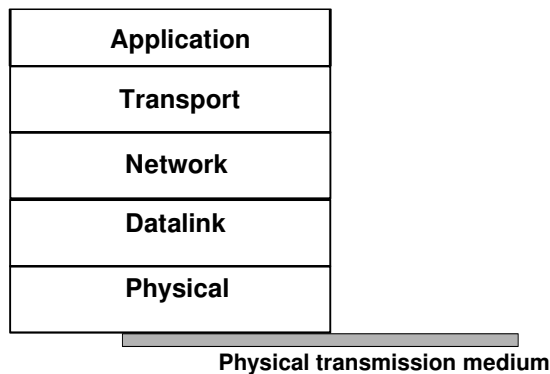


The reference models

Given the growing complexity of computer networks, during the 1970s network researchers proposed various reference models to facilitate the description of network protocols and services. Of these, the Open Systems Interconnection (OSI) model [Zimmermann80] was probably the most influential. It served as the basis for the standardisation work performed within the ISO to develop global computer network standards. The reference model that we use in this book can be considered as a simplified version of the OSI reference model [2].

The five layers reference model

Our reference model is divided into five layers, as shown in the figure below.



The five layers of the reference model

Starting from the bottom, the first layer is the Physical layer. Two communicating devices are linked through a physical medium. This physical medium is used to transfer an electrical or optical signal between two directly connected devices. Several types of physical mediums are used in practice :

- *electrical cable*. Information can be transmitted over different types of electrical cables. The most common ones are the twisted pairs that are used in the telephone network, but also in enterprise networks and coaxial cables. Coaxial cables are still used in cable TV networks, but are no longer used in enterprise networks. Some networking technologies operate over the classical electrical cable.
- *optical fiber*. Optical fibers are frequently used in public and enterprise networks when the distance between the communication devices is larger than one kilometer. There are two main types of optical fibers : multimode and monomode. Multimode is much cheaper than monomode fiber because a LED can be used to send a signal over a multimode fiber while a monomode fiber must be driven by a laser. Due to the different modes of propagation of light, monomode fibers are limited to distances of a few kilometers while multimode fibers can be used over distances greater than several tens of kilometers. In both cases, repeaters can be used to regenerate the optical signal at one endpoint of a fiber to send it over another fiber.
- *wireless*. In this case, a radio signal is used to encode the information exchanged between the communicating devices. Many types of modulation techniques are used to send information over a wireless channel and there is lot of innovation in this field with new techniques appearing every year. While most wireless networks rely on radio signals, some use a laser that sends light pulses to a remote detector. These optical techniques allow to create point-to-point links while radio-based techniques, depending on the directionality of the antennas, can be used to build networks containing devices spread over a small geographical area.

An important point to note about the Physical layer is the service that it provides. This service is usually an unreliable connection-oriented service that allows the users of the Physical layer to exchange bits. The unit of information transfer in the Physical layer is the bit. The Physical layer service is unreliable because :

- the Physical layer may change, e.g. due to electromagnetic interferences, the value of a bit being transmitted
- the Physical layer may deliver *more* bits to the receiver than the bits sent by the sender
- the Physical layer may deliver *fewer* bits to the receiver than the bits sent by the sender

The last two points may seem strange at first glance. When two devices are attached through a cable, how is it possible for bits to be created or lost on such a cable ?

This is mainly due to the fact that the communicating devices use their own clock to transmit bits at a given bit rate. Consider a sender having a clock that ticks one million times per second and sends one bit every tick. Every microsecond, the sender sends an electrical or optical signal that encodes one bit. The sender's bit rate is thus 1 Mbps. If the receiver clock ticks exactly [3] every microsecond, it will also deliver 1 Mbps to its user. However, if the receiver's clock is slightly faster (resp. slower), than it will deliver slightly more (resp. less) than one million bits every second. This explains why the physical layer may lose or create bits.

