Application Layer Functionality and Protocols

Network Fundamentals – Chapter 3

Cisco | Networking Academy®
Mind Wide Open™

Objectives

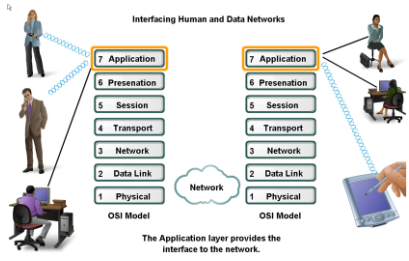
Define the application layer as the source and destination of data for communication across networks.

Explain the role of protocols in supporting communication between server and client processes.

Describe the features, operation, and use of well-known TCP/IP application layer services (HTTP, DNS, SMTP).

Applications – The Interface Between Human and Data Networks

Explain that applications provide the means for generating and receiving data that can be transported on the network



Typical Data Network

In this model, information is passed from one layer to the next, starting at the Application layer on the transmitting host, proceeding down the hierarchy to the Physical layer, then passing over the communications channel to the destination host, where the information proceeds back up the hierarchy, ending at the Application layer.

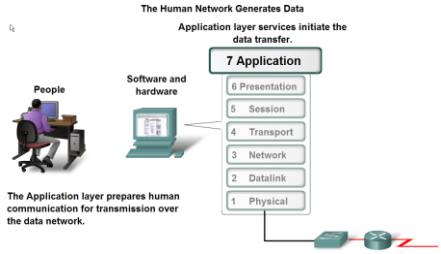
The Application layer, Layer seven, is the top layer of both the OSI and TCP/IP models. It is the layer that provides the interface between the applications we use to communicate and the underlying network over which our messages are transmitted.

Application layer protocols are used to exchange data between programs running on the source and destination hosts.

There are many Application layer protocols and new protocols are always being developed.

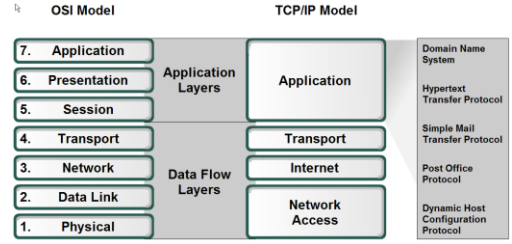
Applications – The Interface Between Human and Data Networks

Explain the role of applications, services and protocols in converting communication to data that can be transferred across the data network



Applications – The Interface Between Human and Data Networks

Define the separate roles applications, services and protocols play in transporting data through networks



TCP/IP and OSI

Although the TCP/IP protocol suite was developed prior to the definition of the OSI model, the functionality of the TCP/IP Application layer protocols fit roughly into the framework of the top three layers of the OSI model: Application, Presentation and Session layers.

Most TCP/IP Application layer protocols were developed before the emergence of personal computers, graphical user interfaces and multimedia objects.

As a result, these protocols implement very little of the functionality that is specified in the OSI model Presentation and Session layers.

Presentation Layer

The Presentation layer has three primary functions:

1. Coding and conversion of Application layer data to ensure that data from the source device can be interpreted by the appropriate application on the destination device.
2. Compression of the data in a manner that can be decompressed by the destination device.
3. Encryption of the data for transmission and the decryption of data upon receipt by the destination

Other Presentation Layer Protocols

Presentation layer implementations are not typically associated with a particular protocol stack.

The standards for video and graphics are examples. Some well-known standards for video include QuickTime and Motion Picture Experts Group (MPEG). QuickTime is an Apple Computer specification for video and audio, and MPEG is a standard for video compression and coding.

Among the well-known graphic image formats are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), and Tagged Image File Format (TIFF).

GIF and JPEG are compression and coding standards for graphic images, and TIFF is a standard coding format for graphic images.

The Session Layer

As the name of the Session layer implies, functions at this 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.

Most applications, like web browsers or e-mail clients, incorporate functionality of the OSI layers 5, 6 and 7.

TCP/IP Protocols

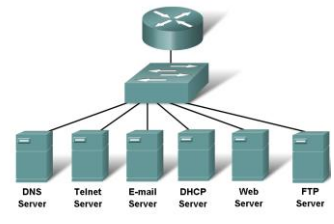
The most widely-known TCP/IP Application layer protocols are those that provide for the exchange of user information.

