

Internet protocol suite

The **Internet protocol suite** is the conceptual model and set of communications protocols used on the Internet and similar computer networks. It is commonly known as **TCP/IP** because the foundational protocols in the suite are the Transmission Control Protocol (TCP) and the Internet Protocol (IP). It is occasionally known as the **Department of Defense (DoD) model**, because the development of the networking method was funded by the United States Department of Defense through DARPA.

The Internet protocol suite provides end-to-end data communication specifying how data should be packetized, addressed, transmitted, routed, and received. This functionality is organized into four abstraction layers which classify all related protocols according to the scope of networking involved.^{[1][2]} From lowest to highest, the layers are the link layer, containing communication methods for data that remains within a single network segment (link); the internet layer, providing internetworking between independent networks; the transport layer handling host-to-host communication; and the application layer, which provides process-to-process data exchange for applications.

Technical standards specifying the Internet protocol suite and many of its constituent protocols are maintained by the Internet Engineering Task Force (IETF). The Internet protocol suite predates the OSI model, a more comprehensive reference framework for general networking systems.

Contents

History

- Early research
- Specification
- Adoption

Key architectural principles

- Link layer
- Internet layer
- Transport layer
- Application layer

Layer names and number of layers in the literature

Comparison of TCP/IP and OSI layering

Implementations

See also

Bibliography

References

External links

History

Early research

The Internet protocol suite resulted from research and development conducted by the Defense Advanced Research Projects Agency (DARPA) in the late 1960s.^[3] After initiating the pioneering ARPANET in 1969, DARPA started work on a number of other data transmission technologies. In 1972, Robert E. Kahn joined the DARPA Information Processing Technology Office, where he worked on both satellite packet networks and ground-based radio packet networks, and recognized the value of being able to communicate across both. In the spring of 1973, Vinton Cerf, the developer of the existing ARPANET Network Control Program (NCP) protocol, joined Kahn to work on open-architecture interconnection models with the goal of designing the next protocol generation for the ARPANET.

such as [FTP Software](#) and the [Wollongong Group](#), began offering TCP/IP stacks for [DOS](#) and [MS Windows](#).^[12] The first [VM/CMS](#) TCP/IP stack came from the University of Wisconsin.^[13]

Some of these TCP/IP stacks were written single-handedly by a few programmers. Jay Elinsky and [Oleg Vishnepolsky](#) of IBM Research wrote TCP/IP stacks for VM/CMS and OS/2, respectively.^[14] In 1984 Donald Gillies at MIT wrote a 'ntcp' multi-connection TCP which ran atop the IP/PacketDriver layer maintained by John Romkey at MIT in 1983-4. Romkey leveraged this TCP in 1986 when FTP Software was founded.^{[15][16]} Phil Karn created KA9Q TCP (a multi-connection TCP for ham radio applications) starting in 1985.^[17]

The spread of TCP/IP was fueled further in June 1989, when AT&T agreed to place the TCP/IP code developed for [UNIX](#) into the public domain. Various vendors, including IBM, included this code in their own TCP/IP stacks. Many companies sold TCP/IP stacks for Windows until Microsoft released a native TCP/IP stack in Windows 95. This event was a little late in the evolution of the Internet, but it cemented TCP/IP's dominance over other protocols, which began to lose ground. These protocols included [IBM Systems Network Architecture](#) (SNA), [Digital Equipment Corporation's DECnet](#), [Open Systems Interconnection](#) (OSI), and [Xerox Network Systems](#) (XNS).

Key architectural principles

An early architectural document, [RFC 1122](#), emphasizes architectural principles over layering.^[18]

The [end-to-end principle](#) has evolved over time. Its original expression put the maintenance of state and overall intelligence at the edges, and assumed the Internet that connected the edges retained no state and concentrated on speed and simplicity. Real-world needs for firewalls, network address translators, web content caches and the like have forced changes in this principle.^[19]

The [robustness principle](#) states: "In general, an implementation must be conservative in its sending behavior, and liberal in its receiving behavior. That is, it must be careful to send well-formed datagrams, but must accept any datagram that it can interpret (e.g., not object to technical errors where the meaning is still clear)."^[20] "The second part of the principle is almost as important: software on other hosts may contain deficiencies that make it unwise to exploit legal but obscure protocol features."^[21] Postel famously summarized the principle as, "Be conservative in what you do, be liberal in what you accept from others"—a saying that came to be known as "Postel's Law."

