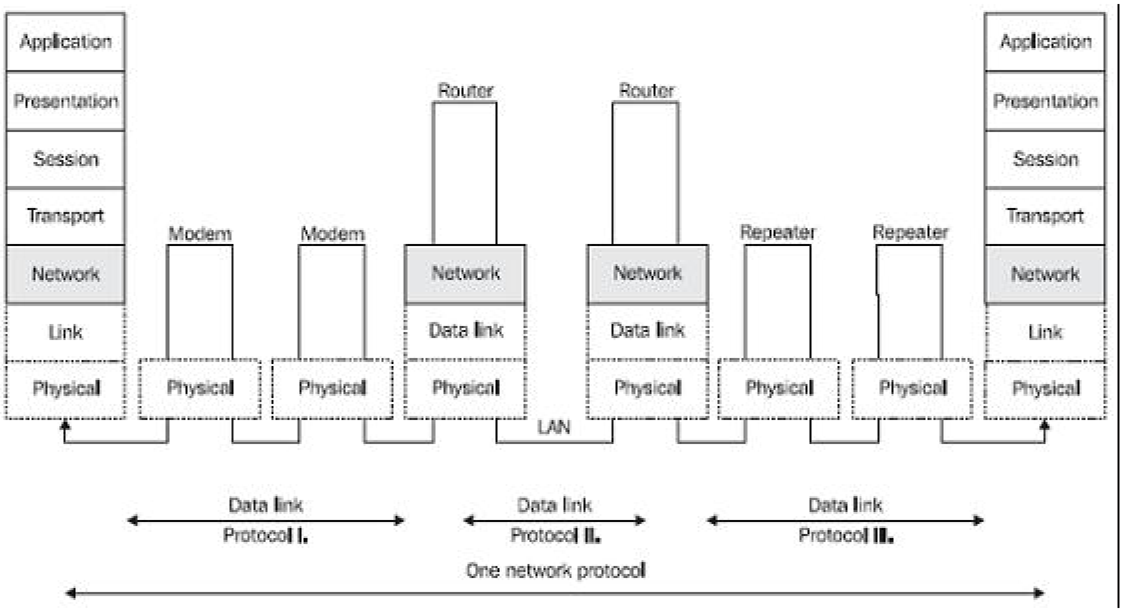


**Figure 1.7:** Network layer communication

A serial port or an Ethernet card can be used as a network interface. A network interface has a one or more unique address within a particular WAN.

## 1.1.4 Transport Layer

A network layer facilitates the connection between two remote computers. As far as the transport layer is concerned, it acts as if there were no modems, repeaters, bridges, or routers along the way. The transport layer relies completely on the services of lower layers. It also expects that the connection between two computers has been established, and it can therefore fully dedicate its efforts to the cooperation between two distant computers. Generally, the transport layer is responsible for communication between two applications running on different computers.

There can be several transport connections between two computers at any given time (for example, one for a virtual terminal and another for email). On the network layer, the transport packets are directed based on the address of the computer (or its network interface). On the transport layer, individual applications are addressed. Applications use unique addresses within one computer, so the transport address is usually composed of both the network and transport addresses.

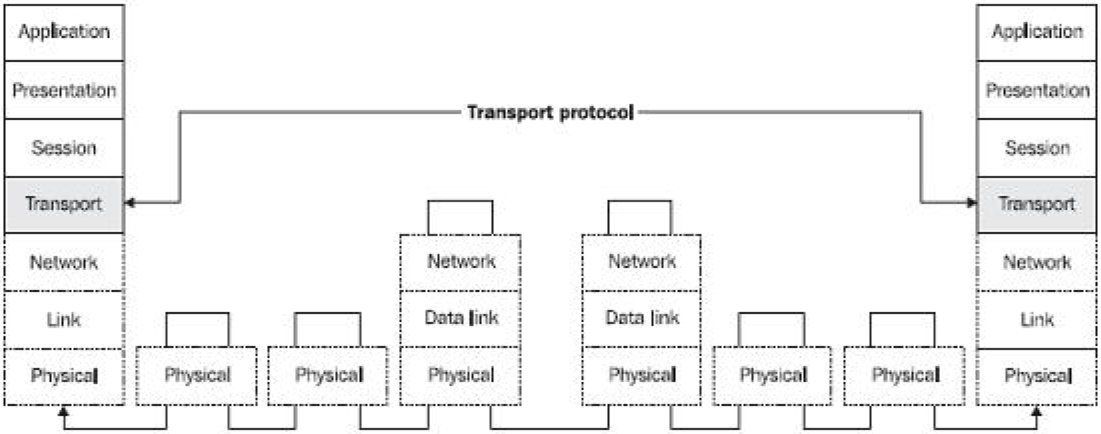


**Figure 1.8:** Transport layer connection

In this case, the basic transmission unit is the segment that is composed of a header and payload. The transport packet is transmitted within the payload of the network packet.

