Network: - A network is simply a collection of computers or other hardware devices that are connected, either physically or logically, using special hardware and software, to allow them to exchange information and cooperate.

Networking: - Networking is the term that describes the processes involved in designing, implementing, upgrading, managing and otherwise working with networks an d network technologies.

Key Concept: A network is a set of hardware devices connected, either physically or logically to allow them to exchange information.

Advantage: -

Key Concept: At a high level, networks are advantageous because they allow computers and people to be connected, so they can share resources. Some of the specific benefits of networking include communication, data sharing, Internet access, data security and management, application performance enhancement, and entertainment.

Disadvantage: -

Key Concept: Networking has a few drawbacks that balance against its many positive aspects. Setting up a network has costs in hardware, software, maintenance and administration. It is also necessary to manage a network to keep it running smoothly, and to address possible misuse abuse. Data security or also becomes a much bigger concern when computers are connected.

Networking Layers: -Networking technologies are most often compartmentalized in this manner by dividing their functions into layers, each of which contains

hardware and/or software elements. Each layer is responsible for performing a task, as well as interacting with the layers above it and below it.

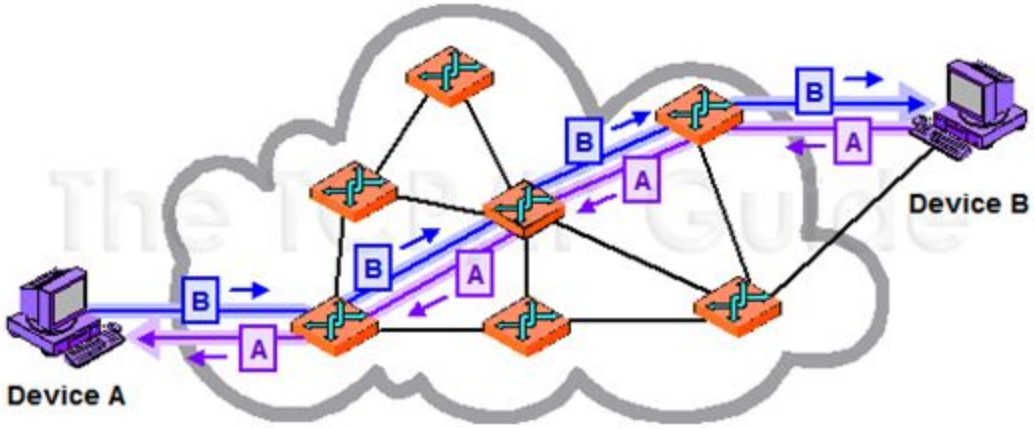
Networking Models: -It is necessary for everyone to agree on how layers will be defined and used. The most common tool for this purpose is a networking model. The model describes what the different layers are in the network, what each is responsible for doing, and how they interact like (OSI) Open Systems Interconnection

Networking Architecture: - An architecture is essentially a set of rules that describes the function of some portion of the hardware and software that constitute a stack of layers.

Protocols: - An architecture is essentially a set of rules that describes the function of some portion of the hardware and software that constitute a stack of layers.

Key Concept: A networking protocol defines a set of rules, algorithms, messages and other mechanisms that enable software and hardware in networked devices to communicate effectively. A protocol usually describes a means for communication between corresponding entities at the same OSI Reference Model layer in two or more devices

Circuit Switching: - In this networking method, a connection called a circuit is set up between two devices, which is used for the whole communication. Information about the nature of the circuit is maintained by the network.



Packet Switching: - In this network type, no specific path is used for data transfer. Instead, the data is chopped up into small pieces called packets and sent over the network.

Key Concept : One important issue in selecting a switching method is whether the network medium is shared or dedicated. Your phone line can be used for establishing a circuit because you are the only one who can use it.

Connection-Oriented and Connectionless Protocols

Key Concept: A connection-oriented protocol is one where a logical connection is first established between devices prior to data being sentby following a specific set of rules that specify how a connection should be initiated, negotiated, managed and eventually terminated. In a connectionless protocol, data is just sent without a connection being created.

The Relationship Between Connection Orientation and Circuits

Key Concept: In order to establish a circuit between two devices, they must also be connected. For this reason, circuit-switched networks are inherently based on connections. This has led to the terms "**circuit-switched**" and "**connection-oriented**" being **used interchangeably**.

Key Concept: Circuit-switched networking technologies are inherently connectionoriented, but not all connection-oriented technologies use circuit switching. Logical connection-oriented protocols can in fact be implemented on top of packet switching networks to provide higher-layer services to applications that require connections.

Connection-Oriented and Connectionless Protocols in TCP/IP

It has two main protocols that operate at the transport layer of the OSI Reference Model. One is Transmission Control Protocol (TCP), which is connection oriented; the other, the User Datagram Protocol(UDP), is connectionless.

Messages: Packets, Frames, Datagrams and Cells

Packet: - This term is considered by many to most correctly refer to a message sent by protocols operating at the network layer of the OSI Reference Model

Datagram: -It is also often used to refer to a message that is sent at a higher level of the OSI Reference Model (more often than "packet" is).

Frame: -It is most commonly seen used in reference to data link layer messages. It is occasionally also used to refer to physical layer messages, when message formatting is performed by a layer one technology.

Cell: Frames and packets, in general, can be of variable length, depending on their contents; in contrast, a cell is most often a message that is fixed in size.

Key Concept: Communication between devices on packet-switched networks is based on in items most generically called messages. These pieces of information also go by other names such as packets, datagrams, frames and cells, which often

correspond to protocols at particular layers of the OSI Reference Model. The formal OSI terms for messages are protocol data unit (PDU) and service data unit (SDU).

Message Formatting: Headers, Payloads and Footers

Key Concept: The general format of a networking message consists of a header, followed by the data or payload of the message, followed optionally by a footer. Header and footer information is functionally the same except for position in the message; footer fields are only sometimes used, especially in cases where the data in the field is calculated based on the values of the data being transmitted.

Message Addressing and Transmission Methods: Unicast, Broadcast and Multicast Messages

Message Transmission Methods

Unicast Messages: -These are messages that are sent from one device to another device; they are not intended for others.

Broadcast Messages: As the name suggests, these messages are sent to every device on a network.

Multicast Messages: These are a compromise between the previous two types: they are sent to a group of stations that meet a particular set of criteria.

Message Addressing Methods

Unicast Addressing: Unicast delivery requires that a message be addressed to a specific recipient.

Broadcast Addressing: Broadcasts are normally implemented via a special address that is reserved for that function.

Multicast Addressing: Multicasts are the most complex type of message because they require a means of identifying a set of specific devices to receive a message.

Key Concept: Three basic methods are used to address and transmit data between networked devices. A unicast transmission goes from one device to exactly one other; this is the "normal" method used for most message transactions. A broadcast transmission is sent from one device to all connected devices on a network. A multicast transmission is addressed and sent to a select group of devices.

Note: A new type of message addressing method was defined as part of IP version 6: the anycast message. This term identifies a message that should be sent to the closest member of a group of devices

Resource Sharing Roles and Structural Models

Peer-to-Peer Networking: -In a strict peer-to-peer networking setup, every computer is an equal, a peer in the network. Each machine can have resources that are shared with any other machine. There is no assigned role for any device, and each of the devices usually runs similar software.

Client/Server Networking: -In this design, a small number of computers are designated as centralized servers and given the task of providing services to a larger number of user machines called clients.

Key Concept: Networks are usually configured to share resources using one of two basic structural models. In a peer-to-peer network, each device is an equal and none are assigned particular jobs. In a client/server network, however, devices are assigned particular roles—a small number of powerful computers are set up as servers and respond to requests from the other devices, which are clients. Client/server computing also refers to the interaction between complementary

protocol elements and software programs and is rising in popularity due to its prevalence in TCP/IP and Internet applications.

Types and Sizes of Networks

Fundamental Network Classifications

Key Concept: Networks are often divided by size and general communication method into three classes. Local area networks (LANs) generally connect proximate devices, usually using cables. Wireless LANs (WLANs) are like cabled LANs but use radio frequency or light technology to connect devices without wires. Wide area networks (WANs) connect distant devices or LANs to each other. Campus area networks (CANs) and metropolitan area networks (MANs) fall between LANs and WANs in terms of overall size; personal area networks (PANs) are like very small LANs and often appear as wireless PANs (WPANs).

Segments, Networks, Subnetworks and Internetworks

Network: -A network can be of pretty much any size, from two devices to thousands.

Subnetwork: -A subnetwork is a portion of a network, or a network that is part of a larger internetwork.

Segment: - More often, however, the term “segment” implies something smaller than a subnetwork.

Internetwork(Internet): -This refers to a larger networking structure that is formed by connecting together smaller ones.

The Internet, Intranets and Extranets

The term intranet was coined to refer to an internal network that functioned like a “private Internet”. It comes from the prefix “intra”, which means “within”. Of course, “inter” is the opposite of “intra”, so this makes some people think that an

“intranet” is the opposite of an “internet”. In fact, most intranets are internetworks as well!

An extranet is an extended intranet, which is really a type of internet that works like the Internet. An extranet isn't public and open to all—it is controlled by a private organization. At the same time, it isn't entirely private either.

Network Performance Issues and Concepts

Key Concept: While performance is one of the most important characteristics of any network, there are others that are equally important. In many cases, the cost, quality, reliability, expandability, maintainability and other attributes of a network may in fact trade off against overall performance. The faster you want your network to go, the more difficult it is to ensure these other attributes are kept at sufficiently high levels.

Bandwidth: -Bandwidth is a widely-used term that usually refers to the data-carrying capacity of a network or data transmission medium.

Throughput: -Throughput is a measure of how much actual data can be sent per unit of time across a network, channel or interface.

Latency: -This very important, often overlooked term, refers to the timing of data transfers on a communications channel or network.

Key Concept: The units baud and bps are often treated equivalently, but are not the same. Baud measures not the throughput of a network but its signaling rate, meaning the number of times that the signal changes value in each second. Since modern encoding and modulation techniques often encode either greater or fewer than one bit value into each such transition, the throughput and baud rate of network technologies are usually different.

Key Concept: There are three basic operating modes that describe how data is sent between connected devices on a network. In simplex operation, data can flow in only one direction between two devices. Half-duplex networks allow any device to transmit, but only one may do so at a time. Full-duplex operation means two

attached devices can each transmit and receive simultaneously—this offers the greatest potential performance, since throughput is not decreased by forcing one device to wait for another before sending data.

International Networking Standards Organizations

International Organization for Standardization (ISO): Probably the biggest standards organization in the world, the ISO is really a federation of standards organizations from dozens of nations.

American National Standards Institute (ANSI): ANSI is the main organization responsible for coordinating and publishing computer and information technology standards in the United States.

Information Technology Industry Council (ITIC): ITIC is a group of several dozen companies in the information technology (computer) industry. ITIC is the SDO approved by ANSI to develop and process standards related to many computerrelated topics.

Institute of Electrical and Electronics Engineers (IEEE): The IEEE (pronounced "eye-triple-e") is a well-known professional organization for those in the electrical or electronics fields, including computers and networking.

Key Concept: A group of related organizations is responsible for the development of TCP/IP standards and Internet technologies. The Internet Society (ISOC) has overall responsibility for many Internet activities including standards development. It oversees the Internet Architecture Board (IAB), which makes high-level

decisions about Internet technology development. Most of the actual work of creating current Internet standards is performed by the Internet Engineering Task Force (IETF), which is managed by the Internet Engineering Steering Group (IESG). Longer-term research is done by the IETF's sibling organization, the Internet Research Task Force (IRTF), led by the Internet Research Steering Group (IRSG).

Backgrounder: Data Representation and the Mathematics of Computing

Key Concept: Formally, an octet is the correct term for exactly eight bits, while a byte is the smallest number of bits that can be accessed in a computer system, which may or may not equal eight. In practice, modern computers use 8-bit bytes, and the terms are used interchangeably (with byte being more common in North America, and octet often being preferred in Europe).

Note: As an interesting "sidebar", the term hexadecimal was not the first one used for base-16 numbers in computing. Originally, these were called sexadecimal numbers. This is actually the correct term, since Latin prefixes (sexa-) are normally used for numbers, not Greek ones (hexa-). However, in the early 1950s, IBM decided that the word "sexadecimal" was just a little too "provocative" for their tastes, so they changed it to hexadecimal.

OSI (Open System Interconnection)

Key Concept: The message used to communicate information for a particular protocol is called its protocol data unit (PDU) in OSI model terminology. That PDU is passed down to the next lower layer for transmission; since that layer is providing the service of handling that PDU, it is called the lower layer's service data unit (SDU). The SDU is encapsulated into that layer's own PDU and in turn sent to the next lower layer in the stack, proceeding until the physical layer is reached. The process is reversed on the recipient device. In summary: a layer N PDU is a layer N-1 SDU, which is encapsulated into a layer N-1 PDU.

Key Concept: In the OSI model, the process of routing occurs when data is sent not directly from transmitter to ultimate recipient, but indirectly through the use of an

intermediate system. That device, normally called a router, connects to two or more physical networks and thus has multiple interfaces to layer two. When it receives data, the data passes up only to the network layer, where it is repackaged and then sent on the next leg of its journey over the appropriate layer two interface

Physical Layer (Layer 1): -It is the only one where data is physically moved across the network interface. All the other layers perform useful functions to create messages to be sent, but they must all be transmitted down the protocol stack to the physical layer, where they are actually sent out over the network.

Data Link Layer (Layer 2): -The data link layer, also sometimes just called the link layer, is where many wired and wireless local area networking (LAN) technologies primarily function.

Data Framing: The data link layer is responsible for the final encapsulation of higher-level messages into frames that are sent over the network at the physical layer.

Key Concept: The second OSI Reference Model layer is the data link layer. This is the place where most LAN and wireless LAN technologies are defined. Layer two is responsible for logical link control, media access control, hardware addressing, error detection and handling, and defining physical layer standards. It is often divided into the logical link control (LLC) and media access control (MAC) sublayers, based on the IEEE 802 Project that uses that architecture.

Network Layer (Layer 3): -The network layer is the lowest one in the OSI model that is concerned with actually getting data from one computer to another even if it is on a remote network; in contrast, the data link layer only deals with devices that are local to each other.

Logical Addressing: Every device that communicates over a network has associated with it a logical address, sometimes called a layer three address.

Routing: Moving data across a series of interconnected networks is probably the defining function of the network layer. It is the job of the devices and software routines that function at the network layer to handle incoming packets from various sources, determine their final destination, and then figure out where they need to be sent to get them where they are supposed to go.

Datagram Encapsulation: The network layer normally encapsulates messages received from higher layers by placing them into datagrams (also called packets) with a network layer header.

Key Concept: The OSI Reference Model's third layer is called the network layer. This is one of the most important layers in the model; it is responsible for the tasks that link together individual networks into internetworks. Network layer functions include internetwork-level addressing, routing, datagram encapsulation, fragmentation and reassembly, and certain types of error handling and diagnostics. The network layer and transport layer are closely related to each other.

Transport Layer (Layer 4): -The transport layer's overall job is to provide the necessary functions to enable communication between software application processes on different computers. This encompasses a number of different but related duties.

Key Concept: The fourth and middle OSI Reference Model layer is the transport layer. This is another very important conceptual layer in the model; it represents the transition point between the lower layers that deal with data delivery issues, and the higher layers that work with application software. The transport layer is responsible for enabling end-to-end communication between application processes, which it accomplishes in part through the use of process-level addressing and multiplexing/demultiplexing. Transport layer protocols are responsible for dividing application data into blocks for transmission, and may be either

connection-oriented or connectionless. Protocols at this layer also often provide data delivery management services such as reliability and flow control.

Session Layer (Layer 5): -The name of this layer tells you much about what it is designed to do: to allow devices to establish and manage sessions. In general terms, a session is a persistent logical linking of two software application processes, to allow them to exchange data over a prolonged period.

Presentation Layer (Layer 6): -. It is different from the other layers in two key respects. First, it has a much more limited and specific function than the other layers; it's somewhat easy to describe, hurray! Second, it is used much less often than the other layers; in many types of connections it is not required.The presentation layer handles the job of hiding these differences between machines.Some types of encryption (and decryption) are performed at the presentation layer.

Application Layer (Layer 7): -The application layer provides services for user applications to employ. These programs are what implement the functions performed by users to accomplish various tasks over the network.

There are dozens of different application layer protocols that enable various functions at this layer. Some of the most popular ones include HTTP, FTP, SMTP, DHCP, NFS, Telnet, SNMP, POP3, NNTP and IRC

TCP/IP Protocol Suite and Architecture

TCP/IP was at one time just “one of many” different sets of protocols that could be used to provide network-layer and transport-layer functionality. Today there are still other options for internetworking protocol suites, but TCP/IP is the universally accepted world-wide standard.

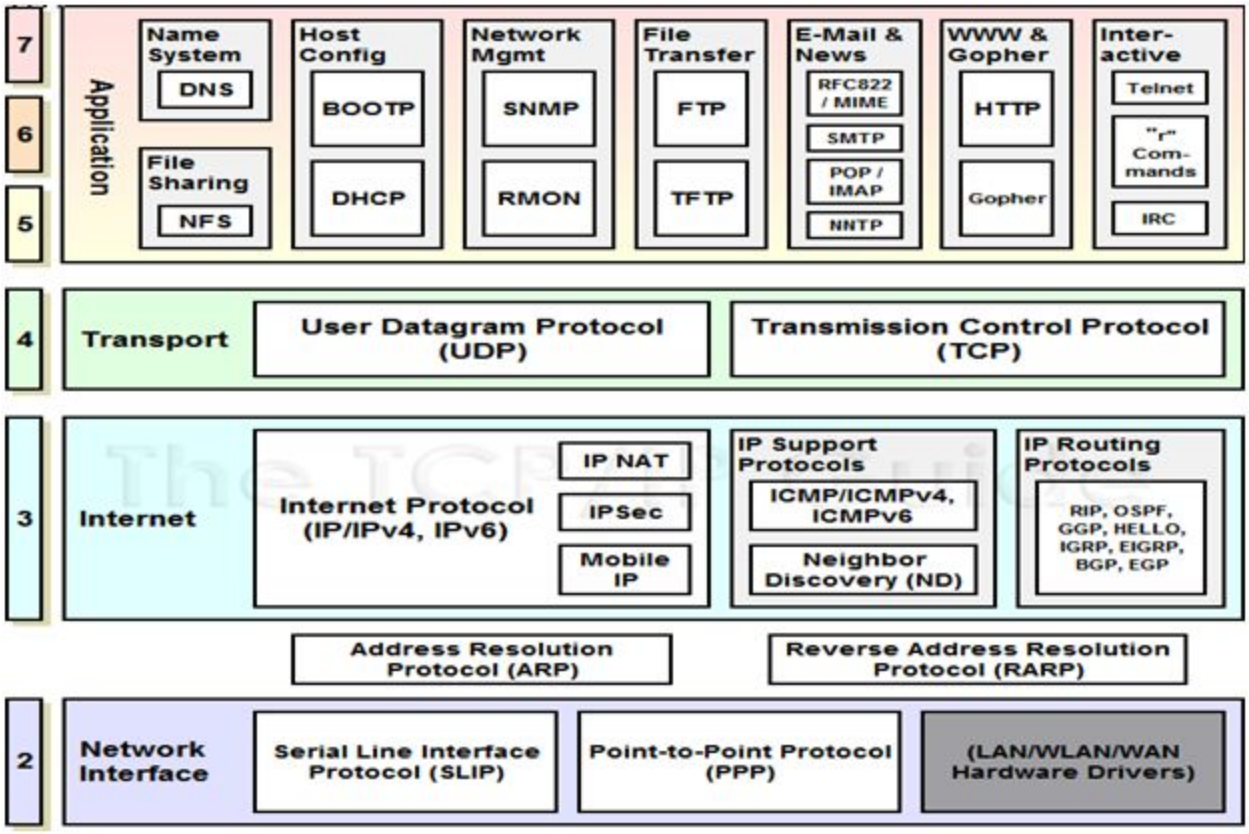
Key Concept: The TCP/IP protocol suite is strongly oriented around the notion of client/server network communication. Rather than all devices and protocol software elements being designed as peers, they are constructed as matched sets. Clients normally initiate communications by sending requests, and servers respond to such requests, providing the client with the desired data or an informative reply

Network Interface Layer: -This layer represents the place where the actual TCP/IP protocols running at higher layers interface to the local network.

Internet Layer: -It is responsible for typical layer three jobs, such as logical device addressing, data packaging, manipulation and delivery, and last but not least, routing. At this layer we find the Internet Protocol (IP), arguably the heart of TCP/IP, as well as support protocols such as ICMP and the routing protocols (RIP, OSFP, BGP, etc.) The new version of IP, called IP version 6, will be used for the Internet of the future and is of course also at this layer.

(Host-to-Host) Transport Layer: -This primary job of this layer is to facilitate end-to-end communication over an internetwork. It is in charge of allowing logical connections to be made between devices to allow data to be sent either unreliably (with no guarantee that it gets there) or reliably (where the protocols keep track of the data sent and received to make sure it arrives, and re-sends it if necessary). It is also here that identification of the specific source and destination application process is accomplished.

Application Layer: -This is the highest layer in the TCP/IP model. It is a rather broad layer, encompassing layers five through seven in the OSI model.



TCP/IP Network Interface Layer (OSI Data Link Layer) Protocols

Serial Line Internet Protocol (SLIP): - A very simple layer two protocol that provides only basic framing for IP datagrams for transmission across the serial line.

Point-to-Point Protocol (PPP): -It defines a complete method for robust data link connectivity between units using serial lines or other physical layers. It includes numerous capabilities and features, including error detection, compression, authentication, encryption and much more.

PPP Encapsulation Method: The primary job of PPP is to take higher-layer messages such as IP datagrams and encapsulate them for transmission over the underlying physical layer link

TCP/IP Network Interface / Internet "Layer Connection" Protocols

The third layer is the network layer, also called the internet layer in the TCP/IP model, where internetworking protocols are defined, the most notable being the Internet Protocol.

Address Resolution and the TCP/IP Address Resolution Protocol (ARP): -Communication on an internetwork is accomplished by sending data at layer three using a network layer address, but the actual transmission of that data occurs at layer two using a data link layer address.

Layer two addresses (such as IEEE 802 MAC addresses) are used for local transmissions between hardware devices that can communicate directly.Layer three addresses (most commonly Internet Protocol or IP addresses) are used in internetworking, to create the equivalent of a massive “virtual” network at the network layer which are indirectly connected devices.

TCP/IP Address Resolution Protocol (ARP): -The Address Resolution Protocol (ARP). ARP is a full-featured dynamic resolution protocol used to match IP addresses to underlying data link layer addresses. Originally developed for Ethernet, it has now been generalized to allow IP to operate over a wide variety of layer two technologies.

Key Concept: ARP was developed to facilitate dynamic address resolution between IP and Ethernet and can now be used on other layer two technologies as well. It works by allowing an IP device to send a broadcast on the local network,

requesting that another device on the same local network respond with its hardware address.

