Ch 10: Application Layer

Computer Networks Course

BY

Dr. Essam Halim Houssein

Cisco | Networking Academy®
Mind Wide Open™

Chapter 10: Application Layer

Applications, such as web browsers, online gaming, chatting with and emailing friends, enable us to send and receive data with relative ease. Typically we can access and use these applications without knowing how they work. However, for network professionals, it is important to know how an application is able to format, transmit and interpret messages that are sent and received across the network.

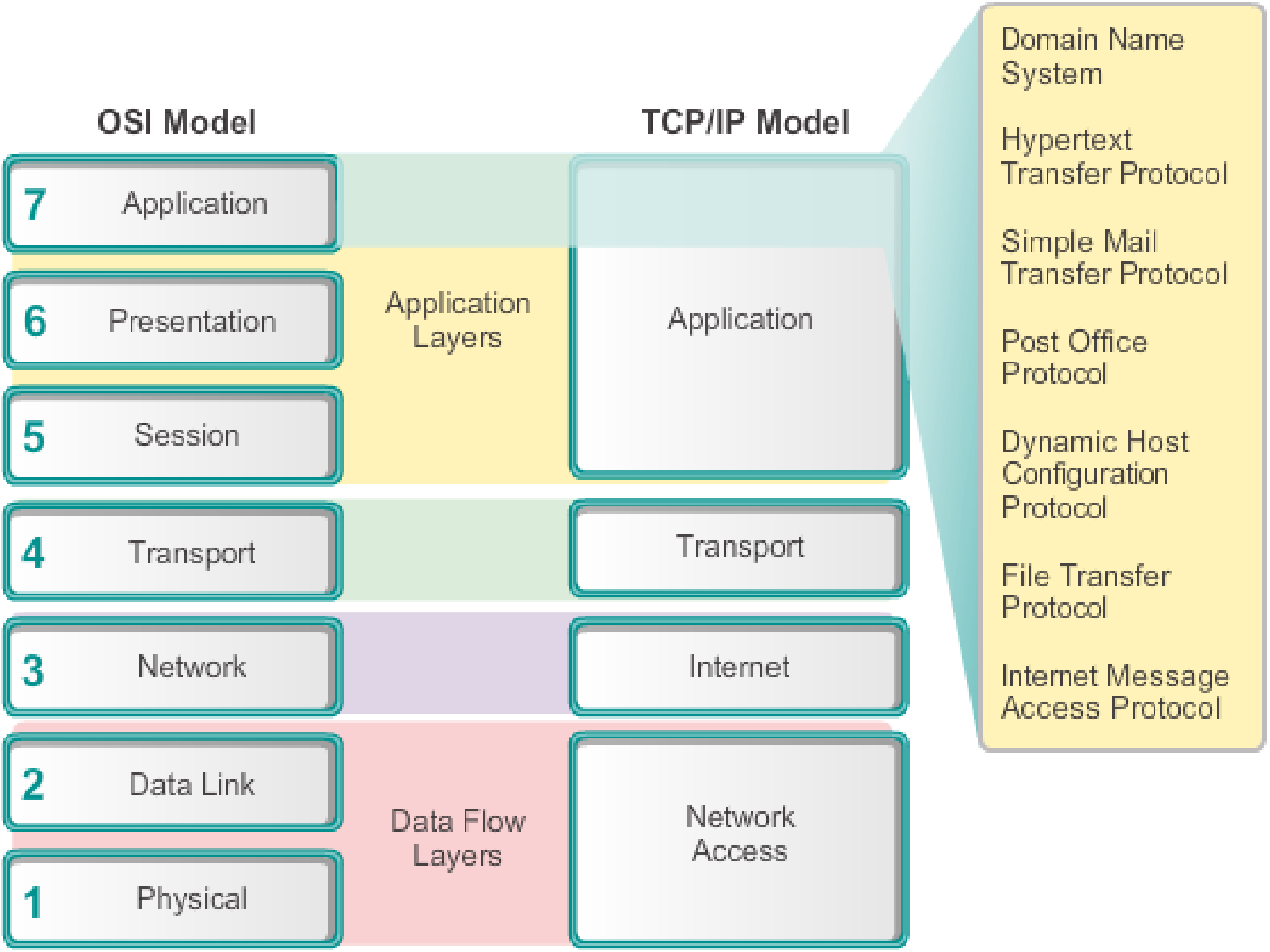
Visualizing the mechanisms that enable communication across the network is made easier if we use the layered framework of the OSI model.

The Application Layer

The application layer is closest to the end user. It is the layer that provides the interface between the applications used to communicate and the underlying network over which messages are transmitted. Application layer protocols are used to exchange data between programs running on the source and destination hosts.

The upper three layers of the OSI model (application, presentation, and session) define functions of the single TCP/IP application layer.

There are many application layer protocols, and new protocols are always being developed. Some of the most widely known application layer protocols include Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), Internet Message Access Protocol (IMAP), and Domain Name System (DNS) protocol.



Presentation and Session Layer

The Presentation Layer

The presentation layer has three primary functions:

Formatting, or presenting, data at the source device into a compatible form for receipt by the destination device

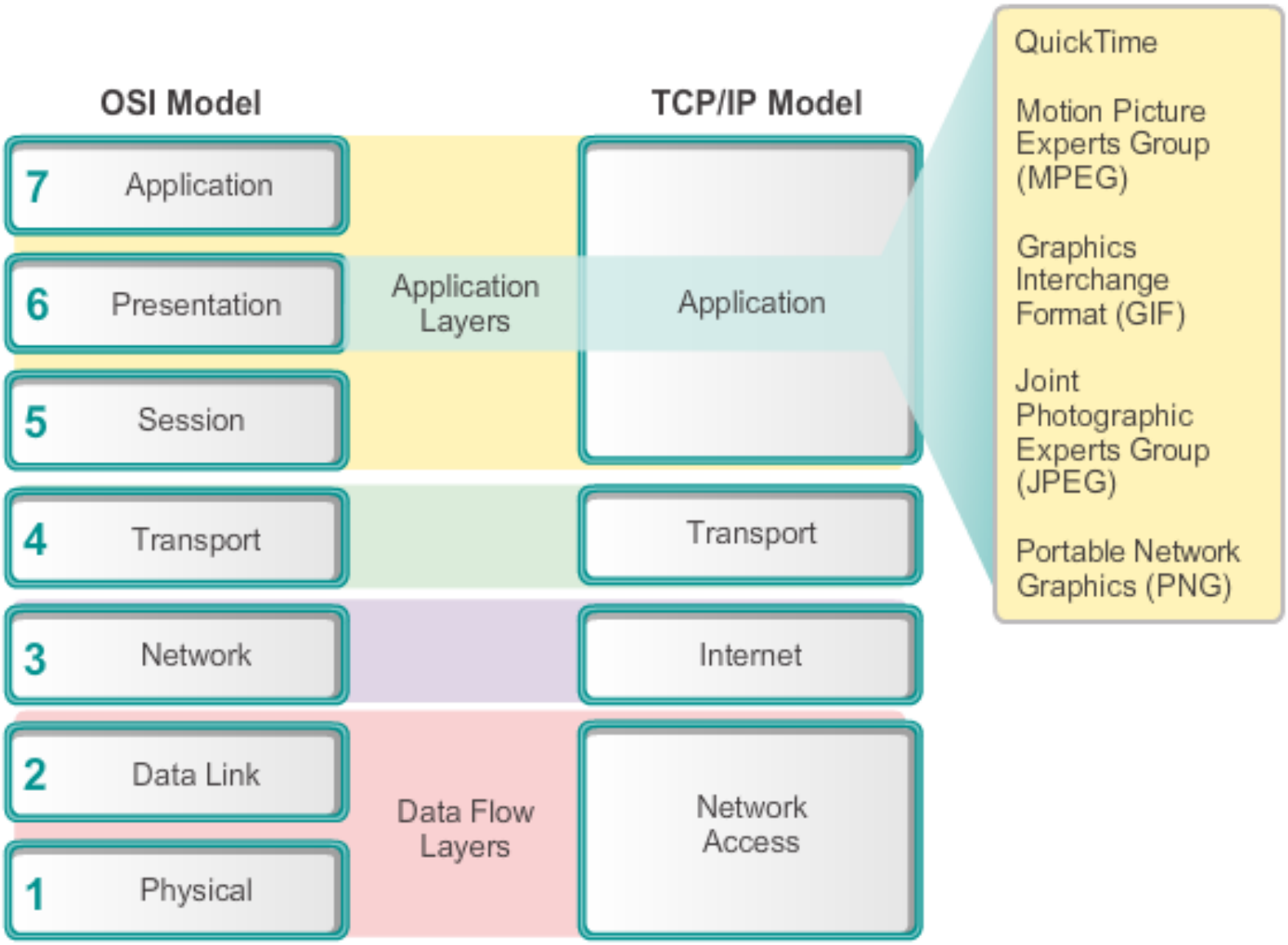
Compressing data in a way that can be decompressed by the destination device

Encrypting data for transmission and decrypting data upon receipt

Some well-known format standards for video include QuickTime and Motion Picture Experts Group (MPEG). Some well-known graphic image formats that are used on networks are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), and Portable Network Graphics (PNG) format.

