WIKIPEDIA

Internet protocol suite

The **Internet protocol suite** is the conceptual model and set of communications protocols used in 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). During its development, versions of it were 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. Its implementation is a protocol stack.

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, providing process-to-process data exchange for applications.

The technical standards underlying the Internet protocol suite and 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

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, who helped develop 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. They drew on the experience from the ARPANET research community and the International Networking Working Group, which Cerf chaired.[4]

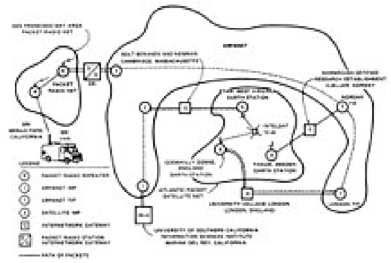


Diagram of the first internetworked connection

By the summer of 1973, Kahn and Cerf had worked out a fundamental reformulation, in which the differences between local network protocols were hidden by using a common internetwork protocol, and, instead of the network being responsible for reliability, as in the existing ARPANET protocols, this function was delegated to the hosts. Cerf credits Hubert Zimmermann and Louis Pouzin, designer of the CYCLADES network, with important influences on this design.[5][6] The new protocol was implemented as the Transmission Control Program in 1974.[7]

An SRI International Packet Radio Van, used for the first three-way internetworked transmission.

Initially, the Transmission Control Program managed both datagram transmissions and routing, but as experience with the protocol grew, collaborators recommended division of functionality into layers of distinct protocols. Advocates included Jonathan Postel of the University of Southern California's Information Sciences Institute, who edited the Request for Comments (RFCs), the technical and strategic document series that has both documented and catalyzed Internet development,[8] and the research group of Robert Metcalfe at Xerox PARC.[9][10] Postel stated, "We are screwing up in our design of Internet protocols by violating the principle of layering."[11] Encapsulation of different mechanisms was intended to create an environment where the upper layers could access only what was needed from the lower layers. A monolithic design would be inflexible and lead to scalability issues. In version 3 of TCP, written in 1978, the Transmission Control Program was split into two distinct protocols, the Internet Protocol as connectionless layer and the Transmission Control Protocol as a reliable connection-oriented service.[12]

The design of the network included the recognition that it should provide only the functions of efficiently transmitting and routing traffic between end nodes and that all other intelligence should be located at the edge of the network, in the end nodes. This design is known as the end-to-end principle. Using this design, it became possible to connect other networks to the ARPANET that used the same principle, irrespective of other local characteristics, thereby solving Kahn's initial internetworking problem. A popular expression is that TCP/IP, the eventual product of Cerf and Kahn's work, can run over "two tin cans and a string." Years later, as a joke, the IP over Avian Carriers formal protocol specification was created and successfully tested.

DARPA contracted with BBN Technologies, Stanford University, and the University College London to develop operational versions of the protocol on several hardware platforms.[13] During development of the protocol the version number of the packet routing layer progressed from version 1 to version 4, the latter of which was installed in the ARPANET in 1983. It became known as *Internet Protocol version 4* (IPv4) as the protocol that is still in use in the Internet, alongside its current successor, Internet Protocol version 6 (IPv6).

Early implementation

In 1975, a two-network TCP/IP communications test was performed between Stanford and University College London. In November 1977, a three-network TCP/IP test was conducted between sites in the US, the UK, and Norway. Several other TCP/IP prototypes were developed at multiple research centers between 1978 and 1983.

A computer called a router is provided with an interface to each network. It forwards network packets back and forth between them.[14] Originally a router was called *gateway*, but the term was changed to avoid confusion with other types of gateways.[15]

Adoption

In March 1982, the US Department of Defense declared TCP/IP as the standard for all military computer networking.[16] In the same year, NORSAR and Peter Kirstein's research group at University College London adopted the protocol.[13][17][18] The migration of the ARPANET to TCP/IP was officially completed on flag day January 1, 1983, when the new protocols were permanently activated.[19]