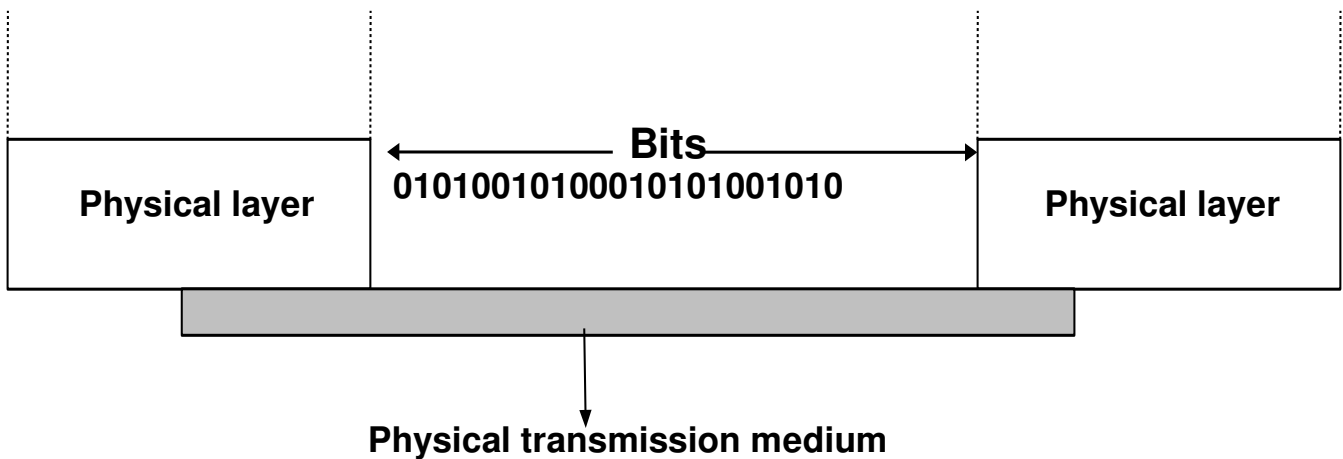


Note

Bit rate

In computer networks, the bit rate of the physical layer is always expressed in bits per second. One Mbps is one million bits per second and one Gbps is one billion bits per second. This is in contrast with memory specifications that are usually expressed in bytes (8 bits), KiloBytes (1024 bytes) or MegaBytes (1048576 bytes). Thus transferring one MByte through a 1 Mbps link lasts 8.39 seconds.

Bit rate	Bits per second
1 Kbps	10^3
1 Mbps	10^6
1 Gbps	10^9
1 Tbps	10^{12}



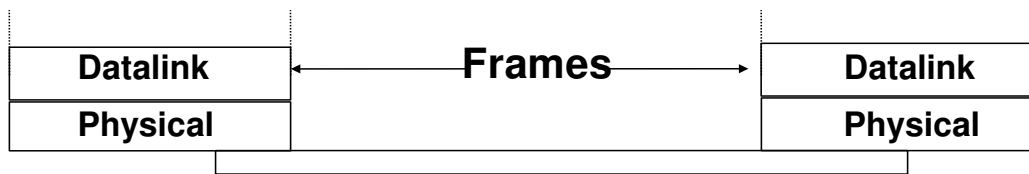
The Physical layer

The physical layer allows thus two or more entities that are directly attached to the same transmission medium to exchange bits. Being able to exchange bits is important as virtually any information can be encoded as a sequence of bits. Electrical engineers are used to processing streams of bits, but computer scientists usually prefer to deal with higher level concepts. A similar issue arises with file storage. Storage devices such as hard-disks also store streams of bits. There are hardware devices that process the bit stream produced by a hard-disk, but computer scientists have designed filesystems to allow applications to easily access such storage devices. These filesystems are typically divided into several layers as well. Hard-disks store sectors of 512 bytes or more. Unix filesystems group sectors in larger blocks that can contain data or *inodes* representing the structure of the filesystem. Finally, applications manipulate files and directories that are translated in blocks, sectors and eventually bits by the operating system.