Proxy ARP: -If device A and device B are separated by a router, they would not be considered local to each other. Device A would not send directly to B or vice-versa; they would send to the router instead at layer two, and would be considered “two hops apart” at layer three.

A cannot send directly to B anyway. Instead, the router sends A its own hardware address. A then sends to the router, which forwards the message to B on the other network. Of course, the router also does the same thing on A's behalf for B, and for every other device on both networks, when a broadcast is sent that targets a device not on the same actual physical network as the resolution initiator.

Key Concept: Address resolution in IPv6 uses the new Neighbor Discovery (ND) protocol instead of the Address Resolution Protocol. A device trying to send an IPv6 datagram sends a Neighbor Solicitation message to get the address of another device, which responds with a Neighbor Advertisement. When possible, the request is sent using a special type of multicast address rather than broadcast, to improve efficiency.

Key Concept: The Reverse Address Resolution Protocol (RARP) is the earliest and simplest protocol designed to allow a device to obtain an IP address for use on a TCP/IP network. It is based directly on ARP and works in basically the same way, but in reverse: a device sends a request containing its hardware address and a device set up as an RARP server responds back with the device’s assigned IP address.

TCP/IP Internet Layer (OSI Network Layer) Protocols

Key Concept: While the Internet Protocol has many functions and characteristics, it can be boiled down to one primary purpose: the delivery of datagrams across an internetwork of connected networks.

IP Network Address Translation (IP NAT / NAT): This protocol provides IP address translation capabilities to allow private networks to be interfaced to public networks in a flexible manner.

Key Concept: IP addresses serve the dual function of device identification and routing. Each network interface requires one IP address, which is network specific. IP addresses can be either statically or dynamically allocated, and come in unicast, multicast and broadcast forms.

Key Concept: Since IP addresses are 32 bits long, the total address space of IPv4 is 2^32 or 4,294,967,296 addresses. However, not all of these addresses can be used, for a variety of reasons.

Key Concept:The "classful" IP addressing scheme divides the IP address space into five classes, A through E, of differing sizes. Classes A, B and C are the most important ones, designated for conventional unicast addresses and comprising 7/8ths of the address space. Class D is reserved for IP multicasting, and Class E for experimental use.

Key Concept: In the "classful" IP addressing scheme, a Class A network contains addresses for about 16 million network interfaces; a Class B about 65,000; and a Class C, 254.

Key Concept: Subnet addressing adds an additional hierarchical level to how IP addresses are interpreted, by dividing an organization's IP network into subnets.

This allows each organization to structure its address space to match its internal physical networks, rather than being forced to treat them a flat block. This solves a number of problems with the original “classful” addressing scheme, but requires changes to how addressing and routing work, as well as modifications to several TCP/IP protocols.

Key Concept: The number of hosts allowed in each subnet is the binary power of the number of host ID bits remaining after subnetting, less two. The reduction by two occurs because the all-zeroes and all-ones host IDs within each subnet are reserved for two “special meaning” addresses: to refer to the subnetwork itself and its local broadcast address. In some implementations, the number of subnets is also reduced by two because the all-zeroes and all-ones subnet IDs were originally not allowed to be used.

Key Concept: The delivery of IP datagrams is divided into two categories: direct and indirect. Direct delivery is possible when two devices are on the same physical network. When they are not, indirect delivery, more commonly called routing, is required to get the datagrams from source to destination. A device can tell which type of delivery is required by looking at the IP address of the destination, in conjunction with supplemental information such as the subnet mask that tells the device what network or subnet it is on.

The Benefits of Next-Hop Routing: - Indirect delivery of IP datagrams is accomplished using a process called next-hop routing, where each message is handed from one router to the next until it reaches the network of the destination. The main advantage of this is that each router needs only to know which neighbouring router should be the next recipient of a given datagram, rather than needing to know the exact route to every destination network.

IP Network Address Translation (NAT) Protocol

A very similar technique can be used for connecting an organization's computers to the Internet. In TCP/IP networks, this technology was first formalized in RFC 1631, The IP Network Address Translator (NAT), adopted in May 1994. The word "translator" refers to the device (router) that implements NAT.

Key Concept: IP Network Address Translation (IP NAT or NAT) is a technique that allows an organization to set up a network using private addresses, while still being able to communicate on the public Internet. A NAT-capable router translates private to public addresses and vice-versa as needed. This allows a small number of public IP addresses to be shared amongst a large number of devices, and provides other benefits as well (but also has some drawback).

IP NAT Static and Dynamic Address Mappings

When static mappings are used, a permanent, fixed relationship is defined between a global and a local representation of the address of either an inside or an outside device.

With dynamic mappings, global and local address representations are generated automatically by the NAT router, used as needed, and then discarded.

IP NAT Unidirectional (Traditional/Outbound) Operation

Key Concept: In unidirectional (traditional) NAT, the NAT router translates the source address of an outgoing request from inside local to inside global form. It then transforms the destination address of the response from inside global to inside local. The outside local and outside global addresses are the same in both request and reply.

IP NAT Bidirectional (Two-Way/Inbound) Operation

Key Concept: In traditional NAT, a transaction must begin with a request from a client on the local network, but in bidirectional (two-way/inbound) NAT, it is possible for a device on the public Internet to access a local network server. This requires the use of either static mapping or DNS to provide to the outside client the address of the server on the inside network. Then the NAT transaction is pretty much the same as in the unidirectional case, except in reverse: the incoming request has its destination address changed from inside global to inside local; the response has its source changed from inside local to inside global.

IP NAT Port-Based ("Overloaded") Operation: Network Address Port Translation (NAPT) / Port Address Translation (PAT)

The combination of an address and port uniquely identifies a connection. As a datagram passes from the private network to the public one, we can change not just the IP address but also the port number in the TCP or UDP header. The datagram will be sent out with a different source address and port. The response will come back to this same address and port combination (called a socket) and can be translated back again.

Key Concept: In port-based NAT, the NAT router translates the source address and port of an outgoing request from inside local to inside global form. It then transforms the destination address and port of the response from inside global to inside local. The outside local and outside global addresses are the same in both request and reply.

Key Concept: “Overlapping” NAT is used in situations where both the source and destination addresses in a datagram are private addresses or otherwise cannot be used regularly on the public Internet. In this case, unlike the other types of NAT,

the NAT router translates both the source and destination addresses of incoming and outgoing datagrams. On outgoing messages, inside local addresses are changed to inside global and outside local to outside global; on incoming messages, inside global addresses are changed to inside local and outside global to outside local.

IP Security (IPSec) Protocols

IPSec is not a single protocol, but rather a set of services and protocols that provide a complete security solution for an IP network. These services and protocols combine to provide various types of protection. Since IPSec works at the IP layer, it can provide these protections for any higher layer TCP/IP application or protocol without the need for additional security methods, which is a major strength. Some of the kinds of protection services offered by IPSec include:

Encryption of user data for privacy.

Authentication of the integrity of a message to ensure that it is not changed en route.

Protection against certain types of security attacks, such as replay attacks.

The ability for devices to negotiate the security algorithms and keys required to meet their security needs.

Two security modes, tunnel and transport, to meet different network needs

IPSec Authentication Header (AH): -This protocol provides authentication services for IPSec. What this means is that it allows the recipient of a message to verify that the supposed originator of a message was in fact the one that sent it.

Encapsulating Security Payload (ESP): The Authentication Header ensures integrity of the data in datagram, but not its privacy.

IPSec Implementation Methods

End Host Implementation

Putting IPSec into all host devices provides the most flexibility and security. It enables “end-to-end” security between any two devices on the network.

Router Implementation

It only provides protection between pairs of routers that implement IPSec The routers can be used to provide protection only for the portion of the route that datagrams take outside the organization, leaving connections between routers and local hosts unsecured

IPSec Architectures

Integrated Architecture

we would integrate IPSec's protocols and capabilities directly into IP itself no extra hardware or architectural layers are needed.

IPv6 was design to support IPSec, making this a viable option for hosts or routers. With IPv4, integration would require making changes to the IP implementation on each device, which is often impractical.

“Bump In The Stack” (BITS) Architecture

In this technique, IPSec is made a separate architectural layer between IP and the data link layer; the cute name refers to the fact that IPSec is an extra element in the networking protocol stack.

“Bump In The Wire” (BITW) Architecture

In this method we add a hardware device that provides IPSec services.

IPSec Modes: Transport and Tunnel

Key Concept: IPSec has two basic modes of operation. In transport mode, IPSec AH and/or ESP headers are added as the original IP datagram is created; this mode is associated with integrated IPSec architectures. In tunnel mode, the original IP datagram is created normally, then the entire datagram is encapsulated into a new IP datagram containing the AH/ESP IPSec headers. This mode is most commonly used with "Bump In The Stack" and "Bump In The Wire" implementations

Internet Protocol Mobility Support (Mobile IP)

To support IP in a mobile environment, a new protocol called IP Mobility Support, or more simply, Mobile IP, was developed.

Key Concept: The basic problem with supporting mobile devices in IP internetworks is that routing is performed using the IP address, which means the IP address of a device is tied to the network where that device is located. If a device changes networks, data sent to its old address cannot be delivered by conventional means. Traditional workarounds such as routing by the full IP address or changing IP addresses manually often create more problems.

Key Concept: Mobile IP solves the problems associated with devices that change network locations, by setting up a system where datagrams sent to the mobile node's home location are forwarded to it wherever it may be located. It is particularly useful for wireless devices but can be used for any device that moves between networks periodically.

Key Concept: Mobile IP agent discovery is the process by which a mobile node determines where it is located and establishes contact with a home or foreign agent. Routers that can function as agents regularly send Agent Advertisement messages to indicate their capabilities, which are modified versions of regular Router

Advertisements. A mobile node can also send an Agent Solicitation to request the sending of an Advertisement, which is the same as regular Router Solicitation.

Key Concept: Mobile IP home agent registration is the process by which a mobility binding is created between a home agent and a traveling mobile node to enable datagram forwarding to be performed. Registration is performed by the mobile node sending a Registration Request message, and the home agent returning a Registration Reply. The foreign agent may be required to act as a "middleman" to facilitate the transaction, but is otherwise not involved.

Internet Control Message Protocol(ICMP)

In TCP/IP, the Internet Protocol is the executive, and ICMP is its "administrative assistant". IP focuses on its core activities, such as addressing, datagram packaging and routing. ICMP provides critical support to IP in the form of ICMP messages that allow different types of communication to occur between IP devices.These messages use a common general format, and are encapsulated in IP datagrams for transmission.

However, in the standard that first defined it, ICMP is specifically declared to be not only part of the network layer, but: "actually an integral part of IP, [that] must be implemented by every IP module".

Key Concept: In TCP/IP, diagnostic, test and error-reporting functions at the internet/network layer are performed by the Internet Control Message Protocol (ICMP), which is like the Internet Protocols "administrative assistant". The original version, now called ICMPv4, is used with IPv4, and the newer ICMPv6 with IPv6.

Key Concept: ICMP is not like most other TCP/IP protocols in that it does not perform a specific task. It defines a mechanism by which various control messages can be transmitted and received to implement a variety of functions.

One interesting general characteristic of ICMP's operation is that when errors are detected, they can be reported using ICMP, but only back to the original source of a datagram. This is actually a big drawback in how ICMP works.

Key Concept: ICMP error-reporting messages sent in response to a problem seen in an IP datagram can only be sent back to the originating device. Intermediate devices cannot be the recipient of an ICMP message because their addresses are normally not carried in the IP datagram's header.

Key Concept: ICMP messages are divided into two general categories: error messages that are used to report problem conditions, and informational messages that are used for diagnostics, testing and other purposes.

Key Concept: A total of 256 different possible message types can be defined for each of ICMPv4 and ICMPv6. The Type field that appears in the header of each message specifies the kind of ICMP message. In ICMPv4 there is no relationship between Type value and message type; in ICMPv6 error messages have a Type value of 0 to 127, informational messages 128 to 255.

An ICMP error message must not be generated in response to any of the following:

An ICMP Error Message: This prevents loops of the type just mentioned. Note, however, that an ICMP error message can be generated in response to an ICMP informational message.

A Broadcast or Multicast Datagram:What would happen if a datagram were broadcast to 5,000 hosts and each of them found an error in it and tried to send a report back to the source? Something unpleasant.

Key Concept: In order to prevent excessive numbers of ICMP messages from being sent on a network, a special set of rules is put into place to govern when and how they may be created. Most of these are designed to eliminate situations where

very large numbers of ICMP error messages would be generated in response to certain occurrences.

Error Message(ICMPv4)

Key Concept: ICMPv4 Destination Unreachable messages are used to inform a sending device of a failure to deliver an IP datagram. The message's Code field provides information about the nature of the delivery problem.

Key Concept: ICMPv4 Source Quench messages are sent by a device to request that another reduce the rate at which it is sending datagrams. They are a rather crude method of flow control compared to more capable mechanisms such as that provided by TCP.

Key Concept: ICMPv4 Time Exceeded messages are sent in two different "time related" circumstances. The first is if a datagram's Time To Live (TTL) field is reduced to zero, causing it to expire and the datagram to be dropped. The second is when all the pieces of a fragmented message are not received before the expiration of the recipient's reassembly timer

Key Concept: ICMPv4 Redirect messages are used by a router to inform a host of a preferred router to use for future datagrams sent to a particular host or network. They are not used to alter routes between routers.

Key Concept: The ICMPv4 Parameter Problem message is a generic "catch all" that can be used to convey an error of any type in an IP datagram. A special Pointer field is normally used to indicate to the recipient of the message where the problem was in the original datagram.

Informational Message (ICMPv4)

Key Concept: IICMPv4 Echo (Request) and Echo Reply messages are used to facilitate network reachability testing. A device can test its ability to perform basic communication with another one by sending an Echo message and waiting for an

Echo Reply to be returned by the other device. The ping utility, a widely-used diagnostic tool in TCP/IP internetworks, makes use of these messages.

Key Concept: ICMP Router Advertisement messages are sent regularly by IP routers to inform hosts of their presence and characteristics, so hosts know to use them for delivery of datagrams to distant hosts. A host that is new to a network and wants to find out immediately what routers are present may send a Router Solicitation, which will prompt listening routers to send out Router Advertisements.

TCP/IP IPv6 Neighbor Discovery Protocol (ND)

Key Concept: The new IPv6 Neighbor Discovery protocol formalizes for IPv6 a number of functions related to communication between devices on a local network that are performed in IPv4 by protocols such as ARP and ICMP. ND is considered another "helper" protocol for IPv6, and is closely related to ICMPv6.

Key Concept: The Neighbor Discovery protocol encompasses nine individual functions, many of which are related to each other. They are organized into three functional groups: host-router discovery functions, host-host communication functions, and the redirect function.

Key Concept: One of the two main functional groups of the Neighbor Discovery protocol is the set of host-router discovery functions. They allow hosts on a local network to discover the identity of a local router and learn important parameters about how the network is to be used. Host-router discovery operations are performed using ICMPv6 Router Advertisement and Router Solicitation messages.

Key Concept: The second of the two main functional groups of the Neighbor Discovery protocol is the set of host-host communication functions. Two ICMPv6 messages are defined, Neighbor Advertisement and Neighbor Solicitation, which enable a variety of types of essential communication between adjacent hosts on a

local network. These include address resolution, determining the next hop to which a datagram should be sent, and also the assessment of a neighboring device's reachability.

TCP/IP Routing Protocols (Gateway Protocols)

Routing is not just one of the most important activities that takes place at the network layer: it is the function that really defines layer three of the OSI Reference Model. Routing is what enables small local networks to be linked together to form potentially huge internetworks that can span cities, countries or even the entire globe. It is the job done by special devices called routers, which forward datagrams from network to network, allowing any device to send to any other even if the source has no idea where the destination is.

Key Concept: Large, modern TCP/IP internetworks can contain thousands of routers. To better manage routing in such an environment, routers are grouped into constructs called autonomous systems. Each autonomous system (AS) consists of a group of routers managed independently by a particular organization or entity.

Key Concept: Interior routing protocols are used to share routing information within an autonomous system; each AS may use a different interior routing protocol because the system is, as the name says, autonomous. Exterior routing protocols convey routing data between autonomous systems; each AS must use the same exterior protocol to ensure that they can communicate.

TCP/IP Interior Routing Protocols (RIP, OSPF, GGP, HELLO, IGRP, EIGRP)

The routing protocols used to facilitate the exchange of routing information between routers within an AS are called interior routing protocols (or historically, interior gateway protocols).

TCP/IP Routing Information Protocol (RIP, RIP-2 and RIPng)

The Routing Information Protocol (RIP) has been the most popular interior routing protocol in the TCP/IP suite for many years published in June 1988.

On a regular basis, each router in the internetwork sends out its routing table in a special message on each of the networks to which it is connected, using UDP.

Other routers receive these tables and use them to update their own tables. This is done by taking each of the routes they receive and adding an extra hop.

RIP only supports a maximum of 15 hops between destinations, making it unsuitable for very large autonomous systems, and this cannot be changed.

Key Concept: The Routing Information Protocol (RIP) is one of the oldest and most popular interior routing protocols. It uses a distance-vector algorithm with each router maintaining a table indicating how to reach various networks in the autonomous system and the distance to it in hops. RIP is popular because it is well-established and simple,but has a number of important limitations.

Development of RIP Version 2 (RIP-2) and RIPng for IPv6

RIP-2 was created in the early 1990s. RIP-2 defines a new message format for RIP and includes a number of new features, including support for classless addressing,

authentication, and the use of multicasting instead of broadcasting to improve network performance.

In order to ensure that RIP can work with TCP/IP in the future, it was necessary to define a version that would work with the new Internet Protocol version 6 (IPv6). In 1997, RFC 2080 was published, titled RIPng for IPv6. The ng stands for next generation—recall that IPv6 is also sometimes called IPng.

RIP Routing Information and Route Distance Metric

The job of RIP, like any routing protocol, is to provide a mechanism for exchanging information about routes so routers can keep their routing tables up to date. For each network or host, the device includes a variety of information, of which the following are the most important:

1. The address of the network or host.

2. The distance from that router to the network or host.

3. The first hop for the route: the device to which datagrams must first be sent to eventually get to the network or host.

Key Concept: Routing information is propagated between routers in RIP using a simple algorithm. On a regular basis, each router sends out RIP messages that specify what networks it can reach, and how many hops to reach them. Other routers directly connected to that one know that they can then reach those networks through that router at a cost of one additional hop. So if router A sends a message saying it can reach network X for a cost of N hops, each other router that connects directly to A can reach network X for a cost of N+1 hops. It will put that information into its routing table, unless it knows of an alternate route through another router that has a lower cost.

The RIP software in each router sends messages and takes other actions both in reaction to certain events and in response to triggers set off by timers. Timers are

also used to determine when routing information should be discarded if not updated.

RIP messagesare sent using the User Datagram Protocol (UDP), with reserved UDP port number 520 for RIP-1 and RIP-2, and 521 for RIPng.

Key Concept: RIP uses two basic message types, the RIP Request and RIP Response, both of which are sent using the User Datagram Protocol (UDP). RIP Response messages, despite their name, are used both for routine periodic routing table updates as well as to reply to RIP Request messages. Requests are sent only in special circumstances, such as when a router first joins a network.

RIP Protocol Limitations and Problems

Slow Convergence:- It takes a long time for all routers to get the same information, and in particular, it takes a long time for information about topology changes to propagate.

Routing Loops: - A routing loop occurs when Router A has an entry telling it to send datagrams for Network 1 to Router B, and Router B has an entry saying that datagrams for Network 1 should be sent to Router A.

"Counting To Infinity": -A special case of slow convergence can lead to a routing loop situation where one router passes bad information to another router, which sends more bad information to another router and so on. This causes a situation where the protocol is sometimes described as unstable.; the problem is called counting to infinity.

"Small Infinity": -The use of a relatively small value of "infinity" limits the slow convergence problem. Even in a situation where we "count to infinity" as we just saw, the total amount of time elapsed is at least manageable; imagine if "infinity" were defined as say, 1000! Unfortunately, the drawback of this is that it limits the size of the internetwork that can be used for RIP.

Key Concept: Four special features represent changes to RIP operation that ameliorate or eliminate the problems with the operation of the basic protocol. Split

horizon and split horizon with poisoned reverse prevent having a router send invalid route information back to the router from which it originally learned the route. Triggered updates reduce the slow convergence problem by causing immediate propagation of changed route information. Finally, hold-down may be used to provide robustness when information about a failed route is received.

Key Concept: RIP-2 is the most recent version of RIP used in IPv4. It includes a number of enhancements over the original RIP-1, including support for subnet masks and classless addressing, explicit next-hop specification, route tagging, authentication and multicast. For compatibility, it uses the same basic message format as RIP-1, putting the extra information required for its new features into some of the unused fields of the RIP-1 message format.

Key Concept: RIPng is the version of RIP that was developed for use on IPv6 internetworks. It is technically a distinct protocol from RIP-1 and RIP-2 but is very similar to both. It retains the enhancements to RIP made in RIP-2, making changes to these features and to the RIP message format where needed for compatibility with IPv6.

Open Shortest Path First (OSPF)

A new routing protocol was developed in the late 1980s that uses the more capable (and more complex) link-state or shortest path first routing algorithm. This protocol is called Open Shortest Path First (OSPF). It fixes many of the issues with RIP and allows routes to be selected dynamically based on the current state of the network, not just a static picture of how routers are connected.

Key Concept: Open Shortest Path First (OSPF) was developed in the late 1980s to provide a more capable interior routing protocol for larger or more complex autonomous systems that were not being served well by RIP. It uses the dynamic shortest path first or link state routing algorithm, with each router maintaining a database containing information about the state and topology of the internetwork. As changes to the internetwork occur, routers send out updated state information,

which allows each router to dynamically calculate the best route to any network at any point in time. OSPF is a complement to RIP in that RIP is simple but limited, where OSPF is more capable but more complicated.

When OSPF basic topology is used, all the routers in the AS function as peers. Each router communicates routing information with each other one, and each maintains a copy of the key OSPF data structure: the link-state database (LSDB).

Key Concept: In basic OSPF topology, each of the routers running OSPF is considered a peer of the others. Each maintains a link-state database (LSDB) that contains information about the topology of the entire autonomous system. Each link between a router and network or between two routers is represented by an entry in the LSDB that indicates the cost to send data over the link. The LSDB is updated regularly through the exchange of OSPF link-state advertisements (LSAs).

Key Concept: To allow better control and management over larger internetworks, OSPF allows a large autonomous system to be structured into a hierarchical form. Contiguous routers and networks are grouped into areas that connect together using a logical backbone. These areas act as the equivalent of smaller autonomous systems within the larger AS, yielding the same benefits of localized control and traffic management that autonomous systems provide for a large internetwork between organizations.

Router Roles in OSPF Hierarchical Topology

There are three different labels applied to routers in this configuration:

1. *Internal Routers:* - These are routers that are only connected to other routers or networks within a single area. They maintain an LSDB for only that area, and really have no knowledge of the topology of other areas.

