Internet Protocols

by Sophia

WHAT'S COVERED

In this lesson, you will learn about the various protocols, what they do, and how they might impact web development work. You will also learn about the internal components of the HyperText Transfer Protocol Secure Protocol (HTTPS).

Specifically, this lesson will cover the following:

1. Protocols of the Internet

Protocols are crucial to the everyday operations of the Internet and the World Wide Web. As a web developer, it is helpful to be aware of the various protocols, what they do, and how they might impact our work. Since protocols can also have an impact when a website project is made available on the World Wide Web, it is helpful to understand which layers of the OSI Model the different protocols align with. This can help with identifying what protocol is causing problems and systematically troubleshooting the issues by examining the protocol stack.

1a. Internet Layer Protocols: IP

The **Internet Protocol (IP)** is a network protocol used in computer networks. It is responsible for addressing and routing network packets to the correct location.

The Internet Protocol’s primary role is two-fold:

1. Provide unique identifiers for networks and hosts.
2. Provide routing capabilities (the ability to communicate across multiple interconnected networks).

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

KEY CONCEPT

In fact, every system that directly connects to the Internet must possess a valid IP address that is unique in the world, whether the device is a network router, a web server, or a database, a unique IP address is required.

WATCH

View this video to review the difference between IPv4 and IPv6 IP addresses.

There are two current versions of the IP protocol, thus two different types of IP addresses; Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6). While almost everything is capable of using IPv4 addresses on local networks and the Internet, more systems and devices are becoming IPv6 compatible as well. The goal of the ICANN (Internet Corporation for Assigned Names and Numbers), the non-profit organization responsible for maintaining assigned IP addresses, is to move everything to IPv6 due to the larger address capacity and improved security and features.

IP addresses identify the network ID that the system is connected to as well as the unique host ID of that system. The second purpose of IP addresses and the IP protocol is to allow network devices, called **routers**, to forward data packets from one network to the next until the packet reaches the network containing the destination system. This process is called routing and allows devices and systems to communicate with other devices and systems that are outside of the local network.

IN CONTEXT

The IP addresses of the source system and the destination system act like your home address and the address of your friend who lives in another state. When you want to send a letter to your friend using a postal service, the postal offices act like a network router. The address's state and ZIP, much like the network ID of the IP address, tell the postal service which state and region to forward the letter to—just like a router has to decide which network to forward the packet to. The difference is that a letter goes from one postal office to the next via a single airplane flight to the distributing postal office in the region where your friend lives. Data packets, on the other hand, are continuously forwarded from one network to the next, each time getting the packet one step closer to the destination network.

TRY IT

There are multiple ways to discover your computer's current IP address. The simplest method on Windows is to use a command prompt. For Mac users, it is easier to use the system Preferences menu.

Directions: Based on your operating system, follow the instructions below to discover your IP address using a command prompt of the terminal:

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

Windows Operating System +

Follow these steps:

1. Click on the start menu or press the Windows key on your keyboard.
2. Type the letters “CMD” and you should see the Command Prompt appear in the menu. Press Enter.
3. In the command prompt window that appears, type in the command “ipconfig” and press Enter.
4. In the list of information printed to the Command Prompt, locate the entry labeled: “IPv4 Address”. This is your computer’s IPv4 address.

You may also find multiple entries labeled “IPv6 Address”. If you see these and they are blank, this means that your computer and your network are using both versions of IP.

Mac Operating System +

Follow these steps:

1. Click on the Apple menu on the top toolbar.
2. Click on System Settings…
3. Locate and click on the Network icon.
4. Select the “connected” network adapter and click the "details" button.
5. You should see the label “IP Address”. This is your IPv4 address.

You may also find the IPv6 Address if you scroll down. Again, this means that your computer and your network are using both versions of IP.

More details about IP addresses will be provided in a future lesson.