Computer networks use a similar approach. Each layer provides a service that is built above the underlying layer and is closer to the needs of the applications.

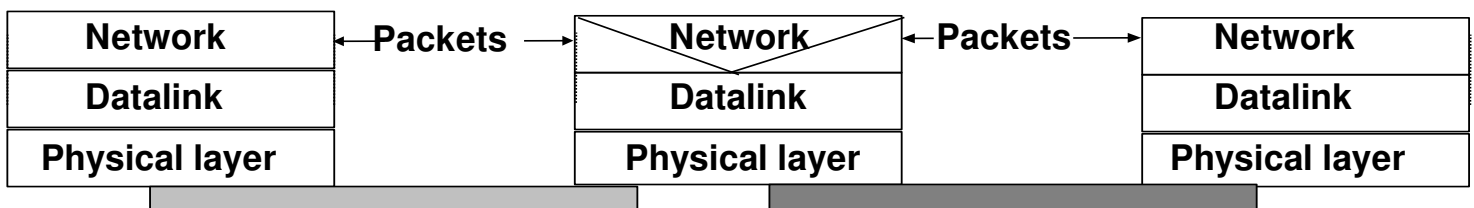
The *Datalink layer* builds on the service provided by the underlying physical layer. The *Datalink layer* allows two hosts that are directly connected through the physical layer to exchange information. The unit of information exchanged between two entities in the *Datalink layer* is a frame. A frame is a finite sequence of bits. Some *Datalink layers* use variable-length frames while others only use fixed-length frames. Some *Datalink layers* provide a connection-oriented service while others provide a connectionless service. Some *Datalink layers* provide reliable delivery while others do not guarantee the correct delivery of the information.

An important point to note about the *Datalink layer* is that although the figure below indicates that two entities of the *Datalink layer* exchange frames directly, in reality this is slightly different. When the *Datalink layer* entity on the left needs to transmit a frame, it issues as many *Data.request* primitives to the underlying *physical layer* as there are bits in the frame. The physical layer will then convert the sequence of bits in an electromagnetic or optical signal that will be sent over the physical medium. The *physical layer* on the right hand side of the figure will decode the received signal, recover the bits and issue the corresponding *Data.indication* primitives to its *Datalink layer* entity. If there are no transmission errors, this entity will receive the frame sent earlier.



The Datalink layer

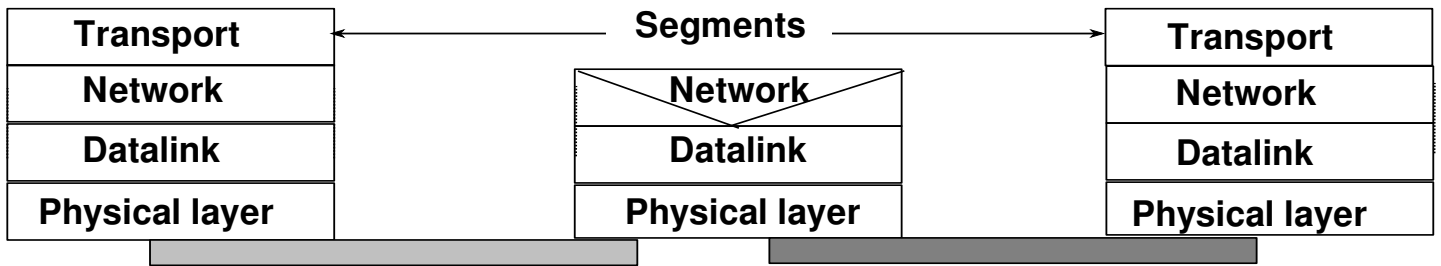
The *Datalink layer* allows directly connected hosts to exchange information, but it is often necessary to exchange information between hosts that are not attached to the same physical medium. This is the task of the *network layer*. The *network layer* is built above the *datalink layer*. Network layer entities exchange *packets*. A *packet* is a finite sequence of bytes that is transported by the datalink layer inside one or more frames. A packet usually contains information about its origin and its destination, and usually passes through several intermediate devices called routers on its way from its origin to its destination.