The Session Layer

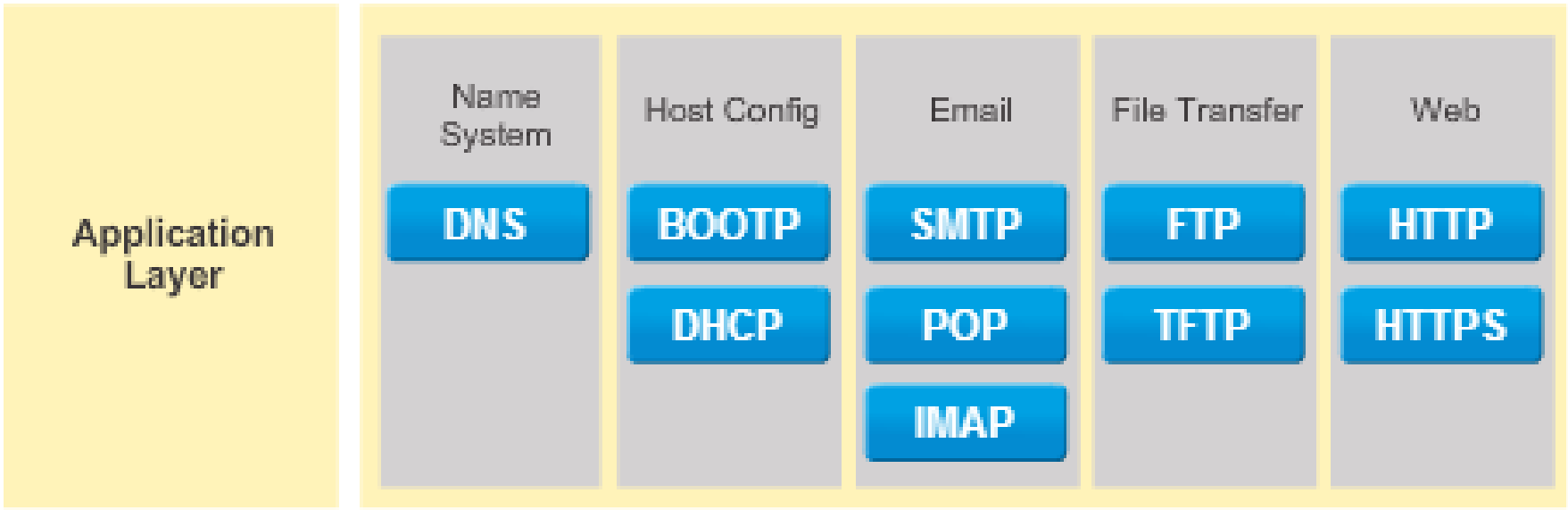
Create and maintain dialogs between source and destination applications. The session layer handles the exchange of information to initiate dialogs, keep them active, and to restart sessions that are disrupted or idle for a long period of time.



TCP/IP Application Layer Protocols

The TCP/IP application protocols specify the **format** and **control** information necessary for many common Internet communication functions.

Application layer protocols are used by both the source and destination devices during a communication session. For the communications to be successful the application layer protocols implemented on the source and destination host must be compatible.

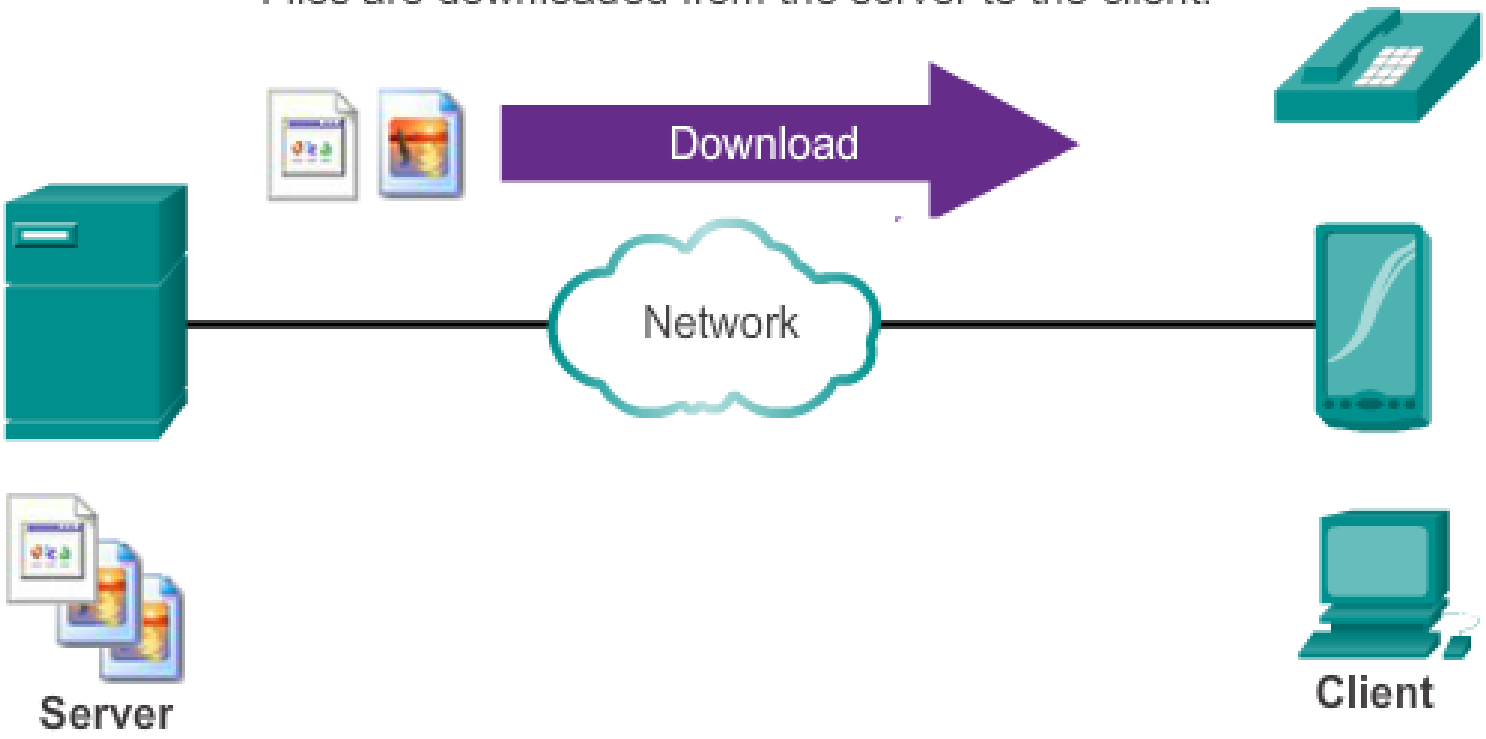


Client-Server Model

The client begins the exchange by requesting data from the server, which responds by sending one or more streams of data to the client. **Application layer protocols describe the format of the requests and responses between clients and servers.** In addition to the actual data transfer, this exchange may also require user authentication and the identification of a data file to be transferred. One example of a client-server network is using an ISP's email service to send, receive and store email. The email client on a home computer issues a request to the ISP's email server for any unread mail. The server responds by sending the requested email to the client.

Client/Server Model

Files are downloaded from the server to the client.



Resources are stored on the server.

A client is a hardware/software combination that people use directly.

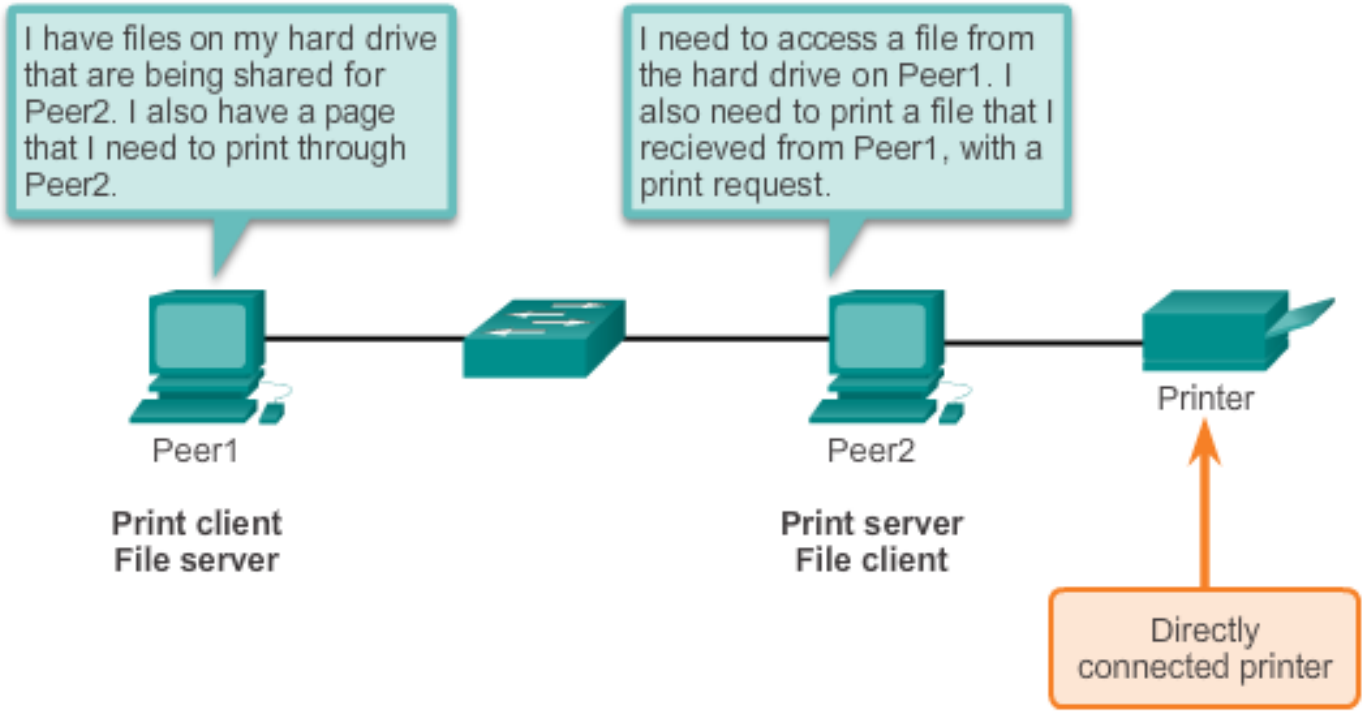
Peer-to-Peer Networks

In P2P networking model, the data is accessed from a peer device without the use of a dedicated server.