TERMS TO KNOW

Internet Protocol (IP)
Part of the TCP/IP protocol suite, Internet Protocol is responsible for addressing computer devices for the purpose of identifying devices connected to a network and locating devices that are connected to remote networks.

Network Router
Device connected to multiple networks that forwards packets onto other networks based on its configured routing rules.

1b. Transport Layer Protocols: TCP and UDP

The Transport Layer Protocols consist of TCP and UDP. As you may recall, TCP played a critical role in the early development of the Internet. Now, let’s take a closer look at the Transport Layer Protocols and the differences

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

between TCP and UDP protocols.

TCP is a specific type of communication protocol, called a **connection-oriented protocol**, that provides reliable, ordered, and error-checked delivery of data packets over IP networks. It establishes a connection between two endpoints (sender and receiver) before transmitting data across a consistent pathway. TCP is designed for data transmissions that require accuracy and that must arrive in order and uncorrupted. File transfers and database access require an exact transmission of the entire data set.

EXAMPLE

HTTP natively runs on top of the TCP protocol since every character of a webpage must arrive as is without modification, corruption, or data loss.

User Datagram Protocol (UDP) is a **connectionless** communication protocol that offers a lightweight and low-overhead alternative to TCP. It provides a best-effort delivery of data packets without guaranteeing reliability or ordering. As a result, lost or damaged packets are not retransmitted, and the origin system does not wait for acknowledgment of receipt before sending the next packet.

KEY CONCEPT

When uploading a file to a website, transferring data from a database, or performing any business transactions online, TCP is the method required to ensure the successful and accurate transmission of critical information. On the other hand, UDP was designed for fault-tolerant applications such as audio and video streaming including web conferencing and Internet phone calls, wherein a small loss of data or data corruption would not have a detrimental impact on the user’s overall experience.

TERMS TO KNOW

Connection-Oriented Protocol

A type of protocol that establishes a connection between two endpoints (sender and receiver) before transmitting data across a consistent pathway.

User Datagram Protocol (UDP)

Similar communication method to HTTP but differs in how it communicates by avoiding error checking and correction in favor of improved transmission.

Connectionless Protocol

A type of protocol that provides a best-effort delivery of data packets without guaranteeing reliability or ordering.

1c. Application Layer Protocols: FTP, SMTP, DNS

Protocols that reside on the upper layers of the OSI Model are more relevant to web developers. The **FTP (file transfer protocol)** for example, is a standard network protocol used for transferring files between a client and a server over the Internet. It provides a reliable and efficient means of file transfer as it relies on TCP and is often used by web developers to easily upload files to a web server.

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

DNS (domain name service) is a protocol and distributed server that translates domain names into the IP address of the associated server. This service is helpful because it is typically easier for us to remember a name than a string of numbers. **Name servers** host **DNS records** for individual domains. Each DNS record contains entries that tie specific domain names and subdomains to a specific IP address.

EXAMPLE

DNS makes it possible for you to reach Google's website by typing in either "google.com" or "142.251.46.206" into your web browser's address or search bar.

DNS records have a number of different types of entries, but the basic types are **A** and **AAAA entries** as well as **CNAME** and **DNAME** entries.

KEY CONCEPT

When dealing with DNS records, it is important to consider how changes to a record will spread to other connected DNS servers. When a DNS record is created or updated, other DNS servers have to be made aware of the change. This process can take anywhere from 24 hours (usually less) up to a week to make it through the Internet. As a result, you may receive inconsistent feedback regarding whether a site has gone live or not just after updating a DNS.

The **SMTP (simple mail transfer protocol)** is a protocol for sending and receiving emails and is commonly implemented on web applications and sites to easily send confirmation and notification emails. While SMTP is still commonly used, more elegant mail applications use the modern IMAP (Internet message access protocol) which allows users to connect to and manage their email accounts. Many of the common **server-side scripting** languages include a native implementation of SMTP and can easily send an email.

TERMS TO KNOW

FTP (File Transfer Protocol)
An alternative communication method to HTTP and part of the Internet Protocol suite, this transmission standard focuses on moving files between computers.

