NETWORKING PROTOCOLS

TCP/IP - TCP stands for Transmission Control Protocol, a communications standard that enables application programs and computing devices to exchange messages over a network. It is designed to send packets across the internet and ensure the successful delivery of data and messages over networks.

TCP organizes data so that it can be transmitted between a server and a client. It guarantees the integrity of the data being communicated over a network. Before it transmits data, TCP establishes a connection between a source and its destination, which it ensures remains live until communication begins. It then breaks large amounts of data into smaller packets, while ensuring data integrity is in place throughout the process.

As a result, high-level protocols that need to transmit data all use TCP Protocol. Examples include peer-to-peer sharing methods like File Transfer Protocol (FTP), Secure Shell (SSH), and Telnet. It is also used to send and receive email through Internet Message Access Protocol (IMAP), Post Office Protocol (POP), and Simple Mail Transfer Protocol (SMTP), and for web access through the Hypertext Transfer Protocol (HTTP).

An alternative to TCP in networking is the User Datagram Protocol (UDP), which is used to establish low-latency connections between applications and decrease transmissions time. TCP can be an expensive network tool as it includes absent or corrupted packets and protects data delivery with controls like acknowledgments, connection startup, and flow control.

UDP does not provide error connection or packet sequencing nor does it signal a destination before it delivers data, which makes it less reliable but less expensive. As such, it is a good option for time-sensitive situations, such as Domain Name System (DNS) lookup, Voice over Internet Protocol (VoIP), and streaming media.

What is IP?

The Internet Protocol (IP) is the method for sending data from one device to another across the internet. Every device has an IP address that uniquely identifies it and enables it to communicate with and exchange data with other devices connected to the internet. Today, it’s considered the standard for fast and secure communication directly between mobile devices.

IP is responsible for defining how applications and devices exchange packets of data with each other. It is the principal communications protocol responsible for the formats and rules for exchanging data and messages between computers on a single network or several internet-connected networks. It does this through the Internet Protocol Suite (TCP/IP), a group of communications protocols that are split into four abstraction layers.

IP is the main protocol within the internet layer of the TCP/IP. Its main purpose is to deliver data packets between the source application or device and the destination using methods and structures that place tags, such as address information, within data packets.

TCP vs. IP: What is the Difference?

TCP and IP are separate protocols that work together to ensure data is delivered to its intended destination within a network. **IP obtains and defines the address—the IP address—of the application or device the data must be sent to. TCP is then responsible for transporting and routing data through the network architecture and ensuring it gets delivered to the destination application or device that IP has defined.** Both technologies working together allow communication between devices over long distances, making it possible to transfer data where it needs to go in the most efficient way possible.

In other words, **the IP address is akin to a phone number assigned to a smartphone. TCP is the computer networking version of the technology used to make the smartphone ring and enable its user to talk to the person who called them.**

How Does TCP/IP Work?

The TCP/IP model is the default method of data communication on the Internet. It was developed by the United States Department of Defense to enable the accurate and correct transmission of data between devices. It breaks messages into packets to avoid having to resend the entire message in case it encounters a problem during transmission. Packets are automatically reassembled once they reach their destination. Every packet can take a different route between the source and the destination computer, depending on whether the original route used becomes congested or unavailable.

TCP/IP divides communication tasks into layers that keep the process standardized, without hardware and software providers doing the management themselves. The data packets must pass through four layers before they are received by the destination device, then TCP/IP goes through the layers in reverse order to put the message back into its original format.

As a connection-based protocol, the TCP establishes and maintains a connection between applications or devices until they finish exchanging data. It determines how the original message should be broken into packets, numbers and reassembles the packets, and sends them on to other devices on the network, such as routers, security gateways, and switches, then on to their destination. TCP also sends and receives packets from the network layer, handles the transmission of any dropped packets, manages flow control, and ensures all packets reach their destination.