In 1985, the Internet Advisory Board (later Internet Architecture Board) held a three-day TCP/IP workshop for the computer industry, attended by 250 vendor representatives, promoting the protocol and leading to its increasing commercial use. In 1985, the first Interop conference focused on network interoperability by broader adoption of TCP/IP. The conference was founded by Dan Lynch, an early Internet activist. From the beginning, large corporations, such as IBM and DEC, attended the meeting.[20]

IBM, AT&T and DEC were the first major corporations to adopt TCP/IP, this despite having competing proprietary protocols. In IBM, from 1984, Barry Appelman's group did TCP/IP development. They navigated the corporate politics to get a stream of TCP/IP products for various IBM systems, including MVS, VM, and OS/2. At the same time, several smaller companies, such as FTP Software and the Wollongong Group, began offering TCP/IP stacks for DOS and Microsoft Windows.[21] The first VM/CMS TCP/IP stack came from the University of Wisconsin.[22]

Some of the early 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. 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.[23][24] Starting in 1985, Phil Karn created a multi-connection TCP application for ham radio systems (KA9Q TCP).[25]

The spread of TCP/IP was fueled further in June 1989, when the University of California, Berkeley agreed to place the TCP/IP code developed for BSD UNIX into the public domain. Various corporate vendors, including IBM, included this code in commercial TCP/IP software releases. Microsoft released a native TCP/IP stack in Windows 95. This event helped cement TCP/IP's dominance over other protocols on Microsoft-based networks, which included IBM's Systems Network Architecture (SNA), and on other platforms such as Digital Equipment Corporation's DECnet, Open Systems Interconnection (OSI), and Xerox Network Systems (XNS).

Nonetheless, for a period in the late 1980s and early 1990s, engineers, organizations and nations were polarized over the issue of which standard, the OSI model or the Internet protocol suite would result in the best and most robust computer networks.[26][27][28]

Formal specification and standards

The technical standards underlying the Internet protocol suite and its constituent protocols have been delegated to the Internet Engineering Task Force (IETF).

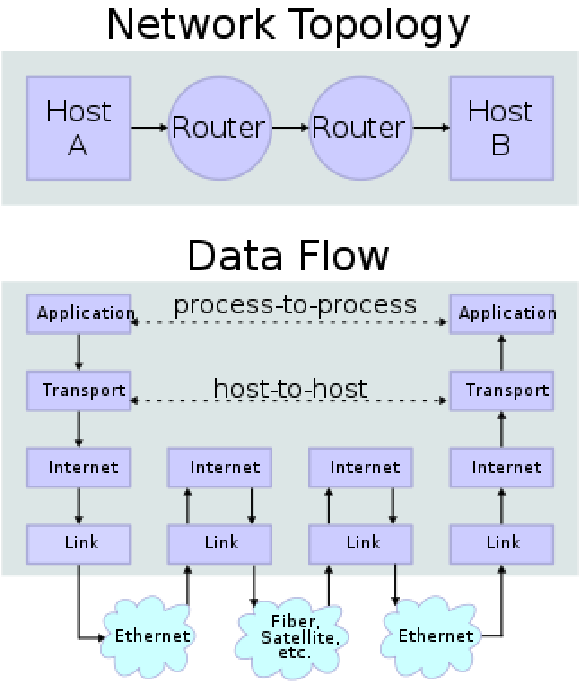
The characteristic architecture of the Internet Protocol Suite is its broad division into operating scopes for the protocols that constitute its core functionality. The defining specification of the suite is RFC 1122, which broadly outlines four abstraction layers.[1] These have stood the test of time, as the IETF has never modified this structure. As such a model of networking, the Internet Protocol Suite predates the OSI model, a more comprehensive reference framework for general networking systems.

Key architectural principles

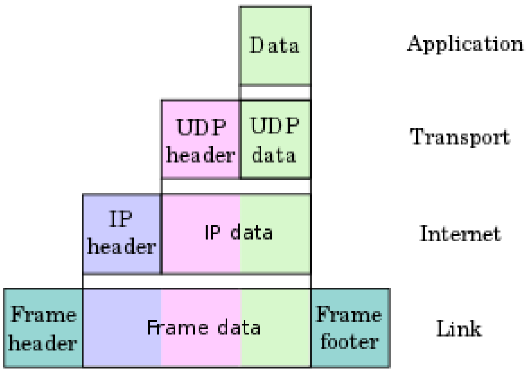
The end-to-end principle has evolved over time. Its original expression put the maintenance of state and overall intelligence at the edges, and assumed the Internet that connected the edges retained no state and concentrated on speed and simplicity. Real-world needs for firewalls, network address translators, web content caches and the like have forced changes in this principle.[29]

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)."[30] "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."[31]

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. The data is further encapsulated at each level.