The network layer

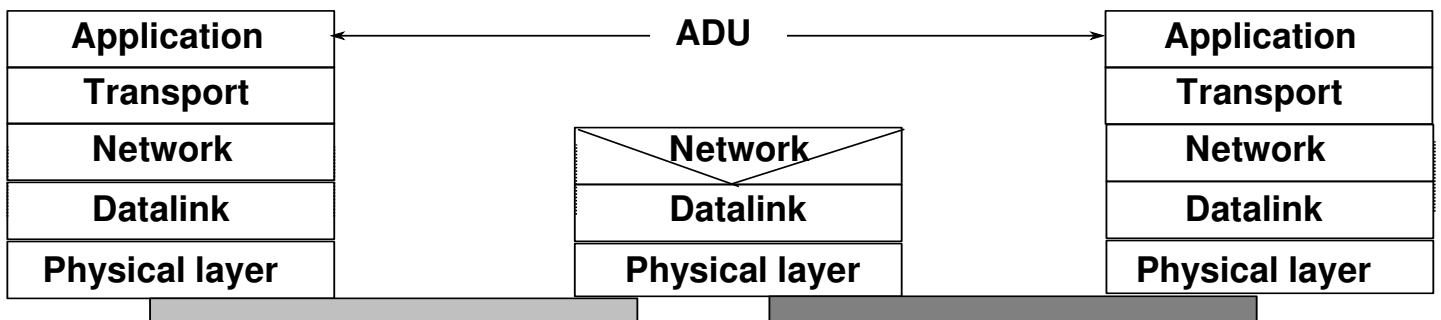
Most realisations of the network layer, including the internet, do not provide a reliable service. However, many applications need to exchange information reliably and so using the network layer service directly would be very difficult for them. Ensuring the reliable delivery of the data produced by applications is the task of the *transport layer*. *Transport layer* entities exchange *segments*. A segment is a finite sequence of bytes that are transported inside one or more packets. A transport layer entity issues segments (or sometimes part of segments) as *Data.request* to the underlying network layer entity.

There are different types of transport layers. The most widely used transport layers on the Internet are **TCP**, that provides a reliable connection-oriented bytestream transport service, and **UDP**, that provides an unreliable connection-less transport service.



The transport layer

The upper layer of our architecture is the *Application layer*. This layer includes all the mechanisms and data structures that are necessary for the applications. We will use Application Data Unit (ADU) to indicate the data exchanged between two entities of the Application layer.



The Application layer

The TCP/IP reference model

In contrast with OSI, the TCP/IP community did not spend a lot of effort defining a detailed reference model; in fact, the goals of the Internet architecture were only documented after TCP/IP had been deployed [Clark88]. RFC 1122, which defines the requirements for Internet hosts, mentions four different layers. Starting from the top, these are :

- an Application layer
- a Transport layer
- an Internet layer which is equivalent to the network layer of our reference model
- a Link layer which combines the functionalities of the physical and datalink layers of our five-layer reference model

Besides this difference in the lower layers, the TCP/IP reference model is very close to the five layers that we use throughout this document.

The OSI reference model

Compared to the five layers reference model explained above, the **OSI** reference model defined in [X200] is divided in seven layers. The four lower layers are similar to the four lower layers described above. The OSI reference model refined the application layer by dividing it in three layers :