These protocols specify the format and control information necessary for many of the common Internet communication functions. Among these TCP/IP protocols are:

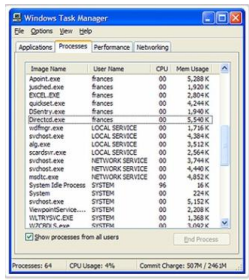
1. Domain Name Service Protocol (DNS) is used to resolve Internet names to IP addresses.
2. Hypertext Transfer Protocol (HTTP) is used to transfer files that make up the Web pages of the World Wide Web.
3. Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.
4. Telnet, a terminal emulation protocol, is used to provide remote access to servers and networking devices.
5. File Transfer Protocol (FTP) is used for interactive file transfer between systems.

Applications – The Interface Between Human and Data Networks

Describe the role protocols play in networking and be able to identify several message properties that can be defined by a protocol



Applications V Software



Examples of processes running in the Windows operating system

Applications

Within the Application layer, there are two forms of software programs or processes that provide access to the network: applications and services.

1. ***Network-Aware Applications***

Applications are the software programs used by people to communicate over the network. Some end-user applications are network-aware, meaning that they implement the Application layer protocols and are able to communicate directly with the lower layers of the protocol stack. E-mail clients and web browsers are examples of these types of applications.

Services

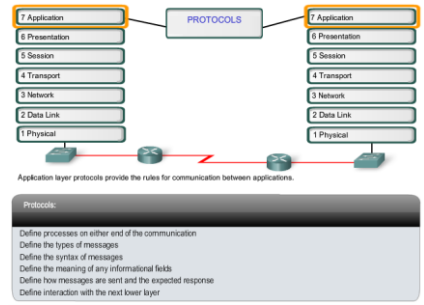
2. Application-layer Services

Other programs may need the assistance of Application layer services to use network resources, like file transfer or network print spooling.

Though transparent to the user, these services are the programs that interface with the network and prepare the data for transfer.

Different types of data - whether it is text, graphics, or video - require different network services to ensure that it is properly prepared for processing by the functions occurring at the lower layers of OSI model.

Application Layer Protocols

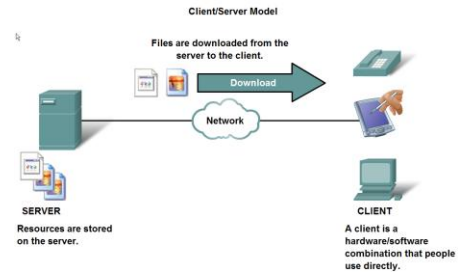


How Protocols are Used

1. Application layer protocols are used by both the source and destination devices during a communication session. In order for the communications to be successful, the Application layer protocols implemented on the source and destination host must match.
2. Protocols establish consistent rules for exchanging data between applications and services loaded on the participating devices.
3. Protocols specify how data inside the messages is structured and the types of messages that are sent between source and destination.
4. These messages can be requests for services, acknowledgments, data messages, status messages, or error messages. Protocols also define message dialogues, ensuring that a message being sent is met by the expected response and the correct services are invoked when data transfer occurs.
5. *Applications and services may also use multiple protocols in the course of a single conversation. One protocol may specify how to establish the network connection and another describe the process for the data transfer when the message is passed to the next lower layer.*

The Role of Protocols in Supporting Communication

Describe the roles of client and server processes in data networks



Client/Server Model

In the client/server model, the device requesting the information is called a client and the device responding to the request is called a server.

Client and server processes are considered to be in the Application layer.

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 control information, such as user authentication and the identification of a data file to be transferred.

Simple Example

One example of a client/server network is a corporate environment where employees use a company e-mail server to send, receive and store e-mail.

The e-mail client on an employee computer issues a request to the e-mail server for any unread mail. The server responds by sending the requested e-mail to the client.

Although data is typically described as flowing from the server to the client, some data always flows from the client to the server.

Data flow may be equal in both directions, or may even be greater in the direction going from the client to the server.

For example, a client may transfer a file to the server for storage purposes. Data transfer from a client to a server is referred to as an upload and data from a server to a client as a download.

Servers

In a general networking context, any device that responds to requests from client applications is functioning as a server. A server is usually a computer that contains information to be shared with many client systems

Different types of server applications may have different requirements for client access.

Some servers may require authentication of user account information to verify if the user has permission to access the requested data or to use a particular operation.

