# REFERENCE MODELS

Now that we have discussed layered networks in the abstract, it is time to look at some examples. We will discuss two important network architectures: the **OSI reference model** and the **TCP/IP reference model**. Although the protocols associated with the OSI model are not used any more, the model itself is actually quite general and still valid, and the features discussed at each layer are still very important. The TCP/IP model has the opposite properties: the model itself is not of much use but the protocols are widely used.

**The OSI Reference Model**

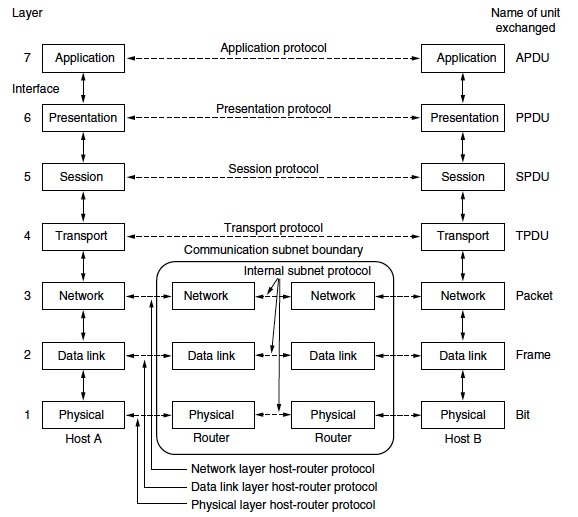
The OSI model is shown in Figure below. This model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers. It was revised in 1995. The model is called the ISO OSI (Open Systems Interconnection) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems. We will just call it the OSI model for short.

**The Physical Layer**

The physical layer is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit it is received by the other side as a 1 bit, not as a 0 bit. Typical questions here are what electrical signals should be used to represent a 1 and a 0, how many nanoseconds a bit lasts, whether transmission may proceed simultaneously in both directions, how the initial connection is established, how it is torn down when both sides are finished, how many pins the network connector has, and what each pin is used for. These design issues largely deal with mechanical, electrical, and timing interfaces, as well as the physical transmission medium, which lies below the physical layer.

**The Data Link Layer**

The main task of the data link layer is to **transform a raw transmission facility into a line that appears free of undetected transmission errors**. It does so by masking the real errors so the network layer does not see them. It accomplishes this task by having the sender break up the input data into data frames (typically a few hundred or a few thousand bytes) and transmit the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an acknowledgement frame.

Another issue that arises in the data link layer is how to keep a **fast transmitter** from drowning a slow receiver in data. Some traffic regulation mechanism may be needed to let the transmitter know when the receiver can accept more data. Broadcast networks have an additional issue in the data link layer: how to control access to the shared channel. A special sublayer of the data link layer, the **medium access control sublayer**, deals with this problem.

**The Network Layer**

The network layer controls the operation of the subnet. A key design issue is **determining how packets are routed from source to destination**. Routes can be based on static tables that are ‘‘wired into’’ the network and rarely changed, or more often they can be updated automatically to avoid failed components. They can also be determined at the start of each conversation. Finally, they can be highly dynamic, being determined anew for each packet to reflect the current network load. If too many packets are present in the subnet at the same time, they will get in one another’s way, forming bottlenecks. **Handling congestion** is also a responsibility of the network layer, in conjunction with higher layers that adapt the load they place on the network. More generally, the **quality of service provided** (delay, transit time, jitter, etc.) is also a network layer issue. In broadcast networks, the **routing problem** is simple, so the network layer is often thin or even nonexistent.

**The Transport Layer**

The basic function of the transport layer is to **accept data from above it, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end.** The most popular type of transport connection is an **error-free point-to-point** channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport service exist, such as the transporting of isolated messages with no guarantee about the order of delivery, and the broadcasting of messages to multiple destinations. The type of service is determined when the connection is established.

The transport layer is a **true end-to-end layer**; it carries data all the way from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages. In the lower layers, each protocols is between a machine and its immediate neighbors, and not between the ultimate source and destination machines, which may be separated by many routers.

**The Session Layer**

The session layer **allows users on different machines to establish sessions between them.** Sessions offer various services, including **dialog control** (keeping track of whose turn it is to transmit), **token management** (preventing two parties from attempting the same critical operation simultaneously), and **synchronization** (check pointing long transmissions to allow them to pick up from where they left off in the event of a crash and subsequent recovery).

**The Presentation Layer**

Unlike the lower layers, which are mostly concerned with moving bits around, the presentation layer is concerned with the **syntax and semantics of the information transmitted**. In order to make it possible for computers with different internal data representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used ‘‘on the wire.’’ The presentation layer manages these abstract data structures and allows higher-level data structures (e.g., banking records) to be defined and exchanged.

**The Application Layer**

The application layer contains a variety of protocols that are commonly needed by users. One widely used application protocol is HTTP (HyperText Transfer Protocol), which is the basis for the World Wide Web. When a browser wants a Web page, it sends the name of the page it wants to the server hosting the page using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail, and network news.

# TCP/IP Model

The first layered protocol model for internetwork communications was created in the early 1970s and is referred to as the Internet model, TCP/IP is stand for Transmission Control Protocol and Internet Protocol. It defines four layers that must occur for communications to be successful that illustrated in figure below:



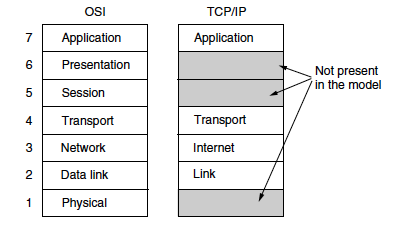
These protocols, which are implemented on both the sending and receiving hosts, interact to provide end-to-end delivery of applications over a network.

**The Link Layer**

All these requirements led to the choice of a packet-switching network based on a connectionless layer that runs across different networks. The lowest layer in the model, the link layer describes what links such as serial lines and classic Ethernet must do to meet the needs of this connectionless internet layer. It is not really a layer at all, in the normal sense of the term, but rather an interface between hosts and transmission links. Early material on the TCP/IP model has little to say about it.

**The Internet Layer**

The internet layer is the linchpin that holds the whole architecture together. It is shown in Figure below as corresponding roughly to the OSI network layer. Its job is to permit hosts to inject packets into any network and have them travel independently to the destination (potentially on a different network). They may even arrive in a completely different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired.

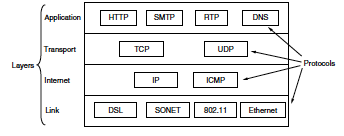


The internet layer defines an official packet format and protocol called IP

(Internet Protocol), plus a companion protocol called ICMP (Internet Control Message Protocol) that helps it function. The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly a major issue here, as is congestion (though IP has not proven effective at avoiding congestion).

**The Transport Layer**

The layer above the internet layer in the TCP/IP model is now usually called the transport layer. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as in the OSI transport layer. Two end-to-end transport protocols have been defined here. The first one, **TCP** (Transmission Control Protocol), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It segments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle. The second protocol in this layer, **UDP** (User Datagram Protocol), is an unreliable, connectionless protocol for applications that do not want TCP’s sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown in Figure below. Since the model was developed, IP has been implemented on many other networks.



**The Application Layer**

The TCP/IP model does not have session or presentation layers. No need for them was perceived. Instead, applications simply include any session and presentation functions that they require. Experience with the OSI model has proven this view correct: these layers are of little use to most applications. On top of the transport layer is the application layer. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP). Many other protocols have been added to these over the years. Some important ones that we will study, shown in Figure above.