[Encapsulation](#) is used to provide abstraction of protocols and services. Encapsulation is usually aligned with the division of the protocol suite into layers of general functionality. In general, an application (the highest level of the model) uses a set of protocols to send its data down the layers, being further encapsulated at each level.

The layers of the protocol suite near the top are logically closer to the user application, while those near the bottom are logically closer to the physical transmission of the data. Viewing layers as providing or consuming a service is a method of [abstraction](#) to isolate upper layer protocols from the details of transmitting bits over, for example, [Ethernet](#) and [collision detection](#), while the lower layers avoid having to know the details of each and every application and its protocol.

Even when the layers are examined, the assorted architectural documents—there is no single architectural model such as ISO 7498, the [Open Systems Interconnection \(OSI\) model](#)—have fewer and less rigidly defined layers than the OSI model, and thus provide an easier fit for real-world protocols. One frequently referenced document, [RFC 1958](#), does not contain a stack of layers. The lack of emphasis on layering is a major difference between the IETF and OSI approaches. It only refers to the existence of the internetworking layer and generally to *upper layers*; this document was intended as a 1996 snapshot of the architecture: "The Internet and its architecture have grown in evolutionary fashion from modest beginnings, rather than from a Grand Plan. While this process of evolution is one of the main reasons for the technology's success, it nevertheless seems useful to record a snapshot of the current principles of the Internet architecture."

[RFC 1122](#), entitled *Host Requirements*, is structured in paragraphs referring to layers, but the document refers to many other architectural principles not emphasizing layering. It loosely defines a four-layer model, with the layers having names, not numbers, as follows:

Application layer

The [application layer](#) is the scope within which applications create user data and communicate this data to other applications on another or the same host. The applications, or processes, make use of the services provided by the underlying, lower layers, especially the Transport Layer which provides reliable or unreliable *pipes* to other processes. The communications partners are characterized by the application architecture, such as the [client-server model](#) and [peer-to-peer](#) networking. This is the layer in

which all *higher level* protocols, such as SMTP, FTP, SSH, HTTP, operate. Processes are addressed via ports which essentially represent services.

Transport layer

The transport layer performs host-to-host communications on either the same or different hosts and on either the local network or remote networks separated by routers. It provides a channel for the communication needs of applications. UDP is the basic transport layer protocol, providing an unreliable datagram service. The Transmission Control Protocol provides flow-control, connection establishment, and reliable transmission of data.

Internet layer

The internet layer exchanges datagrams across network boundaries. It provides a uniform networking interface that hides the actual topology (layout) of the underlying network connections. It is therefore also referred to as the layer that establishes internetworking, indeed, it defines and establishes the Internet. This layer defines the addressing and routing structures used for the TCP/IP protocol suite. The primary protocol in this scope is the Internet Protocol, which defines IP addresses. Its function in routing is to transport datagrams to the next IP router that has the connectivity to a network closer to the final data destination.

Link layer

The link layer defines the networking methods within the scope of the local network link on which hosts communicate without intervening routers. This layer includes the protocols used to describe the local network topology and the interfaces needed to effect transmission of Internet layer datagrams to next-neighbor hosts.

The Internet protocol suite and the layered protocol stack design were in use before the OSI model was established. Since then, the TCP/IP model has been compared with the OSI model in books and classrooms, which often results in confusion because the two models use different assumptions and goals, including the relative importance of strict layering.

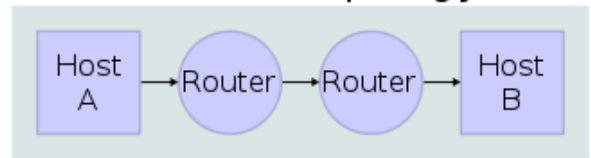
This abstraction also allows upper layers to provide services that the lower layers do not provide. While the original OSI model was extended to include connectionless services (OSIRM CL),^[22] IP is not designed to be reliable and is a best effort delivery protocol.

This means that all transport layer implementations must choose whether or how to provide reliability. UDP provides data integrity via a checksum but does not guarantee delivery; TCP provides both data integrity and delivery guarantee by retransmitting until the receiver acknowledges the reception of the packet.