Conceptual data flow in a simple network topology of two hosts (*A* and *B*) connected by a link between their respective routers. 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. After establishment of this pipe, most details of the communication are hidden from each process, as the underlying principles of communication are implemented in the lower protocol layers. In analogy, at the transport layer the communication appears as host-to-host, without knowledge of the application data structures and the connecting routers, while at the internetworking layer, individual network boundaries are traversed at each router.



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

An early architectural document, RFC 1122 (https://tools.ietf.org/html/rfc1122), emphasizes architectural principles over layering.[32] RFC 1122, titled *Host Requirements*, is structured in paragraphs referring to layers, but the document refers to many other architectural principles and does not emphasize layering. It loosely defines a four-layer model, with the layers having names, not numbers, as follows:

- The application layer is the scope within which applications, or processes, create user data and communicate this data to other applications on another or the same host. The applications 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 application protocols, such as SMTP, FTP, SSH, HTTP, operate. Processes are addressed via ports which essentially represent services.
- The transport layer performs host-to-host communications on either the local network or remote networks separated by routers.[33] It provides a channel for the communication needs of applications. UDP is the basic transport layer protocol, providing an unreliable connectionless datagram service. The Transmission Control Protocol provides flow-control, connection establishment, and reliable transmission of data.
- 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 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 host, functioning as an IP router, that has the connectivity to a network closer to the final data destination.
- 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 affect the transmission of Internet layer datagrams to next-neighbor hosts.

Link layer

The protocols of the link layer operate within the scope of the local network connection to which a host is attached. This regime is called the *link* in TCP/IP parlance and is the lowest component layer of the suite. The link includes all hosts accessible without traversing a router. The size of the link is therefore determined by the networking hardware design. In principle, TCP/IP is designed to be hardware independent and may be implemented on top of virtually any link-layer technology. This includes not only hardware implementations, but also virtual link layers such as virtual private networks and networking tunnels.

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 the link can be controlled in the device driver for the network card, as well as in firmware or by specialized chipsets. These perform functions, such as framing, to prepare the Internet layer packets for transmission, and finally transmit the frames to the physical layer and over a transmission medium. The TCP/IP model includes specifications for 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, and are not explicitly defined in the TCP/IP model.

The link layer in the TCP/IP model has corresponding functions in Layer 2 of the OSI model.

Internet layer

Internetworking requires sending data from the source network to the destination network. This process is called routing and is supported by host addressing and identification using the hierarchical IP addressing system. The internet layer provides an unreliable datagram transmission facility between hosts located on potentially different IP networks by forwarding datagrams to an appropriate next-hop router for further relaying to its destination. The internet layer has the responsibility of sending packets across potentially multiple networks. With this functionality, the internet layer makes possible internetworking, the interworking of different IP networks, and it essentially establishes the Internet.

The internet layer does not distinguish between 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.

The Internet Protocol is the principal component of the internet layer, and it defines two addressing systems to identify network hosts 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 host-to-host connectivity in the form of end-to-end message transfer services that are independent of the underlying network and independent of the structure of user data and the logistics of exchanging information. Connectivity at the transport layer can be categorized as either connection-oriented, implemented in TCP, or connectionless, implemented in UDP. The protocols in this layer may provide error control, segmentation, flow control, congestion control, and application addressing (port numbers).

For the purpose of providing process-specific transmission channels for applications, the layer establishes the concept of the network 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 service discovery or directory services.

Because IP provides only a best-effort delivery, some transport-layer protocols offer reliability.

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 multihoming 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).

Reliability can also be achieved by running IP over a reliable data-link protocol such as the High-Level Data Link Control (HDLC).