A good example of how this works in practice is when an email is sent using SMTP from an email server. To start the process, the TCP layer in the server divides the message into packets, numbers them, and forwards them to the IP layer, which then transports each packet to the destination email server. When packets arrive, they are handed back to the TCP layer to be reassembled into the original message format and handed back to the email server, which delivers the message to a user's email inbox.

TCP/IP uses a three-way handshake to establish a connection between a device and a server, which ensures multiple TCP socket connections can be transferred in both directions concurrently. Both the device and server must synchronize and acknowledge packets before communication begins, then they can negotiate, separate, and transfer TCP socket connections.

The 4 Layers of the TCP/IP Model

The TCP/IP model defines how devices should transmit data between them and enables communication over networks and large distances. The model represents how data is exchanged and organized over networks. It is split into four layers, which set the standards for data exchange and represent how data is handled and packaged when being delivered between applications, devices, and servers.

The four layers of the TCP/IP model are as follows:

Datalink layer: The datalink layer defines how data should be sent, handles the physical act of sending and receiving data, and is responsible for transmitting data between applications or devices on a network. This includes defining how data should be signaled by hardware and other transmission devices on a network, such as a computer's device driver, an Ethernet cable, a network interface card (NIC), or a wireless network. It is also referred to as the link layer, network access layer, network interface layer, or physical layer and is the combination of the physical and data link layers of the Open Systems

Interconnection (OSI) model, which standardizes communications functions on computing and telecommunications systems.

Internet layer: The internet layer is responsible for sending packets from a network and controlling their movement across a network to ensure they reach their destination. It provides the functions and procedures for transferring data sequences between applications and devices across networks.

Transport layer: The transport layer is responsible for providing a solid and reliable data connection between the original application or device and its intended destination. This is the level where data is divided into packets and numbered to create a sequence. The transport layer then determines how much data must be sent, where it should be sent to, and at what rate. It ensures that data packets are sent without errors and in sequence and obtains the acknowledgment that the destination device has received the data packets.

Application layer: The application layer refers to programs that need TCP/IP to help them communicate with each other. This is the level that users typically interact with, such as email systems and messaging platforms. It combines the session, presentation, and application layers of the OSI model.

Are Your Data Packets Private Over TCP/IP?

Data packets sent over TCP/IP are not private, which means they can be seen or intercepted. For this reason, it is vital to avoid using public Wi-Fi networks for sending private data and to ensure information is encrypted. One way to encrypt data being shared through TCP/IP is through a virtual private network (VPN).

What is My TCP/IP Address?

A TCP/IP address may be required to configure a network and is most likely required in a local network.

Finding a public IP address is a simple process that can be discovered using various online tools. These tools quickly detect the IP address of the device being used, along with the user's host IP address, internet service provider (ISP), remote port, and the type of browser, device, and operating system they are using.

Another way to discover the TCP/IP is through the administration page of a router, which displays the user's current public IP address, the router's IP address, subnet mask, and other network information.

FAQs

What is TCP used for?

TCP enables data to be transferred between applications and devices on a network and is used in the TCP IP model. It is designed to break down a message, such as an email, into packets of data to ensure the message reaches its destination successfully and as quickly as possible.

What does TCP mean?

TCP meaning Transmission Control Protocol, is a communications standard for delivering data and messages through networks. TCP is a basic standard that defines the rules of the internet and is a common protocol used to deliver data in digital network communications.

What is TCP and what are its types?

TCP is a protocol or standard used to ensure data is successfully delivered from one application or device to another. TCP is part of the Transmission Control Protocol/Internet Protocol (TCP/IP), which is a suite of protocols originally developed by the U.S. Department of Defense to support the construction of the internet. The TCP/IP model consists of several types of protocols, including TCP and IP, Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), Reverse Address Resolution Protocol (RARP), and User Datagram Protocol (UDP).

TCP is the most commonly used of these protocols and accounts for the most traffic used on a TCP/IP network. UDP is an alternative to TCP that does not provide error correction, is less reliable, and has less overhead, which makes it ideal for streaming.

