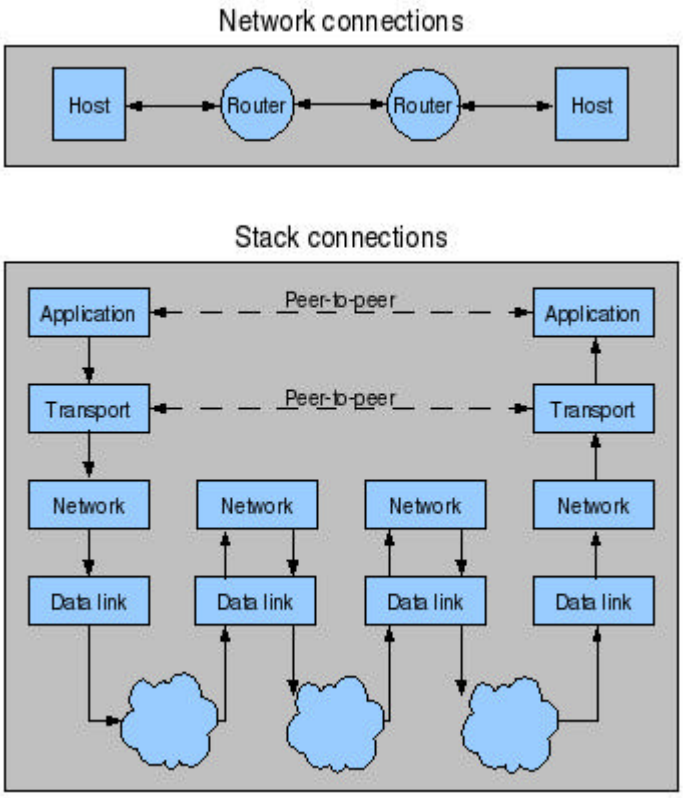
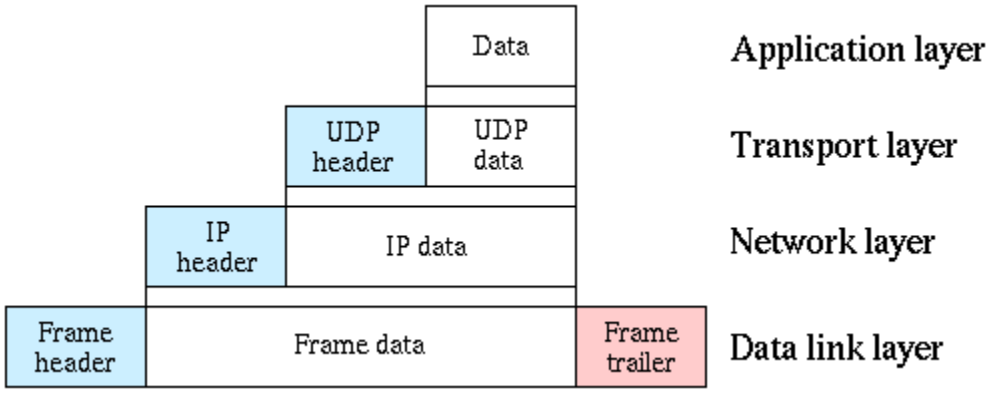
Layers in the internet protocol suite stack



IP suite stack showing the physical network connection of two hosts via two routers and the corresponding layers used at each hop



Sample encapsulation of data within a UDP datagram within an IP packet

The IP suite uses encapsulation to provide abstraction of protocols and services. Generally a protocol at a higher level uses a protocol at a lower level to help accomplish its aims. The internet protocol stack can be roughly fitted into the four fixed layers given below. The stack consists of four layers:

4	**Application**	DNS, TLS/SSL, TFTP, FTP, HTTP, IMAP, IRC, NNTP, POP3, SIP, SMTP, SNMP, SSH, TELNET, ECHO, BitTorrent, RTP, rlogin, ENRP, … Routing protocols like BGP and RIP, which for a variety of reasons run over TCP and UDP respectively, may also be considered part of the application or network layer.
3	**Transport**	TCP, UDP, DCCP, SCTP, IL, RUDP, … Routing protocols like OSPF, which run over IP, may also be considered part of the transport or network layer. ICMP and IGMP run over IP may be considered part of the network layer.
2	**Network**	IP (IPv4, IPv6) ARP and RARP operate underneath IP but above the link layer so they belong somewhere inbetween.
1	**Link**	Ethernet, Wi-Fi, Token ring, PPP, SLIP, FDDI, ATM, Frame Relay, SMDS, …