The User Datagram Protocol (UDP) is a connectionless datagram protocol. Like IP, it is a best-effort, unreliable protocol. Reliability is addressed through error detection using a 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 used over UDP and is designed for real-time data such as streaming media.

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 roughly to the fourth layer in the 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 routing protocols 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).[34] Data coded according to application layer protocols are encapsulated into transport layer protocol units (such as TCP streams or UDP datagrams), 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 OSI model.

Furthermore, the TCP/IP model distinguishes between *user protocols* and *support protocols*.[35] Support protocols provide services to a system of network infrastructure. 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[36]	Kurose,[37] Forouzan[38]	Comer,[39] Kozierok[40]	Stallings[41]	Tanenbaum[42]	Arpanet Reference Model (RFC 871)	OSI model
Four layers	*Four layers*	*Five layers*	*Four+one layers*	*Five layers*	*Five layers*	*Three layers*	*Seven layers*
"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.[43]

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, referring to the Internet architecture, contains a section entitled: "Layering Considered Harmful".

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, an early layered network model
- 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 978-0-07-246060-5.
- Craig Hunt *TCP/IP Network Administration*. O'Reilly (1998) ISBN 1-56592-322-7
- Maufer, Thomas A. (1999). *IP Fundamentals*. Prentice Hall. ISBN 978-0-13-975483-8.
- 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". *Symposium proceedings on Communications architectures and protocols - SIGCOMM '88* (http://www.cs.princeton.edu/~jrex/teaching/spring2005/reading/clark88.pdf) (PDF). *SIGCOMM '88 Symposium Proceedings on Communications Architectures and Protocols*. ACM. pp. 106–114. doi:10.1145/52324.52336 (https://doi.org/10.1145%2F52324.52336). ISBN 978-0897912792. S2CID 6156615 (https://api.semanticscholar.org/CorpusID:6156615). Retrieved October 16, 2011.

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", *Computer Networks*, 7, North-Holland, pp. 307–318, CiteSeerX 10.1.1.88.7505 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.88.7505)
4. Abbate, Janet (2000). *Inventing the Internet* (https://books.google.com/books?id=E2BdY6WQo4AC&pg=PA123). MIT Press. pp. 123–4. ISBN 978-0-262-51115-5.
5. Cerf, V.; Kahn, R. (1974). "A Protocol for Packet Network Intercommunication" (https://www.cs.princeton.edu/courses/archive/fall06/cos561/papers/cerf74.pdf) (PDF). *IEEE Transactions on Communications*. **22** (5): 637–648. doi:10.1109/TCOM.1974.1092259 (https://doi.org/10.1109%2FTCOM.1974.1092259). ISSN 1558-0857 (https://www.worldcat.org/issn/1558-0857). "The authors wish to thank a number of colleagues for helpful comments during early discussions of international network protocols, especially R. Metcalfe, R. Scantlebury, D. Walden, and H. Zimmerman; D. Davies and L. Pouzin who constructively commented on the fragmentation and accounting issues; and S. Crocker who commented on the creation and destruction of associations."
6. "The internet's fifth man" (https://www.economist.com/news/technology-quarterly/21590765-louis-pouzin-helped-create-internet-now-he-campaigning-ensure-its). *Economist*. December 13, 2013. Retrieved September 11, 2017. "In the early 1970s Mr Pouzin created an innovative data network that linked locations in France, Italy and Britain. Its simplicity and efficiency pointed the way to a network that could connect not just dozens of machines, but millions of them. It captured the imagination of Dr Cerf and Dr Kahn, who included aspects of its design in the protocols that now power the internet."
7. Vinton Cerf, Yogen Dalal, Carl Sunshine (December 1974), RFC 675 (https://tools.ietf.org/html/rfc675), *Specification of Internet Transmission Control Protocol* (December 1974)
8. Internet Hall of Fame
9. Panzaris, Georgios (2008). *Machines and romances: the technical and narrative construction of networked computing as a general-purpose platform, 1960-1995* (https://books.google.com/books?id=9yMhAQAAIAAJ). Stanford University. p. 128.
10. Pelkey, James L. (2007). "Yogen Dalal". *Entrepreneurial Capitalism and Innovation: A History of Computer Communications, 1968-1988* (http://www.historyofcomputercommunications.info/Individuals/abstracts/yogen-dalal.html). Retrieved September 5, 2019.
11. Postel, Jon (1977), "Section 3.3.3.2" (https://www.rfc-editor.org/ien/ien2.txt), *Comments on Internet Protocol and TCP*
12. "The TCP/IP Guide - TCP/IP Overview and History" (http://www.tcpipguide.com/free/t_TCPIPOverviewandHistory-2.htm). *www.tcpipguide.com*. Retrieved February 11, 2020.
13. by Vinton Cerf, as told to Bernard Aboba (1993). "How the Internet Came to Be" (https://web.archive.org/web/20170926042220/http://elk.informatik.hs-augsburg.de/tmp/cdrom-oss/CerfHowInternetCame2B.html). Archived from the original (http://elk.informatik.hs-augsburg.de/tmp/cdrom-oss/CerfHowInternetCame2B.html) on September 26, 2017. Retrieved September 25, 2017. "We began doing concurrent implementations at Stanford, BBN, and University College London. So effort at developing the Internet protocols was international from the beginning. ... Mar '82 - Norway leaves the ARPANET and become an Internet connection via TCP/IP over SATNET. Nov '82 - UCL leaves the ARPANET and becomes an Internet connection."
14. RFC 1812, *Requirements for IP Version 4 Routers*, F. Baker (June 1995)