WHAT IS USER DATAGRAM PROTOCOL (UDP)

User datagram protocol (UDP) operates on top of the Internet Protocol (IP) to transmit datagrams over a network. UDP does not require the source and destination to establish a three-way handshake before transmission takes place. Additionally, there is no need for an end-to-end connection.

Since UDP avoids the overhead associated with connections, error checks and the retransmission of missing data, it's suitable for real-time or high performance applications that don't require data verification or correction. If verification is needed, it can be performed at the application layer.

UDP is commonly used for Remote Procedure Call (RPC) applications, although RPC can also run on top of TCP. RPC applications need to be aware they are running on UDP, and must then implement their own reliability mechanisms.

The benefits and downsides of UDP

UDP has a number of benefits for different types of applications, including:

- **No retransmission delays** – UDP is suitable for time-sensitive applications that can't afford retransmission delays for dropped packets. Examples include Voice over IP (VoIP), online games, and media streaming.
- **Speed** – UDP's speed makes it useful for query-response protocols such as DNS, in which data packets are small and transactional.
- **Suitable for broadcasts** – UDP's lack of end-to-end communication makes it suitable for broadcasts, in which transmitted data packets are addressed as receivable by all devices on the internet. UDP broadcasts can be received by large numbers of clients without server-side overhead.

At the same time, UDP's lack of connection requirements and data verification can create a number of issues when transmitting packets. These include:

- No guaranteed ordering of packets.
- No verification of the readiness of the computer receiving the message.
- No protection against duplicate packets.
- No guarantee the destination will receive all transmitted bytes. UDP, however, does provide a checksum to verify individual packet integrity.

UDP header packet structure

UDP wraps datagrams with a UDP header, which contains four fields totaling eight bytes.

The fields in a UDP header are:

- **Source port** – The port of the device sending the data. This field can be set to zero if the destination computer doesn't need to reply to the sender.
- **Destination port** – The port of the device receiving the data. UDP port numbers can be between 0 and 65,535.
- **Length** – Specifies the number of bytes comprising the UDP header and the UDP payload data. The limit for the UDP length field is determined by the underlying IP protocol used to transmit the data.
- **Checksum** – The checksum allows the receiving device to verify the integrity of the packet header and payload. It is optional in IPv4 but was made mandatory in IPv6.

UDP DDoS threats and vulnerabilities

UDP's lack of a verification mechanism and end-to-end connections makes it vulnerable to a number of DDoS attacks. Attackers can spoof packets with arbitrary IP addresses, and reach the application directly with those packets.

This is in contrast to TCP, in which a sender must receive packets back from the receiver before communication can start.

UDP specific DDoS attacks include:

- **UDP Flood**

A UDP flood involves large volumes of spoofed UDP packets being sent to multiple ports on a single server, knowing that there is no way to verify the real source of the packets. The server responds to all the requests with ICMP 'Destination Unreachable' messages, overwhelming its resources.

In addition to the traditional UDP flood, DDoS perpetrators often stage generic network layer attacks by sending mass amounts of fake UDP packets to create network congestion. These attacks can only be mitigated by scaling up a network's resources on demand, as is done when using a cloud DDoS mitigation solution.

- **DNS Amplification**

A DNS amplification attack involves a perpetrator sending UDP packets with a spoofed IP address, which corresponds to the IP of the victim, to its DNS resolvers. The DNS resolvers then send their response to the victim. The attack is crafted such that the DNS response is much larger than the original request, which creates amplification of the original attack.

When done on a large scale with many clients and multiple DNS resolvers, it can overwhelm the target system. A DDoS attack with capacity of 27Gbps can be amplified to as much as 300Gbps using amplification.

- **UDP Port Scan**

Attackers send UDP packets to ports on a server to determine which ports are open. If a server responds with an ICMP 'Destination Unreachable' message, the port is not open. If there is no such response, the attacker infers that the port is open, and then use this information to plan an attack on the system.