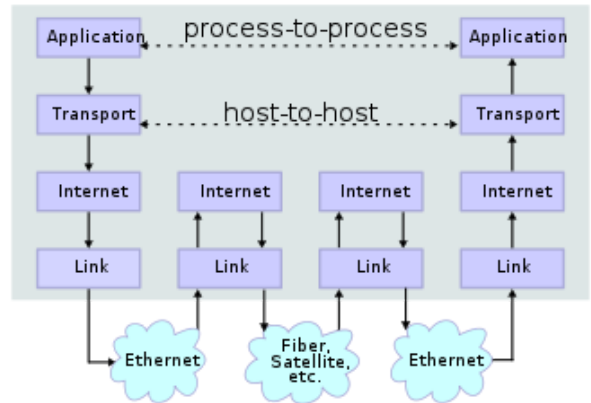
This model lacks the formalism of the OSI model and associated documents, but the IETF does not use a formal model and does not consider this a limitation, as illustrated in the comment by David D. Clark, "We reject: kings, presidents and voting. We believe in: rough consensus and running code." Criticisms of this model, which have been made with respect to the OSI model, often do not consider ISO's later extensions to that model.

For multi-access 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. While the IETF does not use the terminology, this is a subnetwork dependent convergence facility according to an extension to the OSI model, the internal organization of the network layer (IONL).^[23]

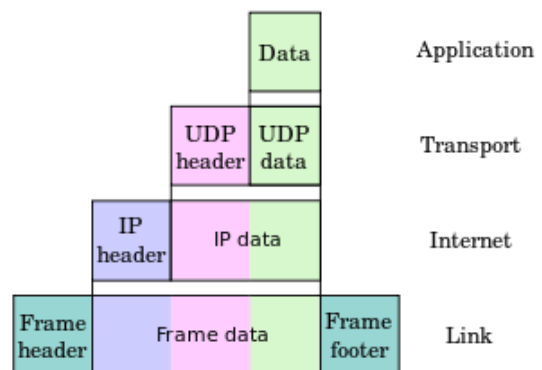
Network Topology



Data Flow



Two Internet hosts connected via two routers and the corresponding layers used at each hop. The application on each host executes read and write operations as if the processes were directly connected to each other by some kind of data pipe. Every other detail of the communication is hidden from each process. The underlying mechanisms that transmit data between the host computers are located in the lower protocol layers.



Encapsulation of application data descending through the layers described in RFC 1122

ICMP & IGMP operate on top of IP but do not transport data like UDP or TCP. Again, this functionality exists as layer management extensions to the OSI model, in its *Management Framework* (OSIRM MF)^[24]

The SSL/TLS library operates above the transport layer (uses TCP) but below application protocols. Again, there was no intention, on the part of the designers of these protocols, to comply with OSI architecture.

The link is treated as a black box. The IETF explicitly does not intend to discuss transmission systems, which is a less academic^[25] but practical alternative to the OSI model.

The following is a description of each layer in the TCP/IP networking model starting from the lowest level.

Link layer

The link layer has the networking scope of the local network connection to which a host is attached. This regime is called the *link* in TCP/IP literature. It is the lowest component layer of the Internet protocols, as TCP/IP is designed to be hardware independent. As a result, TCP/IP may be implemented on top of virtually any hardware networking technology.

The link layer is used to move packets between the Internet layer interfaces of two different hosts on the same link. The processes of transmitting and receiving packets on a given link can be controlled both in the software device driver for the network card, as well as on firmware or specialized chipsets. These perform data link functions such as adding a packet header to prepare it for transmission, then actually transmit the frame over a physical medium. The TCP/IP model includes specifications of translating the network addressing methods used in the Internet Protocol to link layer addresses, such as Media Access Control (MAC) addresses. All other aspects below that level, however, are implicitly assumed to exist in the link layer, but are not explicitly defined.

This is also the layer where packets may be selected to be sent over a virtual private network or other networking tunnel. In this scenario, the link layer data may be considered application data which traverses another instantiation of the IP stack for transmission or reception over another IP connection. Such a connection, or virtual link, may be established with a transport protocol or even an application scope protocol that serves as a tunnel in the link layer of the protocol stack. Thus, the TCP/IP model does not dictate a strict hierarchical encapsulation sequence.

The TCP/IP model's link layer corresponds to the Open Systems Interconnection (OSI) model physical and data link layers, layers one and two of the OSI model.

Internet layer

The internet layer has the responsibility of sending packets across potentially multiple networks. Internetworking requires sending data from the source network to the destination network. This process is called routing.

The Internet Protocol performs two basic functions:

- Host addressing and identification: This is accomplished with a hierarchical IP addressing system.
- Packet routing: This is the basic task of sending packets of data (datagrams) from source to destination by forwarding them to the next network router closer to the final destination.