Domain Name Service (DNS)
An architecture of systems and protocols that link human-friendly addresses called domains to Internet Protocol (IP) addresses of web servers.

Name Servers
Host DNS records for individual domains and each record contains entries that tie specific domain names and subdomains to IP addresses.

DNS Records
Each record pertains to one domain name and contains entries that tie the domain and subdomains to IP addresses and other domains.

A Record
A DNS entry that associates a domain name or subdomain to an IPv4 address.

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

AAAA Record
A DNS entry that associates a domain name or subdomain to an IPv6 address.

CNAME
A DNS entry that creates a subdomain and associates it to another domain or subdomain.

DNAME
A DNS entry that maps a subdomain to another external domain.

SMTP (Simple Mail Transfer Protocol)
A protocol for sending and receiving emails.

Server-Side Scripting
Programming languages that reside on and are executed by the server prior to being communicated to the client.

1d. HTTP

You learned in a previous lesson that of all the various protocols web developers need to be aware of and familiar with, HTTP is likely the most important. When an HTTP Client wants to access a webpage or resource, it sends a request to a web server using TCP. The web server listens for incoming requests and handles them by responding with either the resource or an error message.

Let’s revisit the important internal components HTTP contains:

HTTP Component	Description
Requests	HTTP clients are typically browsers that can create and transmit an **HTTP request**. Request objects contain a method, the Uniform Resource Identifier (URI), headers, and an optional body. The method indicates the type of request being made. The URI points to a specific resource on a specific domain. Headers convey additional information about the request which could include authentication data, content type, and more.
Response	Once a server receives an HTTP request, it then processes the request based on its method and generates a response containing the requested resource and a status code. The status code is a 3-digit number indicating the outcome of the request, along with a textual status message. **HTTP responses** also contain any necessary headers, as well as the body containing the actual requested content, whether it be an HTML page, JSON or XML data structures, etc.
Method	On the front end, HTTP request methods are used to specify what kind of operation the client wants the server to perform. These methods include the following: • GET is used to retrieve resources from the server. • POST is used to transmit data to the server, usually to create a new resource. • PUT is used to transmit data to the server to update an existing resource.

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

	• DELETE is used to remove a resource from a server. • PATCH is used to partially modify a resource on the server. On the backend, HTTP request methods are used by web servers to group handler functions together based on the HTTP request.

DID YOU KNOW

GET and POST methods are two options for submitting a web form. GET is faster, but insecure due to the fact that field name and value pairs are transmitted in plain text in the HTTP request URI field, even over secure connections. POST is a little slower but more secure as it places the field name and value pairs in the body, which would be encrypted over secure connections.

When a server responds to an HTTP request, the response contains a 3-digit status code that indicates the outcome of the request. While there are over a hundred different status codes they can easily be categorized by their left-most digit:

- 1xx: indicates the request was received and is currently being processed.
- 2xx: indicates the request was successfully received and accepted.
- 3xx: indicates that additional actions are needed to fulfill the request.
- 4xx: indicates a client error due to bad syntax or could not be fulfilled.
- 5xx: indicates a server error while processing an otherwise valid request.

A final note about the **HTTPS protocol** is that due to the fact it is used for transferring files and data securely using encryption, web developers can opt to utilize an **HTTP management interfaces**, such as Plesk or CPanel, to manage their web server’s configuration as well as upload and download site files and assets. HTTP management interfaces not only inherently employ encryption, but also centralize various utilities needed for managing different aspects of a web server’s resources. Without such an interface, developers must obtain and use a variety of independent applications and tools.

DID YOU KNOW

You may see the protocol HTTPS written in a few different ways including HTTP/S and HTTP(S).

BIG IDEA

Being knowledgeable of these Internet protocols can help to avoid and troubleshoot issues, should they arise. For example, understanding how the DNS protocol operates and being able to edit DNS records properly won’t improve the development of a web application but will help during the implementation process if the site fails to go live on the web as expected. On the other hand, understanding the operations and different request methods of the HTTP protocol is definitely important to a web developer.