The P2P network model involves two parts: **P2P networks and P2P applications.** Both parts have similar features, but in practice work quite differently.

In a P2P network, two or more computers are connected via a network and can share resources without having a dedicated server. Every connected end device can function as both a server and a client.

Peer-to-Peer Networking



In a peer-to-peer exchange, both devices are considered equal in the communication process.

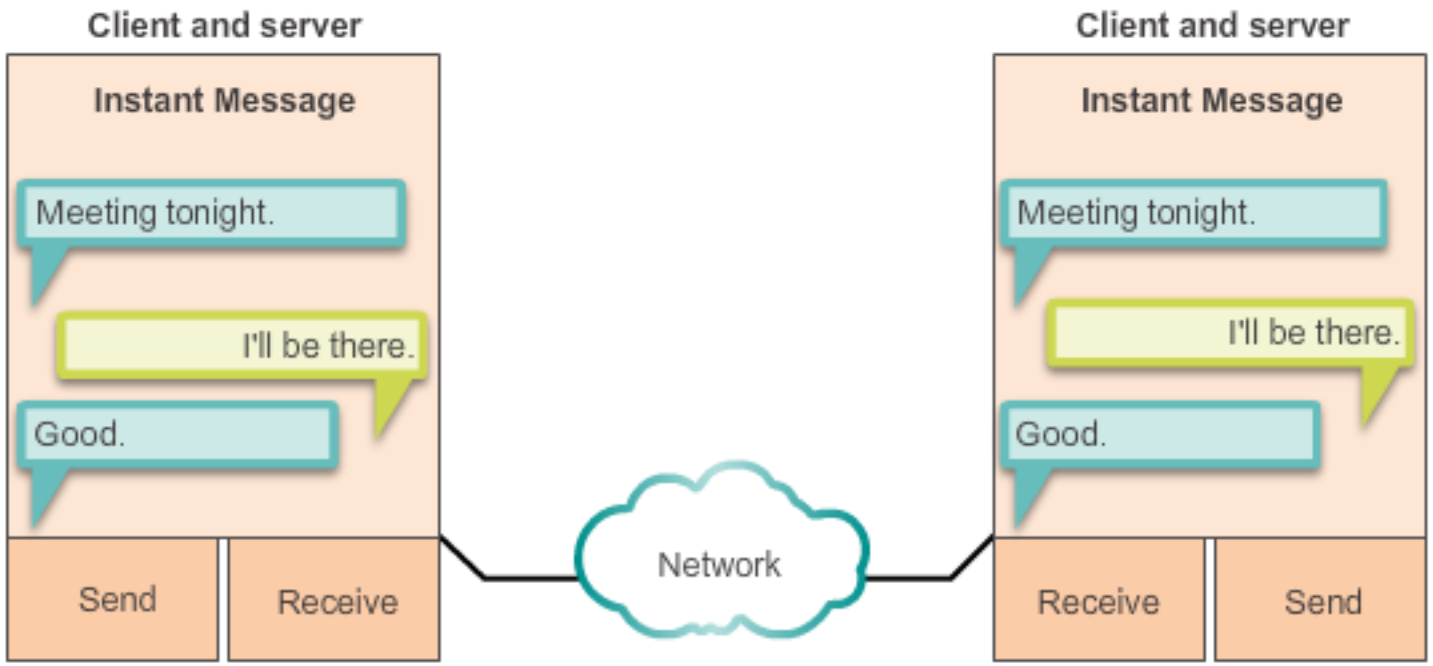
Peer-to-Peer Applications

A P2P application allows a device to act as both a client and a server within the same communication. In this model, every client is a server and every server a client. P2P applications require that each end device provide a user interface and run a background service.

Some P2P applications use a hybrid system where resource sharing is decentralized, but the indexes that point to resource locations are stored in a centralized directory. In a hybrid system, each peer accesses an index server to get the location of a resource stored on another peer.

Peer-to-Peer Applications

Client and server in the same communication



Both clients simultaneously
- Initiate a message
- Receive a message

Hypertext Transfer Protocol and Hypertext Markup Language

When a web address or uniform resource locator (URL) is typed into a web browser, the web browser establishes a connection to the web service running on the server using the HTTP protocol. URLs and Uniform Resource Identifier (URIs) are the names most people associate with web addresses.

To better understand how the web browser and web server interact, we can examine how a web page is opened in a browser. For this example, use the http://www.cisco.com/index.html URL.

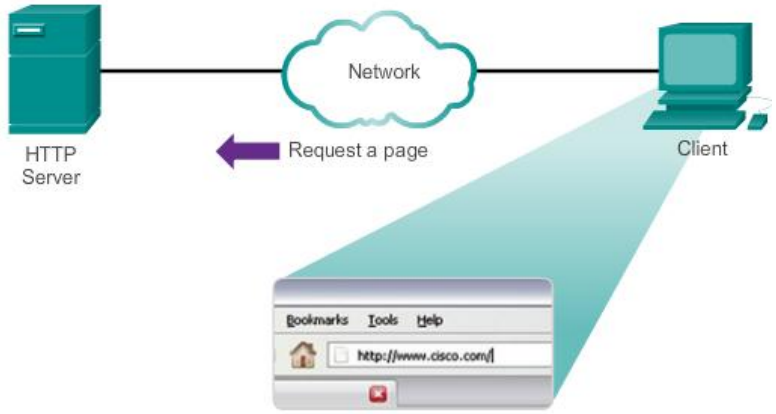
First, the browser interprets the three parts of the URL:

1. **http** (the protocol or scheme)
2. **www.cisco.com** (the server name)
3. **index.html** (the specific filename requested)

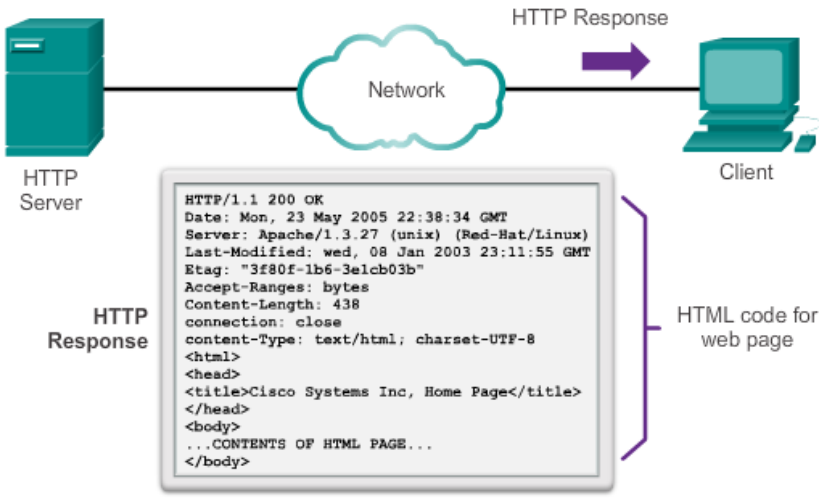
Then, the browser checks with a name server to convert www.cisco.com into a numeric address, which it uses to connect to the server. Using HTTP requirements, the browser sends a GET request to the server and asks for the **index.html** file. The server, sends the HTML code for this web page to the browser. Finally, the browser deciphers the HTML code and formats the page for the browser window.

HTTP Protocol

HTTP Protocol Step 1



HTTP Protocol Step 2



HTTP Protocol Step 3

In response to the request, the HTTP server returns code for a web page.

HTTP and HTTPS

HTTP is a request/response protocol. When a client, typically a web browser, sends a request to a web server, HTTP specifies the message types used for that communication. The three common message types are GET, POST, and PUT:

GET - A client request for data. A client (web browser) sends the GET message to the web server to request HTML pages.