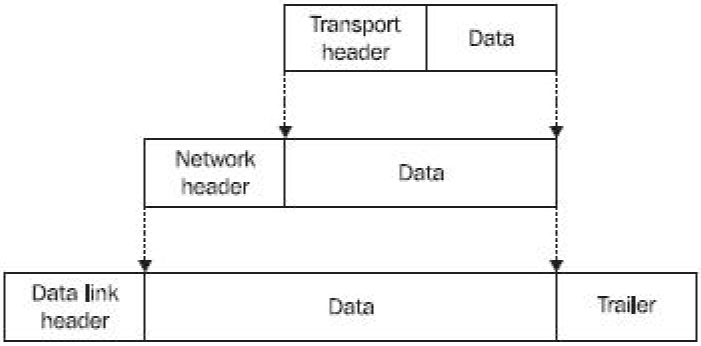


**Figure 1.9:** Inserting transport packets into network packets that are then inserted into link frames

## 1.1.5 Session Layer

The session layer facilitates exchange of data between two applications. In other words, it serves as a checkpoint and is involved in synchronizing transactions, correctly closing files, and so on. Sharing a network disk is a good example of a session. The disk can be shared for a certain period of time, but the disk is not used for the entire time. When we need to work with a file on the network disk, a connection is established on the transport layer from the time when the file is opened to when it is closed. The session, however, exists on the session layer for the entire time the disk is being shared.

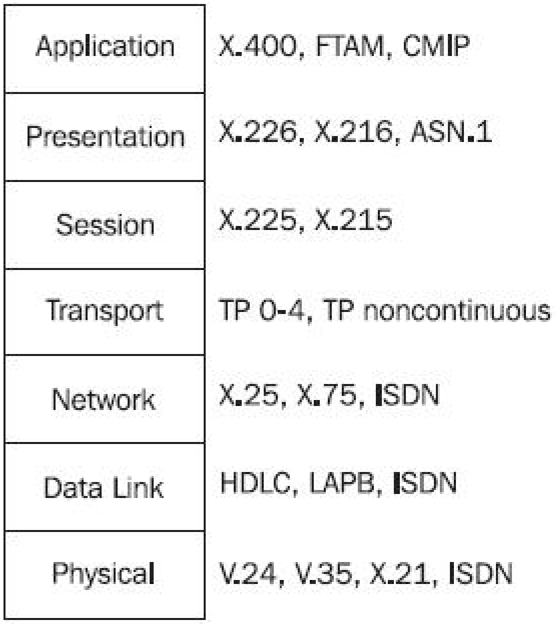
The basic unit is a session layer PDU (Protocol Data Unit), which is inserted in a segment. Other books often illustrate this with a figure of a session-layer PDU, composed of the session header and payload, being inserted in the segment. Starting with the session layer, however, this does not necessarily have to be the case. The session layer information can be transmitted inside the payload. This situation is even more noticeable if, for example, the presentation layer encrypts the data, and thus changes the whole content of the session-layer PDU.

## 1.1.6 Presentation Layer

The presentation layer is responsible for representing and securing data. The representation can differ on different computers. For example, it deals with the problem of whether the highest bit is in the byte on the right or on the left. By securing, we mean encrypting, ensuring data integrity, digital signing, and so forth.

## 1.1.7 Application Layer

The application layer defines the format in which the data should be received from or handed over to the applications. For example, the OSI Virtual Terminal protocol describes how data should be formatted as well as the dialogue used between the two ends of the connection.



| | |
|---|---|
| Application | X.400, FTAM, CMIP |
| Presentation | X.226, X.216, ASN.1 |
| Session | X.225, X.215 |
| Transport | TP 0-4, TP noncontinuous |
| Network | X.25, X.75, ISDN |
| Data Link | HDLC, LAPB, ISDN |
| Physical | V.24, V.35, X.21, ISDN |

**Figure 1.10:** Examples of network protocols from the ISO OSI protocols family

## 1.2 TCP/IP

With a few exceptions, the TCP/IP family does not deal with the physical or link layers. In practice, Internet protocols often use protocols that adhere to the ISO OSI standards for the physical and link layers.