Such servers rely on a central list of user accounts and the authorizations, or permissions, (both for data access and operations) granted to each user. When using an FTP client, for example, if you request to upload data to the FTP server, you may have permission to write to your individual folder but not to read other files on the site.

Server Daemon

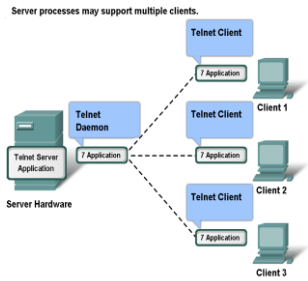
In a client/server network, the server runs a service, or process, sometimes called a server daemon.

Like most services, daemons typically run in the background and are not under an end user's direct control.

Daemons are described as "listening" for a request from a client, because they are programmed to respond whenever the server receives a request for the service provided by the daemon.

When a daemon "hears" a request from a client, it exchanges appropriate messages with the client, as required by its protocol, and proceeds to send the requested data to the client in the proper format.

The Role of Protocols in Supporting Communication



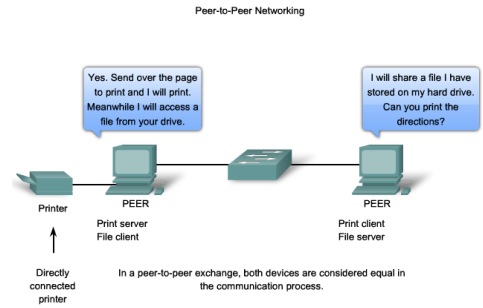
Additionally, servers typically have multiple clients requesting information at the same time.

For example, a Telnet server may have many clients requesting connections to it.

These individual client requests must be handled simultaneously and separately for the network to succeed.

The Application layer processes and services rely on support from lower layer functions to successfully manage the multiple conversations.

Peer-to-Peer Networking



P2P Network

In a peer-to-peer network, two or more computers are connected via a network and can share resources (such as printers and files) without having a dedicated server.

Every connected end device (known as a peer) can function as either a server or a client. One computer might assume the role of server for one transaction while simultaneously serving as a client for another.

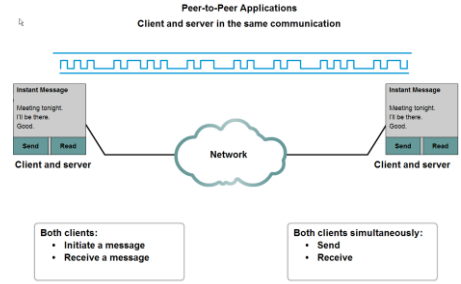
The roles of client and server are set on a per request basis.

A simple home network with two connected computers sharing a printer is an example of a peer-to-peer network.

Because peer-to-peer networks usually do not use centralized user accounts, permissions, or monitors, it is difficult to enforce security and access policies in networks containing more than just a few computers. User accounts and access rights must be set individually on each peer device.

The Role of Protocols in Supporting Communication

Compare and contrast client server networking with peer-to-peer networking and peer-to-peer applications



P2P Applications

A peer-to-peer application (P2P), unlike a peer-to-peer network, 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.

Both can initiate a communication and are considered equal in the communication process.

However, peer-to-peer applications require that each end device provide a user interface and run a background service.

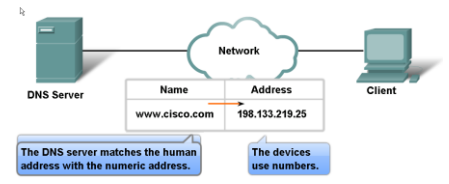
When you launch a specific peer-to-peer application it invokes the required user interface and background services. After that the devices can communicate directly.

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.

Features, Operation, and Use of TCP/IP Application Layer Services

Describe the features of the DNS protocol and how this protocol supports DNS services



DNS

In data networks, devices are labelled with numeric IP addresses, so that they can participate in sending and receiving messages over the network.

On the Internet these domain names, such as www.cisco.com, are much easier for people to remember than 198.133.219.25, which is the actual numeric address for this server.

Also, if Cisco decides to change the numeric address, it is transparent to the user, since the domain name will remain www.cisco.com.

The new address will simply be linked to the existing domain name and connectivity is maintained

The Domain Name System (DNS) was created for domain name to address resolution for these networks. DNS uses a distributed set of servers to resolve the names associated with these numbered addresses.