2. *Area Border Routers:* - These are routers that connect to routers or networks in more than one area. They maintain an LSDB for each area of which they are a part. They also participate in the backbone.

3. *Backbone Routers:* - These are routers that are part of the OSPF backbone. By definition, this includes all area border routers, since those routers pass routing information between areas.

Key Concept: To determine what routes it should use to reach networks in its autonomous system, a router generates a shortest path first tree (SPF tree) from its linkstate database. This tree contains the same basic information as the LSDB but presents it from the point of view of the router doing the calculation so that router can see the costs of various paths to different networks.

OSPF Message Types: -Unlike RIP, OSPF does not send its information using the User Datagram Protocol (UDP). Instead, OSPF forms IP datagrams directly, packaging them using protocol number 89 for the IP Protocol field.

1. Hello: - These messages are used as a form of greeting, to allow a router to discover other adjacent routers on its local links and networks.

2. Database Description: - These messages contain descriptions of the topology of the AS or area. That is, they convey the contents of the link-state database for the autonomous system or area from one router to another.

3. Link State Request: - These messages are used by one router to request updated information about a portion of the LSDB from another router.

4. Link State Update: - These messages contain updated information about the state of certain links on the LSDB.

5. Link State Acknowledgment: - These messages provide reliability to the link-state exchange process, by explicitly acknowledging receipt of a Link State Update message.

Key Concept: The operation of OSPF involves five message types. Hello messages are used to establish contact between routers, and Database Description messages to initialize a router's link-state database. Routine LSDB updates are sent using Link State Update messages, which are acknowledged using Link State Acknowledgments. A device may also request a specific update using a Link State Request.

Intermediate System - Intermediate System (IS-IS)

The IS-IS protocol is one of a family of IP Routing protocols, and is an Interior Gateway Protocol (IGP) for the Internet, used to distribute IP routing information throughout a single Autonomous System (AS) in an IP network.

IS-IS is a link-state routing protocol, which means that the routers exchange topology information with their nearest neighbors. The topology information is flooded throughout the AS, so that every router within the AS has a complete picture of the topology of the AS. This picture is then used to calculate end-to-end paths through the AS, normally using a variant of the Dijkstra algorithm.

The main advantage of a link state routing protocol is that the complete knowledge of topology allows routers to calculate routes that satisfy particular criteria.

Increasing the number of routers increases the size and frequency of the topology updates, and also the length of time it takes to calculate end-to-end routes.

TCP/IP Gateway-to-Gateway Protocol (GGP)

GGP is similar in general operation to the Routing Information Protocol (RIP) in that it uses a distance-vector algorithm to determine the best routes between devices. Like RIP, the metric is a simple hop count, so GGP will select a route with the shortest number of hops.

Every 15 seconds, the router sends a GGP Echo message to each of its neighbors. If the neighbor receives the message, it responds with a GGP Echo Reply message.

Key Concept: The Gateway-to-Gateway Protocol (GGP) was used to communicate route information between core routers on the early Internet. It is a distance-vector protocol that operates in a manner very similar to RIP. Each router periodically sends out its routing table to neighboring routers, so each router can learn the cost, in hops, to reach every network in the autonomous system. GGP is now considered a historical protocol and is no longer part of TCP/IP.

The HELLO Protocol (HELLO)

The HELLO protocol uses a distance-vector algorithm, like the Routing Information Protocol (RIP) and the Gateway-to-Gateway Protocol (GGP). What's interesting about it, however, is that unlike RIP and GGP, HELLO does not use hop count as a metric. Instead, it attempts to select the best route by assessing network delays and choosing the path with the shortest delay.

Interior Gateway Routing Protocol (IGRP)

Like RIP, IGRP is a distance-vector routing protocol designed for use with an autonomous system, and thus uses the same basic mechanism for route determination. Each router routinely sends out on each local network to which it is attached a message containing a copy of its routing table.

An important difference between RIP and IGRP, however, is that where RIP only allows the cost to reach a network to be expressed in terms of hop count, IGRP provides a much more sophisticated metric.

Key Concept: In the 1980s, Cisco Systems created the Interior Gateway Routing Protocol (IGRP) as an improvement over the industry standard protocol RIP. IGRP is a distance-vector protocol like RIP and similar to it in many ways, but includes several enhancements. Amongst the most important of these is an elimination of the 15 hop limit between routers, and the ability to use metrics other than hop count to determine optimal routes.

Enhanced Interior Gateway Routing Protocol (EIGRP)

EIGRP is still a distance-vector protocol, but is more sophisticated than other distance-vector protocols like IGRP or RIP, and includes certain features that are more often associated with link-state routing protocols like OSPF than distance-vector algorithms.

EIGRP is based on a new route calculation algorithm called the Diffusing Update Algorithm (DUAL), developed at a company called SRI International by Dr. J. J. GarciaLuna-Aceves.

DUAL differs from a typical distance-vector algorithm primarily in that it maintains more topology information about the internetwork than that used by protocols like RIP or IGRP. It uses this information to automatically select least-cost, loop-free routes between networks.

TCP/IP Exterior Gateway/Routing Protocols (BGP and EGP)

If interior routing protocols are used within ASes, we need another set of routing protocolsto send that information between ASes. These are called exterior routing protocols.

The details of what happens within an AS are hidden from other ASes, which allows the administrator of an AS to have the independence to control how he or she runs it, including the selection of one or more from a variety of different interior routing protocols.. In contrast, to reliably connect ASes together, it is essential that each one be running the same exterior routing protocol.

The result of this is that in TCP/IP there is generally only one exterior routing protocol in widespread use at a given time.

TCP/IP Border Gateway Protocol (BGP/BGP-4)

BGP is a critically important protocol to the operation of larger internetworks and the Internet itself. It is the "glue" that binds smaller internetworks (autonomous systems) together, and it ensures that every organization is able to share routing information.

"To exchange network reachability information between autonomous systems and from this information determine routes to networks".

It is also possible to use BGP to communicate between BGP routers within the same autonomous system.

Key Concept: The exterior routing protocol used in modern TCP/IP internetworks is the Border Gateway Protocol (BGP). Initially developed in the late 1980s as a successor to EGP, BGP has been revised many times; the current version is 4, so BGP is also commonly called BGP-4. BGP's primary function is the exchange of

network reachability information between autonomous systems to allow each AS on an internetwork to send messages efficiently to every other one.

Key Concept: BGP supports an arbitrary topology of autonomous systems. Each autonomous system using BGP assigns one or more routers to implement the protocol. These devices then exchange messages to establish contact with each other and share information about rates through the internetwork using TCP. BGP employs a sophisticated path vector route calculation algorithm that determines routes from path attributes that describe how different networks can be reached.

Key Concept: Each router configured to use BGP is called a BGP speaker; these devices exchange route information using the BGP messaging system. Routers that only connect to other routers in the same autonomous system are called internal routers, while those that connect to other ASes are border routers. Neighboring BGP speakers in the same AS are called internal peers, while those in different ASes are external peers.

Key Concept: One important issue in BGP is how to handle the flow of traffic between autonomous systems. Each autonomous system in a BGP internetwork is either a stub AS, if it connects to only one other AS, or a multihomed AS if it connects to two or more others. BGP allows the administrators of a multihomed AS to establish routing policies that specify under what conditions the AS is willing to handle transit traffic (messages sent over the AS whose source and destination are both external to that AS.)

The heart of BGP's system of routing information management and handling is the database where routes are stored. This database is collectively called the Routing Information Base (RIB), but it is in fact not a monolithic entity.

Key Concept: The routine operation of BGP requires BGP speakers to store, update, select and advertise routing information. The central data structure used for this purpose is the BGP Routing Information Base (RIB). The RIB actually consists of three sections: a set of input databases (Adj-RIBs-In) that hold routing information received from peers, a local database (Loc-RIB) that contains the router's current routes, and a set of output databases (Adj-RIBs-Out) used by the router to send its routing information to other routers.

The information about the path to each route is stored in the Routing Information Base (RIB) of each BGP speaker in the form of BGP path attributes. These attributes are used to advertise routes to networks when BGP devices send out Update messages.

Key Concept: Unlike simpler routing protocols that store only limited information about how to reach a network, such as its cost and the next hop to reach it, BGP stores detailed information about complete routes to various networks. This information takes the form of path attributes that describe various characteristics of a path (route) through the ASes that connect a router to a destination network.

Key Concept: The method used by a BGP speaker to determine what new routes to accept from its peers and what routes to advertise back them is called the BGP Decision Process. It is a complex algorithm in three phases that involves the computation of the best route based on both pre-existing and incoming path information.

Key Concept: As an exterior routing protocol, BGP operates at the autonomous system level. Its routes are calculated based on paths between ASes, not individual routers. Since BGP, by definition, does not know the internal structure of routers within an AS, it cannot know for certain the cost to send a datagram across a given AS. This in turn means that BGP cannot always guarantee that it will select the absolute lowest-cost route between any two networks.

Key Concept: BGP is implemented through the exchange of four different message types between BGP speakers. A BGP session begins with a TCP connection being established between two routers and each sending an Open message to the other. BGP Update messages are the primary mechanism by which routing information is exchanged between devices. Small BGP Keepalive messages are used to maintain communication between devices between periods that they need to exchange information. Finally, Notification messages are used for problem reporting.

Key Concept: All four BGP message types use a general message format that contains three fixed header fields—Marker, Length and Type—and room for a message body that differs for each message type. The large Marker field is used to denote the start of a new BGP message, and is also used to facilitate the BGP authentication method

Key Concept: BGP sessions begin with each peer in a connection sending the other a BGP Open message. The purpose of this message is to establish contact between devices, identify the sender of the message and its autonomous system, and negotiate important parameters that dictate how the session will be conducted.

Key Concept: The most important message type in BGP is the Update message, which is used to send detailed information about routes between BGP devices. It uses a complex structure that allows a BGP speaker to efficiently specify new routes, update existing ones, and withdraw routes that are no longer valid. Each message may include the full description of one existing route, and may also withdraw from use a list of multiple routes.

Key Concept: BGP Keepalive messages are sent periodically during idle periods when no real information needs to be sent between connected BGP speakers. They

serve only to keep the session alive, and thus contain only a BGP header and no data.

Key Concept: BGP Notification messages are used for error reporting between BGP peers. Each message contains an Error Code field that indicates what type of problem occurred. For certain Error Codes, an Error Subcode field provides additional details about the specific nature of the problem. Despite these field names, Notification messages are also used for other types of special non-error communication, such as terminating a BGP connection.

TCP/IP Exterior Gateway Protocol (EGP)

EGP was developed by Internet pioneers Bolt, Beranek and Newman (BBN) in the early 1980s, published in October 1982.

EGP is responsible for communication of network reachability information between neighboring routers, which may or may not be in different autonomous systems. The operation of EGP is somewhat similar to that of BGP. Each EGP router maintains a database of information regarding what networks it can reach and how to reach them. It sends this information out on a regular basis to each router to which it is directly connected. Routers receive these messages and update their routing tables, and then use this new information to update other routers. Information about how to reach each network propagates across the entire internetwork.

Routing Information Exchange Process

Neighbor Acquisition: - Each router attempts to establish a connection to each of its neighboring routers by sending Neighbor Acquisition Request messages. A

neighbor hearing a request can respond with a Neighbor Acquisition Confirm to say that it recognized the request and wishes to connect.

Neighbor Reachability:- After acquiring a neighbor, a router checks to make sure the neighbor is reachable and functioning properly on a regular basis. This is done by sending an EGP Hello message to each neighbor for which a connection has been established.

Network Reachability Update: - A router sends Poll messages on a regular basis to each of its neighbors. The neighbor responds with an Update message, which contains details about the networks that it is able to reach.

It may be sent by a neighbor in response to receipt of an EGP message either when the message itself has a problem. Unlike the BGP Notification message, an EGP router does not necessarily close the connection when sending an Error message.

TCP/IP Transport Layer Protocols

The transport layer, called the host-to-host transport layer in the TCP/IP model. Itresides in the very architectural center of the modeland represents an important transition point between the hardware-associated layers below it thatdo the “grunt work”, and the layers above that are more software-oriented and abstract.

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)

TCP/IP includes a large set of protocols that operate at the network layer and above. The suite as a whole is anchored at layer three by the Internet Protocol (IP), which many people consider the single most important protocol in the world of networking.

IP is the protocol that performs the bulk of the functions needed to make an internetwork. In TCP/IP these tasks are performed by a pair of protocols that

operate at the transport layer: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

TCP and UDP Overview and Role In TCP/IP

The most important of these are that IP is connectionless, unreliable and unacknowledged. Data is sent over an IP internetwork without first establishing a connection, using a "best effort" paradigm.

Applications need to be able to count on the fact that the data they send will get to its destination without loss or error."In the beginning" there was just a single protocol called "TCP" that combined the tasks of the Internet Protocol with the reliability and session management features just mentioned.

Key Concept: To suit the differing transport requirements of the many TCP/IP applications, two TCP/IP transport layer protocols exist. The Transmission Control Protocol (TCP) is a full-featured, connection-oriented protocol that provides acknowledged delivery of data while managing traffic flow and handling issues such as congestion and transmission loss. The User Datagram Protocol (UDP), in contrast, is a much simpler protocol that concentrates only on delivering data, to maximize the speed of communication when the features of TCP are not required.

Key Concept: Most classical applications, especially ones that send files or messages, require that data be delivered reliably, and therefore use TCP for transport. Applications using UDP are usually those where loss of a small amount of data is not a concern, or that use their own application-specific procedures for dealing with potential delivery problems that TCP handles more generally.

Key Concept: TCP/IP is designed to allow many different applications to send and receive data simultaneously using the same Internet Protocol software on a given device. To accomplish this it is necessary to multiplex transmitted data from many sources as it is passed down to the IP layer. As a stream of IP datagrams is received, it is demultiplexed and the appropriate data passed to each application software instance on the receiving host.

Key Concept: Well-known and registered port numbers are needed for server processes since a client must know the server's port number to initiate contact. In contrast, client processes can use any port number. Each time a client process initiates a UDP or TCP communication it is assigned a temporary, or ephemeral, port number to use for that conversation. These port numbers are assigned in a pseudo-random way, since the exact number used is not important, as long as each process has a different number.

Key Concept: Each device may have multiple TCP connections active at any given time. Each connection is uniquely identified using the combination of the client socket and server socket, which in turn contains four elements: the client IP address and port, and the server IP address and port.

Key Concept: The User Datagram Protocol (UDP) was developed for use by application protocols that do not require reliability, acknowledgment or flow control features at the transport layer. It is designed to be simple and fact, providing only transport layer addressing in the form of UDP ports and an optional checksum capability, and little else.

Key Concept: UDP is most often used by a protocol instead of TCP in two situations. The first is when an application values timely delivery over reliable delivery, and where TCP's retransmission of lost data would be of limited or even no value. The second is when a simple protocol is able to handle the potential loss of an IP datagram itself at the application layer using a timer/retransmit strategy, and where the other features of TCP are not required. UDP is also used for applications that require multicast or broadcast transmissions, since these are not supported by TCP.

Functions Performed By TCP

Addressing/Multiplexing: - An important job for TCP is multiplexing the data received from these different processes so they can be sent out using the underlying network-layer protocol. At the same time, these higher-layer application processes are identified using TCP ports.

Connection Establishment and Termination: - TCP provides a set of procedures that devices follow to negotiate and establish a TCP connection over which data can travel.

Data Handling and Packaging :-TCP defines a mechanism by which applications are able to send data to it from higher layers. This data is then packaged into messages to be sent to the destination TCP software. The destination software unpackage the data and gives it to the application on the destination machine.

Data Transfer: -Conceptually, the TCP implementation on a transmitting device is responsible for the transfer of packaged data to the TCP process on the other device.

Providing Reliability and Transmission Quality Services: -TCP includes a set of services and features that allow an application to consider the sending of data using the protocol to be "reliable".

Providing Flow Control and Congestion Avoidance Features:- TCP allows the flow of data between two devices to be controlled and managed.

Key Concept: A basic technique for ensuring reliability in communications uses a rule that requires a device to send back an acknowledgment each time it successfully receives a transmission. If a transmission is not acknowledged after a period of time, it is retransmitted by its sender. This system is called positive acknowledgment with retransmission (PAR). One drawback with this basic scheme is that the transmitter cannot send a second message until the first has been acknowledged.

Key Concept: The normal process of establishing a connection between a TCP client and server involves three steps: the client sends a SYN message; the server sends a message that combines an ACK for the client's SYN and contains the server's SYN; and then the client sends an ACK for the server's SYN. This is called the TCP three-way handshake.

There are two types of the connection termination

1. Normal Connection Termination: -

Key Concept: A TCP connection is normally terminating using a special procedure where each side independently closes its end of the link. It normally begins with one of the application processes signalling to its TCP layer that the session is no longer needed. That device sends a FIN message to tell the other device that it wants to end the connection, which is acknowledged. When the responding device is ready, it too sends a FIN that is acknowledged; after waiting a period of time for the ACK to be received, the session is closed.

2. Simultaneous Connection Termination: -

Key Concept: Just as two devices can simultaneously open a TCP session, they can terminate it simultaneously as well. In this case a different state sequence is followed, with each device responding to the other's FIN with an ACK, waiting for receipt of its own ACK, and pausing for a period of time to ensure that its ACK is received by the other device before ending the connection.

Key Concept: There are two approaches to handling retransmission in TCP. In the more "conservative" approach, only the segments whose timers expire are retransmitted; this saves bandwidth but may cause performance degradation if many segments in a row are lost. The alternative is that when a segment's retransmission timer expires, both it and all subsequent unacknowledged segments are retransmitted. This provides better performance if many segments are lost but may waste bandwidth on unnecessary retransmissions.

Key Concept: The optional TCP selective acknowledgment feature provides a more elegant way of handling subsequent segments when a retransmission timer expires. When a device receives a non-contiguous segment it includes a special Selective Acknowledgment (SACK) option in its regular acknowledgment that identifies noncontiguous segments that have already been received, even if they are not yet acknowledged. This saves the original sender from having to retransmit them.

Characteristic / Description	UDP	TCP
General Description	Simple, high-speed, low-functionality "wrapper" that interfaces applications to the network layer and does little else.	Full-featured protocol that allows applications to send data reliably without worrying about network layer issues.
Protocol Connection Setup	Connectionless; data is sent without setup.	Connection-oriented; connection must be established prior to transmission.
Data Interface To Application	Message-based; data is sent in discrete packages by the application.	Stream-based; data is sent by the application with no particular structure.
Reliability and Acknowledgments	Unreliable, best-effort delivery without acknowledgments.	Reliable delivery of messages; all data is acknowledged.
Retransmissions	Not performed. Application must detect lost data and retransmit if needed.	Delivery of all data is managed, and lost data is retransmitted automatically.
Features Provided to Manage Flow of Data	None	Flow control using sliding windows; window size adjustment heuristics; congestion avoidance algorithms.
Overhead	Very low	Low, but higher than UDP
Transmission Speed	Very high	High, but not as high as UDP
Data Quantity Suitability	Small to moderate amounts of data (up to a few hundred bytes)	Small to very large amounts of data (up to gigabytes)
Types of Applications That Use The Protocol	Applications where data delivery speed matters more than completeness, where small amounts of data are sent; or where multicast/broadcast are used.	Most protocols and applications sending data that must be received reliably, including most file and message transfer protocols.
Well-Known Applications and Protocols	Multimedia applications, DNS, BOOTP, DHCP, TFTP, SNMP, RIP, NFS (early versions)	FTP, Telnet, SMTP, DNS, HTTP, POP, NNTP, IMAP, BGP, IRC, NFS (later versions)

Name Systems and TCP/IP Name Registration and Name Resolution

Key Concept: Networking name systems are important because they allow devices to be assigned efficient numeric addresses, while still enabling humans to access them using names that are easier to remember. Name systems become more important as you increase the size of the network, the address or the user base. They are also more essential when the user base is limited in skill or experience.

Key Concept: A name system consists of three theoretical high-level functions: the name space, which describes how names are created and organized; the name registration technique, which is used to set up relationships between names and addresses; and the name resolution method, which is responsible for translating names to addresses.

Key Concept: Name registration is the process by which names are linked to addresses in a name system. It encompasses activities such as central registry authority designation and delegation, and name space structure management. The most common methods of name registration, in order of both increasing capability and complexity, are manual table maintenance, broadcast registration and database registration.

Key Concept: Name resolution is arguably the most important of the main functional elements of a name system, because it is the part of the system that actually converts names into addresses. The two main components of name resolution are name resolvers, which act as clients in the resolution process, and name servers. The three main name resolution methods—table-based, broadcast and client/server—correspond closely to the table, broadcast and database methods of name registration.

Key Concept: The top of the DNS name space is the root; under the root come toplevel domains, and within these are second-level domains and then subdomains. In theory, any number of levels of subdomains can be created. A branch is any contiguous portion of the DNS tree; a leaf is a domain with nothing underneath it in the structure, and usually represents a single device.

DNS Labels and Label Syntax Rules

Length: Each label can theoretically be from 0 to 63 characters in length. In practice, a length of 1 to about 20 characters is most common, with a special exception for the label assigned to the root of the tree

Symbols: Letters, numbers are allowed, as well as the dash symbol ("-"). No other punctuation is permitted, including the underscore ("_").

Case: Labels are not case-sensitive. This means that "Jabberwocky" and "jabberwocky" are both permissible domain name labels, but they are equivalent.

Key Concept: A fully-qualified domain name (FQDN) is a complete domain name that uniquely identifies a node in the DNS name space by giving the full path of labels from the root of the tree down to that node. It defines the absolute location of a domain. In contrast, a partially-qualified domain name (PQDN) only specifies a portion of a domain name. It is a relative name that has meaning only within a particular context; the partial name must be interpreted within that context to fully identify the node.

Generic TLD	Abbreviation For	Authority	Current Use / Description
.AERO	Aerospace	Société Internationale de Télécommunications Aéronautiques (SITA)	Members of the aerospace industry, such as airlines and airports. (Yes, that ***is*** French!)
.ARPA	Address and Routing Parameter Area	Internet Assigned Numbers Authority (IANA) / Internet Corporation for Assigned Names and Numbers (ICANN)	First defined as a temporary domain for migration from the older host table system, the "ARPA" of course originally stood for the Advanced Research Projects Agency, creators of the predecessors of the Internet. Today, the .ARPA domain is used for internal Internet management purposes; the abbreviation at left was, I believe, "manufactured" to fit the letters "ARPA". ☺ The best-known use of this domain is for reverse DNS lookups.
.BIZ	Business	NeuLevel, Inc.	Businesses. Intended as a competitor to .COM.
.COM	Commercial Organizations	VeriSign Global Registry Services	Originally intended for corporations and other commercial interests, .COM is also widely used for other purposes, including small businesses and even individuals who like the popularity of the .COM domain.
.COOP	Cooperative Associations	Dot Cooperation LLC	Cooperative associations.
.EDU	Education	Educause	Originally intended for all types of educational organizations, now used only for degree-granting higher education institutions accredited in the United States. Other educational institutions such as public schools usually use the country code TLDs.
.GOV	Government	US General Services Administration	Reserved for the United States federal government.
.INFO	Information	Afilias Limited	A very generic TLD designed for information resources of various sorts. It is unrestricted, in that anyone can register any sort of organization in .INFO. Also positioned as an alternative to .COM.
.INT	International	IANA .int Domain Registry	Used only for large organizations established by international treaty.