What is the correlation between the ISO OSI protocols and TCP/IP? Each group of protocols has its definition of its own layers as well as the protocols used on these layers. Generally speaking, ISO OSI protocols and TCP/IP are incompatible. In practice, ISO OSI-compliant communication appliances need to be used for transferring IP datagrams, or on the other hand, services based on ISO OSI need to be provided via the Internet.

## 1.2.1 Internet Protocol

**Internet Protocol** (**IP**) basically corresponds to the network layer. IP is used for transmitting IP datagrams between remote computers. Each IP datagram header contains the destination address, which is the complete routing information used for delivering the IP datagram to its destination. Therefore, the network can only transmit each datagram individually. IP datagrams of one session can be transmitted through different paths and can thus be received by the destination in a different order than they were sent.

Each network interface on the large Internet network has one or more IP address that is unique worldwide. (One network interface can have several IP addresses, but one IP address cannot be used by many network interfaces.) The Internet is composed of individual networks that are interconnected via routers. Routers are also referred to as gateways in old literature.

## 1.2.2 TCP and UDP

TCP and UDP correspond to the transportation layer. TCP transports data using TCP segments that are addressed to individual applications. UDP transports data using UDP datagrams.

TCP and UDP arrange a connection between applications that run on remote computers. TCP and UDP can also facilitate communication between processes running on the same computer, but this is not very interesting for our purposes.

The difference between TCP and UDP is that TCP is a connection-oriented service—the destination confirms the data received. If some data (TCP segments) gets lost, the destination requests a retransmission of the lost data. UDP transports data using datagrams (the delivery is not guaranteed). In other words, the source party sends the datagram without worrying about whether it has been received. UDP is connectionless-oriented service.

The port is used as the address. To understand the difference between an IP address and port number, think of it as a mailing address. The IP address corresponds to the address of a house, while the port tells you the name of the person that should receive the letter.

TCP is described in Chapter 9 and UDP in Chapter 10.

## 1.2.3 Application Protocols

Application protocols correspond to several ISO OSI layers. The session, presentation, and application ISO OSI layers are reduced to one TCP/IP application layer.

The absence of a presentation layer is made up for by introducing specialized presentation-application protocols such as SSL and S/MINE that specialize in securing data or the Virtual Terminal and ASN.1 protocols that are designed for presenting data. The Virtual Terminal protocol (not to be confused with the ISO OSI protocol of the same name) specifies the network data presentation for character-oriented network protocols (Telnet, FTP, SMTP, and, partly, HTTP). Similarly, ASN.1 is often used for binary-oriented network transport. ASN.1 (including BER or DER encoding) was initially used by SNMP, but today it is also used by S/MINE.

There are many different application protocols. For practical purposes, they can be divided into two groups:

- User protocols utilized by user applications (HTTP, SMTP, Telnet, FTP, IMAP, PIP3, and so on).
- Service protocols, i.e., the protocols that ordinary Internet users rarely encounter. These protocols make sure the Internet functions correctly. For example, these could be routing protocols that are used for mutual communication by routers to correctly set their routing tables. Another example is SNMP usage in network administration.