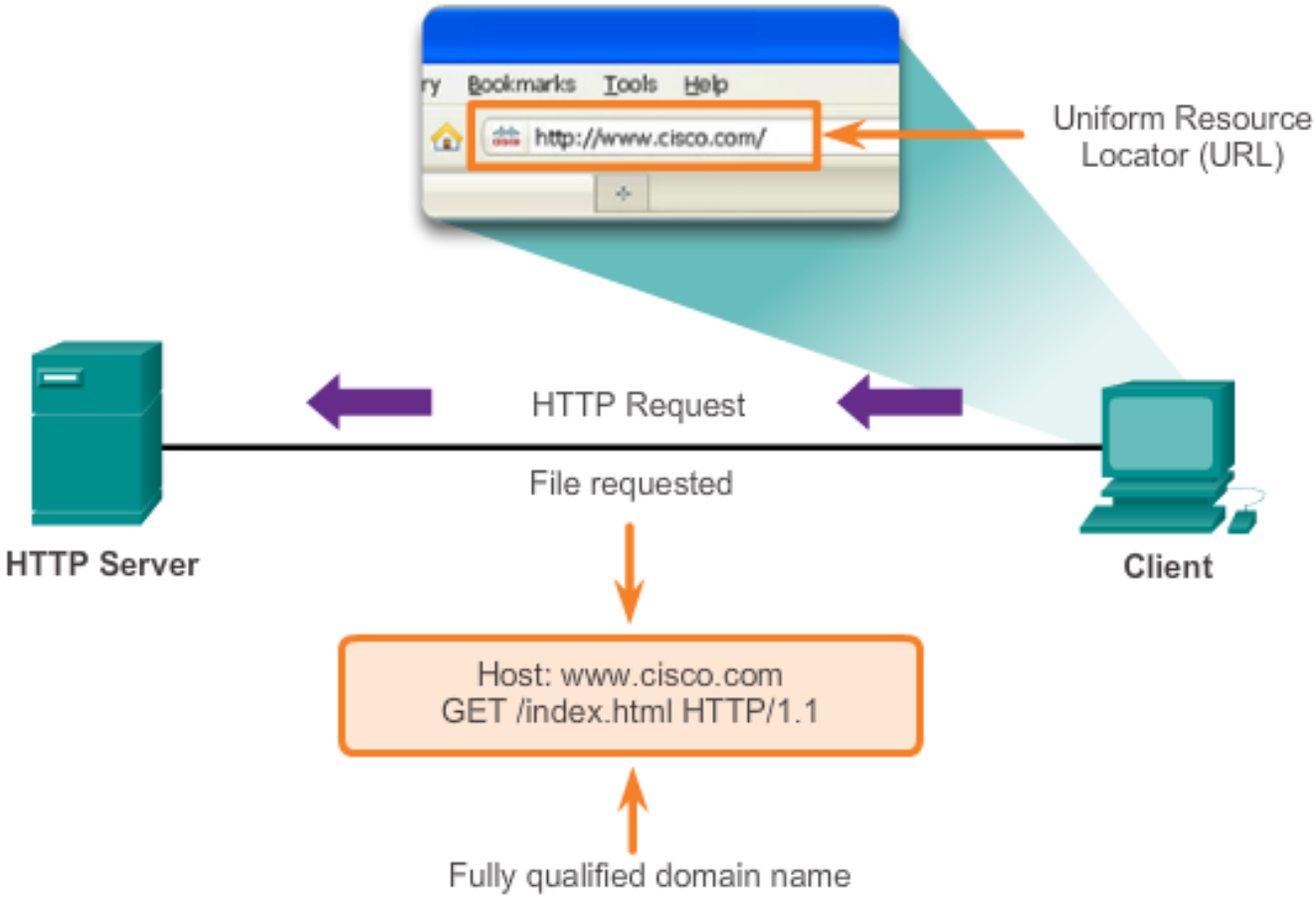
POST - Uploads data files to the web server such as form data.

PUT - Uploads resources or content to the web server such as an image.

Although HTTP is remarkably flexible, it is not a secure protocol. The request messages send information to the server in plain text that can be intercepted and read. The server responses, typically HTML pages, are also unencrypted.

For secure communication across the Internet, the HTTP Secure (HTTPS) protocol is used. HTTPS uses authentication and encryption to secure data as it travels between the client and server. HTTPS uses the same client request-server response process as HTTP, but the data stream is encrypted with Secure Socket Layer (SSL) before being transported across the network.

HTTP Protocol using GET



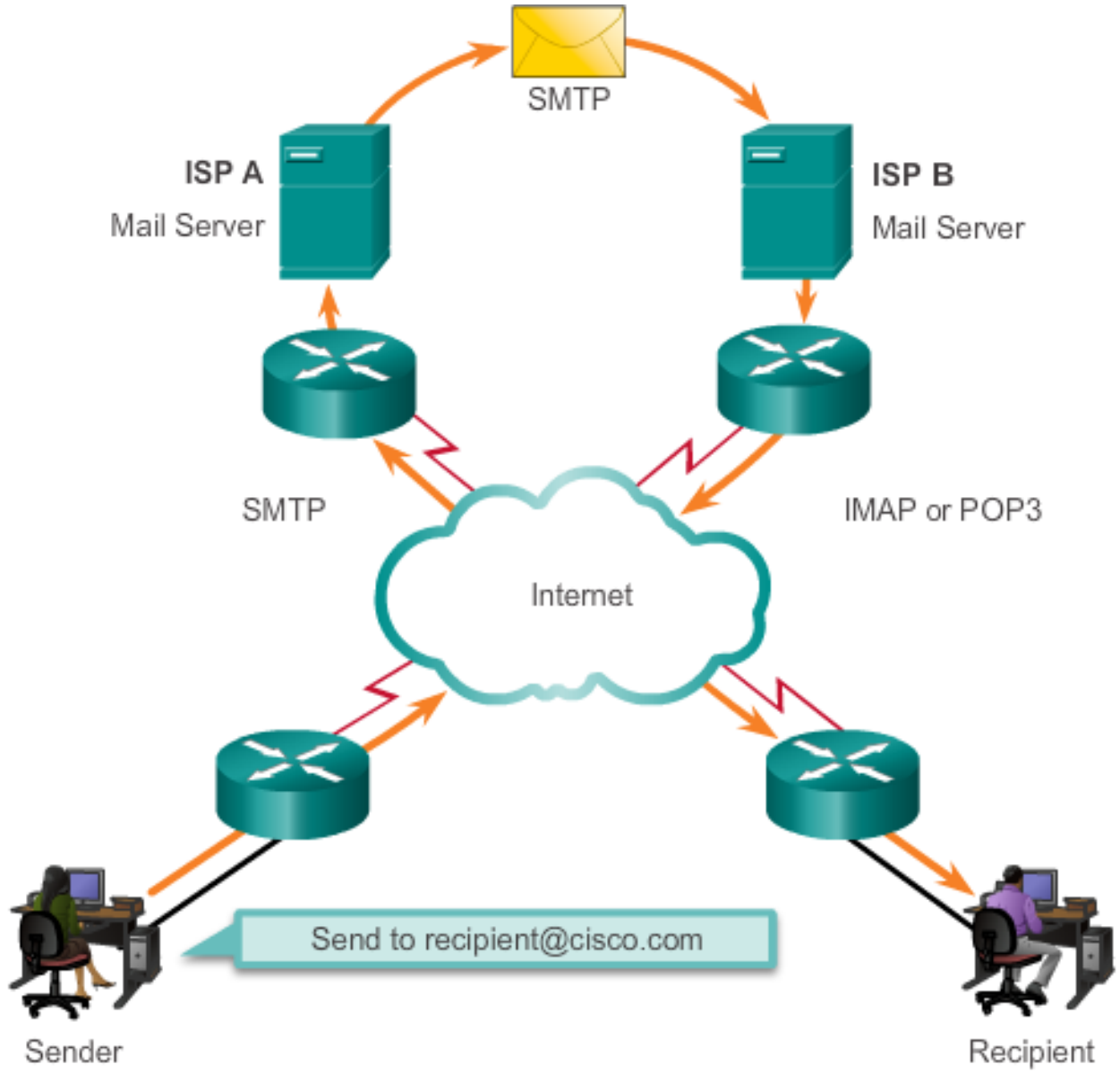
Entering 'http://www.cisco.com' in the address bar of a web browser generates the HTTP 'GET' message.

Email Protocols

Email requires several applications and services. Email is a store-and-forward method of sending, storing, and retrieving electronic messages across a network. Email messages are stored in databases on mail servers.

Email clients communicate with mail servers to send and receive email. Mail servers communicate with other mail servers to transport messages from one domain to another. An email client does not communicate directly with another email client when sending email. Instead, both clients rely on the mail server to transport messages.

Email supports three separate protocols for operation: **Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), and IMAP. The application layer process that sends mail uses SMTP. A client retrieves email, however, using one of the two application layer protocols: POP or IMAP.**