The internet layer is not only agnostic of data structures at the transport layer, but it also does not distinguish between operation of the various transport layer protocols. IP carries data for a variety of different upper layer protocols. These protocols are each identified by a unique protocol number: for example, Internet Control Message Protocol (ICMP) and Internet Group Management Protocol (IGMP) are protocols 1 and 2, respectively.

Some of the protocols carried by IP, such as ICMP which is used to transmit diagnostic information, and IGMP which is used to manage IP Multicast data, are layered on top of IP but perform internetworking functions. This illustrates the differences in the architecture of the TCP/IP stack of the Internet and the OSI model. The TCP/IP model's internet layer corresponds to layer three of the Open Systems Interconnection (OSI) model, where it is referred to as the network layer.

The internet layer provides an unreliable datagram transmission facility between hosts located on potentially different IP networks by forwarding the transport layer datagrams to an appropriate next-hop router for further relaying to its destination. With this functionality, the internet layer makes possible internetworking, the interworking of different IP networks, and it essentially establishes the Internet. The Internet Protocol is the principal component of the internet layer, and it defines two addressing systems to identify network hosts' computers, and to locate them on the network. The original address system of the ARPANET and its

successor, the Internet, is [Internet Protocol version 4](#) (IPv4). It uses a 32-bit [IP address](#) and is therefore capable of identifying approximately four billion hosts. This limitation was eliminated in 1998 by the standardization of [Internet Protocol version 6](#) (IPv6) which uses 128-bit addresses. IPv6 production implementations emerged in approximately 2006.

Transport layer

The transport layer establishes basic data channels that applications use for task-specific data exchange. The layer establishes process-to-process connectivity, meaning it provides end-to-end services that are independent of the structure of user data and the logistics of exchanging information for any particular specific purpose. Its responsibility includes end-to-end message transfer independent of the underlying network, along with [error control](#), [segmentation](#), [flow control](#), [congestion control](#), and application addressing (port numbers). End-to-end message transmission or connecting applications at the transport layer can be categorized as either [connection-oriented](#), implemented in TCP, or [connectionless](#), implemented in UDP.

For the purpose of providing process-specific transmission channels for applications, the layer establishes the concept of the [port](#). This is a numbered logical construct allocated specifically for each of the communication channels an application needs. For many types of services, these [port numbers](#) have been standardized so that client computers may address specific services of a server computer without the involvement of service announcements or directory services.

Because IP provides only a [best effort delivery](#), some transport layer protocols offer reliability. However, IP can run over a reliable data link protocol such as the [High-Level Data Link Control](#) (HDLC).

For example, the TCP is a connection-oriented protocol that addresses numerous reliability issues in providing a [reliable byte stream](#):

- data arrives in-order
- data has minimal error (i.e., correctness)
- duplicate data is discarded
- lost or discarded packets are resent
- includes traffic congestion control

The newer [Stream Control Transmission Protocol](#) (SCTP) is also a reliable, connection-oriented transport mechanism. It is message-stream-oriented—not byte-stream-oriented like TCP—and provides multiple streams multiplexed over a single connection. It also provides [multi-homing](#) support, in which a connection end can be represented by multiple IP addresses (representing multiple physical interfaces), such that if one fails, the connection is not interrupted. It was developed initially for telephony applications (to transport [SS7](#) over IP), but can also be used for other applications.

The User Datagram Protocol is a connectionless [datagram](#) protocol. Like IP, it is a best effort, "unreliable" protocol. Reliability is addressed through [error detection](#) using a weak checksum algorithm. UDP is typically used for applications such as streaming media (audio, video, [Voice over IP](#) etc.) where on-time arrival is more important than reliability, or for simple query/response applications like [DNS](#) lookups, where the overhead of setting up a reliable connection is disproportionately large. [Real-time Transport Protocol](#) (RTP) is a datagram protocol that is designed for real-time data such as [streaming audio and video](#).

The applications at any given network address are distinguished by their TCP or UDP port. By convention certain *well known ports* are associated with specific applications.

The TCP/IP model's transport or host-to-host layer corresponds to the fourth layer in the Open Systems Interconnection (OSI) model, also called the transport layer.