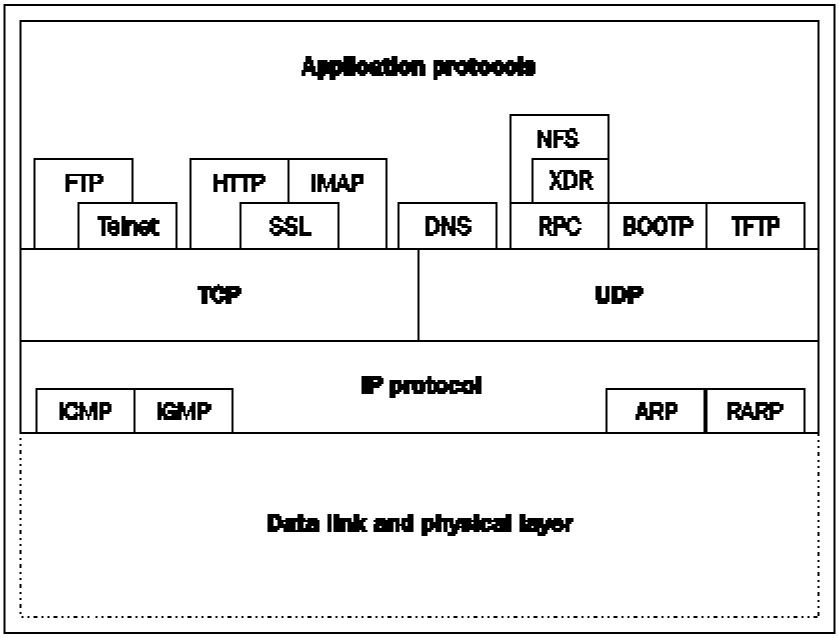


**Figure 1.11:** Some protocols of the TCP/IP family

# 1.3 Methods of Information Transmission

There are many different network protocols and several protocols can be available even on a single layer. Especially with lower-layer protocols, we distinguish between the types of transmission that they facilitate, whether they provide connection-oriented or connection-less services, if the protocol uses virtual circuits, and so on. We also distinguish between synchronous, packet, and asynchronous transmission.

## 1.3.1 Synchronous Transmission

Synchronous transmission  is needed when it is necessary to provide a stable (guaranteed) bandwidth, for example, in audio and video. If the source does not use the provided bandwidth it remains unused. Synchronous transmission uses frames that are of fixed length and are transmitted at constant speeds.

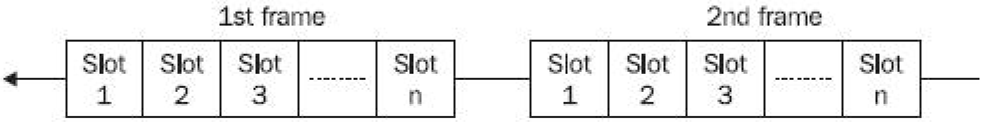


**Figure 1.12:** Frames divided into slots in synchronous transmission

In synchronous transmission, the guaranteed bandwidth is established by dividing the transmitted frames into slots (see Figure 1.12). One or more slots in any transmitted frame are reserved for a particular connection. Let's say that each frame has slot 1 reserved for our connection. Since the frames follow each other steadily in a network, our application has a guaranteed bandwidth consisting of the number of slot 1s that can be transmitted through the network in one second.

The concept becomes even clearer if we draw several frames under each other, creating a 'super-frame' (see Figure 1.13). The slots located directly under each other belong to the same connection.

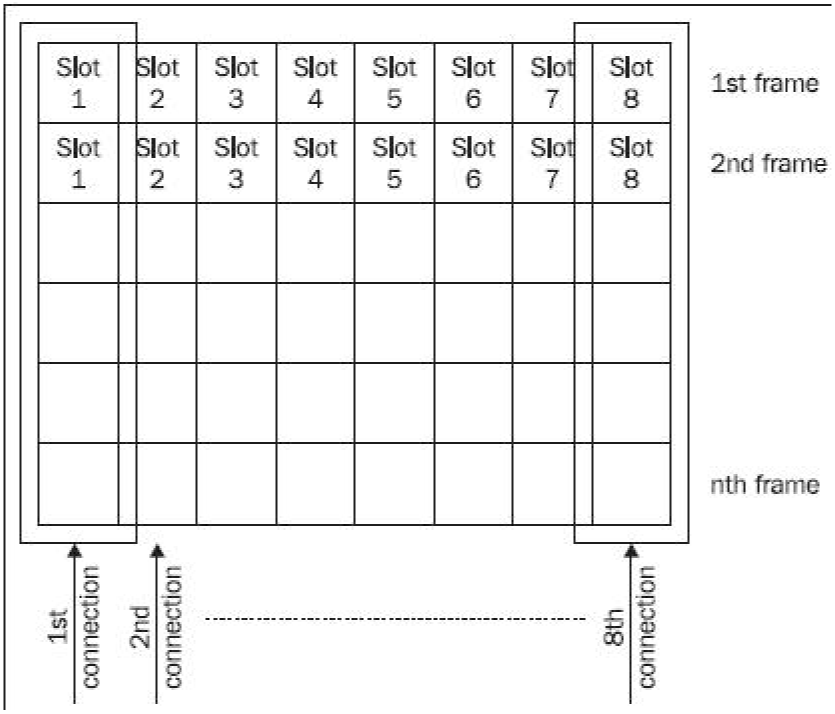


**Figure 1.13:** Super-frame

Synchronous transmission is used to connect your company switchboard to the phone company exchange. In this case, we use an E1(or T1 in United States) link containing 32 slots of 64 Kbps each. A slot can be used for making a phone call. Therefore, in theory, 32 calls are guaranteed at the same time (although some slots are probably used for servicing).