HTTP

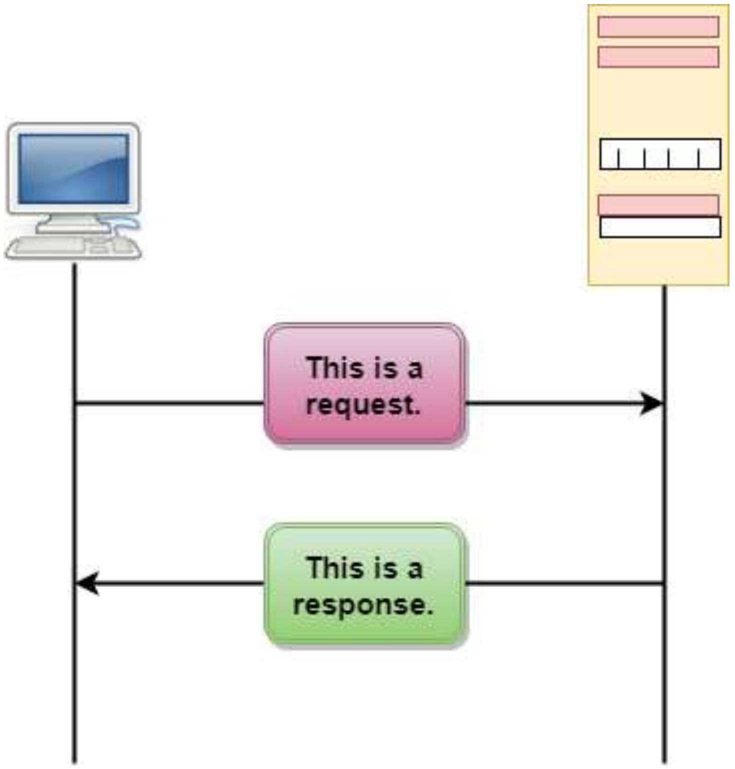
- HTTP stands for **HyperText Transfer Protocol**.
- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- This protocol is known as HyperText Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.
- HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files.
- HTTP is used to carry the data in the form of MIME-like format.

- HTTP is similar to SMTP as the data is transferred between client and server. The HTTP differs from the SMTP in the way the messages are sent from the client to the server and from server to the client. SMTP messages are stored and forwarded while HTTP messages are delivered immediately.

Features of HTTP:

- **Connectionless protocol:** HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. The connection between client and server exist only during the current request and response time only.
- **Media independent:** HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.
- **Stateless:** HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

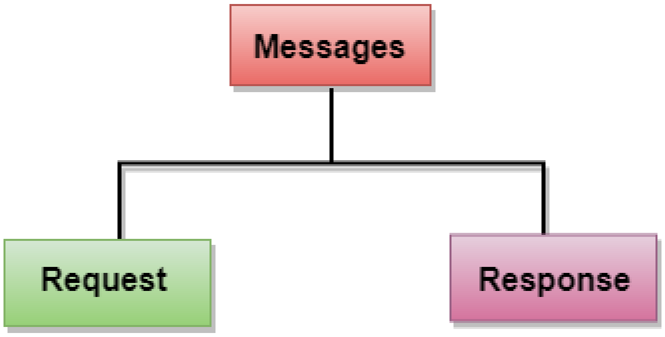
HTTP Transactions



The above figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.

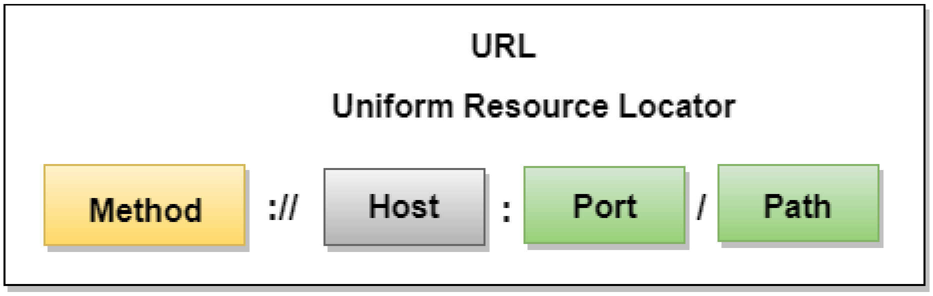
Messages

HTTP messages are of two types: request and response. Both the message types follow the same message format.