15. Crowell, William; Contos, Brian; DeRodeff, Colby (2011). *Physical and Logical Security Convergence: Powered By Enterprise Security Management*. Syngress. p. 99. ISBN 9780080558783.
16. Ronda Hauben. "From the ARPANET to the Internet" (http://www.columbia.edu/~rh120/other/tcpdigest_paper.txt). TCP Digest (UUCP). Retrieved July 5, 2007.
17. Martin, Olivier (2012). *The "Hidden" Prehistory of European Research Networking* (https://books.google.com/books?id=eTRYAAAAQBAJ&pg=PT17). Trafford Publishing. ISBN 978-1466938724.
18. Kirstein, Peter T. "Early experiences with the ARPANET and Internet in the UK" (http://nrg.cs.ucl.ac.uk/internet-history.html). Department of Computer Science, Systems and Networks Research Group, University College London. Retrieved April 13, 2016.
19. "TCP/IP Internet Protocol" (https://web.archive.org/web/20180101082256/https://www.livinginternet.com/i/ii_tcpip.htm). Archived from the original (https://www.livinginternet.com/i/ii_tcpip.htm) on January 1, 2018. Retrieved December 31, 2017.
20. Leiner, Barry M.; 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
21. Wollongong (http://support.microsoft.com/kb/108007)
22. "A Short History of Internet Protocols at CERN" (https://web.archive.org/web/20161110200124/http://www.weblab.isti.cnr.it/education/ssfs/lezioni/slides/archives/cern.htm). Archived from the original (http://www.weblab.isti.cnr.it/education/ssfs/lezioni/slides/archives/cern.htm) on November 10, 2016. Retrieved September 12, 2016.
23. Baker, Steven; Gillies, Donald W. "Desktop TCP/IP at middle age" (http://www.ece.ubc.ca/~gillies/9802net.html).
24. Romkey, John (February 17, 2011). "About" (http://www.romkey.com/about/). Retrieved September 12, 2016.
25. Phil Karn, *KA9Q TCP Download Website*
26. Andrew L. Russell (July 30, 2013). "OSI: The Internet That Wasn't" (https://spectrum.ieee.org/computing/networks/osi-the-internet-that-wasnt). *IEEE Spectrum*. Vol. 50 no. 8.
27. Russell, Andrew L. "Rough Consensus and Running Code' and the Internet-OSI Standards War" (https://web.archive.org/web/20191117080112/https://www2.cs.duke.edu/courses/common/compsci092/papers/govern/consensus.pdf) (PDF). IEEE Annals of the History of Computing. Archived from the original (https://www2.cs.duke.edu/courses/common/compsci092/papers/govern/consensus.pdf) (PDF) on November 17, 2019.
28. Davies, Howard; Bressan, Beatrice (April 26, 2010). *A History of International Research Networking: The People who Made it Happen* (https://books.google.com/books?id=DN-t8MpZ0-wC&q=%22protocol+wars%22&pg=PA106). John Wiley & Sons. ISBN 978-3-527-32710-2.
29. 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
30. Jon Postel, ed. (September 1981). *Internet Protocol DARPA Internet Program Protocol Specification* (https://tools.ietf.org/html/rfc791#page-23). p. 23. doi:10.17487/RFC0791 (https://doi.org/10.17487%2FRFC0791). RFC 791 (https://tools.ietf.org/html/rfc791).
31. R. Braden, ed. (October 1989). *Requirements for Internet Hosts – Communication Layers* (https://tools.ietf.org/html/rfc1122#page-13). p. 13. doi:10.17487/RFC1122 (https://doi.org/10.17487%2FRFC1122). RFC 1122 (https://tools.ietf.org/html/rfc1122).
32. B. Carpenter, ed. (June 1996). *Architectural Principles of the Internet* (https://tools.ietf.org/html/rfc1958). doi:10.17487/RFC1958 (https://doi.org/10.17487%2FRFC1958). RFC 1958 (https://tools.ietf.org/html/rfc1958).
33. Hunt, Craig (2002). *TCP/IP Network Administration* (3rd ed.). O'Reilly. pp. 9–10. ISBN 9781449390785.
34. *TCP/IP Illustrated: the protocols* (http://www.kohala.com/start/tcpipiv1.html), ISBN 0-201-63346-9, W. Richard Stevens, February 1994
35. RFC 1122 (https://tools.ietf.org/html/rfc1122#page-9), *Requirements for Internet Hosts – Communication Layers*, 1.1.3 *Internet Protocol Suite*, 1989
36. Dye, Mark; McDonald, Rick; Rufi, Antoon (October 29, 2007). *Network Fundamentals, CCNA Exploration Companion Guide* (https://books.google.com/books?id=JVAk7r6jHF4C). Cisco Press. ISBN 9780132877435. Retrieved September 12, 2016 – via Google Books.
37. James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach, 2008 (http://www.pearsonhighered.com/educator/academic/product/0,,0321497708,00%2ben-USS_01DBC.html) ISBN 0-321-49770-8