Application layer

The [application layer](#) includes the protocols used by most applications for providing user services or exchanging application data over the network connections established by the lower level protocols. This may include some basic network support services such as protocols for routing and host configuration. Examples of application layer protocols include the [Hypertext Transfer Protocol](#) (HTTP), the [File Transfer Protocol](#) (FTP), the [Simple Mail Transfer Protocol](#) (SMTP), and the [Dynamic Host Configuration Protocol](#) (DHCP).^[26] Data coded according to application layer protocols are [encapsulated](#) into transport layer protocol units (such as TCP or UDP messages), which in turn use [lower layer protocols](#) to effect actual data transfer.

The TCP/IP model does not consider the specifics of formatting and presenting data, and does not define additional layers between the application and transport layers as in the OSI model (presentation and session layers). Such functions are the realm of libraries and application programming interfaces.

Application layer protocols generally treat the transport layer (and lower) protocols as black boxes which provide a stable network connection across which to communicate, although the applications are usually aware of key qualities of the transport layer connection such as the end point IP addresses and port numbers. Application layer protocols are often associated with particular client-server applications, and common services have *well-known* port numbers reserved by the Internet Assigned Numbers Authority (IANA). For example, the HyperText Transfer Protocol uses server port 80 and Telnet uses server port 23. Clients connecting to a service usually use ephemeral ports, i.e., port numbers assigned only for the duration of the transaction at random or from a specific range configured in the application.

The transport layer and lower-level layers are unconcerned with the specifics of application layer protocols. Routers and switches do not typically examine the encapsulated traffic, rather they just provide a conduit for it. However, some firewall and bandwidth throttling applications must interpret application data. An example is the Resource Reservation Protocol (RSVP). It is also sometimes necessary for network address translator (NAT) traversal to consider the application payload.

The application layer in the TCP/IP model is often compared as equivalent to a combination of the fifth (Session), sixth (Presentation), and the seventh (Application) layers of the Open Systems Interconnection (OSI) model.

Furthermore, the TCP/IP reference model distinguishes between *user protocols* and *support protocols*.^[27] Support protocols provide services to a system. User protocols are used for actual user applications. For example, FTP is a user protocol and DNS is a support protocol.

Layer names and number of layers in the literature

The following table shows various networking models. The number of layers varies between three and seven.

RFC 1122, Internet STD 3 (1989)	Cisco Academy^[28]	Kurose,^[29] Forouzan^[30]	Comer,^[31] Kozierok^[32]	Stallings^[33]	Tanenbaum^[34]	Arpanet Reference Model (RFC 871)	<u>OSI model</u>
<i>Four layers</i>	<i>Four layers</i>	<i>Five layers</i>	<i>Four+one layers</i>	<i>Five layers</i>	<i>Five layers</i>	<i>Three layers</i>	<i>Seven layers</i>
"Internet model"	"Internet model"	"Five-layer Internet model" or "TCP/IP protocol suite"	"TCP/IP 5- layer reference model"	"TCP/IP model"	"TCP/IP 5- layer reference model"	"Arpanet reference model"	OSI model
Application	Application	Application	Application	Application	Application	Application/Process	Application
							Presentation
							Session
Transport	Transport	Transport	Transport	Host-to-host or transport	Transport	Host-to-host	Transport
Internet	Internetwork	Network	Internet	Internet	Internet		Network
Link	Network interface	Data link	Data link (Network interface)	Network access	Data link	Network interface	Data link
		Physical	(Hardware)	Physical	Physical		Physical

Some of the networking models are from textbooks, which are secondary sources that may conflict with the intent of RFC 1122 and other IETF primary sources.^[35]

Comparison of TCP/IP and OSI layering

The three top layers in the OSI model, i.e. the application layer, the presentation layer and the session layer, are not distinguished separately in the TCP/IP model which only has an application layer above the transport layer. While some pure OSI protocol applications, such as X.400, also combined them, there is no requirement that a TCP/IP protocol stack must impose monolithic

architecture above the transport layer. For example, the NFS application protocol runs over the eXternal Data Representation (XDR) presentation protocol, which, in turn, runs over a protocol called Remote Procedure Call (RPC). RPC provides reliable record transmission, so it can safely use the best-effort UDP transport.

Different authors have interpreted the TCP/IP model differently, and disagree whether the link layer, or the entire TCP/IP model, covers OSI layer 1 (physical layer) issues, or whether a hardware layer is assumed below the link layer.

Several authors have attempted to incorporate the OSI model's layers 1 and 2 into the TCP/IP model, since these are commonly referred to in modern standards (for example, by IEEE and ITU). This often results in a model with five layers, where the link layer or network access layer is split into the OSI model's layers 1 and 2.