The Internet does not use synchronous transmission, i.e., in general, does not guarantee bandwidth. Quality audio or video transmission on the Internet is usually achieved by over-dimensioning the transmission lines. Recently, there has been a steady increase in requests for audio and video transmission via the Internet, so more and more often we come across systems that guarantee bandwidth even on the Internet with the help of Quality of Service (QoS). In order for us to reach the expected results, however, all appliances on route from the source to the destination must support these services. Today, we are more likely to get involved with only those areas on the Internet that guarantee bandwidth such as within a particular Internet provider.

## 1.3.2 Packet Transmission

(From now onwards we will use the term **packet** to refer to 'packet', 'datagram', 'segment', 'protocol data unit'.) Packet transmission is especially valuable for transferring data. Packets usually carry data of variable size.

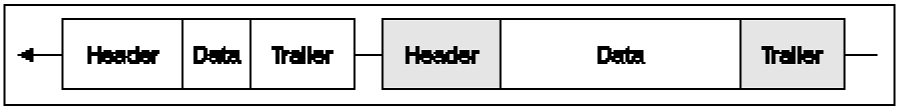


**Figure 1.14:** Packet data transmission

One packet always carries data of one particular application (of one connection). It is not possible to guarantee bandwidth, because the packets are of various lengths. On the other hand, we can use the bandwidth more effectively because if one application does not transmit data, then other applications can use the bandwidth instead.

## 1.3.3 Asynchronous Transmission

Asynchronous transmission is used in the ATM protocol. This transmission type combines features of packet transmission with features of synchronous transmission.

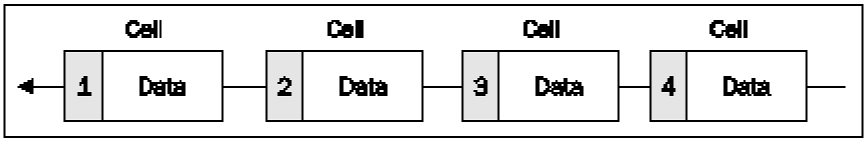


**Figure 1.15:** Asynchronous data transfer

Similarly to synchronous transmission, in asynchronous transmission, the data are transmitted in packets that are rather small, but are all of the same size; these packets are called **cells**. Similarly to packet transmission, data for one application (one connection) is transmitted in one cell. All cells have the same length; so if we guarantee that the nth cell will be available for a certain application (a particular connection), the bandwidth will be guaranteed by this as well. Additionally, it doesn't really matter if the application does not send the cell since a different application's cell might be sent instead.

# 1.4 Virtual Circuit

Some network protocols create virtual circuits in networks. A virtual circuit is conducted through the network and all packets of a particular connection go via the circuit. If the circuit gets interrupted anywhere, then the connection is interrupted, a new circuit is established, and data transmission continues.

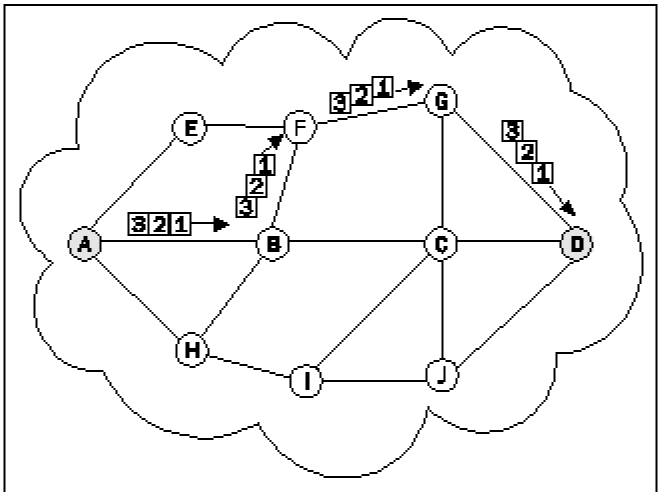


**Figure 1.16:** Virtual circuit

In the figure above, a virtual circuit between nodes A and D is established via nodes B, F, and G. All packets must go through this circuit.

Datagrams can be transmitted via the virtual circuit in two ways:

- The circuit does not guarantee the datagram's delivery to its destination. (If network congestion occurs, the circuit can even throw the datagram away.) An example is the Frame Relay protocol.
- The virtual circuit can establish a connection and guarantee the data delivery, i.e., the data packets transmitted are numbered and the destination confirms their reception. If any data gets lost, a request to resend the data is made. For example, this mechanism is used in the X.25 protocol.