UNIFORM RESOURCE LOCATOR (URL)

- A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL).
- The Uniform Resource Locator (URL) is a standard way of specifying any kind of information on the internet.
- The URL defines four parts: method, host computer, port, and path.



- **Method:** The method is the protocol used to retrieve the document from a server. For example, HTTP.
- **Host:** The host is the computer where the information is stored, and the computer is given an alias name. Web pages are mainly stored in the computers and the computers are given an alias name that begins with the characters "www". This field is not mandatory.
- **Port:** The URL can also contain the port number of the server, but it's an optional field. If the port number is included, then it must come between the host and path and it should be separated from the host by a colon.
- **Path:** Path is the pathname of the file where the information is stored. The path itself contain slashes that separate the directories from the subdirectories and files.

WHAT IS FILE TRANSFER PROTOCOL (FTP) AND WHAT IS IT USED FOR?

What Is File Transfer Protocol (FTP)?
The term file transfer protocol (FTP) refers to a process that involves the transfer of files between devices over a network. The process works when one party allows another to send or receive files over the Internet. Originally used as a way for users to communicate and exchange information between two physical devices, it is now commonly used to store files in the cloud, which is usually a secure location that is held remotely.

FTP may be used by a business or individual to transfer files from one computer system to another or by websites to upload or download files from their servers.

KEY TAKEAWAYS

- File transfer protocol (FTP) is a way to download, upload, and transfer files from one location to another on the Internet and between computer systems.
- FTP enables the transfer of files back and forth between computers or through the cloud.
- Users require an Internet connection in order to execute FTP transfers.
- FTP is an essential tool for those who build and maintain websites.
- Many FTP clients are free to download, although most websites already have the FTP built-in.

How File Transfer Protocol (FTP) Works

File transfer protocol allows individuals and businesses to share electronic files with others without having to be in the same space. This can be done using an FTP client or through the cloud. Regardless of the option, both parties require a working Internet connection.[1]

Most web browsers come with FTP clients that enable users to transfer files from their computer to a server and vice versa. Some users may want to use a third-party FTP client because many of them offer extra features. Examples of FTP clients that are free to download include FileZilla Client, FTP Voyager, WinSCP, CoffeeCup Free FTP, and Core FTP.

Many people have used FTP before without even realizing it. If you have ever downloaded a file from a web page, you've used FTP. The first step is to log in, which may occur automatically or by manually inputting a username and password. FTP will also require you to access an FTP server through a specific port number. Once you access the FTP server through your FTP client, you can now transfer files. Not all public FTP servers require you to sign in because some servers enable you to access them anonymously.

As noted above, FTP was originally developed as a way to send and receive files between two physical computers. But with changes in technology, users can execute file transfers through the cloud. Using the cloud allows transfers to be done conveniently and safely (which could protect individuals and companies from data breaches), and at relatively low cost.[2]

FTP Process

The FTP process can be broken down into just a couple of key steps.

- First, a user logins to an FTP server (although a login might not be required).
- The FTP client interacts with the server upon a request, which is the second step.
- With FTP, a user can then upload, download, or move files on the server.

The term FTP client refers to the software that allows you to transfer files to another party.

History of FTP

File Transfer Protocol (FTP) was first described in a white paper in 1971 by then MIT graduate student Abhay Bhushan.[3] The aim was to allow the transfer of data files over the ARPANET, the early precursor to the modern Internet.

The original protocol has undergone several revisions and upgrades since the 1980s to improve its speed, fidelity, and security.