The IETF protocol development effort is not concerned with strict layering. Some of its protocols may not fit cleanly into the OSI model, although RFCs sometimes refer to it and often use the old OSI layer numbers. The IETF has repeatedly stated that Internet protocol and architecture development is not intended to be OSI-compliant. RFC 3439, addressing Internet architecture, contains a section entitled: "Layering Considered Harmful".^[35]

For example, the session and presentation layers of the OSI suite are considered to be included to the application layer of the TCP/IP suite. The functionality of the session layer can be found in protocols like HTTP and SMTP and is more evident in protocols like Telnet and the Session Initiation Protocol (SIP). Session layer functionality is also realized with the port numbering of the TCP and UDP protocols, which cover the transport layer in the TCP/IP suite. Functions of the presentation layer are realized in the TCP/IP applications with the MIME standard in data exchange.

Conflicts are apparent also in the original OSI model, ISO 7498, when not considering the annexes to this model, e.g., the ISO 7498/4 Management Framework, or the ISO 8648 Internal Organization of the Network layer (IONL). When the IONL and Management Framework documents are considered, the ICMP and IGMP are defined as layer management protocols for the network layer. In like manner, the IONL provides a structure for "subnetwork dependent convergence facilities" such as ARP and RARP.

IETF protocols can be encapsulated recursively, as demonstrated by tunneling protocols such as Generic Routing Encapsulation (GRE). GRE uses the same mechanism that OSI uses for tunneling at the network layer.

Implementations

The Internet protocol suite does not presume any specific hardware or software environment. It only requires that hardware and a software layer exists that is capable of sending and receiving packets on a computer network. As a result, the suite has been implemented on essentially every computing platform. A minimal implementation of TCP/IP includes the following: Internet Protocol (IP), Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Group Management Protocol (IGMP). In addition to IP, ICMP, TCP, UDP, Internet Protocol version 6 requires Neighbor Discovery Protocol (NDP), ICMPv6, and IGMPv6 and is often accompanied by an integrated IPSec security layer.

Application programmers are typically concerned only with interfaces in the application layer and often also in the transport layer, while the layers below are services provided by the TCP/IP stack in the operating system. Most IP implementations are accessible to programmers through sockets and APIs.

Unique implementations include Lightweight TCP/IP, an open source stack designed for embedded systems, and KA9Q NOS, a stack and associated protocols for amateur packet radio systems and personal computers connected via serial lines.

Microcontroller firmware in the network adapter typically handles link issues, supported by driver software in the operating system. Non-programmable analog and digital electronics are normally in charge of the physical components below the link layer, typically using an application-specific integrated circuit (ASIC) chipset for each network interface or other physical standard. High-performance routers are to a large extent based on fast non-programmable digital electronics, carrying out link level switching.

See also

- BBN Report 1822
- FLIP (protocol) (fast local Internet protocol stack)
- List of automation protocols
- List of information technology acronyms
- List of IP protocol numbers

- [List of network protocols](#)
- [List of TCP and UDP port numbers](#)

Bibliography

- Douglas E. Comer. *Internetworking with TCP/IP – Principles, Protocols and Architecture*. ISBN 86-7991-142-9
- Joseph G. Davies and Thomas F. Lee. *Microsoft Windows Server 2003 TCP/IP Protocols and Services*. ISBN 0-7356-1291-9
- Forouzan, Behrouz A. (2003). *TCP/IP Protocol Suite* (2nd ed.). McGraw-Hill. ISBN 0-07-246060-1.
- Craig Hunt *TCP/IP Network Administration*. O'Reilly (1998) ISBN 1-56592-322-7
- Maufer, Thomas A. (1999). *IP Fundamentals*. Prentice Hall. ISBN 0-13-975483-0.
- Ian McLean. *Windows(R) 2000 TCP/IP Black Book*. ISBN 1-57610-687-X
- Ajit Mungale *Pro .NET 1.1 Network Programming*. ISBN 1-59059-345-6
- W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. ISBN 0-201-63346-9
- W. Richard Stevens and Gary R. Wright. *TCP/IP Illustrated, Volume 2: The Implementation*. ISBN 0-201-63354-X
- W. Richard Stevens. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols*. ISBN 0-201-63495-3
- Andrew S. Tanenbaum. *Computer Networks*. ISBN 0-13-066102-3
- Clark, D. (1988). "The Design Philosophy of the DARPA Internet Protocols" (<http://www.cs.princeton.edu/~jrex/teaching/spring2005/reading/clark88.pdf>) (PDF). *SIGCOMM '88 Symposium proceedings on Communications architectures and protocols*. ACM: 106–114. doi:10.1145/52324.52336 (<https://doi.org/10.1145/52324.52336>). ISBN 0897912799. Retrieved 2011-10-16.