DNS Protocol

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

It includes the format for queries, responses, and data formats.

DNS protocol communications use a single format called a message.

This message format is used for all types of client queries and server responses, error messages, and the transfer of resource record information between servers.

DNS a Client/Server Service

DNS is a client/server service; however, it differs from the other client/server services that we are examining.

While other services use a client that is an application (such as web browser, e-mail client), the DNS client runs as a service itself.

The DNS client, sometimes called the DNS resolver, supports name resolution for our other network applications and other services that need it.

When configuring a network device, we generally provide one or more DNS Server addresses that the DNS client can use for name resolution.

Usually the Internet service provider provides the addresses to use for the DNS servers. When a user's application requests to connect to a remote device by name, the requesting DNS client queries one of these name servers to resolve the name to a numeric address.

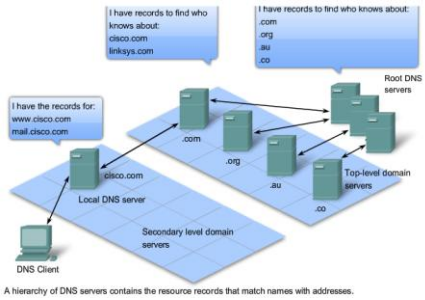
Message Types

The DNS server stores different types of resource records used to resolve names. These records contain the name, address, and type of record.

Some of these record types are:

1. A - an end device address
2. NS - an authoritative name server
3. CNAME - the canonical name (or Fully Qualified Domain Name) for an alias; used when multiple services have the single network address but each service has its own entry in DNS
4. MX - mail exchange record; maps a domain name to a list of mail exchange servers for that domain

DNS Hierarchy



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

System Hierarchy

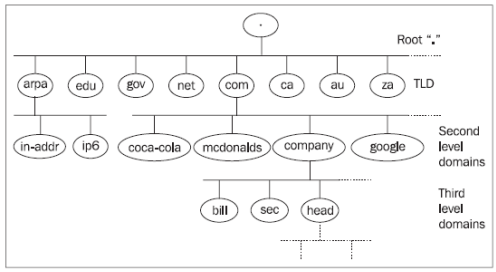
The Domain Name System uses a hierarchical system to create a name database to provide name resolution. The hierarchy looks like an inverted tree with the root at the top and branches below.

At the top of the hierarchy, the root servers maintain records about how to reach the top-level domain servers, which in turn have records that point to the secondary level domain servers and so on.

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

1. .au - Australia
2. .co - Colombia
3. .com - a business or industry
4. .jp - Japan
5. .org - a non-profit organization

DNS Tree Structure



Quick Example

For example, as shown in the figure, the root DNS server may not know exactly where the e-mail server mail.cisco.com is located, but it maintains a record for the "com" domain within the top-level domain.

Likewise, the servers within the "com" domain may not have a record for mail.cisco.com, but they do have a record for the "cisco.com" domain.

The servers within the cisco.com domain have a record (a MX record to be precise) for mail.cisco.com.

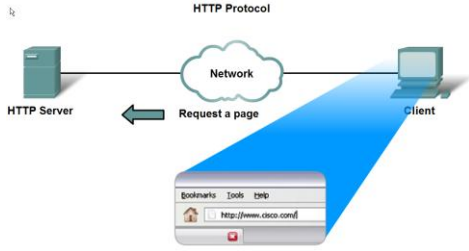
The Domain Name System relies on this hierarchy of decentralized servers to store and maintain these resource records.

The resource records list domain names that the server can resolve and alternative servers that can also process requests.

If a given server has resource records that correspond to its level in the domain hierarchy, it is said to be authoritative for those records.

Features, Operation, and Use of TCP/IP Application Layer Services

Describe the features of the HTTP protocol and how this protocol supports the delivery of web pages to the client



Web Browsers

Web browsers are the client applications our computers use to connect to the World Wide Web and access resources stored on a web server.

As with most server processes, the web server runs as a background service and makes different types of files available.

In order to access the content, web clients make connections to the server and request the desired resources.

The server replies with the resources and, upon receipt, the browser interprets the data and presents it to the user.

How it Works

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

1. http (the protocol or scheme)

2. www.cisco.com (the server name)

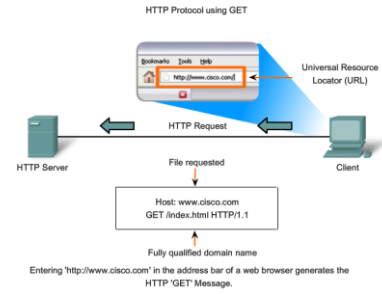
3. web-server.htm (the specific file name requested).