Types of FTP

There are various types of FTPs, including anonymous and password protected. Anonymous allows the transfer of data without encryption or using a password. This is good for files that can be distributed without restrictions.

Meanwhile, password-protected FTP uses a username and password to access the files. FTP secure (FTPS) offers increased security when transferring, allowing for implicit transport layer security (TLS). FTP can also employ explicit TLS, which upgrades the connection to an encrypted connection for added security.

Other Protocols

File transfer protocol is one of many different protocols that dictate how computers and computing systems behave on the Internet. Other such protocols include the following:

- **Hypertext Transfer Protocol (HTTP):** Designed to transmit data across the web [4]
- **Internet Message Access Protocol (IMAP):** Provides access to bulletin board or email messages from a shared service[5]
- **Network Time Protocol (NTP):** Synchronizes clock times on computers over a network[6]

FTP enables computers on the Internet to transfer files back and forth. As such, it is an essential tool for those building and maintaining websites today.

Benefits and Uses of FTP

FTP made handling data across the Internet much easier and intuitive. Without FTP and its later iterations, we would not be able to easily stream video content, use video calls, play online games, share files, or enjoy cloud storage.

Today, FTP operates behind the scenes as a backbone for data transfer from servers around the world to millions of clients every second of every day.

Example of FTP Clients

FTP software is relatively straightforward to set up. FileZilla is a free, downloadable FTP client. Other examples of FTP clients include Transmit, WinSCP, and WS_FTP.

You type in the address of the server you wish to access, the port, and the password for accessing the server. Once access has been granted, the user's files on their local system as well as the accessed server will be visible.

The user can download files from the server to the local system, or upload files from the local system to the server. They can also make changes to files on the server, as long as they have the proper authorization to do so.

WHAT IS DHCP AND HOW DOES IT WORK?

DHCP is an under-the-covers mechanism that automates the assignment of IP addresses to fixed and mobile hosts that are connected wired or wirelessly.

When a device wants access to a network that's using DHCP, it sends a request for an IP address that is picked up by a DHCP server. The server responds be delivering an IP address to the device, then monitors the use of the address and takes it back after a specified time or when the device shuts down. The IP address is then returned to the pool of addresses managed by the DHCP server to be reassigned to another device as it seeks access to the network.

[Related: What is IPv6, and why aren't we there yet?

While the delegation of IP addresses is the central function of the protocol, DHCP also assigns a variety of related networking parameters including subnet mask, default gateway address, and domain name server (DNS). DHCP is an IEEE standard built on top of the older BOOTP (bootstrap protocol), which has become obsolete because it only works on IPv4 networks.

Benefits of DHCP

DHCP provides a range of benefits to network administrators:

Reliable IP address configuration
You can't have two users with the same IP address because it would create a conflict where one or both devices could not connect to the network. DHCP eliminates human error so that address conflicts, configuration errors, or simple typos are minimized.

Reduced network administration
DHCP provides centralized and automated TCP/IP configuration. By deploying a DHCP relay agent, a DHCP server is not needed on every subnet.

['IT has a new 'It Crowd': Join the CIO Tech Talk Community]

Mobility
DHCP efficiently handles IP address changes for users on portable devices who move to different locations on wired or wireless networks.

IP address optimization
DHCP not only assigns addresses, it automatically takes them back and returns them to the pool when they are no longer being used.

Efficient change management
DHCP makes it simple for an organization to change its IP address scheme from one range of addresses to another. DHCP enables network administrators to make those changes without disrupting end users.

DHCP components

When working with DHCP, it's important to understand all of its components. Below is a list of them and what they do:

DHCP server
This is a networked device running the DCHP service that holds IP addresses and related configuration information. This is most typically a server or a router but could be anything that acts as a host, such as an SD-WAN appliance.

DHCP client
This endpoint endpoint software requests and receives configuration information from a DHCP server. This can be installed on a computer, mobile device, IoT endpoint or anything else that requires connectivity to the network. Most are configured to receive DHCP information by default.