References

1. [RFC 1122](#), *Requirements for Internet Hosts – Communication Layers*, R. Braden (ed.), October 1989.
2. [RFC 1123](#), *Requirements for Internet Hosts – Application and Support*, R. Braden (ed.), October 1989
3. Cerf, Vinton G. & Cain, Edward (1983), "The DoD Internet Architecture Model" (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.7505&rep=rep1&type=pdf>), *Computer Networks*, 7, North-Holland, pp. 307–318
4. Cerf, Vinton G. & Kahn, Robert E. (1974), *A Protocol for Packet Network Intercommunication* (<https://www.cs.princeton.edu/courses/archive/fall06/cos561/papers/cerf74.pdf>) (PDF), 5
5. Internet Hall of Fame
6. Postel, Jon (1977), "Section 3.3.3.2", *Comments on Internet Protocol and TCP* (<https://www.rfc-editor.org/ien/ien2.txt>)
7. [RFC 1812](#), *Requirements for IP Version 4 Routers*, F. Baker (June 1995)
8. [RFC 675](#), *Specification of Internet Transmission Control Protocol*, V. Cerf et al. (December 1974)
9. "TCP/IP Internet Protocol" (https://www.livinginternet.com/i/ii_tcpip.htm). Retrieved 2017-12-31.
10. Ronda Hauben. "From the ARPANET to the Internet" (http://www.columbia.edu/~rh120/other/tcpdigest_paper.txt). TCP Digest (UUCP). Retrieved 2007-07-05.
11. Barry M. Leiner, et. al. (1997), *Brief History of the Internet* (https://www.internetsociety.org/wp-content/uploads/2017/09/ISOC-History-of-the-Internet_1997.pdf) (PDF), Internet Society, p. 15
12. [Wollongong](http://support.microsoft.com/kb/108007) (<http://support.microsoft.com/kb/108007>)
13. "A Short History of Internet Protocols at CERN" (<http://www.weblab.isti.cnr.it/education/ssfs/lezioni/slides/archives/cern.htm>). Retrieved 12 September 2016.
14. [Barry Appelman](#)
15. Baker, Steven; Gillies, Donald W. "Desktop TCP/IP at middle age" (<http://www.ece.ubc.ca/~gillies/9802net.html>).
16. Romkey, John (17 February 2011). "About" (<http://www.romkey.com/about/>). Retrieved 12 September 2016.
17. Karn, Phil. "KA9Q TCP Download Website".
18. [RFC 1958](#), *Architectural Principles of the Internet*, B. Carpenter (June 1996)
19. Rethinking the design of the Internet: The end-to-end arguments vs. the brave new world (http://www.csd.uoc.gr/~hy558/papers/Rethinking_2001.pdf), Marjory S. Blumenthal, David D. Clark, August 2001
20. p.23 INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION September 1981 Jon Postel Editor (<http://www.ietf.org/rfc/rfc0791.txt?number=791>)
21. Requirements for Internet Hosts – Communication Layers p.13 October 1989 R. Braden, Editor (<http://tools.ietf.org/html/rfc1122#page-12>)
22. *OSI: Reference Model Addendum 1: Connectionless-mode Transmission*, ISO7498/AD1, ISO7498/AD1, May 1986
23. "Information processing systems – Open Systems Interconnection – Internal organization of the Network Layer" (http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=16011), ISO 8648:1988.
24. "Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management framework" (http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=14258), ISO 7498-4:1989.