The browser then checks with a name server to convert www.cisco.com into a numeric address, which it uses to connect to the server.

Using the HTTP protocol requirements, the browser sends a GET request to the server and asks for the file web-server.htm.

The server in turn 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 Verbs



Verbs Explained

GET is a client request for data. A web browser sends the GET message to request pages from a web server.

Once the server receives the GET request, it responds with a status line, such as HTTP/1.1 200 OK, and a message of its own, the body of which may be the requested file, an error message, or some other information.

POST and PUT are used to send messages that upload data to the web server. For example, when the user enters data into a form embedded in a web page, POST includes the data in the message sent to the server.

PUT uploads resources or content to the web server

HTTP and Security

Although it is remarkably flexible, HTTP is not a secure protocol. The POST messages upload information to the server in plain text that can be intercepted and read.

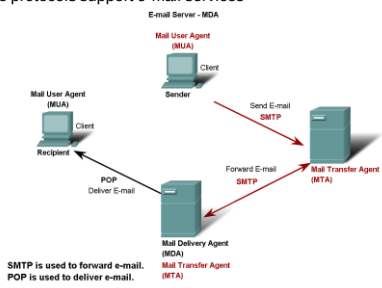
Similarly, the server responses, typically HTML pages, are also unencrypted.

For secure communication across the Internet, the HTTP Secure (HTTPS) protocol is used for accessing or posting web server information.

HTTPS can use authentication and encryption to secure data as it travels between the client and server. HTTPS specifies additional rules for passing data between the Application layer and the Transport Layer.

Features, Operation, and Use of TCP/IP Application Layer Services

Describe the features of the POP and SMTP protocols, and how these protocols support e-mail services



Mail Server Processes

The e-mail server operates two separate processes:

1. Mail Transfer Agent (MTA)
2. Mail Delivery Agent (MDA)

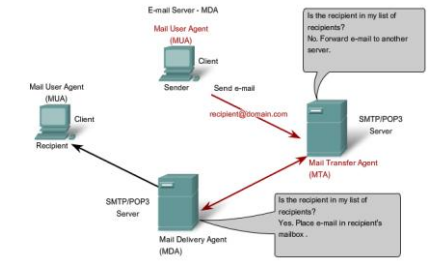
The Mail Transfer Agent (MTA) process is used to forward e-mail. As shown in the figure, the MTA receives messages from the MUA or from another MTA on another e-mail server.

Based on the message header, it determines how a message has to be forwarded to reach its destination.

If the mail is addressed to a user whose mailbox is on the local server, the mail is passed to the MDA.

If the mail is for a user not on the local server, the MTA routes the e-mail to the MTA on the appropriate server.

MUA, MTA, MDA



The Mail Delivery Agent process governs delivery of e-mail between servers and clients.

Roles

In the figure, we see that the Mail Delivery Agent (MDA) accepts a piece of e-mail from a Mail Transfer Agent (MTA) and performs the actual delivery.

The MDA receives all the inbound mail from the MTA and places it into the appropriate users' mailboxes.

The MDA can also resolve final delivery issues, such as virus scanning, spam filtering, and return-receipt handling.

Most e-mail communications use the MUA, MTA, and MDA applications. However, there are other alternatives for e-mail delivery.

Proprietary Email Systems

A client may be connected to a corporate e-mail system, such as IBM's Lotus Notes, Novell's GroupWise, or Microsoft's Exchange.

These systems often have their own internal e-mail format, and their clients typically communicate with the e-mail server using a proprietary protocol.

The server sends or receives e-mail via the Internet through the product's Internet mail gateway, which performs any necessary reformatting.

SMTP

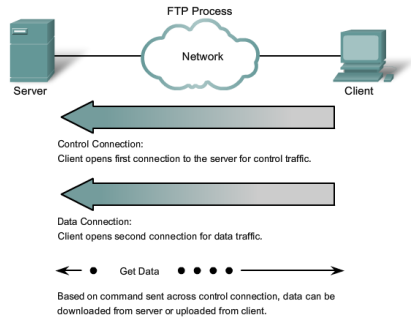
The SMTP protocol message format uses a rigid set of commands and replies.