Generic TLD	Abbreviation For	Authority	Current Use / Description
.MIL	Military	US DoD Network Information Center	Reserved for the United States military.
.MUSEUM	Museum	Museum Domain Management Association	Take a guess. ☺
.NAME	Names	Global Name Registry	In the original generic hierarchy there was no place set aside for individuals to register names for themselves, so people would instead create domains like "jonesfamily.org". This was non-ideal so .NAME was created as a place for individuals and families to register a domain for their names. .NAME also competes with the country code TLDs.
.NET	Network	VeriSign Global Registry Services	This TLD was supposed to be used only for Internet service providers and other organizations working intimately with the Internet or networking. Due to the exhaustion of name space in .COM and .ORG, many .NET domains are registered to other organizations, however.
.ORG	Organizations	Public Interest Registry	Originally intended for organizations not fitting into the other generic TLDs, .ORG quickly became associated with professional and non-profit organizations. It is possible, however, to have a for-profit company use a .ORG name.
.PRO	Professional	RegistryPro	Reserved for credentialed professionals such as lawyers and doctors.

Types of DNS Name Server

Primary/Master: -

Secondary/Slave: - there are several reasons why slave servers are also important:

Redundancy: Slave name servers act as a backup for the masters they support.

Maintenance: With more than one server, we can easily take the primary server down for maintenance when needed without name resolution service being disrupted.

Load Handling: Busy zones can use multiple servers to spread the load of name resolution requests to improve performance.

Key Concept: The master DNS server for a zone is its primary server, which maintains the master copy of DNS information. Most DNS zones also have at least one slave or secondary DNS server. These are important because they serve as backups for the primary server, and they can also help share the load of responding to requests in busy zones. Secondary name servers get their information from primary servers on a routine basis. Both master and slave servers are considered authoritative for the zones whose data they maintain.

Key Concept: Each DNS domain has associated with it a set of three contact names that indicate who is responsible for managing it. The administrative contact is the person with overall responsibility for the domain. The billing contact is responsible for payment issues; this may be the same as the administrative contact. The technical contact is in charge of technical matters for the domain, and is often a different person than the administrative contact, especially when DNS services are out-sourced.

Key Concept: The two methods of name resolution in DNS are iterative resolution and recursive resolution. In iterative resolution, if a client sends a request to a name server that does not have the information the client needs, the server returns a pointer to a different name server and the client sends a new request to that server. In recursive resolution, if a client sends a request to a server that doesn't have the requested information, that server takes on the responsibility for sending requests to other servers to find the necessary records, then returns them to the client. A server doing this takes on the role of client for its requests to other servers.

Key Concept: NFS resides architecturally at the application layer of the TCP/IP model. Its functions are implemented primarily through three distinct functional components that implement the functions of layers five through seven of the OSI reference model: the Remote Procedure Call (RPC), which provide session-layer services; the External Data Representation (XDR) standard, which manages data representation and conversion, and NFS procedures and operations, which allow application-layer tasks to be performed using the other two components.

Host Configuration and TCP/IP Host Configuration Protocols (BOOTP and DHCP)

Key Concept: Host configuration protocols enable administrators to set up hosts so that they can automatically determine their address and other key parameters. They are useful not only because of the effort they save over manual configuration, but because they enable the automatic setup of remote, storageless or mobile devices.

When a device like this is turned on, it is in a difficult position: it needs to use IP to communicate with another device that will tell it how to communicate using IP! This process, called bootstrapping or booting, comes from an analogy to a person "pulling himself up using his own bootstraps".

Key Concept: The first widely-used host configuration protocol for TCP/IP was the Boot Protocol (BOOTP). It was created specifically to enable host configuration while addressing many of the weaknesses of RARP. BOOTP is intended to be used as the first phase of a two-phase boot procedure for storageless devices—after obtaining an IP address and other configuration parameters using BOOTP, the device employs a protocol such as TFTP to download software necessary to function on the network.

Overview of DHCP Features

Rather than using a static table that absolutely maps hardware addresses to IP addresses, a pool of IP addresses is used to dynamically allocate addresses.. Today, DHCP is found on millions of networks worldwide. It is used for everything from assigning IP addresses to multi-thousand-host corporate networks, to allowing a home Internet access router to automatically providing the correct Internet configuration information to a single user's computer.

DHCP Address Assignment and Allocation Mechanisms

Manual Allocation: A particular IP address is pre-allocated to a single device by an administrator. DHCP only communicates the IP address to the device. .Each device has an address that an administrator gives it ahead of time, and all DHCP does is look up the address in a table and send it to the client for which it is intended.. It keeps all the IP address information centralized in the DHCP address database, instead of requiring an administrator to go from machine to machine checking addresses and ensuring there are no duplicates. Updates can also be made in a single place as well.

Automatic Allocation: DHCP automatically assigns an IP address permanently to a device, selecting it from a pool of available addresses. The third option, automatic allocation, can be used in cases where there are enough IP addresses for each device that may connect to the network, but where devices don't really care what IP address they use.

Dynamic Allocation: DHCP assigns an IP address from a pool of addresses for a limited period of time chosen by the server, or until the client tells the DHCP server that it no longer needs the address. Each client that is configured to use DHCP contacts the server when it needs an IP address. The server keeps track of which IP addresses are already assigned, and leases one of the free addresses from the pool to the client. The server decides the amount of time that the lease will last. When the time expires, the client must either request permission to keep using the address (renewing the lease) or must get a new one.

Common Lease Durations

The administrator need not pick from “short” and “long” lease durations. He or she can “compromise” by choosing a number that best suits the network.Some example are as One Hour Or Less, One Day, Three Days, One Week, One Month, Three Months, One Year

The DHCP Lease "Life Cycle

Allocation: A client begins with no active lease, and hence, no DHCP-assigned address. It acquires a lease through a process of allocation.

Reallocation: If a client already has an address from an existing lease, then when it reboots or starts up after being shut down, it will contact the DHCP server that granted it the lease to confirm the lease and acquire operating parameters

Normal Operation: Once a lease is active, the client functions normally, using its assigned IP address and other parameters during the "main part" of the lease.

Renewal: After a certain portion of the lease time has expired, the client will attempt to contact the server that initially granted the lease, to renew the lease so it can keep using its IP address.

Rebinding: If renewal with the original leasing server fails (because, for example, the server has been taken offline), then the client will try to rebind to any active DHCP server, trying to extend its current lease with any server that will allow it to do so.

Release: The client may decide at any time that it no longer wishes to use the IP address it was assigned, and may terminate the lease, releasing the IP address.

Key Concept: If a site has multiple DHCP servers, they can be set up with either overlapping or non-overlapping scopes. Overlapping scopes allow each server to assign from the same pool, providing flexibility, but raising the possibility of two clients being assigned the same address unless a feature such as server conflict detection is employed. Non-overlapping scopes are "safer" because each server has a dedicated set of addresses for its use, but this means one server could run out of addresses while the other still has plenty, and if a server goes down its addresses will be temporarily unallocatable.

Key Concept: DHCP servers are devices programmed to provide DHCP services to clients. They manage address information and other parameters and respond to client configuration requests. DHCP clients are TCP/IP devices that have been set to use DHCP to determine their configuration. They send requests and read responses, and are responsible for managing their own leases, including renewing or rebinding a lease when necessary.

Lease Allocation Process Steps

1. **Client Creates DHCPDISCOVER Message**

The client begins in the INIT (initialization) state. It has no IP address and doesn't even know whether or where a DHCP server may be on the network. To find one, it creates a DHCPDISCOVER message.

2. **Client Sends DHCPDISCOVER**

Message The client broadcasts the DHCPDISCOVER message on the local network. The client transitions to the SELECTING state, where it waits for replies to its message.

3. **Servers Receive and Process DHCPDISCOVER**

Message Each DHCP server on the local network receives the client's DHCPDISCOVER message and examines it. The server looks up the client's hardware address in its database and determines if it is able to offer the client a lease, and what the terms of the lease will be.

4. **Servers Create DHCPOFFER Messages**

Each server that chooses to respond to the client creates a DHCPOFFER message including the following information:

- The IP address to be assigned to the client, in the YIAddr field. If the server previously had a lease for this client it will attempt to reuse the IP address it used last time.
- The length of the lease being offered.
- The same transaction ID (XID) used in the DHCPDISCOVER message.

5. **Servers Probe And/Or Reserve Offered Address (Optional)**

The DHCP standard specifies that before sending a DHCPOFFER to a client, the server “SHOULD” check to see that the IP address isn't already in use by sending an ICMP Echo message to that address. If the probe is made and the address is in use, the server will of course not offer it to the client.

6. **Servers Send DHCPOFFER Messages**

Each server sends its DHCPOFFER message. They of course may not all be sent at exactly the same time. The messages are sent either unicast or broadcast, as mentioned earlier.

7. **Client Collects and Processes DHCPOFFER Messages**

8. **Client Creates DHCPREQUEST Message**

The client creates a DHCPREQUEST message for the server offer it has selected. This message serves two purposes: it tells the server whose offer the client has accepted “yes, I accept your offer, assuming it is still available” and also tells the other servers “sorry, your offer was rejected”.

9. **Client Sends DHCPREQUEST Message**

The client sends the DHCPREQUEST message. Since it is intended for not just the selected DHCP server but all servers, it is broadcast.

10. **Servers Receive and Process DHCPREQUEST Message**

Each of the servers receives and processes the client's request message.The servers not chosen will take the message as a rejection.

11. **Server Sends DHCPACK or DHCPNAK Message**

12. **Client Receives and Processes DHCPACK or DHCPNAK Message**

The client receives either a positive or negative acknowledgment for its request. If the message is a DHCPNAK, the client transitions back to the INIT state and starts over: back to square one.

13. **Client Checks That Address Is Not In Use**

The client device should perform a final check to ensure that the new address isn't already in use before it concludes the leasing process.

14. **Client Finalizes Lease Allocation**

Assuming that the address is not already in use, the client finalizes the lease and transitions to the BOUND state.

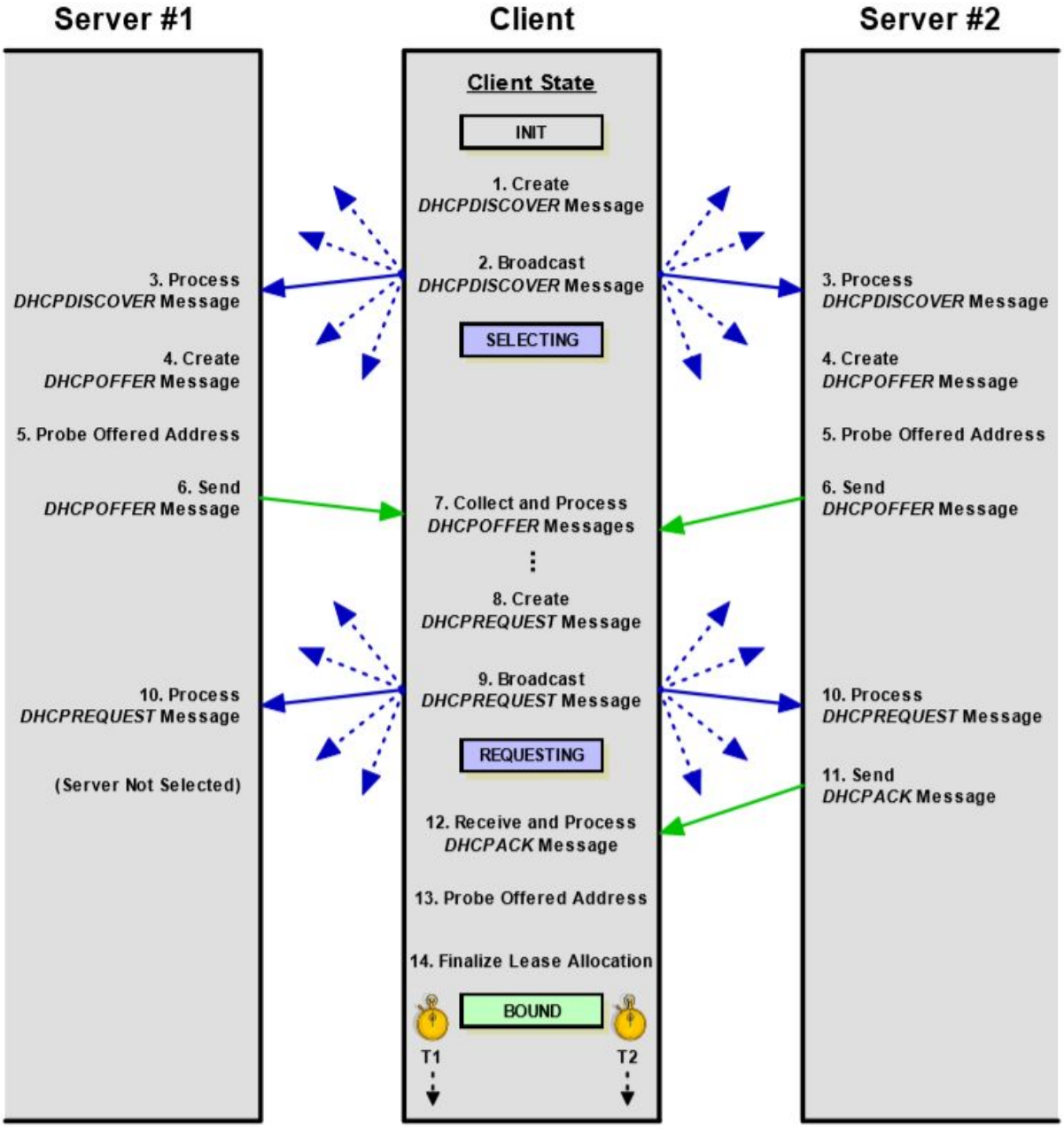


Figure 263: DHCP Lease Allocation Process

This diagram shows the steps involved in DHCP client lease allocation. This diagram is a bit different from

Key Concept: Some DHCP implementations include a feature called server conflict detection. When this feature is activated, it causes each server to always check to make sure an address is not in use before granting it to a client. When conflict detection is used by all DHCP servers on a network, the servers can be given overlapping scopes, so each can assign any of the organization's IP addresses, while at the same time not needing to be concerned about two clients being assigned the same address by different servers.

Adding Security to DHCP

This standard describes an enhancement that replaces the normal DHCP messages with authenticated ones. Clients and servers check the authentication information and reject messages that come from invalid sources. The technology involves the use of a new DHCP option type, the Authentication option, and operating changes to several of the leasing processes to use this option.

Both client and server must be programmed to use authentication for this method to have value. A DHCP server that supports authentication could use it for clients that support the feature and skip it for those that do not. However, the fact that this option is not universal means that it is not widely deployed, and most networks must rely on more conventional security measures.

DHCPv6 Message Exchanges

1. The client sends a multicast **Solicit** message to find a DHCPv6 server and ask for a lease.

2. Any server that can fulfill the client's request responds to it with an **Advertise** message.

3. The client chooses one of the servers and sends a **Request** message to it asking to confirm the offered address and other parameters.

4. The server responds with a **Reply** message to finalize the process.

TCP/IP Network Management Framework and Protocols (SNMP and RMON)

Key Concept: The Simple Network Management Protocol (SNMP) defines a set of technologies that allows network administrators to remotely monitor and manage TCP/IP network devices. The term "SNMP" refers both to a specific communication protocol (sometimes called the SNMP Protocol) and an overall framework for Internet management (the SNMP Framework).

SNMP Device Types

Managed Nodes: Regular nodes on a network that have been equipped with software to allow them to be managed using SNMP. These are, generally speaking, conventional TCP/IP devices; they are also sometimes called managed devices.

Network Management Station (NMS): A designated network device that runs special software to allow it to manage the regular managed nodes mentioned just above. One or more NMSes must be present on the network, as these devices are the ones that really `run" SNMP.

Key Concept: SNMP allows a network administrator using a network management station (NMS) to control a set of managed nodes. Each device incorporates an SNMP entity that implements the technology. In an NMS, the entity consists of an SNMP manager module and a set of SNMP applications; in a managed node, an SNMP agent and management information base (MIB).

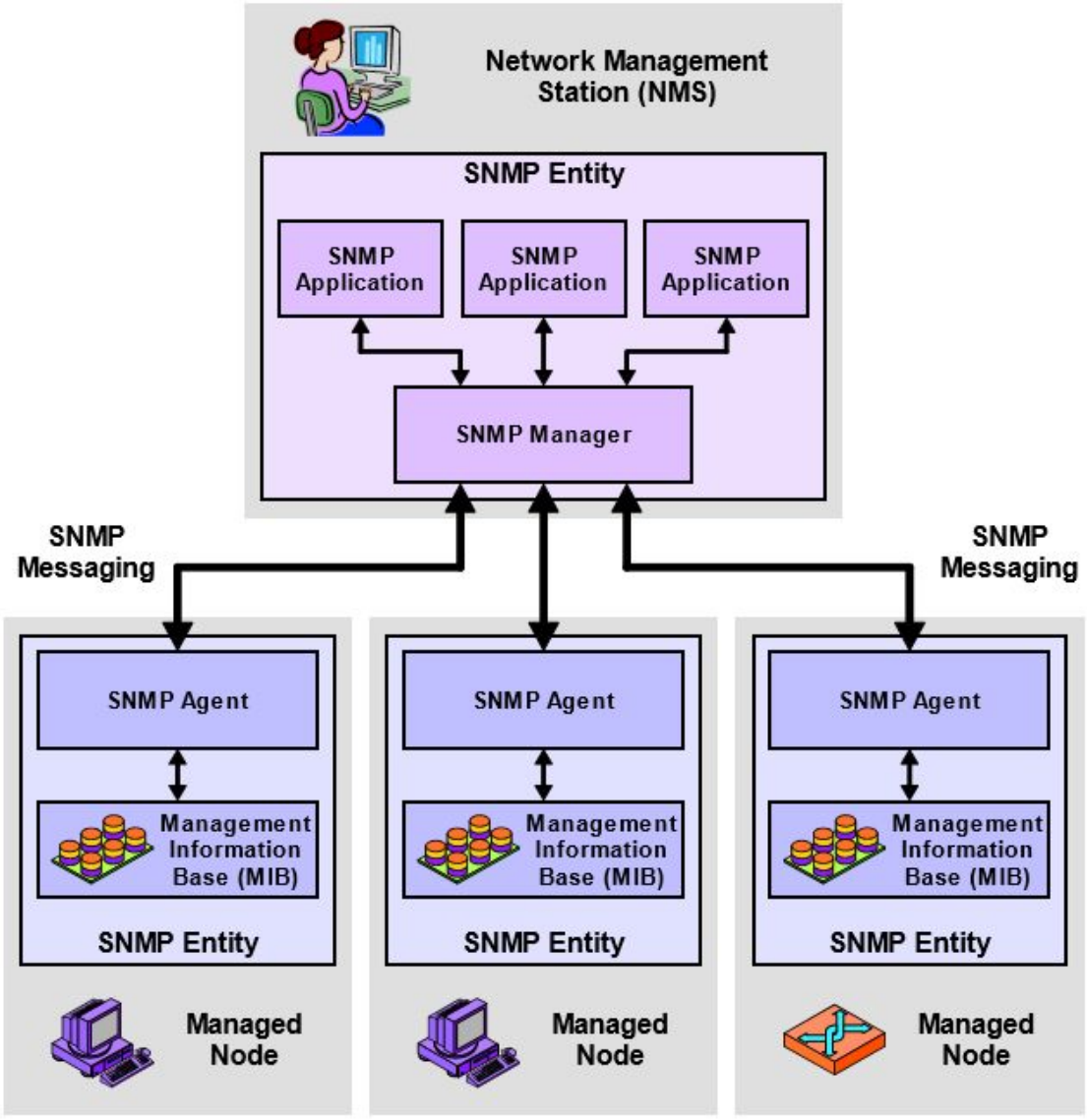


Figure 270: SNMP Operational Model

This diagram shows a simplified implementation of SNMP, with one network management station used to maintain three managed nodes. Each device has an *SNMP Entity*, and they communicate using SNMP messages. The SNMP entity of the NMS consists of the *SNMP Manager* and one or more *SNMP Applications*; the managed nodes each run an *SNMP Agent* and maintain a *Management Information Base (MIB)*.

TCP/IP Internet Standard Management Framework Architecture and Protocol Components

TCP/IP network management is based on the Simple Network Management Protocol, abbreviated SNMP.While it is commonly used to refer to the actual communication protocol used to exchange network management information, the term also refers to the entire set of technologies that enable TCP/IP network management.

SNMP Framework Components

Structure of Management Information (SMI): To ensure interoperability of various devices, we want to have a consistent way of describing the characteristics of devices to be managed using SNMP. The Structure of Management Information (SMI) is a standard that defines the structure, syntax and characteristics of management information in SNMP.

Management Information Bases (MIBs): Each managed device contains a set of variables that is used to manage it. These variables represent information about the operation of the device that is sent to a network management station, and/or parameters sent to the managed device to control it. MIB is the full set of these variables that describe the management characteristics of a particular type of device.

Simple Network Management Protocol (SNMP): It defines how information is exchanged between SNMP agents and network management stations. The SNMP protocol operations define the various SNMP messages and how they are created and used.

Security and Administration: To the three main architectural components above, the SNMP Framework adds a number of supporting elements. These provide enhancements to the operation of the SNMP protocol for security, and address issues related to SNMP implementation, version transition and other administrative issues.