25. Shah, Deven N. (2009). *A Complete Guide To Internet And Web Programming* (https://books.google.co.in/books?id=8vKmPUfQMalC&pg=PA8&lpg=PA8&dq=The+IETF+explicitly+does+not+intend+to+discuss+transmission+systems,+which+is+a+less+academic&source=bl&ots=Cz6yjVCAUq&sig=5N9epUgg8Q94L8aTGKWCzTfj_ws&hl=en&sa=X&ved=0ahUKEwjUjfDz_czYAhUHSY8KHT_gDz8Q6AEIKDAA#v=onepage&q=The%20IETF%20explicitly%20does%20not%20intend%20to%20discuss%20transmission%20systems,%20which%20is%20a%20less%20academic&f=false). Dreamtech Press. ISBN 9788177229257.
26. *TCP/IP Illustrated: the protocols* (<http://www.kohala.com/start/tcpipiv1.html>), ISBN 0-201-63346-9, W. Richard Stevens, February 1994
27. RFC 1122 (<https://tools.ietf.org/html/rfc1122#page-9>), *Requirements for Internet Hosts – Communication Layers*, 1.1.3 *Internet Protocol Suite*, 1989
28. Dye, Mark; McDonald, Rick; Rufi, Antoon (29 October 2007). "Network Fundamentals, CCNA Exploration Companion Guide" (<https://books.google.com/books?id=JVAk7r6jHF4C>). Cisco Press. ISBN 9780132877435. Retrieved 12 September 2016 – via Google Books.
29. James F. Kurose, Keith W. Ross, *Computer Networking: A Top-Down Approach*, 2008 (http://www.pearsonhighered.com/education/academic/product/0,,0321497708,00%2ben-USS_01DBC.html) ISBN 0-321-49770-8
30. Forouzan, Behrouz A.; Fegan, Sophia Chung (1 August 2003). "Data Communications and Networking" (<https://books.google.co.uk/books?id=U3Gcf65Pu9IC>). McGraw-Hill Higher Education. ISBN 9780072923544. Retrieved 12 September 2016 – via Google Books.
31. Comer, Douglas (1 January 2006). "Internetworking with TCP/IP: Principles, protocols, and architecture" (<https://books.google.co.uk/books?id=jonyuTASbWAC>). Prentice Hall. ISBN 0-13-187671-6. Retrieved 12 September 2016 – via Google Books.
32. Kozierok, Charles M. (1 January 2005). "The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference" (<https://books.google.com/books?id=Pm4RgYV2w4YC>). No Starch Press. ISBN 9781593270476. Retrieved 12 September 2016 – via Google Books.
33. Stallings, William (1 January 2007). "Data and Computer Communications" (https://books.google.com/books?id=c_AWmhkovR0C). Prentice Hall. ISBN 0-13-243310-9. Retrieved 12 September 2016 – via Google Books.
34. Tanenbaum, Andrew S. (1 January 2003). "Computer Networks" (https://books.google.com/books?id=Pd-z64SJRBA&pg=PA42&vq=internet%20layer&dq=networks&hl=sv&source=gb_s&sig=ACfU3U3DHANelz0sOsd5NK4VXSrgNFYVAw#PPA42,M1). Prentice Hall PTR. ISBN 0-13-066102-3. Retrieved 12 September 2016 – via Google Books.
35. R. Bush; D. Meyer (December 2002), *Some Internet Architectural Guidelines and Philosophy* (<http://www.ietf.org/rfc/rfc3439.txt>), Internet Engineering Task Force

External links

- Internet History (<http://www.livinginternet.com/i/ii.htm>) – Pages on Robert Kahn, Vinton Cerf, and TCP/IP (reviewed by Cerf and Kahn).
- RFC 675 (<http://www.ietf.org/rfc/rfc0675.txt>) – Specification of Internet Transmission Control Program, December 1974 Version
- RFC 1180 A TCP/IP Tutorial – from the Internet Engineering Task Force (January 1991)
- TCP/IP FAQ (<http://www.itprc.com/tcpipfaq/>)
- The TCP/IP Guide (<http://www.tcpipguide.com/free/>) – A comprehensive look at the protocols and the procedures/processes involved
- A Study of the ARPANET TCP/IP Digest (http://www.columbia.edu/~rh120/other/tcpdigest_paper.txt)
- TCP/IP Sequence Diagrams (<http://www.eventhelix.com/RealtimeMantra/Networking/>)
- Daryl's TCP/IP Primer (<http://www.ipprimer.com/>) – Intro to TCP/IP LAN administration, conversational style
- Introduction to TCP/IP (<http://www.linux-tutorial.info/MContent-142>)
- A Protocol for Packet Network Intercommunication, Cerf & Kahn, IEEE Trans on Comms, Vol Com-22, No 5 May 1974 (<https://www.cs.princeton.edu/courses/archive/fall06/cos561/papers/cerf74.pdf>)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Internet_protocol_suite&oldid=836078780"

This page was last edited on 12 April 2018, at 15:34.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.