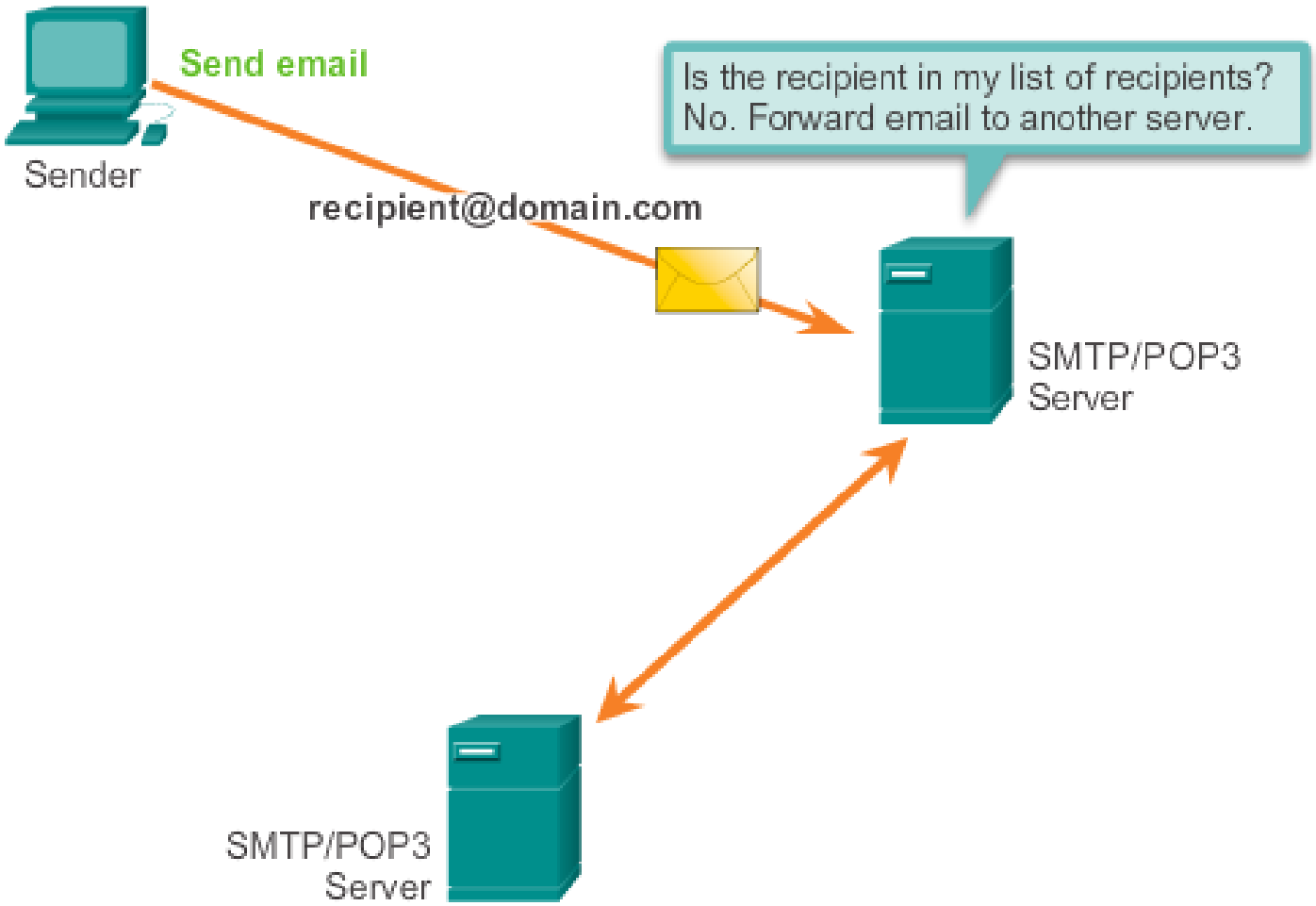


SMTP Operation

When a client sends email, the client SMTP process connects with a server SMTP process on well-known port 25. After the connection is made, the client attempts to send the email to the server across the connection. When the server receives the message, it either places the message in a local account, if the recipient is local, or forwards the message to another mail server for delivery, as shown in the figure.

The destination email server may not be online or may be busy when email messages are sent. Therefore, SMTP spools messages to be sent at a later time. Periodically, the server checks the queue for messages and attempts to send them again. If the message is still not delivered after a predetermined expiration time, it is returned to the sender as undeliverable.

SMTP



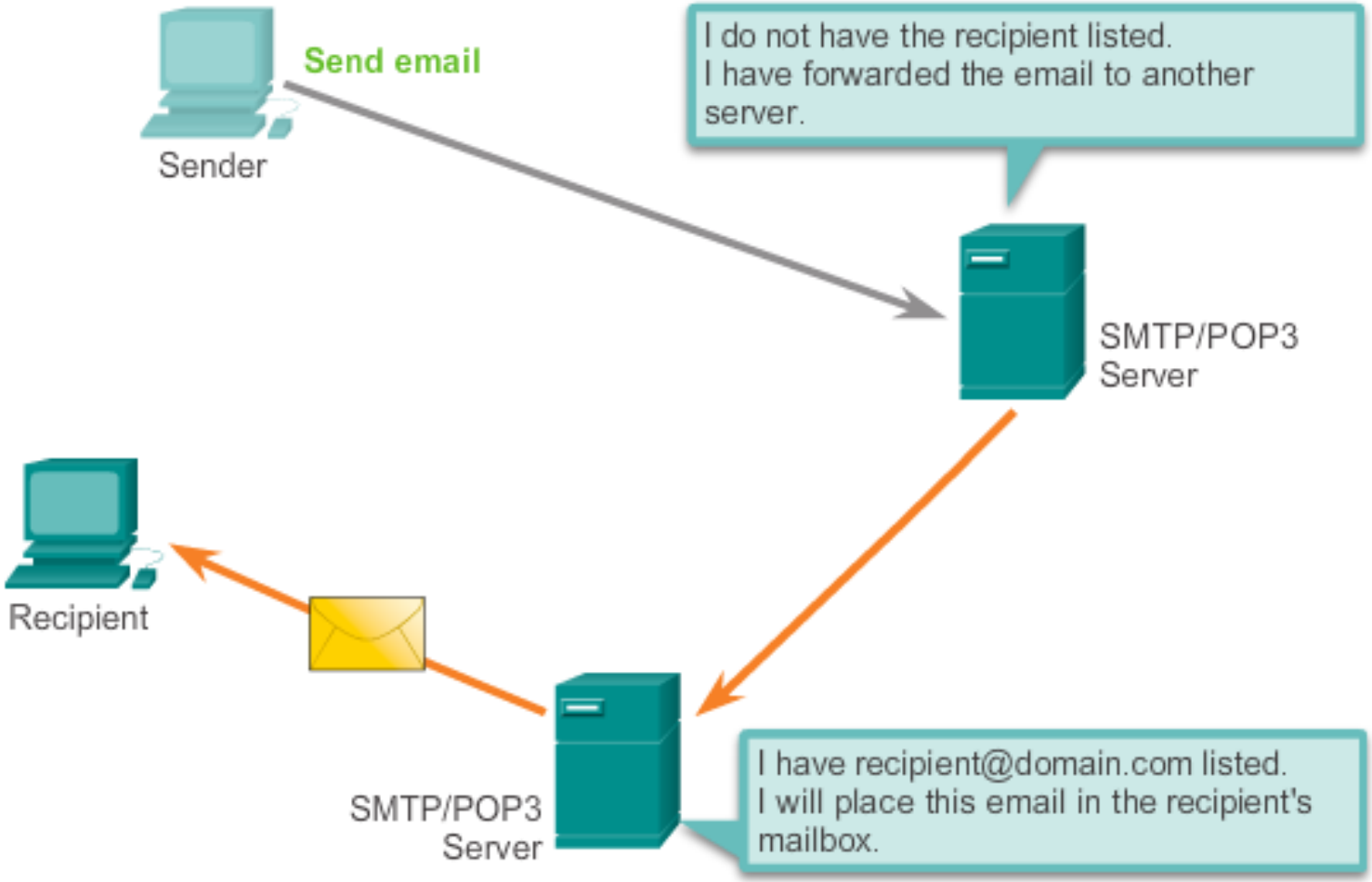
POP Operation

POP is used by an application to retrieve mail from a mail server. **With POP, mail is downloaded from the server to the client and then deleted on the server.** This is how POP operates, by default.

The server starts the POP service by passively listening on TCP port **110** for client connection requests. When a client wants to make use of the service, it sends a request to establish a TCP connection with the server. When the connection is established, the POP server sends a greeting. The client and POP server then exchange commands and responses until the connection is closed or aborted.

With POP, email messages are downloaded to the client and removed from the server, so there is no centralized location where email messages are kept. **Because POP does not store messages, it is undesirable for a small business that needs a centralized backup solution.**