Table 200: SNMP Version 1 (SNMPv1) Standards

Obsolete RFCs	Most Recent RFC	Date of Most Recent RFC	Standard Name
1065	**1155**	May 1990	*Structure and identification of management information for TCP/IP-based internets*
1066	**1156**	May 1990	*Management Information Base for network management of TCP/IP-based internets*
1067, 1098	**1157**	May 1990	*Simple Network Management Protocol (SNMP)*
1158	**1213**	March 1991	*Management Information Base for Network Management of TCP/IP-based internets: MIB-II*

Table 201: SNMP Security (SNMPSec) Standards

Obsolete RFCs	Most Recent RFC	Date of Most Recent RFC	Standard Name
—	**1351**	July 1992	*SNMP Administrative Model*
—	**1352**	July 1992	*SNMP Security Protocols*
—	**1353**	July 1992	*Definitions of Managed Objects for Administration of SNMP Parties*

Table 202: Party-Based SNMP Version 2 (SNMPv2p) Standards *(Page 1 of 2)*

Obsolete RFCs	Most Recent RFC	Date of Most Recent RFC	Standard Name
—	**1441**	April 1993	*Introduction to version 2 of the Internet-standard Network Management Framework*
—	**1442**	April 1993	*Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1443**	April 1993	*Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1444**	April 1993	*Conformance Statements for version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1445**	April 1993	*Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1446**	April 1993	*Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2)*

Table 202: Party-Based SNMP Version 2 (SNMPv2p) Standards *(Page 2 of 2)*

Obsolete RFCs	Most Recent RFC	Date of Most Recent RFC	Standard Name
—	**1447**	April 1993	*Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1448**	April 1993	*Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1449**	April 1993	*Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1450**	April 1993	*Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1451**	April 1993	*Manager-to-Manager Management Information Base*
—	**1452**	April 1993	*Coexistence between version 1 and version 2 of the Internet-standard Network Management Framework*

Table 203: Community-Based SNMP Version 2 (SNMPv2c) Standards

Obsolete RFCs	Most Recent RFC	Date of Most Recent RFC	Standard Name
—	**1901**	January 1996	*Introduction to Community-based SNMPv2*
—	**1902**	January 1996	*Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1903**	January 1996	*Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1904**	January 1996	*Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1905**	January 1996	*Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1906**	January 1996	*Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1907**	January 1996	*Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*
—	**1908**	January 1996	*Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework*

Table 204: User-Based SNMP Version 2 (SNMPv2u) Standards

Obsolete RFCs	Most Recent RFC	Date of Most Recent RFC	Standard Name
—	**1909**	February 1996	*An Administrative Infrastructure for SNMPv2*
—	**1910**	February 1996	*User-based Security Model for SNMPv2*

Table 205: SNMP Version 3 (SNMPv3) Standards

Obsolete RFCs	Most Recent RFC	Date of Most Recent RFC	Standard Name
—	**2576**	March 2000	*Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*
—	**2578**	April 1999	*Structure of Management Information Version 2 (SMIv2)*
—	**2579**	April 1999	*Textual Conventions for SMIv2*
—	**2580**	April 1999	*Conformance Statements for SMIv2*
2570	**3410**	December 2002	*Introduction and Applicability Statements for Internet-Standard Management Framework*
2261, 2271, 2571	**3411**	December 2002	*An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*
2262, 2272, 2572	**3412**	December 2002	*Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*
2263, 2273, 2573	**3413**	December 2002	*Simple Network Management Protocol (SNMP) Applications*
2264, 2274, 2574	**3414**	December 2002	*User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*
2265, 2275, 2575	**3415**	December 2002	*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*
—	**3416**	December 2002	*Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*
—	**3417**	December 2002	*Transport Mappings for the Simple Network Management Protocol (SNMP)*
—	**3418**	December 2002	*Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

In addition to all of these tables, as I mentioned before, there are dozens of supplemental RFCs that describe MIB modules and also clarify various fine points of operation related to SNMP. Listing all of these would… make me go insane. Sorry, those tables were bad enough. ☺ You can find all the MIBs in an online list of RFCs by searching for "MIB" or "SNMP".

SNMP Protocol Basic Request/Response Information Poll Using GetRequest and (Get)Response Messages

1. **SNMP Manager Creates GetRequest-PDU:** Based on the information required by the application and user, the SNMP software on the network management station creates a GetRequest-PDU message. It contains the names of the MIB objects whose values the application wants to retrieve.

2. **SNMP Manager Sends GetRequest-PDU:** The SNMP manager sends the PDU to the device that is being polled.

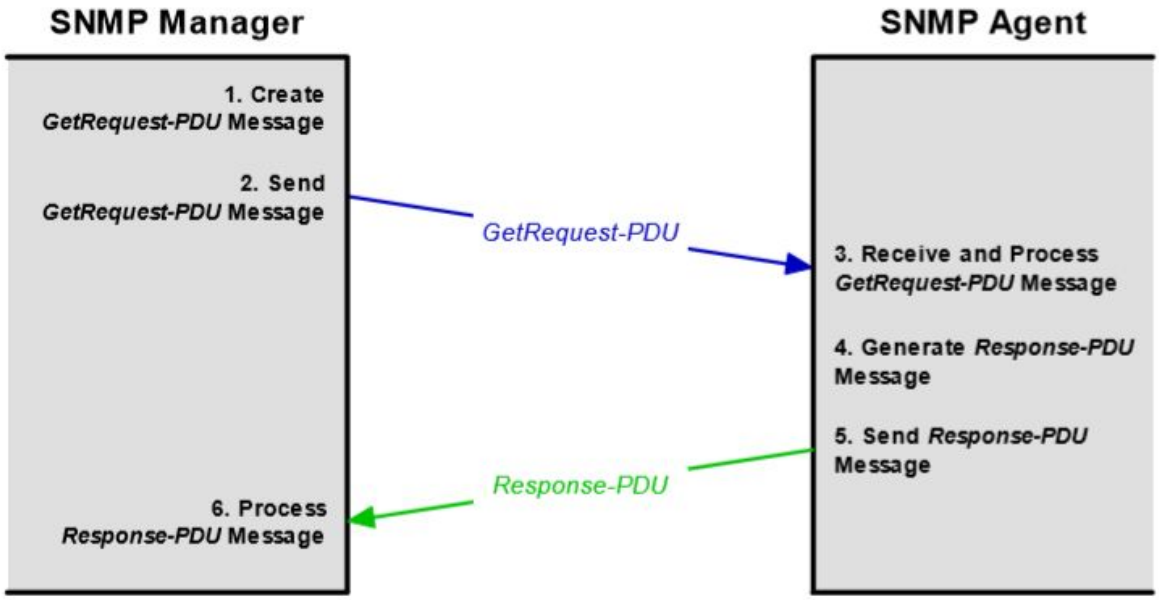
3. **SNMP Agent Receives and Processes GetRequest-PDU:** The SNMP agent receives and processes the request. It looks at the list of MIB object names contained in the message and checks to see if they are valid (ones the agent actually implements). It looks up the value of each variable that was correctly specified.

4. **SNMP Agent Creates Response-PDU:** The agent creates a Response-PDU to send back to the SNMP Manager. This message contains the values of the MIB objects requested and/or error codes to indicate any problems with the request, such as an invalid object name.

5. **SNMP Agent Sends Response-PDU:** The agent sends the response back to the SNMP Manager.

6. **SNMP Manager Processes Response-PDU:** The manager processes the information in the Response-PDU received from the agent.

SNMP Information Poll Process is given below:



TCP/IP Remote Network Monitoring (RMON)

An extension of the Simple Network Management Protocol (SNMP) that allows detailed monitoring of network statistics for Ethernet networks. Remote Network Monitoring (RMON) is defined in Request for Comments (RFC) 1757.

RMON lets you monitor network traffic on a remote Ethernet segment from a central location on the network to detect problem conditions such as traffic congestion, dropped packets, and excessive collisions. You can use RMON to set network traffic thresholds that trigger alarms so that you can correct network problems before they occur. Embedded RMON support for Ethernet switches lets network administrators monitor switched Ethernet networks that cannot easily be monitored using traditional packet-sniffing network analyzers.

TCP/IP Application Layer Addressing: Uniform Resource Identifiers, Locators and Names (URIs, URLs and URNs)

TCP/IP has defined a system of Uniform Resource Identifiers (URIs) that can be used both on the Internet and on private TCP/IP networks. Each URI uniquely specifies how a client can locate a particular resource and access it so it can be used. URIs are subdivided into Uniform Resource Locators (URLs) and Uniform Resource Names (URNs), which serve a similar purpose but work in different ways.

URI Categories

URIs are in fact a general purpose method for referring to many kinds of TCP/IP resources. They are currently divided into two primary categories based on how they describe a resource:

- **Uniform Resource Locators (URLs):** A URL is a uniform resource identifier that refers to a resource through the combination of a protocol or access mechanism and a specific resource location. A URL begins with the name of the protocol to be used for accessing the resource and then contains sufficient information to point to how it can be obtained.
- **Uniform Resource Names (URNs):** A URN is a uniform resource identifier that provides a way of uniquely naming a resource without specifying an access protocol or mechanism, and without specifying a particular location.

Key Concept: Some sort of mechanism is needed on any internetwork to allow resources such as files, directories and programs to be identified and accessed. In TCP/IP, Uniform Resource Identifiers (URIs) are used for this sort of "application layer addressing". The two types of URIs are Uniform Resource Locators (URLs), which specify how to access an object using a combination of an access method and location, and Uniform Resource Names (URNs), which identify an object by name but do not indicate how to access it.

Uniform Resource Locators (URLs)

Uniform Resource Locators (URLs) are text strings that allow a resource such as a file or other object to be labelled based on its location on an internetwork and the primary method or protocol by which it may be accessed.

URL General Syntax

Uniform Resource Locators (URLs) are a subset of Uniform Resource Identifiers (URIs) that consist of two components that identify how to access a resource on a TCP/IP internetwork. These two components are the **location** of the resource, and the **method** to be used to access it.

The most general form of this common Internet scheme syntax is as follows:

<scheme>://<user>:<password>@<host>:<port>/<url-path>;<params>?<query>#<fragment>

- **<scheme>:** The URL scheme, as described above.
- **<user> and <password>:** Authentication information for schemes requiring a login, in the form of a username and password.
- **<host>:** An Internet host, usually specified either as a fully qualified DNS domain name, or an IP address in dotted-decimal notation.
- **<port>:** A TCP or UDP port number to use when invoking the protocol appropriate to the scheme.
- **<url-path>:** A resource location path. This is usually a full directory path expressing the sequence of directories to be traversed from the root directory to the place where the resource is located, and then the resource's name.
- **<params>:** Scheme-specific parameters included to control how the scheme is used to access the resource. Each parameter is generally of the form "<parameter>=<value>", with each parameter specification separated from the next using a semi-colon.

- **<query>:** An optional query or other information to be passed to the server when the resource is accessed.
- **<fragment>:** Identifies a particular place within a resource that the user of the URL is interested in.

SOME COMMON URLs

http://<user>:<password>@<host>:<port>/<url-path>?<query>#<bookmark>

ftp://<user>:<password>@<host>:<port>/<url-path>;type=<typecode>

mailto:<e-mail-address>

gopher://<host>:<port>/<gopher-path>

news://<newsgroup-name>

news://<message-id>

nntp://<host>:<port>/<newsgroup-name>/<article-number>

telnet://<user>:<password>@<host>:<port>

file://<host>:<url-path>

file:///<url-path>

Key Concept: Regular URLs are absolute, meaning that they include all of the information needed to fully specify how to access a resource. In situations where many resources need to be accessed that are approximately in the same place or are related in some way, completely specifying a URL can be inefficient. Instead, relative URLs can be used, which specify how to access a resource relative to the location of another one. A relative URL can only be interpreted within the context of a base URL that provides any information missing from the relative reference.

Uniform Resource Names (URNs)

A URN is intended to label a resource based on its actual identity, rather than where it can be found. To be useful in identifying a particular resource, it is necessary that a URN be globally unique, and that's not always as simple as it may at first appear.

To allow URNs to represent many kinds of resources, numerous URN namespaces are defined. A namespace is referenced using a unique string that tells the person or computer interpreting the URN what type of resource the URN identifies.

The general syntax of a URN is:

URN:<namespace-ID>:<resource-identifier>

TCP/IP File and Message Transfer Applications and Protocols (FTP, TFTP, Electronic Mail, USENET, HTTP/WWW, Gopher)

Information is most often arranged into discrete units called files. When those files are created specifically for the purpose of communication, they are often called messages. One of the most important groups of TCP/ IP applications is the one that describes the basic mechanisms for moving these files between internetworked devices: file and message transfer applications.

The TCP/IP file and message transfer protocols have in common the notion of moving files from one computer to another. Where they differ is in how the files are handled and processed. There are two basic approaches: general file transfer, and message transfer.

General transfer applications normally treat the file as a “black box”, moving them from place to place and paying little or no attention to what the files contain. The TCP/IP **File Transfer Protocol (FTP)** and **Trivial File Transfer Protocol (TFTP)** fall into this category.

Other TCP/IP applications work with particular types of files, processing and interpreting them in various ways. These files are usually designed for the specific purpose of communication, and are thus called messages; these applications allow users to construct, send and receive messages that fit a particular message format.

Like: - E-mail, usenet,World Wide Web

TCP/IP General File Transfer Protocols (FTP and TFTP)

File Transfer Protocol (FTP)

FTP is one of the most widely used application protocols in the world. It was designed to allow the efficient transfer of files between any two devices on a TCP/IP internetwork. It automatically takes care of the details of how files are moved, provides a rich command syntax to allow various supporting file operations to be performed (such as navigating the directory structure and deleting files) and operates using the TCP transport service for reliability.

The developers of early applications conceptually divided methods of network use into two categories: **direct** and **indirect**.

Direct network applications let a user access a remote host and use it as if it were local, creating the illusion that the network doesn't even exist (or at least, minimizing the importance of distance).

Indirect network use meant getting resources from a remote host and using them on the local system, then transferring them back. These two methods of use became the models for the first two formalized TCP/IP networking applications:

Telnet for direct access and the File Transfer Protocol (FTP) for indirect network use.

Overview of FTP Operation

FTP was created with the overall goal of allowing indirect use of computers on a network, by making it easy for users to move files from one place to another. Like most TCP/IP protocols, it is based on a client/server model, with an FTP client on a user machine creating a connection to an FTP server to send and retrieve files to and from the server.

The main objectives of FTP were to make file transfer simple, and to shield the user from implementation details of how the files are actually moved from one place to another. To this end, FTP is designed to automatically deal with many of the issues that can potentially arise due to format differences in files stored on differing systems.

To ensure that files are sent and received without loss of data that could corrupt them, FTP uses the reliable Transmission Control Protocol (TCP) at the transport layer. An authentication system is used to ensure that only authorized clients are allowed to access a server.

Key Concept: The FTP client is sometimes called the user device, since the human user interacts with the client directly. The FTP client software is called the User-FTP Process; the FTP server software, the Server-FTP Process.

The FTP model is designed around two logical channels of communication between the server and user FTP processes:

- **Control Connection:** This is the main logical TCP connection that is created when an FTP session is established. It is maintained throughout the FTP session and is used only for passing control information, such as FTP commands and replies. It is not used to send files.
- **Data Connection:** Each time that data is sent from the server to the client or vice versa, a distinct TCP data connection is established between them. Data is transferred over this connection. When the file transfer is complete, the connection is terminated.

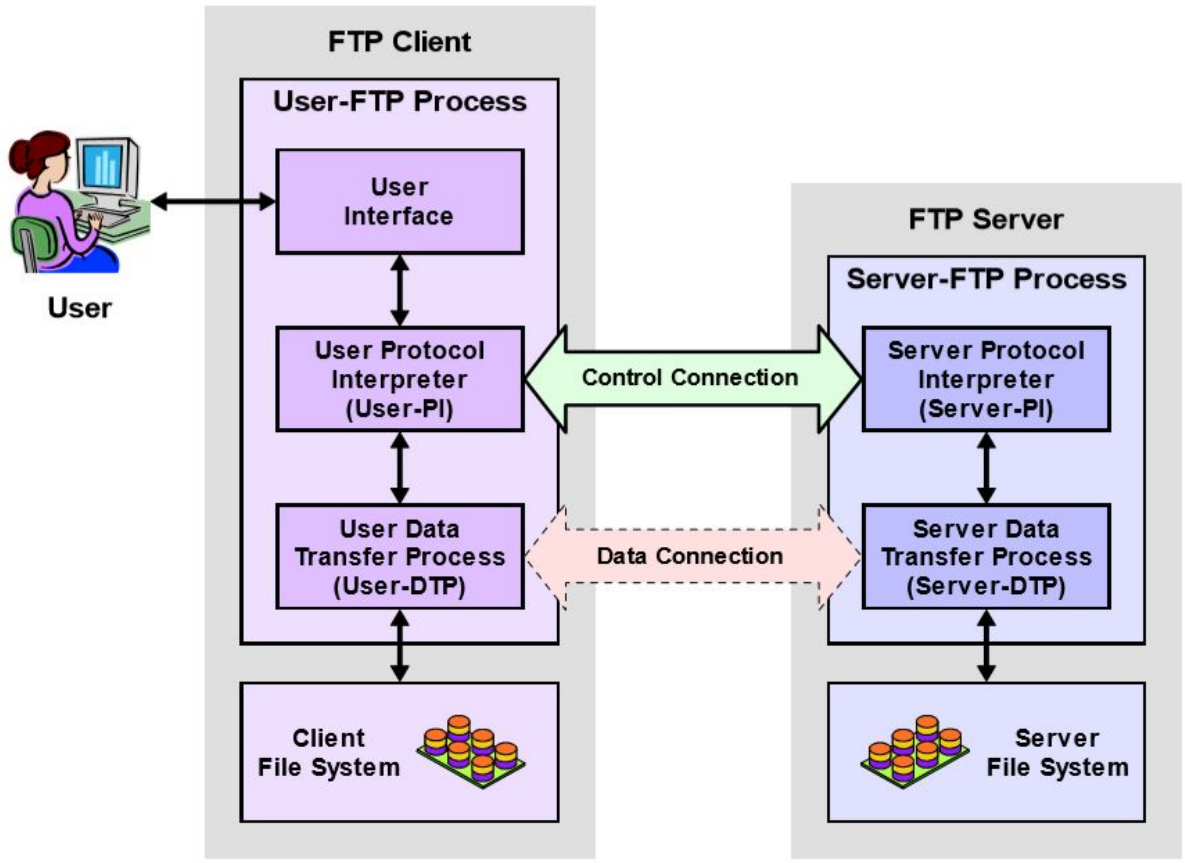


Figure 288: File Transfer Protocol (FTP) Operational Model

Server-FTP Process Components

- **Server Protocol Interpreter (Server-PI):** The protocol interpreter responsible for managing the control connection on the server. It listens on the main reserved FTP port for incoming connection requests from users (clients). Once a connection is established, it receives commands from the User-PI, sends back replies, and manages the server data transfer process.
- **Server Data Transfer Process (Server-DTP):** The DTP on the server side, used to send or receive data to or from the User-DTP. The Server-DTP may either establish a data connection or listen for a data connection coming from the user. It interacts with the server's local file system to read and write files.

User-FTP Process Components

- **User Protocol Interpreter (User -PI):** The protocol interpreter responsible for managing the control connection on the client. It initiates the FTP session by issuing a request to the Server-PI. Once a connection is established, it processes commands received from the user interface, sends them to the Server-PI, and receives back replies. It also manages the user data transfer process.
- **User Data Transfer Process (User-DTP):** The DTP on the user side, which sends or receives data to or from the Server-DTP. The User-DTP may either establish a data connection or listen for a data connection coming from the server. It interacts with the client device's local file system.
- **User Interface:** The user interface provides a more “friendly” FTP interface to a human user. It allows simpler user-oriented commands to be used for FTP functions rather than the somewhat cryptic internal FTP commands, and also allows results and information to be conveyed back to the person operating the FTP session.

Third-Party File Transfer (Proxy FTP)

In this technique, a user on one host performs a file transfer from one server to another. This is done by opening two control connections: one each from the User-PI on the user's machine to the two Server-PI's on the two servers. Then, a Server-DTP is invoked on each server to send data; the User-DTP is not used.

This method, sometimes called third-party file transfer or proxy FTP, is not widely used today. A major reason for this is that it raises security concerns, and has been exploited in the past.

The server protocol interpreter (Server-PI) “listens” on the special well known TCP port reserved for FTP control connections: port 21. The User-PI initiates the connection by opening a TCP connection from the user device to the server on this port. It uses an ephemeral port number as its source port in the TCP connection.

Once TCP has been set up, the control connection between the devices is established, allowing commands to be sent from the User-PI to the Server-PI, and reply codes to be sent back in response. The first order of business after the channel is operating is user authentication, which the FTP standard calls the login sequence. There are two purposes for this process:

- **Access Control:** The authentication process allows access to the server to be restricted to only authorized users. It also lets the server control what types of access each user has.
- **Resource Selection:** By identifying the user making the connection, the FTP server can make decisions about what resources to make available to the user.

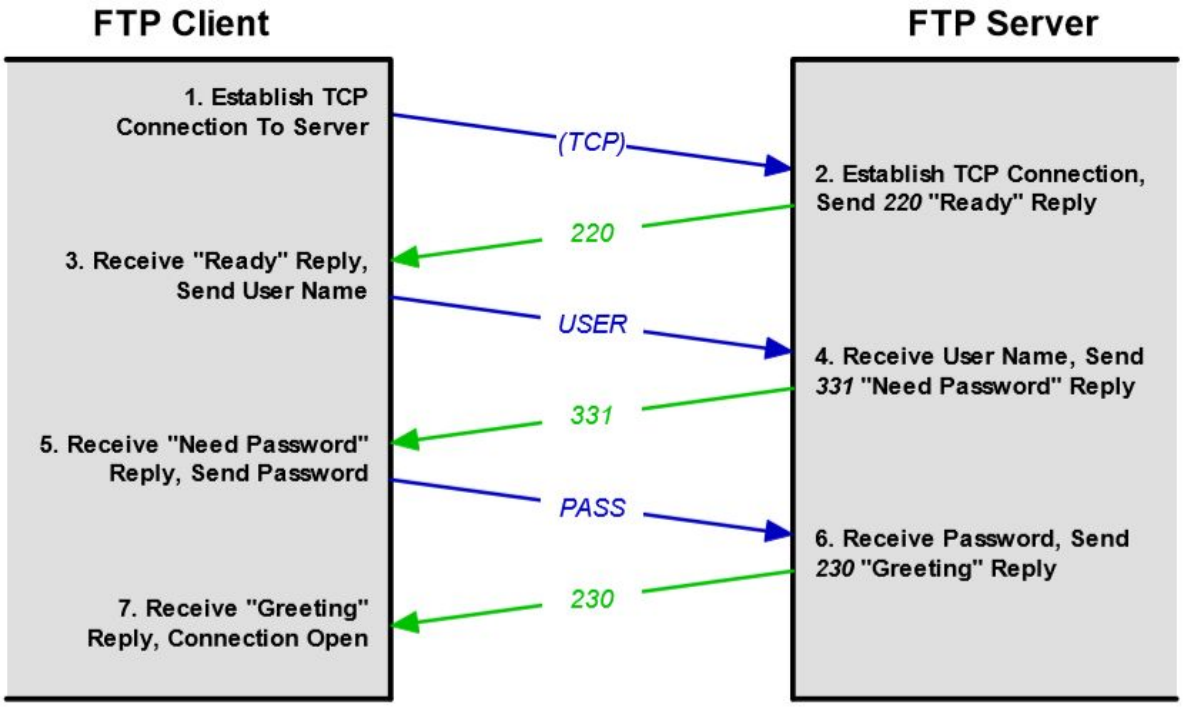


Figure 289: FTP Connection Establishment and User Authentication

Key Concept: An FTP session begins with the establishment of a control connection between an FTP client and server. After the TCP connection be made, the user must authenticate with the server, using a simple user/password exchange between client and server. This provides only rudimentary security, so if more is required, it must be implemented using FTP security extensions or through other means.