TERMS TO KNOW

HTTP Request

A message sent to a server that contains a specified method, a Universal Resource Identifier, headers, and possibly a body of data to request the server perform some type of operation.

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

HTTP Response

A message sent in response to an HTTP Request that contains a status code, brief status message, and the requested content.

HyperText Transfer Protocol Secure Protocol (HTTPS)

A variation of the HTTP protocol that employs encryption to protect the privacy of any information transmitted between the client and server. Also called *HTTP/S* or *HTTP(S)*.

HTTP Management Interfaces

A website designed for managing a web server's various resources, files, plugins, and configurations.

SUMMARY

In this lesson, you learned about various **protocols of the Internet** including **Internet Layer Protocol: IP** for addressing devices on public and private networks and **Transport Layer Protocols: TCP and UDP** for providing reliability or performance for transmissions. You also learned about **Application Layer Protocols: FTP, SMTP, and DNS** that provide different higher-level features and services that are used on the web. Finally, you were introduced to **HTTP**, its internal components, and their meaning and use. Together, these protocols support the Internet and WWW but also help support developers and their projects.

Source: This Tutorial has been adapted from "The Missing Link: An Introduction to Web Development and Programming " by Michael Mendez. Access for free at https://open.umn.edu/opentextbooks/textbooks/the-missing-link-an-introduction-to-web-development-and-programming. License: Creative Commons attribution: CC BY-NC-SA.

TERMS TO KNOW

A Record

A DNS entry that associates a domain name or subdomain to an IPv4 address.

AAAA Record

A DNS entry that associates a domain name or subdomain to an IPv6 address.

CNAME

A DNS entry that creates a subdomain and associates it to another domain or subdomain.

Connection-Oriented Protocol

A type of protocol that establishes a connection between two endpoints (sender and receiver) before transmitting data across a consistent pathway.

Connectionless Protocol

A type of protocol that provides a best-effort delivery of data packets without guaranteeing reliability or ordering.

DNAME

A DNS entry that maps a subdomain to another external domain.

DNS Records

Each record pertains to one domain name and contains entries that tie the domain and subdomains to IP addresses and other domains.

Domain Name Service (DNS)

An architecture of systems and protocols that link human-friendly addresses called domains to Internet Protocol (IP) addresses of web servers.

FTP (File Transfer Protocol)

An alternative communication method to HTTP and part of the Internet Protocol suite, this transmission standard focuses on moving files between computers.

HTTP Management Interfaces

A website designed for managing a web server's various resources, files, plugins, and configurations.

HTTP Request

A message sent to a server that contains a specified method, a Universal Resource Identifier, headers, and possibly a body of data to request the server perform some type of operation.

HTTP Response

A message sent in response to an HTTP Request that contains a status code, brief status message, and the requested content.

HyperText Transfer Protocol Secure Protocol (HTTPS)

A variation of the HTTP protocol that employs encryption to protect the privacy of any information transmitted between the client and server. Also called *HTTP/S* or *HTTP(S)*.

Internet Protocol (IP)

Part of the TCP/IP protocol suite, Internet Protocol is responsible for addressing computer devices for the purpose of identifying devices connected to a network and locating devices that are connected to remote networks.

Name Servers

Host DNS records for individual domains and each record contains entries that tie specific domain names and subdomains to IP addresses.

Network Router

Device connected to multiple networks that forwards packets onto other networks based on its configured routing rules.

SMTP (Simple Mail Transfer Protocol)

A protocol for sending and receiving emails.

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

Server-Side Scripting

Programming languages that reside on and are executed by the server prior to being communicated to the client.

User Datagram Protocol (UDP)

Similar communication method to HTTP but differs in how it communicates by avoiding error checking and correction in favor of improved transmission.

© 2024 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.