The advantage of virtual circuits is that they are first established (using signalization) and then the data is inserted only into the established circuit. Each packet does not have to carry the globally unique address of the destination (complete routing information) in its header. It only needs the circuit ID.

The virtual mechanism is not used on the Internet, which was primarily aimed for use by the U.S. Department of Defense, since the destruction of a node in the virtual circuit would result in the transmission being interrupted—a fact that the authors of TCP/IP did not like. For this reason, IP does not use virtual circuits. Each IP datagram carries a destination IP address (complete routing information) and is therefore transported independently. If a node is destroyed, only the IP datagrams currently being transmitted through that particular node are destroyed. The remaining datagrams are routed via different nodes.

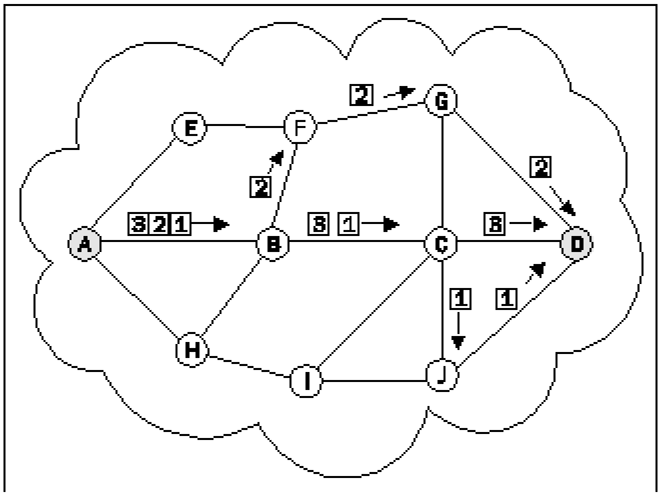


**Figure 1.17:** IP does not use virtual circuits

As the figure above shows, IP datagrams 1, 2, and 3 start from the node A to node B, but from this point, datagrams 1 and 3 are routed through a different path than datagram 2. The destination (node D) is then reached by each of them via a different path. Generally, IP datagrams may reach their destination in a different order than the order in which they were sent. So our IP datagrams could be received in the following order: 2, 1, and then 3.

In the Internet hierarchy, TCP—a higher-layer protocol that establishes a connection and guarantees the delivery of data—is used above the connectionless IP. If some of the data packets are lost, their retransmission is requested. If the data packets were lost due to the destruction of a node along the way and there is another routing possible within the network, then the transmission is automatically repeated using the other path.

Virtual circuits are divided into the following groups:

- Permanent (**Permanent Virtual Circuit** (**PVC**)), i.e., circuits permanently built by the network administrator.
- Switched (**Switched Virtual Circuit** (**SVC**)), i.e., virtual circuits that are created dynamically as the need arises. An SVC is created with the help of signalizing protocols that can be used for communicating between the user and the network itself. The network signalizes to the user various events that can be used for network monitoring and administration. SVC communication consists of two steps: creating the virtual circuit and using it for communication.

PVC corresponds to leased lines and SVC corresponds to the dial-up lines of a phone network.

**Note:**
Protocols using virtual circuits are called **Connection-Oriented Network Services** (**CONS**) and protocols transporting their packets without using virtual circuits are called **Connection-Less Network Services** (**CLNS**).

You are probably wondering whether to refer to this book to understand more about TCP/IP or to read some other good books describing similar topics and containing the word TCP/IP in their titles. Let us explain to you what moved us to write another publication about the TCP/IP protocols on which the Internet is based.

Publications about the Internet are usually of two types:

- Publications involved with concrete operating systems (Microsoft Windows, UNIX, CISCO, etc.). The goal of such publications is to train readers in a particular TCP/IP implementation, while describing the main TCP/IP principles is only their secondary goal.
- Publications written for the academic environment. Even if their main goal is to describe the basic TCP/IP principles, they could be too tedious for many readers.

So we faced the task of creating a basic TCP/IP guide, independent from any concrete environment (for example, Microsoft Windows, UNIX, CISCO, etc.), emphasizing presentation of the text in a clear and apt form to readers so that they understand the main coherences. To explain the basic principles and coherences in the best way, we have used a lot of illustrations. These illustrations were not created by chance. We drew and constantly refined them according to the requirements from our countless TCP/IP courses. First we chalked them on a blackboard, next we drew them on a white blackboard, and finally we drew them in Microsoft Visio. It has been twenty years since we started teaching TCP/IP.