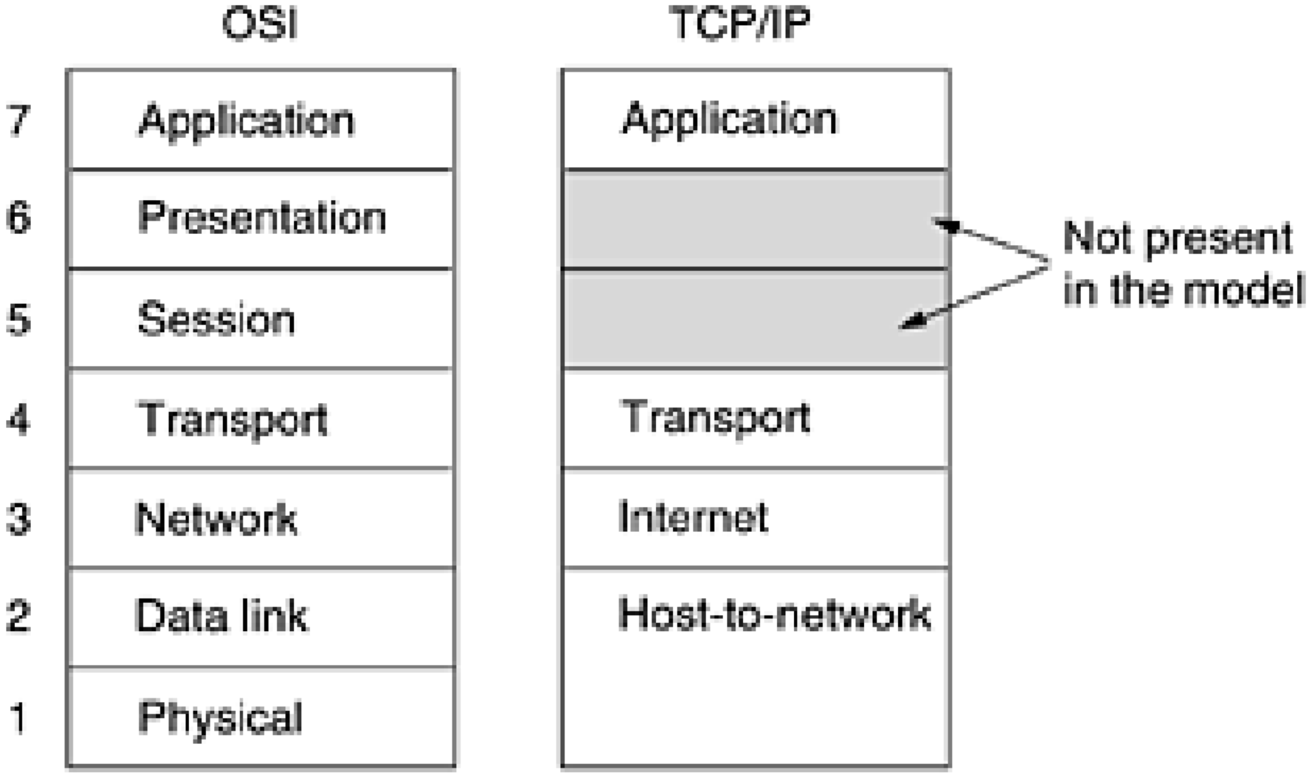
The TCP/IP Reference Model

Let us now turn from the OSI model to the reference mode; used in the grandparent of all wide area computer networks. The ARPANET, and its successor, the world wide internet.

The ARPENET was research network sponsored by the DoD (U.S Department of Defense). It eventually connected hundreds of universities and government installation, using leased telephone lines. When satellite and radio networks were added later, the exiting protocols had trouble inter working with them, so a new reference architecture was needed. Thus, the ability to connect multiple networks in a seamless way was one of the major design goals from the very beginning.

This architecture later became known as the TCP/IP Reference Model, after its two primary protocols. The TCP/IP protocol suite was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application.

However, when TCP/IP is compared to OSI, we can say that the host-to-network layer is equivalent to the combination of the physical and data link layers. The internet layer is equivalent to the network layer, and the application layer is roughly doing the job of the session, presentation, and application layers with the transport layer in TCP/IP taking care of part of the duties of the session layer.



The Host-to-Network Layer

Below the internet layer is a great void. The TCP/IP reference model does not really say much about what happens here. Except to point out that the host has to connect to the network using some protocol so it can send IP packets to it. This protocol is not defined and varies from host has to host and network to network. Books and papers about the TCP/IP model rarely discuss it.

The Internet Layer

internet layer, is the linchpin that holds the whole architecture together. 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 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. Note that "internet" is used here in a generic sense, even though this layer is in the internet.

The internet layer defines an official packet format and protocol called IP (Internet Protocol). IP, in turn, uses four supporting protocols: ARP, RARP, ICMP, and IGMP.