Key Concept: Many FTP servers support anonymous FTP, which allows a guest who has no account on the server to have limited access to server resources. This is often used by organizations that wish to make files available to the public for purposes such as technical support, customer support, or distribution.

FTP Data Connection Management, Normal (Active) and Passive Data Connections and Port Usage

Each time files or other data need to be sent between the server and user FTP processes, a data connection must be created. The data connection links the User-DTP with the Server-DTP. This connection is required both for explicit file transfer actions (getting or receiving a file) and also for implicit data transfers, such as requesting a list of files from a directory on the server.

Normal (Active) Data Connections

In this type of connection, the Server-DTP initiates the data channel by opening a TCP connection to the User-DTP. The server uses the special reserved port number 20 (one less than the well-known control FTP port number, 21) for the data connection. On the client machine, the default port number used is the same as the ephemeral port number used for the control connection, but as we'll see shortly, the client will often choose a different port for each transfer.

Ex:-Let's take an example to see how this works. Suppose the User-PI established a control connection from its ephemeral port number 1678 to the server's FTP control port of 21. Then, to create a data connection for data transfer, the Server-PI would instruct the ServerDTP to initiate a TCP connection from the server's port 20 to the client's port 1678. The client would acknowledge this and then data could be transferred (in either direction — remember that TCP is bidirectional).

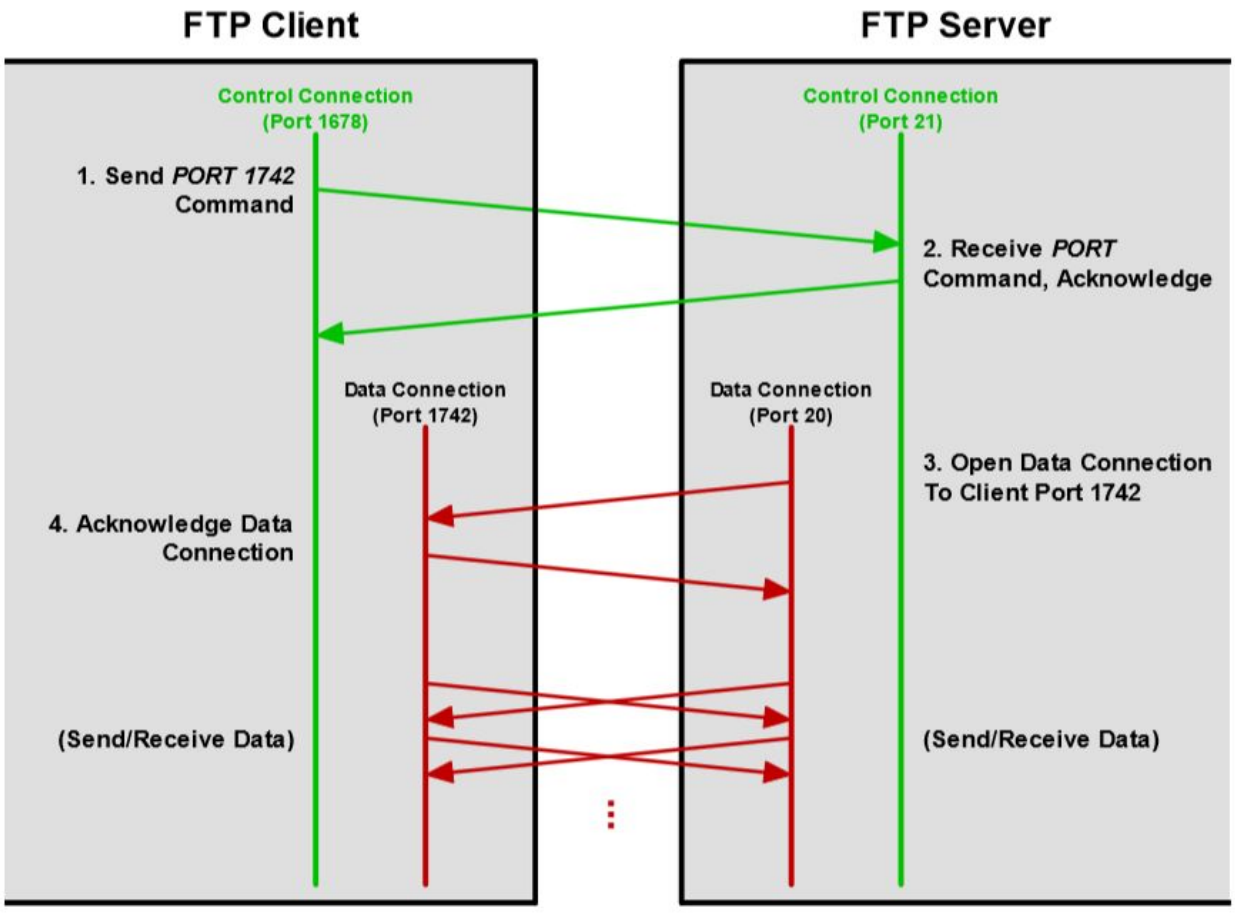


Figure 290: FTP Active Data Connection

Passive Data Connections

The client tells the server to be “passive”, that is, to accept an incoming data connection initiated by the client. The server replies back giving the client the server IP address and port number that it should use. The Server-DTP then listens

on this port for an incoming TCP connection from the User-DTP. By default, the user machine uses the same port number it used for the control connection, as in the active case. However, here again, the client can choose to use a different port number for the data connection if necessary (typically an ephemeral port number.)

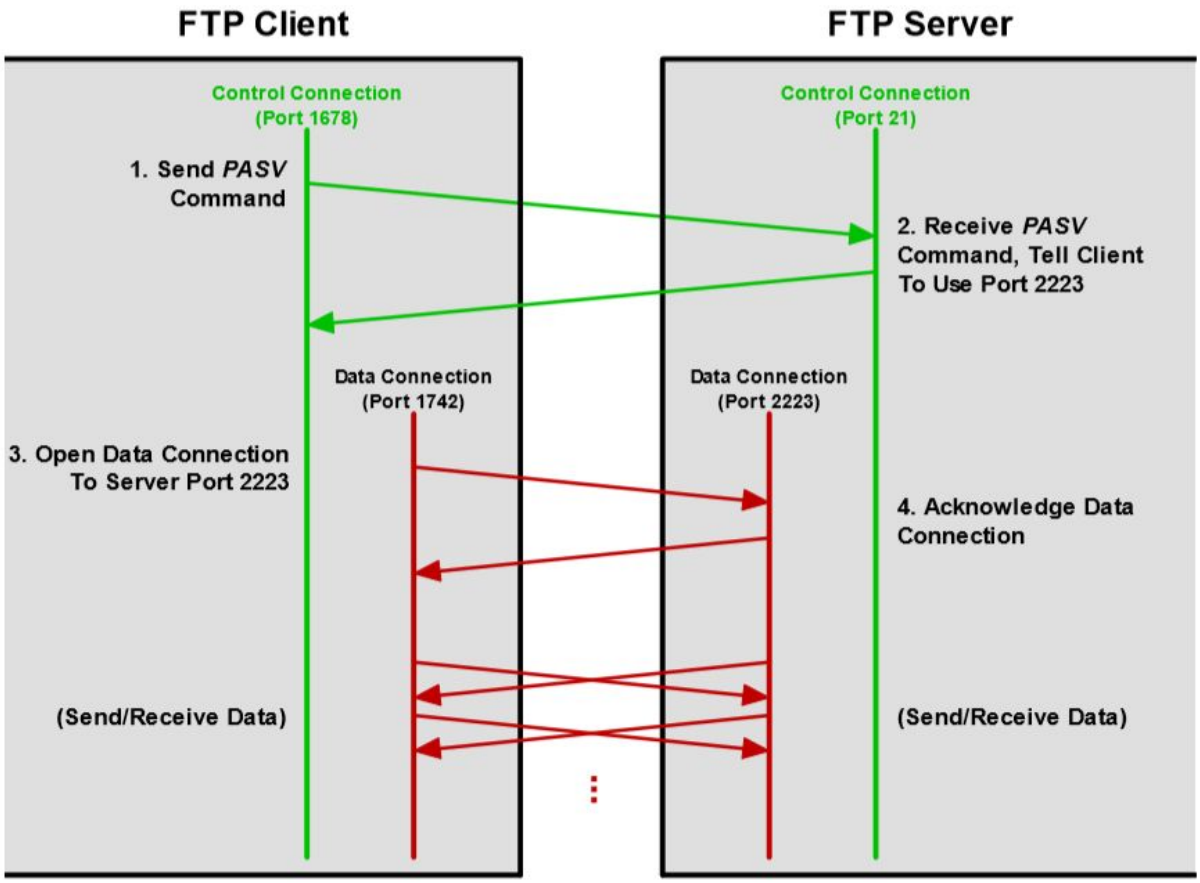


Figure 291: FTP Passive Data Connection

In a passive FTP data connection, the client uses the *PASV* command to tell the server to wait for the client to establish the data connection. The server responds, telling the client what port it should use on the server for the data transmission, in this case port 2223. The client then opens the data connection using that port number on the server and a client port number of its own choosing, in this case 1742. Contrast to Figure 291.

Key Concept: FTP supports two different models for establishing data connections between the client and server. In normal, or active data connections, the server initiates the connection when the client requests a transfer, and the client responds; in a passive data connection, the client tells the server it will initiate the connection, and the server responds. Since TCP is bidirectional, data can flow either way in both cases; the chief difference between the two modes has to do with security. In particular, passive mode is often used because many client devices today are not able to accept incoming connections from servers.

FTP General Data Communication and Transmission Modes

FTP defines three different transmission modes (also called transfer modes) that specify exactly how data is sent from one device to another over an opened data channel: stream mode, block mode, and compressed mode.

Stream Mode: In this mode, data is sent simply as a continuous stream of unstructured bytes. The sending device simply starts pushing data across the TCP data connection to the recipient. No message format with distinct header fields is used, making this method quite different from the way many other protocols send information in discrete chunks. It relies strongly on the data streaming and reliable transport services of TCP. Since there is no header structure, the end of the file is indicated simply by the sending device closing the data connection when it is done.

Block Mode: This is a more “conventional” data transmission mode in which data is broken into data blocks and encapsulated into individual FTP blocks, or records. Each record has a threebyte header that indicates its length and contains information about the data blocks being sent. A special algorithm is used to keep track of the transmitted data and to detect and restart an interrupted transfer.

Compressed Mode: A transmission mode where a relatively simple compression technique called run-length encoding is used to detect repeated patterns in the data being sent, and then represent them in such a way that the overall message takes fewer bytes. The compressed information is then sent in a way similar to block mode, using a header+payload record format.

FTP Data Representation: Data Types, Data Structures and Format Control

If you move a text file from one type of system to another using regular FTP, the data will all get moved exactly as it was. Moving a text file from a UNIX system to a PC as just a set of bytes would mean programs would not properly recognize end of line markers.

FTP Data Types

The first piece of information that can be given about a file is its data type, which dictates the overall representation of the file. There are four different data types specified in the FTP standard:

- **ASCII:** Defines an ASCII text file, with lines marked by some sort of end-of-line marker as described above.
- **EBCDIC:** Conceptually the same as the ASCII type, but for files using IBM's EBCDIC character set.
- **Image:** The file has no formal internal structure and is sent one byte at a time without any processing; this is the “black box” mode I mentioned above.
- **Local:** This data type is used to handle files that may store data in logical bytes containing a number of bits other than 8. Specifying this type along with the way the data is structured allows the data to be stored on the destination system in a manner consistent with its local representation

Key Concept: FTP defines four data types: ASCII, EBCDIC, image and local. ASCII and EBCDIC are used for text files in the ASCII and EBCDIC character sets, respectively; the image type is used for files with no specific structure, and the local type for local representation. The ASCII type is important because it allows text files to be transferred successfully between file systems that may use different methods of indicating the end of a line of text. The image type, also called binary, is used for files that must be sent and received byte-for-byte with no transformation, such as executable files, graphics and files with arbitrary formats.

FTP Format Control

For the ASCII and EBCDIC types, FTP defines an optional parameter called format control. This allows a user to specify a particular representation for how vertical formatting is used to describe a file. The three options are:

- **Non Print:** The default, indicating no vertical formatting.
- **Telnet Format:** Indicates that vertical format control characters, as specified in the Telnet protocol, are used in this file.
- **Carriage Control / FORTRAN:** The file uses format control characters given as the first character of each line, as specified for the FORTRAN programming language.

FTP Data Structures

In addition to specifying a file's data type, it is also possible to specify the file's data structure. There are three possibilities:

- **File Structure:** The file is a contiguous stream of bytes with no internal structure.
- **Record Structure:** The file consists of a set of sequential records, each of which is delimited by an end-of-record marker.
- **Page Structure:** The file contains a set of special indexed data pages.

You work in a ABC Cyber Law firm and investigating the case in which the phone call was made for extortion in a kidnapping case from the stolen mobile phone which primarily belongs to person A. Person A never files a complaint against the same as the cost of the phone was very less. In future analysis you have identified a mail from a X .

You being the cyber law consulted prepare a report with applicable case study laws and conclusion.

FTP Command Groups

Each command is identified by a short three-letter or four-letter command code for convenience, and performs a specific task in the overall functionality of FTP.

There are several dozen of these protocol commands, and to help organize them, the FTP standard categorizes them into three groups, based on overall function type:

- **Access Control Commands:** Commands that are part of the user login and authentication process, are used for resource access, or are part of general session control.
- **Transfer Parameter Commands:** Commands that specify parameters for how data transfers should occur. For example, commands in this group specify the data type of a file to be sent, indicate whether passive or active data connections will be used, and so forth.
- **FTP Service Commands:** This is the largest group, containing all the commands that actually perform file operations, such as sending and receiving files. Commands to implement support functions, such as deleting or renaming files, are also here.

Key Concept: Each command sent by the FTP client results in a reply sent by the FTP server. FTP replies consist of a three-digit numeric reply code, along with a line of descriptive text. The reply code serves to standardize FTP replies, both so they can be interpreted by client software, and so experienced users can see at a glance what the results were of a command. The reply code is structured so that the first two digits indicate the type of reply and to what category it belongs.

Key Concept: The FTP user interface is the component on the FTP client that acts as an intermediary between the human user and the FTP software. The existence of the user interface allows FTP to be used in a friendly manner without requiring knowledge of FTP's internal protocol commands. Most FTP software uses either a command-line interface that understands English-like user commands, or a graphical interface, where mouse clicks and other graphical operations are translated into FTP commands.

Trivial File Transfer Protocol (TFTP)

In cases where only the most basic file transfer functions are needed and simplicity and small program size is of paramount importance, a companion to FTP was created called the Trivial File Transfer Protocol (TFTP).

TFTP is a greatly simplified version of FTP that allows only basic operations and lacks some of FTP's fancy capabilities, in order to keep its implementation easy (even "trivial"!) and its program size small.

Some of the more significant specific differences between FTP and TFTP:

- **Transport:** The comparison to TCP and UDP is apt not only based on the features/ simplicity trade-off, but because FTP uses TCP for transport while TFTP uses UDP.
- **Limited Command Set:** FTP includes a rich set of commands to allow files to be sent, received, renamed, deleted and so forth. TFTP only allows files to be sent and received.

- **Limited Data Representations:** TFTP does not include some of FTP's fancy data representation options; it allows only simple ASCII or binary file transfers.
- **Lack of Authentication:** UDP uses no login mechanism or other means of authentication. This is again a simplification, though it means the operators of TFTP servers must severely restrict the files they make available for access.

Connection Establishment and Identification

The TFTP server listens continuously for requests on well-known UDP port number 69, which is reserved for TFTP. The client chooses for its initial communication an ephemeral port number, as is usually the case in TCP/IP. This port number actually identifies the data transfer, and is called a transfer identifier (TID).

Key Concept: Since TFTP uses UDP rather than TCP, there is no explicit concept of a connection as in FTP. A TFTP session instead uses the concept of a "logical connection", which is opened when a client sends a request to a server to read or write a file. Communication between the client and server is performed in "lock-step" fashion: one device sends data messages and receives acknowledgments so it knows the data messages were received; the other sends acknowledgments and receives data messages so it knows the acknowledgments were received.

TFTP Detailed Operation and Messaging

Initial Message Exchange

The first message sent by the client to initiate TFTP is either a read request (RRQ) message or a write request (WRQ) message. This message serves to implicitly establish the logical TFTP connection, and to indicate if the file is to be sent from

the server to the client (read request) or the client to the server (write request). The message also specifies the type of file transfer to be performed.

Data Block Numbering

Each data message contains a block of between 0 and 512 bytes of data. The blocks are numbered sequentially, starting with 1. The number of each block is placed in the header of the data message carrying that block, and then used in the acknowledgment for that block so the original sender knows it was received. The device sending the data will always send 512 bytes of data at a time for as long as it has enough data to fill the message. When it gets to the end of the file and has fewer than 512 bytes to send, it will send only as many bytes as remain.

TFTP Read Process Steps

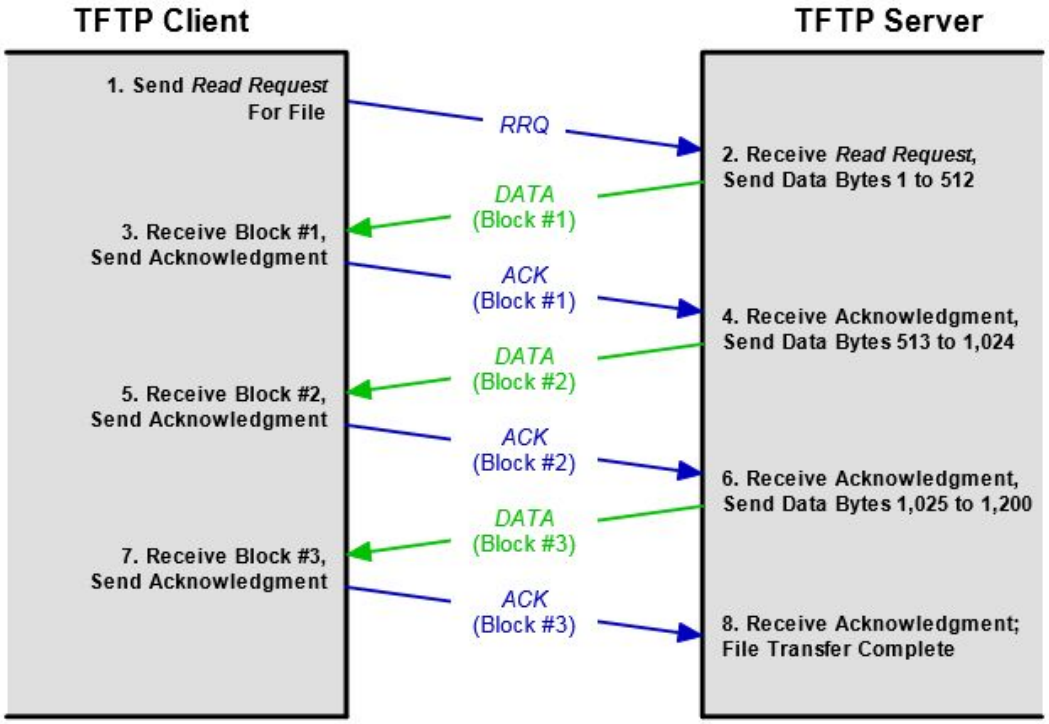


Figure 293: TFTP Read Process

In this example, the client starts the process of reading a file by sending a request for it to the server. The server acknowledges this request by immediately sending a *DATA* message carrying block #1, containing the first 512 bytes of the file. The client acknowledges this with an *ACK* message for block #1. The server then sends block #2, containing bytes 513 to 1,024, which the client acknowledges. When the client receives block #3, it realizes it has only 176 bytes, which marks it as the last block of the file.

Key Concept: TFTP is supposed to be a small and simple protocol, so it includes few "extra" features. One that it does support is option negotiation, where a TFTP client and server attempt to come to agreement on additional parameters that they will use in transferring a file. The TFTP client includes one or more options in its Read Request or Write Request message; the TFTP server then sends an Option Acknowledgment (OACK) message listing each option the server agrees to use. The use of options when reading a file means that an extra acknowledgment must be sent by the client--to acknowledge the OACK--before the server sends the first block of the file.

TFTP Read Process With and Without Option Negotiation

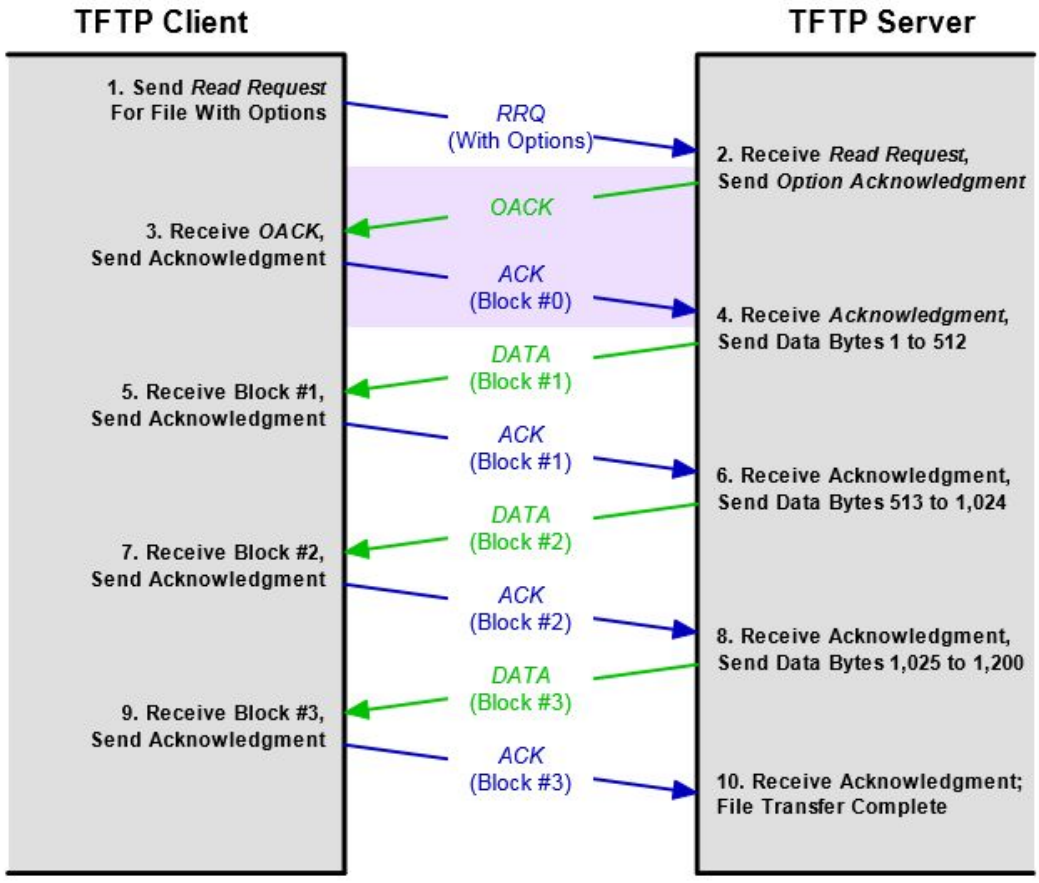


Figure 295: TFTP Read Process With Option Negotiation

This diagram shows the same example illustrated in Figure 293, but with one added message exchange used for option negotiation (purple background). The client's initial *Read Request* here includes options that it wants to use for this transfer. The server responds not immediately with the first data block, but with an *Option Acknowledgment*. The client indicates receipt of the *OACK* by sending an acknowledgment using block #0. The server sends data block #1 and the rest of the exchange proceeds as normal.