POP3

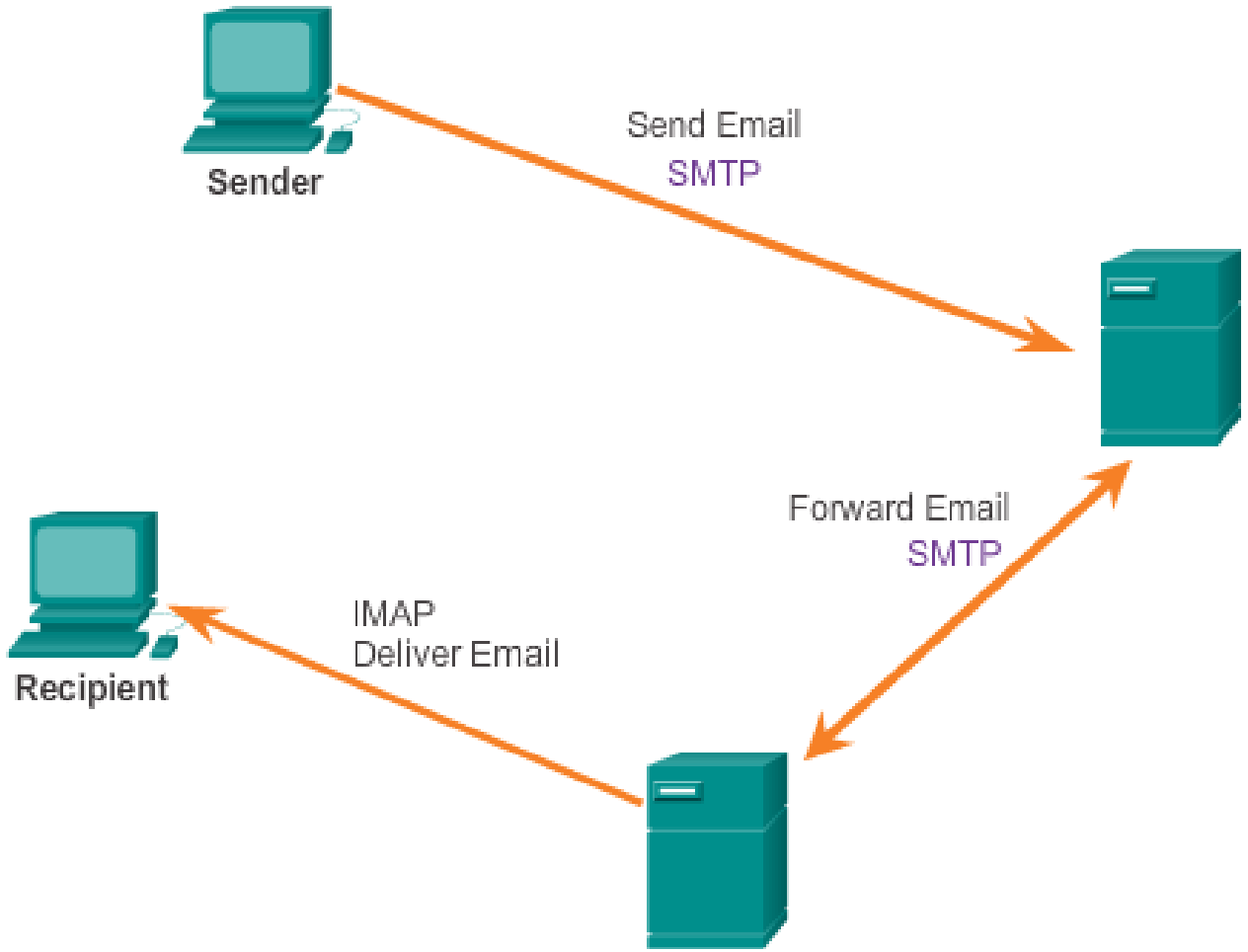


IMAP Operation

IMAP is another protocol that describes a method to retrieve email messages. **Unlike POP, when the user connects to an IMAP-capable server, copies of the messages are downloaded to the client application.** The original messages are kept on the server until manually deleted. Users view copies of the messages in their email client software.

Users can create a file hierarchy on the server to organize and store mail. That file structure is duplicated on the email client as well. When a user decides to delete a message, the server synchronizes that action and deletes the message from the server.

IMAP

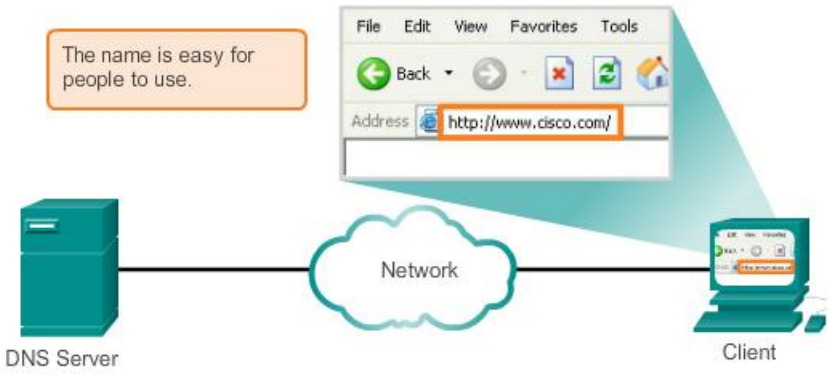


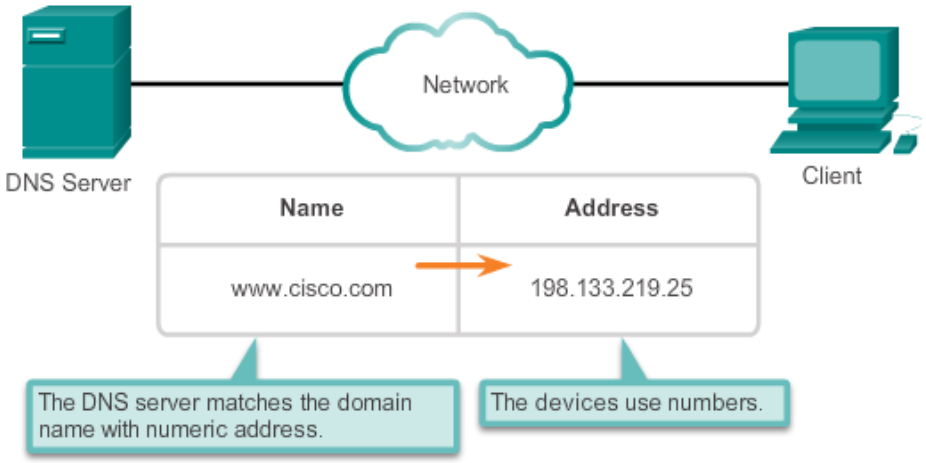
Domain Name Service

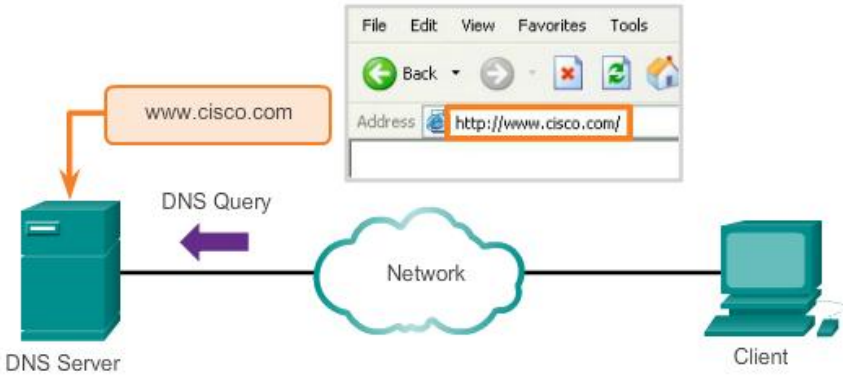
In data networks, devices are labeled with numeric IP addresses to send and receive data over networks. Domain names were created to convert the numeric address into a simple, recognizable name.

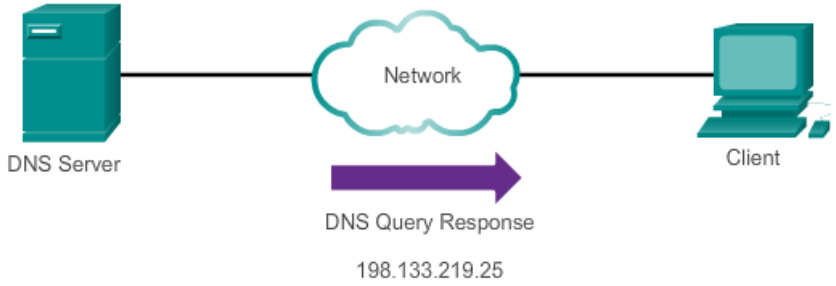
On the Internet, these domain names, such ashttp://www.cisco.com, are much easier for people to remember than 198.133.219.25, which is the actual numeric address for this server. If Cisco decides to change the numeric address of www.cisco.com, it is transparent to the user because the domain name remains the same. The new address is simply linked to the existing domain name and connectivity is maintained.

The DNS protocol defines an automated service that matches resource names with the required numeric network address.









DNS Message Format

When a client makes a query, the server's DNS process first looks at its own records to resolve the name. If it is unable to resolve the name using its stored records, it contacts other servers to resolve the name. After a match is found and returned to the original requesting server, the server temporarily stores the numbered address in the event that the same name is requested again.

The DNS Client service on Windows PCs also stores previously resolved names in memory. The **ipconfig /displaydns** command displays all of the cached DNS entries.

DNS Hierarchy

The DNS protocol uses a hierarchical system to create a database to provide name resolution. The hierarchy looks like an inverted tree with the root at the top and branches below. DNS uses domain names to form the hierarchy.

The naming structure is broken down into small, manageable zones. Each DNS server maintains a specific database file and is only responsible for managing name-to-IP mappings for that small portion of the entire DNS structure. When a DNS server receives a request for a name translation that is not within its DNS zone, the DNS server forwards the request to another DNS server within the proper zone for translation.