The layers near the top are logically closer to the user while those near the bottom are logically closer to the physical transmission of the data. Each layer has an upper layer protocol and a lower layer protocol (except the top/bottom protocols, of course) that either use said layer's service or provide a service, respectively. Viewing layers as providing or consuming a service is a method of abstraction to isolate upper layer protocols from the nitty gritty detail of transmitting bits over, say, ethernet and collision detection while the lower layers avoid having to know the details of each and every application and its protocol.

This abstraction also allows upper layers to provide services that the lower layers cannot, or choose not, to provide. For example, IP is designed to not be reliable and is a best effort delivery protocol. This means that all transport layer must address whether or not to provide reliability and to what degree. UDP provides data integrity (via a checksum) but does not guarantee delivery; TCP provides both data integrity and delivery guarantee (by retransmitted until the receiver receives the packet).

This model is in some ways lacking.

1. For multipoint links with their own addressing systems (e.g. ethernet) an address mapping protocol is needed. Such protocols can be considered to be below IP but above the existing link system.
2. ICMP & IGMP operate on top of IP but does not transport data like UDP or TCP.
3. The SSL/TLS library operates above the transport layer (utilizes TCP) but below application protocols.
4. The link is treated like a black box here. This is fine for discussing ip (since the whole point of IP is it will run over virtually anything) but is less helpful when considering the network as a whole.

The third and fourth examples are perhaps better explained using the OSI model while the first two are still problematic.

[edit]

OSI model comparison

The IP suite (and corresponding stack) were in use before the OSI model was established and since then the IP stack has been compared with the OSI model numerous times in books and classrooms. Also OSI layer numbers are generally used for describing the capabilities of network equipment.

The two can roughly be related but are not a perfect match. The first striking difference is the layer count. The IP stack uses five layers (the physical layer isn't shown above however) and the OSI model uses seven. Strictly comparing names, the two "new" layers are the presentation layer and the session layer. Most comparisons lump these two layers with the OSI application layer and equate to the IP application layer.

Much like the IP stack, the OSI model is also not rich enough at the lower layers to capture the true workings of the IP suite. For example, an "internetworking layer" is needed to fit inbetween the network and transport layers to address where ICMP and IGMP reside. Additionally, a layer between the network and data link layer is needed for ARP and RARP. It also suffers from being designed for simple network setups having only a single data link layer (for example an ADSL user tunneling into a corporate network could have IP over PPTP over IP over PPPoA over the ADSL link)

One example of where the OSI model is better used is showing where SSL/TLS fits in. Typically, SSL/TLS is used as a session protocol that is an upper layer protocol for TCP or UDP but is a lower layer protocol for numerous protocols (HTTP, SFTP, etc.) or any application that operates over an stunnel or secure virtual private network.

7	**Application**	HTTP, SMTP, SNMP, FTP, Telnet, ECHO, SIP, SSH, NFS, RTSP, XMPP, Whois, ENRP
6	**Presentation**	XDR, ASN.1, SMB, AFP, NCP
5	**Session**	ASAP, TLS, SSH, ISO 8327 / CCITT X.225, RPC, NetBIOS, ASP, Winsock, BSD sockets
4	**Transport**	TCP, UDP, RTP, SCTP, SPX, ATP, IL
3	**Network**	IP, ICMP, IGMP, IPX, BGP, OSPF, RIP, IGRP, EIGRP, ARP, RARP, X.25
2	**Data Link**	Ethernet, Token ring, HDLC, Frame relay, ISDN, ATM, 802.11 WiFi, FDDI, PPP
1	**Physical**	wire, radio, fiber optic, Carrier pigeon

There are several mnemonics for remembering the order of the layers in the OSI model.