TCP/IP Electronic Mail System: Concepts and Protocols (RFC 822, MIME, SMTP, POP3, IMAP)

TCP/IP Electronic Mail System Overview and Concepts

Electronic mail in the TCP/IP protocol suite is not implemented as just a single protocol or technology. Rather, it is a complete system that contains a number of related components that work together. These include standards defining methods for addressing and message formatting, and a number of protocols that play different functions in implementing electronic mail messaging.

One of the most important general concepts in the modern electronic mail system is that a distinction is made between protocols that deliver electronic mail between SMTP hosts on the internetwork, and those that let users access received mail on their local hosts.

Key Concept: One of the most important TCP/IP applications is the internetworking equivalent of the real-world postal delivery system, commonly called electronic mail or e-mail. The history of e-mail goes back to the very earliest days of TCP/IP's development; today it is used by millions of people every day to send both simple and complex messages around the world. TCP/IP e-mail is not a single application, but rather a complete system that includes several protocols, software elements and components.

E-Mail Communication Process Steps

1. **Mail Compositio: -** A user begins the e-mail journey by creating an electronic mail message. The message contains two sections: the header and the body. The body of the message is the actual information to be communicated; the header contains data that describes the message and controls how it is delivered and processed.
2. **Mail Submission: -**Electronic mail is different from many other internetworking applications in that the sender and receiver of a message do

not necessarily need to be connected to the network simultaneously, nor even continuously, to use it.

3. **Mail Delivery: -** The electronic mail message is accepted by the sender's local SMTP system for delivery through the mail system to the destination user. Today, this is accomplished by performing a Domain Name System (DNS) lookup of the intended recipient's host system and establishing an SMTP connection to that system. SMTP also supports the ability to specify a sequence of SMTP servers through which a message must be passed to reach a destination.
4. **Mail Receipt and Processing: -**The local SMTP server accepts the e-mail message and processes it. It places the mail into the intended recipient's mail box, where it waits for the user to retrieve it.
5. **Mail Access and Retrieval: -** The intended recipient periodically checks with its local SMTP server to see if any mail has arrived. If so, the recipient retrieves the mail, opens it and reads its content. This is done using not SMTP but a special mail access protocol or method.

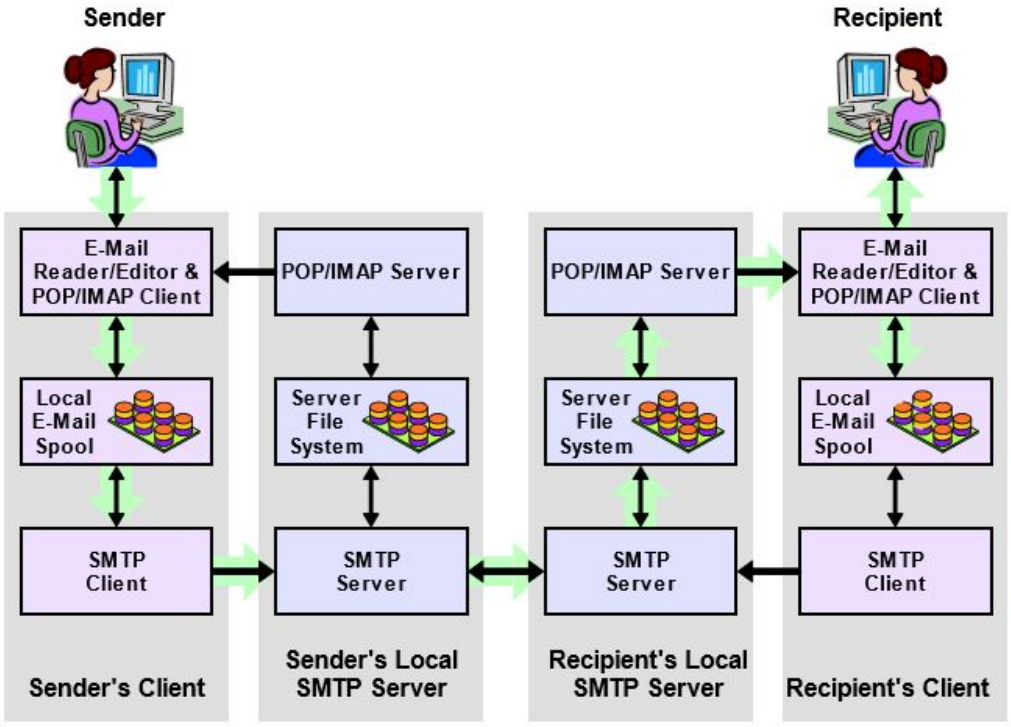


Figure 301: Electronic Mail (E-Mail) Communication Model

This diagram shows the four devices that are involved in a typical e-mail communication between two users. (Yes, they ***are*** identical twins, imagine that! ☺) Each device consists of a number of different elements, which communicate as indicated by the black arrows. Note the inherent asymmetry, because the method used to send an e-mail from a user is not the same as that used to retrieve it from the server. The large green arrows show a typical transaction: the sender composes mail and it goes to her local e-mail spool. It is sent to the sender's local SMTP server using SMTP, and then to the recipient's SMTP server, where it goes into that user's inbox. It is then retrieved, usually using a protocol such as POP or IMAP.

Key Concept: One of the critical requirements of an electronic mail system is that the sender and receiver of a message not be required to both be on the system at the time mail is sent. TCP/IP therefore uses a communication model with several devices that allow the sender and recipient to be decoupled. The sender's client device spools mail and moves it to the sender's local SMTP server when it is ready for transmission; the email is then transmitted to the receiver's SMTP server using SMTP. The email can remain on the recipient's server for an indefinite period of time. When the recipient is ready to read it, he or she retrieves it using one or more of a set of mail access protocols and methods, the two most popular of which are POP and IMAP.

Key Concept: Some form of addressing is required for all network communication; since electronic mail is user-oriented, e-mail addresses are based on users as well. In modern TCP/IP e-mail, standard addresses consist of a user name, which specifies who the recipient is, and a domain name, which specifies the DNS domain where the user is located. A special DNS mail exchange (MX) record is set up for each domain that accepts e-mail, so a sending SMTP server can determine what SMTP server it should use to send mail to a particular recipient.

Key Concept: One of the many benefits of electronic mail is that it is easy to send a message to many people at once, simply by specifying several recipient addresses. This permits easy and simple group communication, because each recipient can then do a group reply to send a response to each of the people who were sent the original message. Electronic mailing lists provide a more formalized way for groups to exchange ideas and information; there are many thousands of such lists in existence on the Internet.

Key Concept: To ensure that every device on a TCP/IP internetwork can read email sent by every other device, all messages are required to adhere to a specific structure. The standard that first specified the form of modern TCP/IP e-mail messages was RFC 822, and as a result, this is now called the RFC 822 message format. An RFC 822 message consist of a set of message headers and a message body, which are separated by a blank line. RFC 822 messages must contain only plain ASCII text characters; each line must be no more than 1000 characters in

length, and the last two characters must be the ASCII characters “CR” and “LF” to mark the end of the line.

Key Concept: Each RFC 822 message begins with a set of headers that carry essential information about the message. These headers are used to manage how the message is processed and interpreted, and also describe the contents of the message body. Each header consists of a header name and a header value. There are over a dozen different standard RFC 822 headers, which are organized into groups; it is also possible for customized user headers to be defined.

TCP/IP Enhanced Electronic Mail Message Format: Multipurpose Internet Mail Extensions (MIME)

To allow e-mail to carry multimedia information, arbitrary files, and messages in languages using character sets other than ASCII, the Multipurpose Internet Mail Extensions (MIME) standard was created.

MIME allows regular RFC 822 e-mail messages to carry the following:

- Non-text information, including graphic files, multimedia clips and all other non-text data examples.
- Arbitrary binary files, including executable programs and files stored in proprietary formats (for example, AutoCAD files, Adobe Acrobat PDF files and so forth);
- Text messages that use character sets other than ASCII. This even includes the ability to use non-ASCII characters in the headers of RFC 822 e-mail messages.

Primary MIME Headers

The first of the five main MIME standards, RFC 2045, describes a set of five primary MIME headers that communicate basic information about the content of each MIME entity

MIME-Version

First, It identifies the e-mail message as being MIME - encoded. Second even through only one version of MIME has been defined so far,having a version number header provides “future-proofing” in case a new version is created later that may have some incompatibilities with the present one.

Content-Type

Describes the nature of the data that is encoded in the MIME entity. This header specifies a content type and a content subtype, which are separated by a slash character.

Content-Transfer-Encoding

For a message using simple structure, specifies the specific method that was used to encode the data in the message body; for a composite message, identifies the encoding method for each MIME body part.

Content-ID

Allows the MIME content to be assigned a specific identification code.

Content-Description

This is an optional header that allows an arbitrary additional text description to be associated with the MIME entity.

- There are also some additional MIME headers such as Content-Disposition, Content-Location, Content-Length

TCP/IP Electronic Mail Delivery Protocol: The Simple Mail Transfer Protocol (SMTP)

Key Concept: The most important component of the TCP/IP electronic mail system is the Simple Mail Transfer Protocol (SMTP). SMTP was derived from the earlier Mail Transfer Protocol (MTP), and is the mechanism used for the delivery of mail between TCP/IP systems and users. The only part of the e-mail system for which SMTP is not used is the final retrieval step by an e-mail recipient.

Modern E-Mail Communication Using DNS and Direct Delivery

In the new system, SMTP communication is much simpler and more direct. The sending SMTP server uses DNS to find the MX record of the domain to which the e-mail is addressed. This gives the sender the DNS name of the recipient's SMTP server. This is resolved to an IP address, and a connection can be made directly from the sender's SMTP server to the recipient's to deliver the e-mail.

Key Concept: SMTP servers both send and receive e-mail; the device sending mail acts as a client for that transaction; the one receiving it acts as a server. To avoid confusion, it is easier to refer to the device sending e-mail as the SMTP sender and the one receiving as the SMTP receiver; these were the terms used when SMTP was originally created.

Key Concept: An SMTP session consists of three basic phases. The session is first established through the creation of a TCP connection and the exchange of identity information between the SMTP sender and receiver using the HELO command. Once established, mail transactions can be performed. When the SMTP sender is done with the session, it terminates it using the QUIT command. If SMTP extensions are supported, the SMTP sender uses the EHLO (extended hello) command instead of HELO, and the SMTP receiver replies with a list of extensions it will allow the SMTP sender to use.

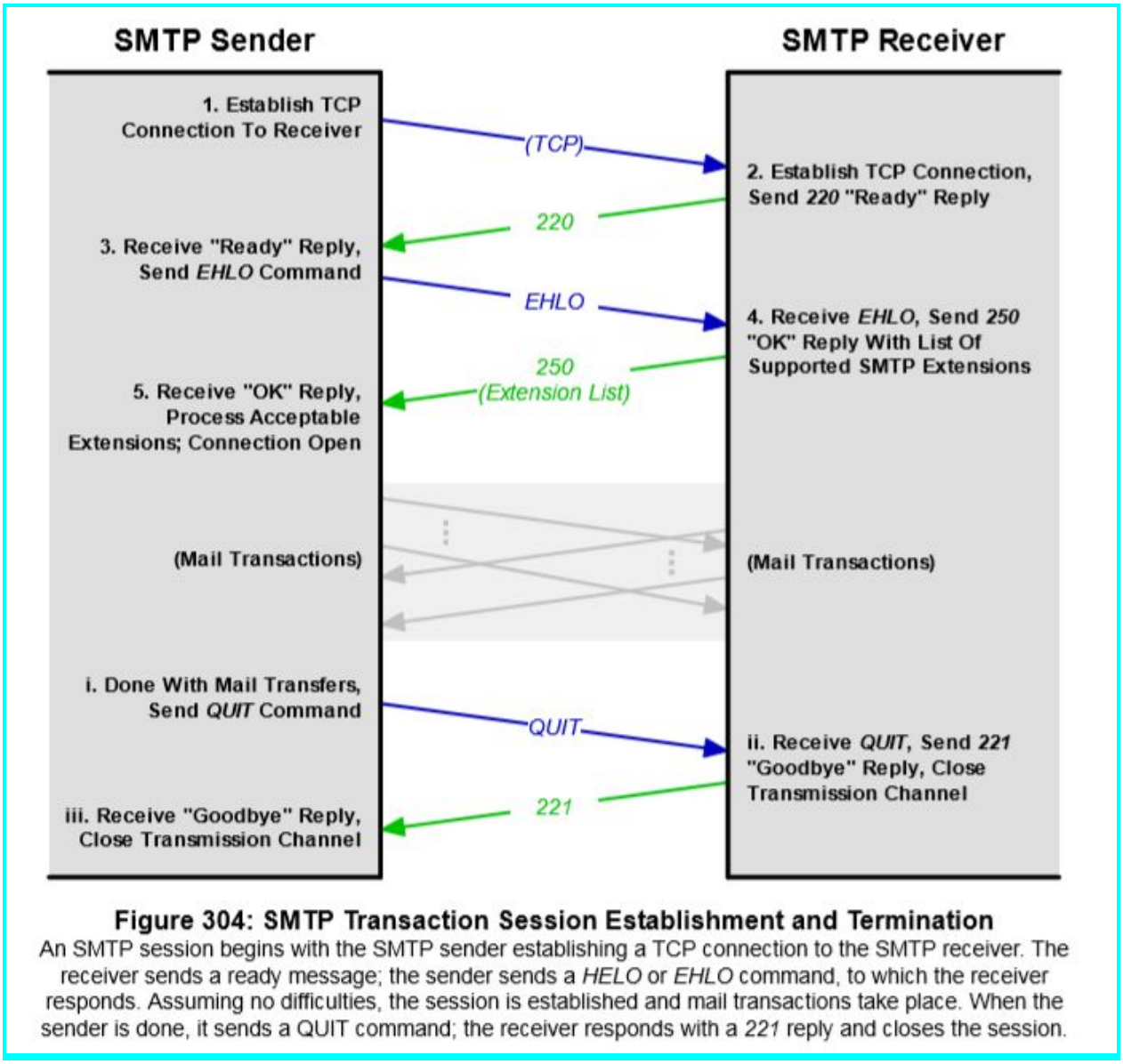


Figure 304: SMTP Transaction Session Establishment and Termination
An SMTP session begins with the SMTP sender establishing a TCP connection to the SMTP receiver. The receiver sends a ready message; the sender sends a *HELO* or *EHLO* command, to which the receiver responds. Assuming no difficulties, the session is established and mail transactions take place. When the sender is done, it sends a QUIT command; the receiver responds with a *221* reply and closes the session.

Common SMTP Server Security Techniques

- Checking the IP address of a device attempting connection and refusing to even start an SMTP session unless it is in a list of authorized client devices.
- Limiting the use of commands such as EXPN to prevent unauthorized users from determining the e-mail addresses of users on mailing lists.
- Limiting the size of e-mail messages that may be sent or the number that may be sent in a given period of time.

- Logging all access to the server to keep records of server use and check for abuse.

TCP/IP Post Office Protocol (POP/POP3)

The Post Office Protocol (POP) was designed for quick, simple and efficient mail access; it is used by millions of people to access billions of e-mail messages every day.

The idea behind POP was to provide a simple way for a client computer to retrieve e-mail from a mailbox on an SMTP server so it could used locally.

It describes only a simple sequence of operations where a user gives a name and password for authentication, and then downloads the entire contents of a mailbox. Simple is good, but there are limits.

POP2 expanded the capabilities of POP by defining a much richer set of commands and replies. This includes the important ability of being able to read only certain messages, rather than dumping a whole mailbox. Of course, this came at the cost of a slight increase in protocol complexity, but POP2 was still quite simple as protocols go.

. POP3 was based closely on POP2, but refined and enhanced with the idea of providing a simple and efficient way for PCs and other clients not normally connected to the Internet to access and retrieve e-mail.

. POP3 is a straight-forward state-based protocol, with a client and server proceeding through three stages during a session. A very small number of commands are defined to perform simple tasks, and even after all the changes and revisions described above, the protocol has a minimum of “fluff”.

POP3 General Operation, Client/Server Communication and Session States

POP3 uses the Transmission Control Protocol (TCP) for communication, to ensure the reliable transfer of commands, responses and message data. POP3 servers "listen" on well-known port number 110 for incoming connection requests from POP3 clients. After a TCP connection is established, the POP3 session is activated. The client sends commands to the server, which replies with responses and/or e-mail message contents.

Session States

1. Authorization State: The server provides a greeting to the client to indicate that it is ready for commands. The client then provides authentication information to allow access to the user's mailbox.
2. *Transaction State*: The client is allowed to perform various operations on the mailbox. These include listing and retrieving messages, and marking retrieved messages for deletion.
3. *Update State*: When the client is done with all of its tasks and issues the QUIT command, the session enters this state automatically, where the server actually deletes the messages marked for deletion in the Transaction state.

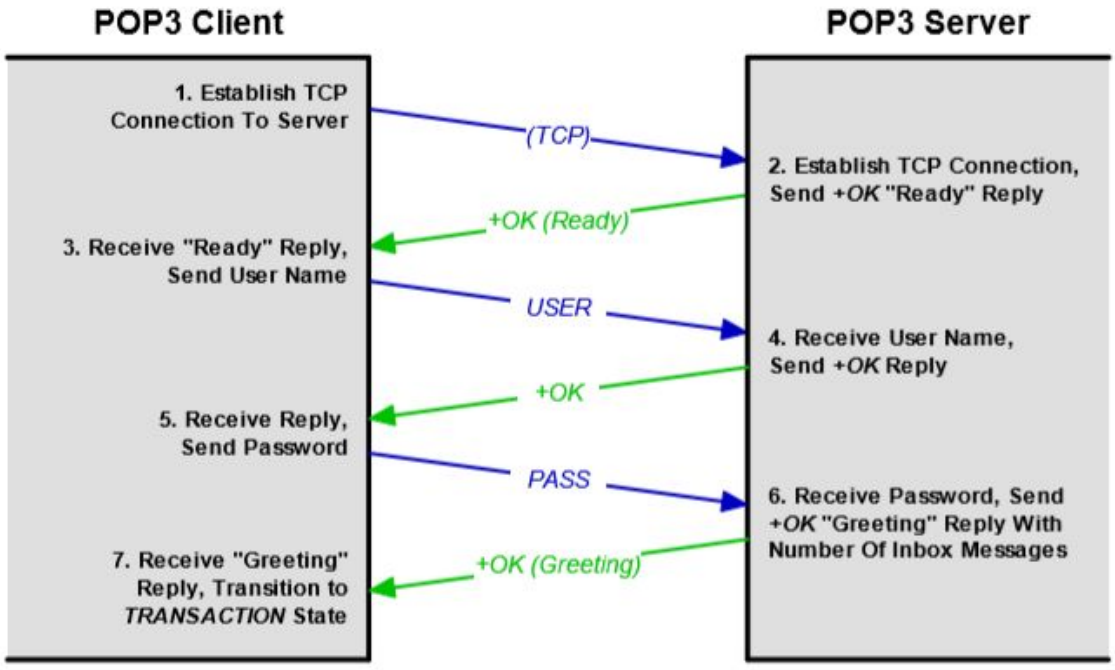


Figure 307: Post Office Protocol (POP3) User Authentication Process

Once the TCP connection is established from the client to the server, the server responds with a greeting message, and the simple POP3 authentication process begins. The client sends a user name and password to the server using the *USER* and *PASS* commands, and the server evaluates the information to determine whether or not it will allow the client access.

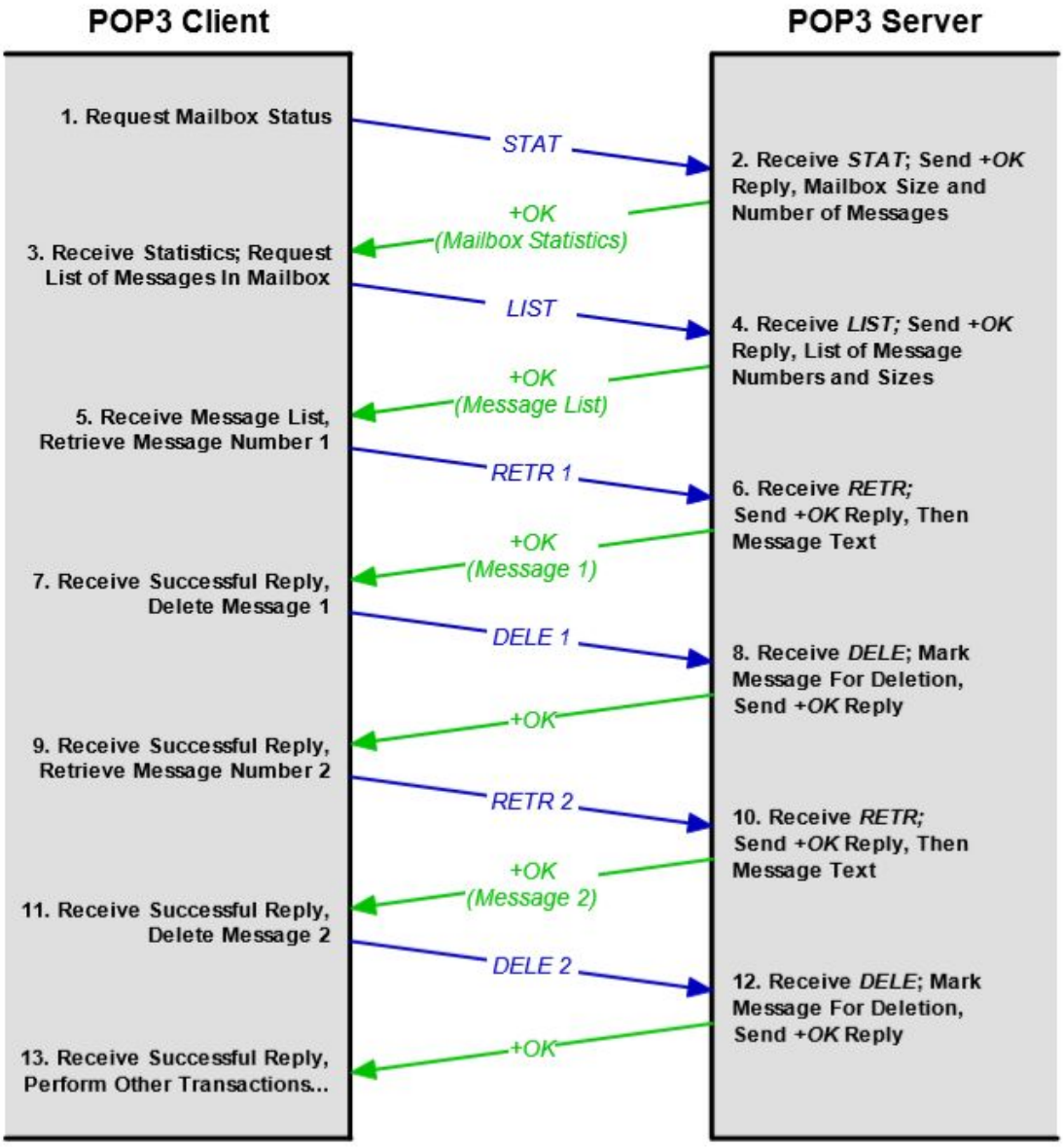


Figure 308: Post Office Protocol (POP3) Mail Exchange Process

This diagram shows the typical exchange of commands and replies employed by a POP3 client to retrieve e-mail from a POP3 server. The *STAT* command is used to get mailbox statistics, followed by the *LIST* command to obtain a list of message numbers. Each message in turn is then retrieved using *RETR* and marked for deletion by *DELE*. (Messages are not actually deleted until the *Update* state is entered.)