These commands support the procedures used in SMTP, such as session initiation, mail transaction, forwarding mail, verifying mailbox names, expanding mailing lists, and the opening and closing exchanges.

Some of the commands specified in the SMTP protocol are:

1. HELO - identifies the SMTP client process to the SMTP server process
2. EHLO - Is a newer version of HELO, which includes services extensions
3. MAIL FROM - Identifies the sender
4. RCPT TO - Identifies the recipient
5. DATA - Identifies the body of the message

FTP – File Transfer Protocol



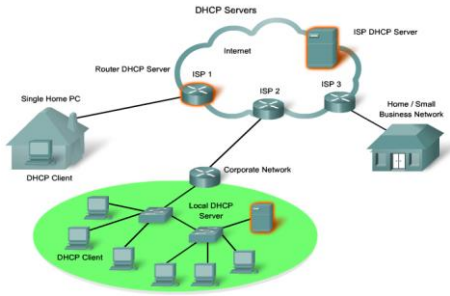
FTP

To successfully transfer files, 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 on TCP port 21. This connection is used for control traffic, consisting of client commands and server replies.

The client establishes the second connection to the server over TCP port 20. This connection is for the actual file transfer and is created every time there is a file transferred.

DHCP – Dynamic Host Control Protocol



DHCP Contd...

The Dynamic Host Configuration Protocol (DHCP) service enables devices on a network to obtain IP addresses and other information from a DHCP server.

This service automates the assignment of IP addresses, subnet masks, gateway and other IP networking parameters.

DHCP allows a host to obtain an IP address dynamically when it connects to the network.

The DHCP server is contacted and an address requested. The DHCP server chooses an address from a configured range of addresses called a pool and assigns ("leases") it to the host for a set period.

DHCP makes it possible for you to access the Internet using wireless hotspots at airports or coffee shops. As you enter the area, your laptop DHCP client contacts the local DHCP server via a wireless connection. The DHCP server assigns an IP address to your laptop.

DHCP Security

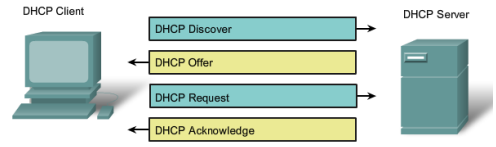
DHCP can pose a security risk because any device connected to the network can receive an address.

This risk makes physical security an important factor when determining whether to use dynamic or manual addressing.

Dynamic and static addressing both have their places in network designs. Many networks use both DHCP and static addressing.

DHCP is used for general purpose hosts such as end user devices, and fixed addresses are used for network devices such as gateways, switches, servers and printers.

DHCP Message Types



How it Works

When a DHCP-configured device boots up or connects to the network, the client broadcasts a DHCP DISCOVER packet to identify any available DHCP servers on the network.

A DHCP server replies with a DHCP OFFER, which is a lease offer message with an assigned IP address, subnet mask, DNS server, and default gateway information as well as the duration of the lease.

The client may receive multiple DHCP OFFER packets if there is more than one DHCP server on the local network, so it must choose between them, and broadcast a DHCP REQUEST packet that identifies the explicit server and lease offer that the client is accepting.

A client may choose to request an address that it had previously been allocated by the server.

How it Works

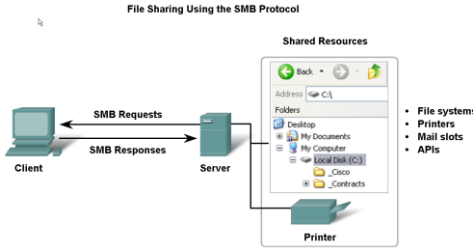
Assuming that the IP address requested by the client, or offered by the server, is still valid, the server would return a DHCP ACK message that acknowledges to the client the lease is finalized

If a DHCP NAK message is returned, then the selection process must begin again with a new DHCP DISCOVER message being transmitted.

The DHCP server ensures that all IP addresses are unique (an IP address cannot be assigned to two different network devices simultaneously).

Features, Operation, and Use of TCP/IP Application Layer Services

Describe the features of the SMB protocol and the role it plays in supporting file sharing in Microsoft-based networks



Server Message Block

The Server Message Block (SMB) is a client/server file sharing protocol. IBM developed Server Message Block (SMB) in the late 1980s to describe the structure of shared network resources, such as directories, files, printers, and serial ports.