IP address pool
The range of IP addresses that are available to DHCP clients is the IP address. Addresses are typically handed out sequentially from lowest to highest.

Subnet
IP networks can be partitioned into segments known as subnets. Subnets help keep networks manageable.

Lease
The length of time for which a DHCP client holds the IP address information is known as the lease. When a lease expires, the client must renew it.

DHCP relay
A router or host that listens for client messages being broadcast on that network and then forwards them to a configured server is the DHCP relay. The server then sends responses back to the relay agent that passes them along to the client. This can be used to centralize DHCP servers instead of having a server on each subnet.

Assigning IP addresses

The existential question associated with DHCP is how does an end user connect to the network in the first place without having an IP address?

The answer is that there’s a complex system of back-and-forth requests and acknowledgments. First, all modern device operating systems include a DHCP client, which is typically enabled by default. In order to request an IP address, the client device sends out a broadcast message—DHCPDISCOVER. The network directs that request to the appropriate DHCP server.

DHCP server functionality is typically assigned to a physical server plus a backup. Other devices can also act as DHCP servers, such as SD-WAN appliances or wireless access points.

The server then determines the appropriate IP address and sends an OFFER packet to the client, which responds with a REQUEST packet. In the final step in the process, the server sends an ACK packet confirming that the client has been given an IP address.

This is all done quickly and automatically and without the need for the end user to take any action. The catch is that the IP address isn’t permanent. It’s only good for a specified period of time, known as the lease time.

Controlling lease time

If all DHCP did was assign IP addresses permanently, it wouldn’t be dynamic, it would be static. Static addresses are appropriate for some devices, such as network printers. However, under the DHCP protocol, every time the DHCP server assigns an address there is an associated lease time. When the lease expires, the client can no longer use the IP address and is essentially kicked off the network.

The protocol is designed so active clients automatically contact the DHCP server halfway through the lease period to renew the lease. If the server doesn’t respond immediately, the client continues to ask the DHCP server for a lease renewal until it is approved.

Typically, when a host shuts down, the lease is automatically terminated, in order to free up its IP address so it can be used by another client on the network.

DHCP networking functionality

In addition to providing the client with the ability to connect to network and internet resources through the IP address, the DHCP server assigns additional networking parameters that provide efficiency and security. These include:

Default gateway
This gateway is responsible for transferring data back and forth between the local network and Internet, or between local subnets.

Subnet mask
IP networking uses a subnet mask for separate the host address and the network address portions of an IP address.

DNS server
Translates domain names (networkworld.com) into IP addresses, which are represented by long strings of numbers.

Scopes and user classes of IP addresses

DHCP assigns addresses dynamically, but not randomly. Since DHCP connects hosts to the network and also assigns networking parameters, there are scenarios in which a network administrator might want to assign certain sets of subnet parameters to specific groups of users.

A scope is a consecutive range of IP addresses that a DHCP server can draw on to fulfill an IP address request from a DHCP client. By defining one or more scopes on the DHCP server, the server can manage the distribution and assignment of IP addresses to DHCP clients. Under the DHCP protocol, network admins can set unlimited numbers of scopes, as needed.

A class is a subset of a scope. Classes are useful if the network administrator wants to separate groups of devices to one segment of a larger scope. For example, SD-WAN clients for employees working remotely.

DHCP security concerns

With DHCP, the initial assignment of an IP address is designed to be fast and efficient. The tradeoff is that the DHCP protocol doesn't require authentication. Of course, enterprises have set up strong authentication requirements for users to access resources once they are on the network, but that still leaves the DHCP server itself as a weak link in the security chain.

An attacker could take over or spoof the DHCP server and hand out bad information to legitimate end users, sending them to a fake site. Or it could hand out legitimate IP addresses to unauthorized users. This could lead to man-in-the-middle attacks and denial of service attacks.

The DHCP specification does address some of these issues. There is a relay-agent information option that enables network engineers to tag DHCP messages as they arrive. This tag can be used to control network access.