38. Forouzan, Behrouz A.; Fegan, Sophia Chung (August 1, 2003). *Data Communications and Networking* (https://books.google.com/books?id=U3Gcf65Pu9IC). McGraw-Hill Higher Education. ISBN 9780072923544. Retrieved September 12, 2016 – via Google Books.
39. Comer, Douglas (January 1, 2006). *Internetworking with TCP/IP: Principles, protocols, and architecture* (https://books.google.com/books?id=jonyuTASbWAC). Prentice Hall. ISBN 0-13-187671-6. Retrieved September 12, 2016 – via Google Books.
40. Kozierok, Charles M. (January 1, 2005). *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference* (https://books.google.com/books?id=Pm4RgYV2w4YC). No Starch Press. ISBN 9781593270476. Retrieved September 12, 2016 – via Google Books.
41. Stallings, William (January 1, 2007). *Data and Computer Communications* (https://books.google.com/books?id=c_AWmhkovR0C). Prentice Hall. ISBN 978-0-13-243310-5. Retrieved September 12, 2016 – via Google Books.
42. Tanenbaum, Andrew S. (January 1, 2003). *Computer Networks* (https://archive.org/details/computernetworks00tane_2). Prentice Hall PTR. p. 42 (https://archive.org/details/computernetworks00tane_2/page/42). ISBN 0-13-066102-3. Retrieved September 12, 2016 – via Internet Archive. "networks."
43. RFC 3439, *Some Internet Architectural Guidelines and Philosophy*, R. Bush, D. Meyer (eds.), December 2002.

External links

- Internet History (https://www.livinginternet.com/i/ii.htm) – Pages on Robert Kahn, Vinton Cerf, and TCP/IP (reviewed by Cerf and Kahn).
- RFC 675 (https://tools.ietf.org/html/rfc675) – 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 (https://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 (https://www.eventhelix.com/RealtimeMantra/Networking/)
- Daryl's TCP/IP Primer (https://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=1008636563"