It is a request-response protocol.

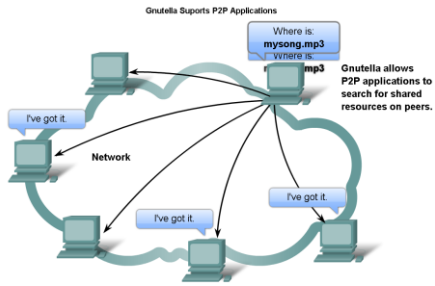
Once the connection is established, the user of the client can access the resources on the server as if the resource is local to the client host.

SMB file-sharing and print services have become the mainstay of Microsoft networking.

The LINUX and UNIX operating systems also provide a method of sharing resources with Microsoft networks using a version of SMB called SAMBA. The Apple Macintosh operating systems also support resource sharing using the SMB protocol.

Features, Operation, and Use of TCP/IP Application Layer Services

Describe the features of the Gnutella protocol and the role it plays in supporting P2P services



P2P Applications

You learned about FTP and SMB as ways of obtaining files, here is another Application protocol.

Sharing files over the Internet has become extremely popular. With P2P applications based on the Gnutella protocol, people can make files on their hard disks available to others for downloading.

Gnutella-compatible client software allows users to connect to Gnutella services over the Internet and to locate and access resources shared by other Gnutella peers.

Many client applications are available for accessing the Gnutella network, including: BearShare, Gnucleus, LimeWire, Morpheus, WinMX and XoloX (see a screen capture of LimeWire in the figure)

P2P Applications

Many P2P applications do not use a central database to record all the files available on the peers.

Instead, the devices on the network each tell the other what files are available when queried and use the Gnutella protocol and services to support locating resources

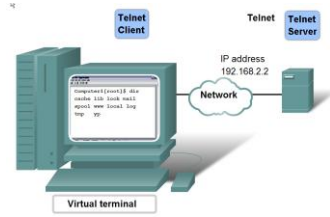
When a user is connected to a Gnutella service, the client applications will search for other Gnutella nodes to connect to.

These nodes handle queries for resource locations and replies to those requests.

They also govern control messages, which help the service discover other nodes. The actual file transfers usually rely on HTTP services.

Features, Operation, and Use of TCP/IP Application Layer Services

Describe the features of the Telnet protocol and identify several of its uses in examining and managing networks



Telnet provides a way to use a computer, connected via the network, to access a network device as if the keyboard and monitor were directly connected to the device.

Telnet

Telnet provides a standard method of emulating text-based terminal devices over the data network. Both the protocol itself and the client software that implements the protocol are commonly referred to as Telnet.

Appropriately enough, a connection using Telnet is called a Virtual Terminal (VTY) session, or connection

To support Telnet client connections, the server runs a service called the Telnet daemon.

A virtual terminal connection is established from an end device using a Telnet client application.

Once a Telnet connection is established, users can perform any authorized function on the server, just as if they were using a command line session on the server itself.

If authorized, they can start and stop processes, configure the device, and even shut down the system.

Telnet and Security

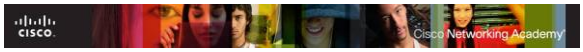
While the Telnet protocol supports user authentication, it does not support the transport of encrypted data.

All data exchanged during a Telnet sessions is transported as plain text across the network. This means that the data can be intercepted and easily understood.

If security is a concern, the Secure Shell (SSH) protocol offers an alternate and secure method for server access.

SSH provides the structure for secure remote login and other secure network services.

It also provides stronger authentication than Telnet and supports the transport of session data using encryption. As a best practice, network professionals should always use SSH in place of Telnet, whenever possible.



Summary

In this chapter, you learned to:

- Describe how the functions of the three upper OSI model layers provide network services to end user applications.
- Describe how the TCP/IP Application layer protocols provide the services specified by the upper layers of the OSI model.
- Define how people use the Application layer to communicate across the information network.
- Describe the function of well-known TCP/IP applications, such as the World Wide Web and email, and their related services (HTTP, DNS, SMB, DHCP, STMP/POP, and Telnet).
- Describe file-sharing processes that use peer-to-peer applications and the Gnutella protocol.
- Explain how protocols ensure services running on one kind of device can send to and receive data from many different network devices.
- Use network analysis tools to examine and explain how common user applications work.