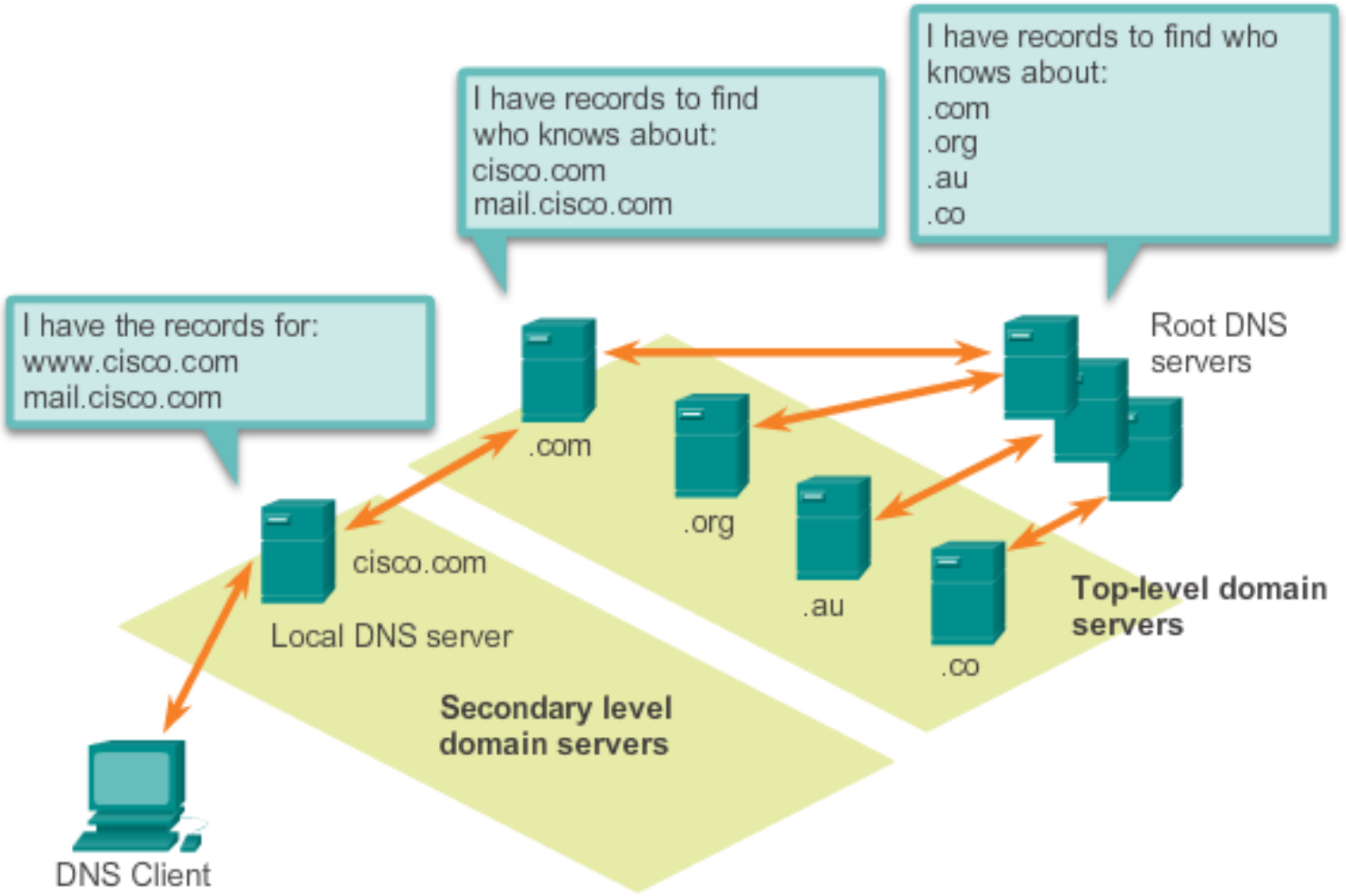
The different top-level domains represent either the type of organization or the country of origin. Examples of top-level domains are:

.com - a business or industry

.org - a non-profit organization

.au - Australia

.co - Colombia



A hierarchy of DNS servers contains the resource records that match names with addresses.

Dynamic Host Configuration Protocol

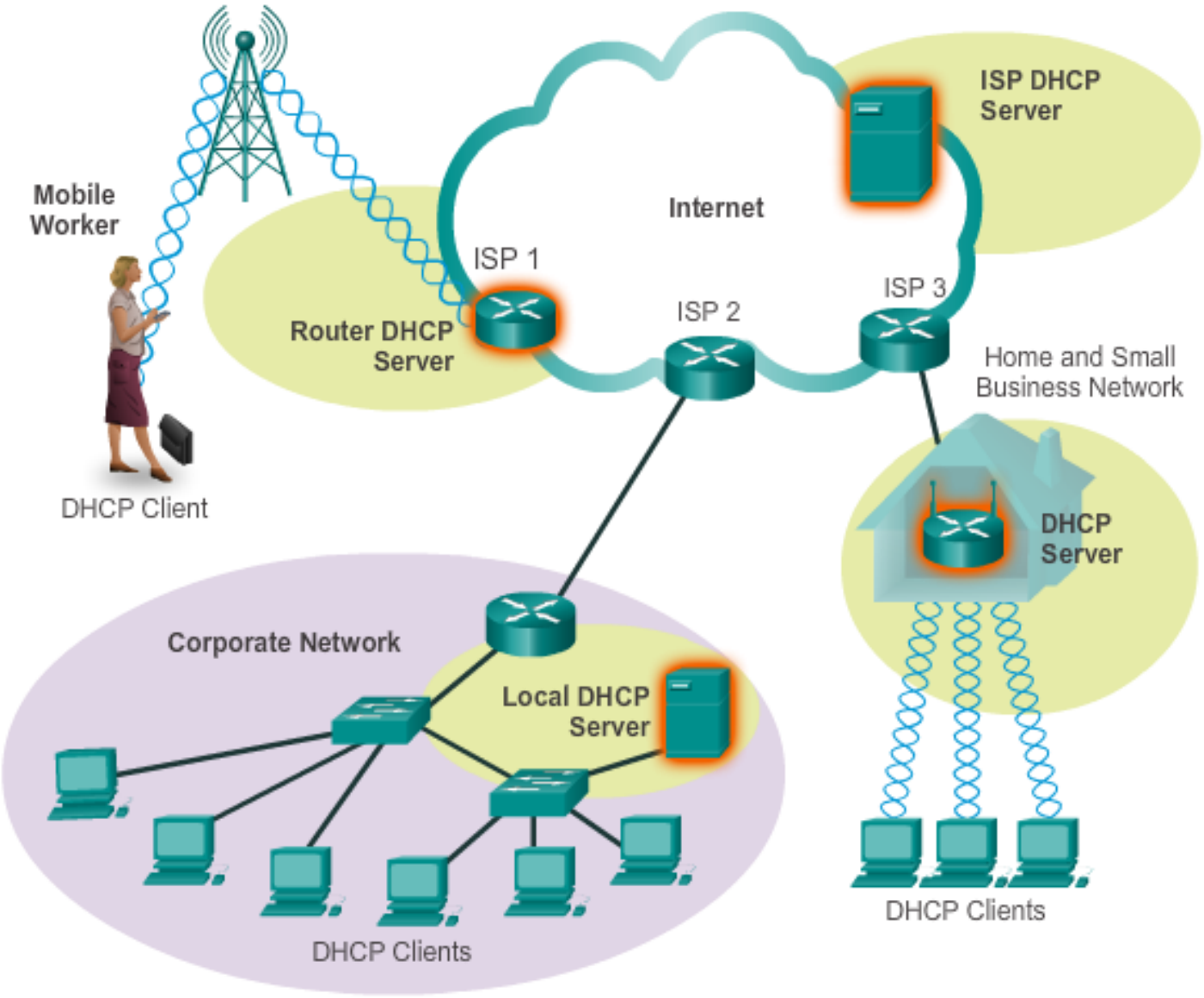
The Dynamic Host Configuration Protocol (DHCP) for IPv4 service automates the assignment of IPv4 addresses, subnet masks, gateways, and other IPv4 networking parameters.

The DHCP server chooses an address from a configured range of addresses called a pool and assigns (leases) it to the host.

The DHCP server in most medium-to-large networks is usually a local, dedicated PC-based server. With home networks, the DHCP server is usually located on the local router that connects the home network to the ISP.

Many networks use both DHCP and static addressing. DHCP is used for general purpose hosts, such as end user devices. Static addressing is used for network devices, such as gateways, switches, servers, and printers.

DHCPv6 (DHCP for IPv6) provides similar services for IPv6 clients. One important difference is that DHCPv6 does not provide a default gateway address. This can only be obtained dynamically from the router's Router Advertisement message.



File Transfer Protocol

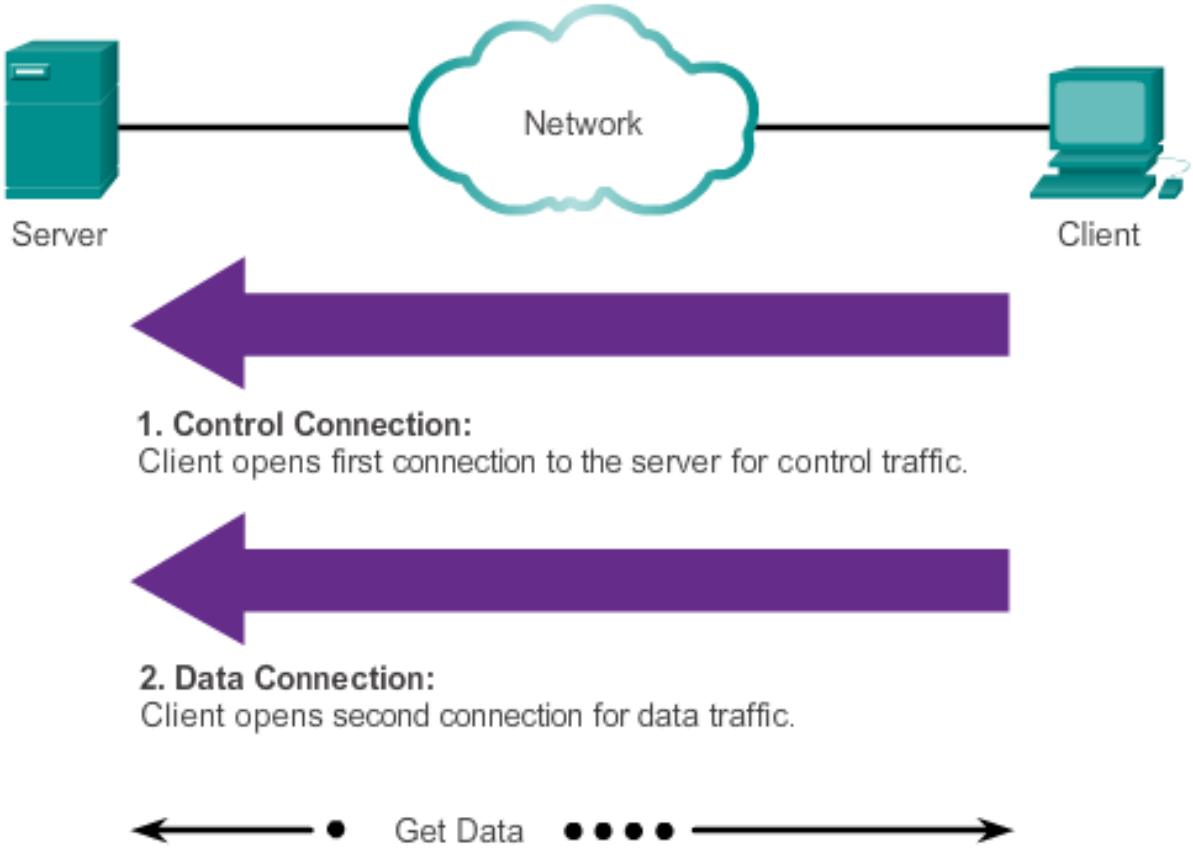
FTP was developed to allow for data transfers between a client and a server. To successfully transfer data, FTP requires two connections between the client and the server, one for commands and replies, the other for the actual file transfer:

The client establishes the first connection to the server for control traffic using TCP port 21, consisting of client commands and server replies.