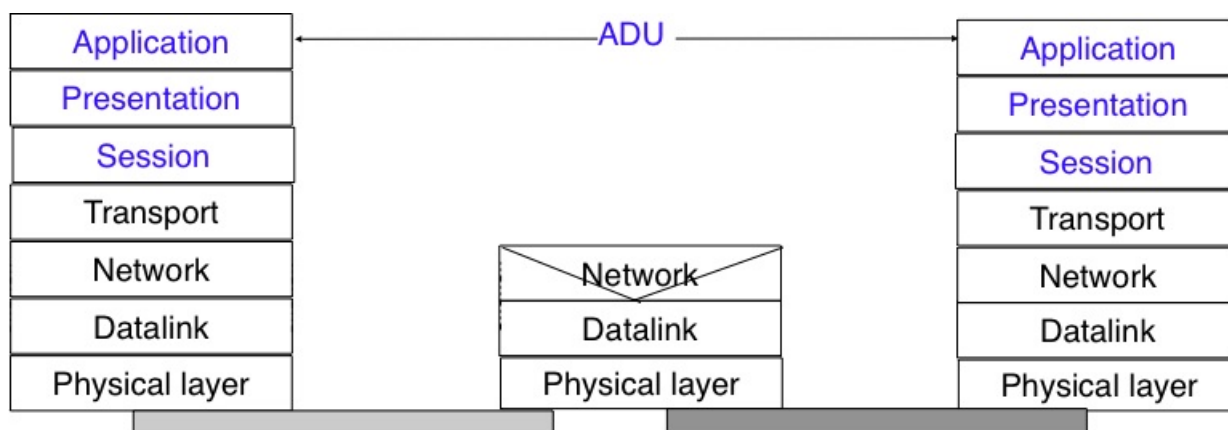
- the Session layer. The Session layer contains the protocols and mechanisms that are necessary to organize and to synchronize the dialogue and to manage the data exchange of presentation layer entities. While one of the main functions of the transport layer is to cope with the unreliability of the network layer, the session's layer objective is to hide the possible failures of transport-level connections to the upper layer higher. For this, the Session Layer provides services that allow to establish a session-connection, to support orderly data exchange (including mechanisms that allow to recover from the abrupt release of an underlying transport connection), and to release the connection in an orderly manner.
- the Presentation layer was designed to cope with the different ways of representing information on computers. There are many differences in the way computer store information. Some computers store integers as 32 bits field, others use 64 bits field and the same problem arises with floating point number. For textual information, this is even more complex with the many different character codes that have been used [1]. The situation is even more complex when considering the exchange of structured information such as database records. To solve this problem, the Presentation layer contains provides for a common representation of the data transferred. The **ASN.1** notation was designed for the Presentation layer and is still used today by some protocols.
- the Application layer that contains the mechanisms that do not fit in neither the Presentation nor the Session layer. The OSI Application layer was itself further divided in several generic service elements.



Note

Where are the missing layers in TCP/IP reference model ?

The TCP/IP reference places the Presentation and the Session layers implicitly in the Application layer. The main motivations for simplifying the upper layers in the TCP/IP reference model were pragmatic. Most Internet applications started as prototypes that evolved and were later standardised. Many of these applications assumed that they would be used to exchange information written in American English and for which the 7 bits US-ASCII character code was sufficient. This was the case for email, but as we'll see in the next chapter, email was able to evolve to support different character encodings. Some applications considered the different data representations explicitly. For example, **ftp** contained mechanisms to convert a file from one format to another and the HTML language was defined to represent web pages. On the other hand, many ISO specifications were developed by committees composed of people who did not all participate in actual implementations. ISO spent a lot of effort analysing the requirements and defining a solution that meets all of these requirements. Unfortunately, some of the specifications were so complex that it was difficult to implement them completely and the standardisation bodies defined recommended profiles that contained the implemented sets of options...



The seven layers of the OSI reference model

Footnotes

- [1] There is now a rough consensus for the greater use of the **Unicode** character format. Unicode can represent more than 100,000 different characters from the known written languages on Earth. Maybe one day, all computers will only use Unicode to represent all their stored characters and Unicode could become the standard format to exchange characters, but we are not yet at this stage today.
- [2] An interesting historical discussion of the OSI-TCP/IP debate may be found in [Russel06]
- [3] Having perfectly synchronised clocks running at a high frequency is very difficult in practice. However, some physical layers

introduce a feedback loop that allows the receiver's clock to synchronise itself automatically to the sender's clock. However, not all physical layers include this kind of synchronisation.