The Internetworking Protocol (IP)

The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly the major issue here, as is avoiding congestion. For these reasons, it is reasonable to say that the TCP/IP internet layer is similar in functionality to the OSI network layer.

IP is an unreliable and connectionless protocol, -a best-effort delivery service.

The term best effort means that IP provides no error checking or tracking. IP assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.

IP transports data in packets called datagrams, each of which is transported separately.

Datagrams can travel along different routes and can arrive out of sequence or be duplicated. IP does not keep track of the routes and has no facility for reordering datagrams once they arrive at their destination. The limited functionality of IP should not be considered a weakness.

Address Resolution Protocol

The Address Resolution Protocol (ARP) is used to associate a logical address with a physical address.

Reverse Address Resolution Protocol

The Reverse Address Resolution Protocol (RARP) allows a host to discover its Internet address when it knows only its physical address.

Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) is a mechanism used by hosts and gateways to send notification of datagram problems back to the sender.

Internet Group Message Protocol

The Internet Group Message Protocol (IGMP) is used to facilitate the simultaneous transmission of a message to a group of recipients.

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 entities in the source and destination hosts to carry on a conversation. Traditionally the transport layer was represented in TCP/IP by two protocols: TCP and UDP. IP is a host-to-host protocol, meaning that it can deliver a packet from one physical device to another. UDP and TCP are transport level protocols responsible or delivery of a message from a process (running program) to another process.

1- 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 fragments the incoming byte steam into discrete messages called segments, Each segment includes a sequence number for reordering after receipt, together with an acknowledgment number for the segments received. Segments are carried across the internet inside of IP datagrams. At the receiving end, TCP collects each datagram as it comes in and reorders the transmission based on sequence numbers.TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages it can handle.

2- UDP (User Datagram Protocol)

The User Datagram Protocol (UDP) is the simpler of the two standard TCP/IP transport protocols. 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 also widely used for one – shot, silent – server – type request – reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. It is a process-to-process protocol that adds only port addresses, checksum error control, and length information to the data from the upper layer.

The application Layer

The TCP/IP model does not have session or presentation layers. No need for them was perceived, so they were not included. Experience with the OSI model has proven this view correct: they are of little use to most applications.

On top of the transport layer is the application layer. It contains the higher level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP).

The virtual terminal protocol allows a user on one. Machine to log onto a distant machine and work there. The file transfer protocol provides a way to move data efficiently from one machine to another. Electronic mail was originally just a kind of the file transfer, but later a specialized protocol (SMTP) was developed for it. Many other protocols have been added to these over the years. The Domain Name System (DNS) for mapping host names onto their network addresses, NNTP, the protocol for moving USENET news articles around, and HTTP, the protocol for fetching pages on the World Wide Web, and many others.

A comparison of the OSI and TCP/IP Reference Models

The OSI and TCP/IP reference models have much in common. Both are based on the concept of a stack of independent protocols. Also the functionality of the layers is roughly similar. For example, in both models the layers up through and including the transport layer are there to provide an end-to-end, network – independent transport service to processes wishing to communicate. These layers from the transport provider. Again in both models, the layers above transport are application-oriented users of the transport service.

Despite these fundamental similarities, the two models also have many differences. In this section we will focus on the key differences between the two references model. It is important to note we are comparing the reference model here. Not the corresponding protocol stacks. The protocol themselves will be discussed later. For an entire book comparing and contrasting TCP/IP and OSI, see (piscitello and Chapin, 1993).

There concepts are central to the OSI model:

1. services
2. interfaces
3. protocols.

Probably the biggest contribution of the OSI model is to make the distinction between these three concepts explicit. Each layer performs some services for the layer above it. The service definition tells what the layer does, not how entities above it access it or how the layer works. It defines the layer's semantics.

A layer's interface tells the processes above it how to access it. It specifies what the parameters are and what results to expect. It, too says nothing about how the layer works inside.

Finally, the peer protocols tells the layer's own business. It can use any protocols it wants t, as long as it gets the job done (i.e., provides the offered services). It can also change them at will without affecting software in higher layers.

These ideas fit very nicely with modern ideas about object-oriented programming. An object, like a layer, has a set of methods (operations) that processes outside the object can invoke. The semantic of these methods define the set of services that the object offers. The methods parameters and results from the object's interface. The code internal to the objects is its protocol and is not visible or of any concern outside the object.

The TCP/IP model did not originally clearly distinguish between service, interface, and protocol, although people have tried to retrofit it after the fact to make it more OSI-like. For example, the only real services offers by the internet by the internet layer are SEND IP PACKET and RECEVICE IP PACKET.

As a consequence, the protocol in the OSI model are better than in the TCP/IP model and can be replaced relatively easily as the technology changes. Being able to make such is one of the main purposes of having layered protocols in the first place.

The OSI reference model was devised before the corresponding protocols were invented. This ordering means that the model was not.