The client establishes the second connection to the server for the actual data transfer using TCP port 20. This connection is created every time there is data to be transferred.

The data transfer can happen in either direction. The client can download (pull) data from the server, or the client can upload (push) data to the server.

FTP Process

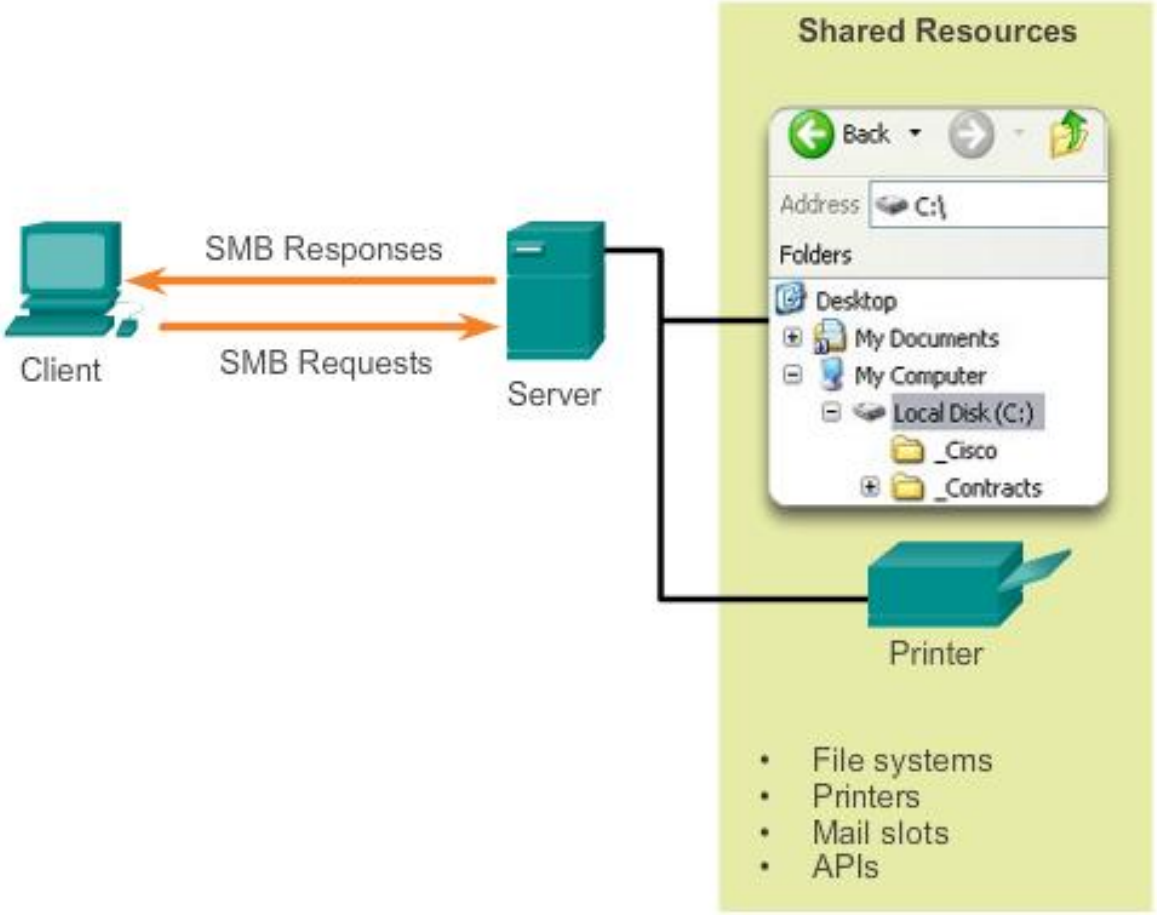


Based on commands sent across control connection, data can be downloaded from server or uploaded from client.

Server Message Block

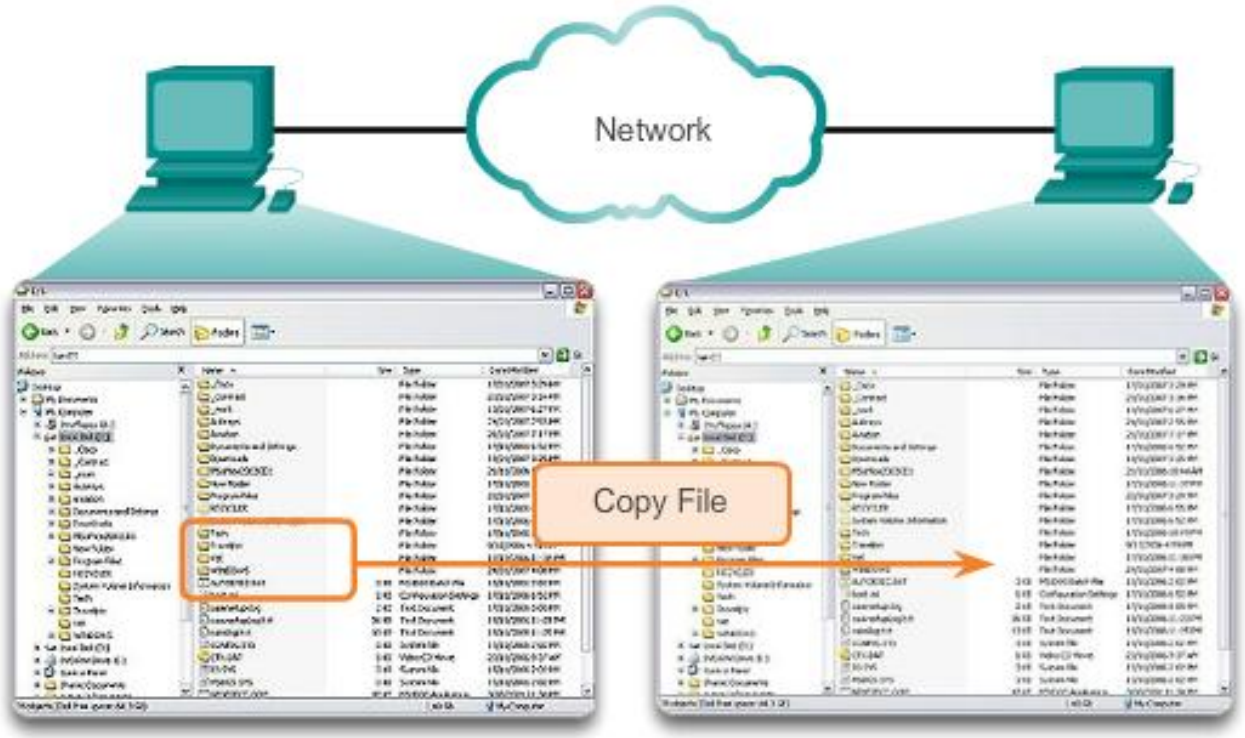
The Server Message Block (SMB) is a client/server file sharing protocol that describes the structure of shared network resources, such as directories, files, printers, and serial ports. It is a request-response protocol. All SMB messages share a common format. This format uses a fixed-sized header, followed by a variable-sized parameter and data component.

SMB Protocol



SMB is a client-server, request-response protocol. Servers can make their resources available to clients on the network.

SMB File Sharing



A file may be copied from PC to PC with Windows Explorer using the SMB protocol.

Summary

The application layer is responsible for directly accessing the underlying processes that manage and deliver communication to the human network. This layer serves as the source and destination of communications across data networks. The application layer applications, services, and protocols enable users to interact with the data network in a way that is meaningful and effective.

Applications are computer programs with which the user interacts and which initiate the data transfer process at the user's request.

Services are background programs that provide the connection between the application layer and the lower layers of the networking model.

Protocols provide a structure of agreed-upon rules and processes that ensure services running on one particular device can send and receive data from a range of different network devices.

Delivery of data over the network can be requested from a server by a client, or between devices that operate in a P2P arrangement. In P2P,the client/server relationship is established according to which device is the source and destination at that time. Messages are exchanged between the application layer services at each end device in accordance with the protocol specifications to establish and use these relationships.

Protocols like HTTP, for example, support the delivery of web pages to end devices. SMTP, IMAP , and POP support sending and receiving email. SMB and FTP enable users to share files. P2P applications make it easier for consumers to seamlessly share media in a distributed fashion. DNS resolves the human-legible names, used to refer to network resources, into numeric addresses usable by the network. Clouds are remote upstream locations that store data and host applications so that users do not require as many local resources, and so that users can seamlessly access content on different devices from any location.

All of these elements work together, at the application layer. The application layer enables users to work and play over the Internet.