TCP/IP Internet Message Access Protocol (IMAP/IMAP4)

To provide better control over how mail is accessed and managed, we must use either the online or disconnected access models. The Internet Message Access Protocol (IMAP) was created to allow these access models to be used, providing rich functionality and flexibility for the TCP/IP e-mail user.

IMAP allows a user to do all of the following:

- Access and retrieve mail from a remote server so it can be used locally while retaining it on the server
- Set message flags so that the user can keep track of which messages he or she has already seen, already answered, and so on.
- Manage multiple mailboxes and transfer messages from one mailbox to another.
- Download only portions of a message, such as one body part from a MIME multipart message. This can be quite helpful in cases where large multimedia files are combined with short text elements in a single message.

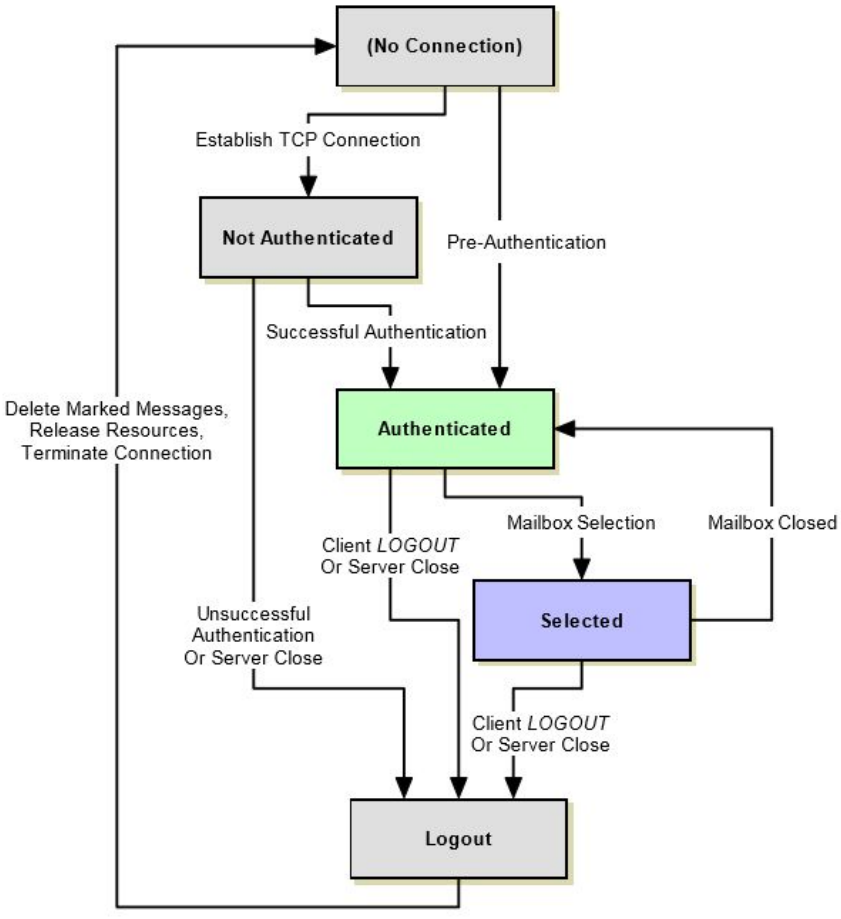


Figure 309: Internet Message Access Protocol (IMAP) Finite State Machine

The following are the IMAP states, in the usual sequence in which they occur for a session:

1. **Not Authenticated State:** The session normally begins in this state after a TCP connection is established, unless the special IMAP preauthentication feature has been used.
2. **Authenticated State:** The client has completed authentication, either through an authentication process in the prior state or through preauthentication. The client is now allowed to perform operations on whole mailboxes. The client must select a mailbox before individual message operations are permitted.
3. **Selected State:** After a mailbox has been chosen, the client is allowed to access and manipulate individual messages within the mailbox. When the client is done with the current mailbox it can close it and return to the Authenticated state to select a new one to work with, or can log out to end the session.
4. **Logout State:** The client may issue a Logout command from any of the other states to request that the IMAP session be ended.

Key Concept: IMAP tags its commands with a unique identifier. These tags can then be used in replies by the server to match replies with the commands to which they correspond. This enables multiple commands to be sent to an IMAP server in succession.

IMAP Authentication Methods

The authentication methods are:

1. **Plain Login:** This is the typical "username / password" technique, using the LOGIN command by itself. In IMAP4 one command is used to send both the username and password. Since the command and parameters are sent in plain text, this is by far the least secure method of authentication

2. **TLS Login:** This is a secure login where the Transport Layer Security (TLS) protocol is first enabled with the STARTTLS command, and then the LOGIN command can be used securely.
3. **Negotiated Authentication Method:** The AUTHENTICATE command allows the client and server to use any authentication scheme that they both support. The server may indicate which schemes it supports in response to a CAPABILITY command. After specifying the authentication mechanism to be used, the server and client exchange authentication information as required by the mechanism specified. This may require one or more additional lines of data to be sent.

Key Concept: After the client opens a specific mailbox, the IMAP session enters the Selected state, where operations such as reading and copying individual e-mail messages may be performed. The two most important commands used in this state are FETCH, which can be used to retrieve a whole message, part of a message, or only certain message headers or flags; and STORE, which sets a message's status information. IMAP also includes a powerful search facility, providing users with great flexibility in finding messages in a mailbox. When the client is done working with a particular mailbox, it may choose a different one and re-enter the Selected state, close the mailbox and return to the Authenticated state, or log out, automatically entering the Logout state.

Key Concept: Instead of using a dedicated protocol like POP3 or IMAP4 to retrieve mail, on some systems it is possible for a user to have direct server access to e-mail. This is most commonly done on UNIX systems, where protocols like Telnet or NFS can give a user shared access to mailboxes on a server. This is the oldest method of e-mail access; it provides the user with the most control over his or her mailbox, and is well-suited to those who must access mail from many locations. The main drawback is that it means the user must be on the Internet to read e-mail, and it also usually requires familiarity with the UNIX operating system, which few people use today

Usenet (Network News) and the TCP/IP Network News Transfer Protocol (NNTP)

For distributing news and other types of general information over internetworks, a messaging system called both Usenet (for user's network) and Network News was created. This application is like e-mail in allowing messages to be written and read by large numbers of users. In Usenet, anyone can write a message that can be read by any number of recipients, and can likewise respond to messages written by others.

Overview of Usenet Operation and Characteristics

Usenet begins with a user writing a message to be distributed. After the message is posted to say, the group on TCP/IP networking, it is stored on that user's local news server, and special software sends copies of it to other connected news servers. The message eventually propagates around the world, where anyone who chooses to read the TCP/IP networking newsgroup can see the message.

Key Concept: One of the very first online electronic communities was set up in 1979 by university students who wanted to keep in touch and share news and other information. Today, this User's Network (Usenet), also called Network News, has grown into a logical network that spans the globe. By posting messages to a Usenet newsgroup, people can share information on a variety of subjects of interest. Usenet was originally implemented in the form of direct connections established between participating hosts; today the Internet is the vehicle for message transport

Usenet Communication Model: Message Composition, Posting, Storage, Propagation and Access

The main issue with e-mail in this respect is that only the individuals who are specified as recipients of a message can read it. There is no facility whereby someone can write a message and put it in an open place where anybody who wants can read it, analogous to posting a newsletter in a public place.

This affects every aspect of how Usenet communication works:

- **Addressing:** Messages are not addressed from a sender to any particular recipient or set of recipients, but rather to a group, which is identified with a newsgroup name.
- **Storage:** Messages are not stored in individual mailboxes but in a central location on a server, where any user of the server can access them.
- **Delivery:** Messages are not conveyed from the sender's system to the recipient's system, but are rather spread over the Internet to all connected systems so anyone can read them.

Table 264: Usenet "Big Eight" Newsgroup Hierarchies

Hierarchy	Description
comp.*	Newsgroups discussing computer-related topics, including hardware, software, operating systems and techniques.
humanities.*	Groups discussing the humanities, such as literature and art.
misc.*	Miscellaneous topics that don't fit into other Big Eight hierarchies.
news.*	Groups discussing Usenet itself and its administration.
rec.*	Recreation topics, such as games, sports and activities.
sci.*	Science newsgroups, covering specific areas such as physics and chemistry, research topics and so forth.

Table 264: Usenet "Big Eight" Newsgroup Hierarchies

Hierarchy	Description
soc.*	Society and social discussions, including groups on specific cultures.
talk.*	Groups primarily oriented around discussion and debate of current events and happenings.

TCP/IP Network News Transfer Protocol (NNTP)

NNTP uses TCP, with servers listening on well-known TCP port 119 for incoming connections, either from client hosts or other NNTP servers. As in SMTP, when two servers communicate using NNTP, the one that initiates the connection plays the role of client for that exchange.

After a connection is established, communication takes the form of commands sent by the client to the server, and replies returned from the server to the client device. NNTP commands are sent as plain ASCII text, NNTP responses take the form of three-digit reply codes as well as descriptive text.

Key Concept: The Network News Transfer Protocol (NNTP) is the protocol used to implement message communication in modern Usenet. It is used for two primary purposes: to propagate messages between NNTP servers, and to permit NNTP clients to post and read articles. It is a standalone protocol, but shares many characteristics with e-mail's Simple Mail Transfer Protocol (SMTP).

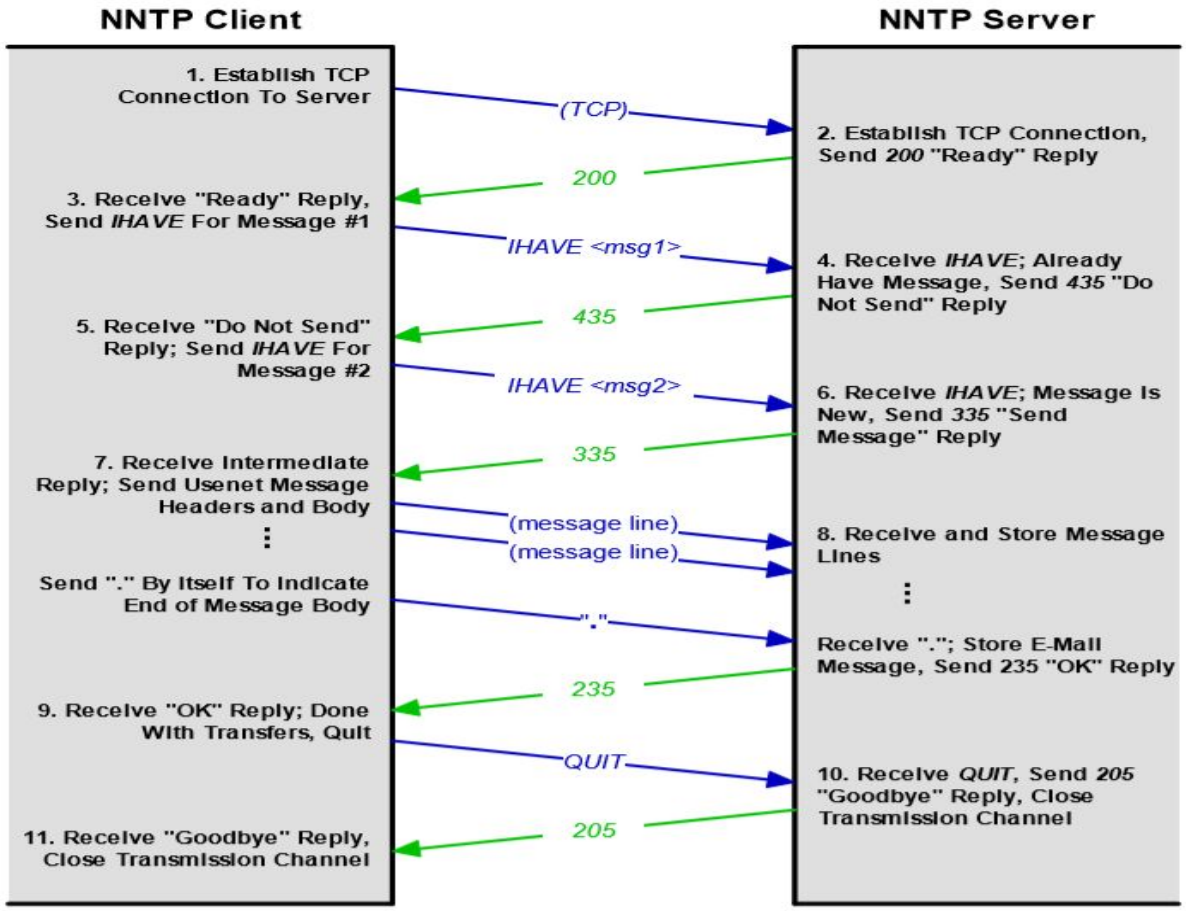


Figure 312: NNTP Article Propagation Using The "Push" Model

TCP/IP World Wide Web (WWW, "The Web") and the Hypertext Transfer Protocol (HTTP)

Key Concept: The World Wide Web (WWW) began in 1989 as a project designed to facilitate the representation of relationships between documents and the sharing of information between researchers. The main feature of the Web that makes it so powerful is hypertext, which allows links to be made from one document to another. The many benefits of the Web caused it to grow in only a few short years from a small application to the largest and arguably most important application in the world of networking; it is largely responsible for bringing the Internet into the mainstream of society.

Major Functional Components of the Web

- **HyperText Markup Language(HTML):** A text language used to define hypertext documents. The idea behind HTML was to add simple constructs, called tags, to regular text documents, to enable the linking of one document to another, as well as to allow special data formatting and the combining of different types of media.
- **Hypertext Transfer Protocol (HTTP):** The TCP/IP application-layer protocol that implements the World Wide Web, by enabling the transfer of hypertext documents and other files between a client and server. HTTP began as a very crude protocol for transferring HTML documents between computers, and has evolved to a full-featured and sophisticated messaging protocol.
- **Uniform Resource Identifiers (URIs):** A method of defining labels that identify resources on an internet so that they can be easily found and referenced. URIs were originally developed to provide a means by which the users of the Web could locate hypertext documents so they could be retrieved.

Web Hardware and Software

These three main components are supplemented by a number of other elements that play “supporting roles” hardware and software used to implement client/server communication that makes the web work: Web servers and Web browsers.

Web servers are computers that run special server software to allow them to provide hypertext documents and other files to clients who request them.

Web browsers are HTTP client software programs that run on TCP/IP client computers to access Web documents on Web servers. These browser programs retrieve hypertext documents and display them, and also implement many of the Web's advanced features, such as caching. Today's browsers support a wide variety of media, allowing the Web to implement many different functions aside from simply hypertext document transfer.

World Wide Web Media and the Hypertext Markup Language (HTML)

The standard language used by the World Wide Web is thus called the Hypertext Markup Language (HTML). HTML is one of the three primary system components of the World Wide Web, and was invented in 1990 by the creator of the Web, Tim Berners-Lee.

For the purposes of hypertext, the most basic type of information in a document is a special instruction that specifies how one document can be linked to another—after all, this linking process is the defining attribute of hypertext. It defines a full set of text codes for describing nearly every aspect of how a document is shown to a user.

HTML Elements and Tags

HTML document is a plain ASCII text file like an e-mail message or other text document. The biggest difference between HTML and regular text, however, is that HTML documents are structured. The document is logically organized into a series of elements that are arranged according to the rules of the language.

Each element is described using special text tags that follow a particular syntax. Each tag begins with the "<" symbol, which is then followed by the (case-insensitive) element name, and optionally, additional parameters that describe the element. The tag ends with the ">" symbol.

World Wide Web Addressing: HTTP Uniform Resource Locators (URLs)

Key Concept: Uniform Resource Identifiers (URIs) were developed to allow World Wide Web resources to be easily and consistently identified; they are also now used for other protocols and applications. The type of URI currently used on the Web is the Uniform Resource Locator (URL), which identifies the use of HTTP to retrieve a resource, and provides information on where and how it can be found and retrieved

TCP/IP Hypertext Transfer Protocol (HTTP)

Some sort of mechanism was needed to allow a client computer to tell a server to send it a document. To fill this function, the early developers of the Web created a new TCP/IP application layer protocol: the Hypertext Transfer Protocol (HTTP).

HTTP/0.9

The original version of HTTP was intended only for the transfer of hypertext documents, and was designed to be very simple to make implementation of the fledgling Web easier. This early HTTP specifies that an HTTP client establishes a connection to an HTTP server using TCP. The client then issues a single "GET" request specifying a resource to be retrieved. The server responds by sending the file as a stream of text bytes, and the connection is terminated

HTTP/1.0

HTTP/1.0 transformed HTTP from a trivial request/response application to a true messaging protocol. It described a complete message format for HTTP, and explained how it should be used for client requests and server responses. One of the most important changes in HTTP/1.0 was the generalization of the protocol to handle many types of different media, as opposed to strictly hypertext documents. This was done by borrowing concepts and header constructs from the **Multipurpose Internet Mail Extensions (MIME)** standard defined for e-mail.'

HTTP/1.1

Some of the more important improvements in version 1.1 are:

- **Multiple Host Name Support:** In HTTP/1.0, there was no way to specify the host name of the server to which the client needed to connect. As a result, the Web server at a particular IP address could only support one domain name. HTTP/1.1 allows one Web server to handle requests for dozens or even hundreds of different virtual hosts.
- **Persistent Connections:** HTTP/1.1 allows a client to send multiple requests for related documents to a server in a single TCP session.
- **Partial Resource Selection:** In HTTP/1.1, a client can ask for only part of a resource rather than the entire document, which reduces the load on the server and saves transfer bandwidth.
- **Better Caching and Proxying Support:** HTTP/1.1 includes many provisions to make caching and proxying more efficient and effective than they were in HTTP/1.0. These techniques can improve performance by providing clients with faster replies to their requests while reducing the load on servers, as well as enhancing security and implementing other functionality.
- **Content Negotiation:** A negotiation feature was added that allows the client and server to exchange information to help select the best resource or version of a resource when multiple variants are available.

- **Better Security:** HTTP/1.1 defines authentication methods and is generally more "security aware" than HTTP/1.0 was.

Key Concept: HTTP is a client/server-oriented, request/reply protocol. Basic communication consists of an HTTP Request message sent by an HTTP client to an HTTP server, which returns an HTTP Response message back to the client.

Key Concept: HTTP/0.9 and HTTP/1.0 only supported transitory connections between an HTTP client and server, where just a single request and response could be exchanged on a TCP connection. This is very inefficient for the modern Web, where clients frequently need to make dozens of requests to a server. HTTP/1.1 operates by default using persistent connections: once a TCP connection is established, the client can send many requests to the server and receive replies to each in turn. This allows files to be retrieved more quickly, and conserves server resources and Internet bandwidth. The client can even pipeline its requests, sending the second one immediately, without having to first wait for a reply to the first one. HTTP/1.1 still supports transitory connections for backwards compatibility, when needed.

HTTP Data Transfer, Content Encodings and Transfer Encodings

Encoding was a significant issue for MIME, because it was created for the specific purpose of sending non-text data using the old RFC 822 e-mail message standard. RFC 822 imposes several significant restrictions on the messages it carries, the most important of which is that data must be encoded using 7-bit ASCII. RFC 822 messages are also limited to lines of no more than 1000 characters that end in a "CRLF" sequence.

HTTP's Two-Level Encoding Scheme

This effort to make HTTP flexible resulted in a system of representing encodings that is actually more complicated than MIME's. The key to understanding it is to recognize that HTTP/1.1 actually splits MIME's notion of a "content transfer encoding" into two different encoding levels:

- **Content Encoding:** This is an encoding that is applied specifically to the entity carried in an HTTP message, to prepare or package it prior to transmission. Content encodings are said to be “end-to-end”, because the encoding of the entity is done once before it sent by the client or server, and only decoded upon receipt by the ultimate recipient: server or client.
- **Transfer Encoding:** This is an encoding that is done specifically for the purpose of ensuring that data can be safely transferred between devices. It is applied across an entire HTTP message, and not specifically to the entity. This type of encoding is “hopby-hop” because a different transfer encoding may be used for each hop of a message that is transmitted through many intermediaries in the request/response chain.

HTTP Data Length Issues, "Chunked" Transfers and Message Trailers

Using The Content-Length Header

There are two usual approaches to dealing with the sort of data length issue:either using an explicit delimiter to mark the end of the message, or including a length header or field to tell the recipient how long each message is.

This method works fine in cases where the size of the entity to be transferred is known in advance, such as when a static file such as a text document, image or executable program needs to be transmitted. However, there are many types of resources that are generated dynamically; the total size of such a resource is not known until it has been completely processed.

Using "Chunked" Transfers

The problem of unknown message length could be resolved by buffering the entire resource before transmission. However, this would be wasteful of server memory and would delay the transmission of the entity unnecessarily—no part could be sent until the entire entity was ready.

When this technique is used, instead of sending an entity as a raw sequence of bytes, it is broken into, well, chunks.

This allows HTTP to send a dynamically-generated resource, such as output from a script, a piece at a time as the data becomes available from the software processing it. To indicate that this method has been used, the special header "Transfer-Encoding: chunked" is placed in the message.

Gopher Protocol (Gopher)

The Gopher Protocol was developed in the late 1980s to provide a mechanism for organizing documents for easy access by students and faculty at the university. The core principle that guided the development of the system was simplicity. Gopher is designed on the basis of a small number of core principles, and uses a very straight-forward mechanism for passing information between client and server devices.

Information Storage on Gopher Servers

Information accessible by Gopher is stored as files on Gopher servers. It is organized in a hierarchical manner similar to the file system tree of a computer such as a Windows PC or UNIX workstation.

Gopher Client/Server Operation

Typical use of Gopher begins with a user on a client machine creating a TCP connection to a Gopher server using well-known TCP port number 70. After the connection is established, the server waits for the client to request a particular resource by sending the server a piece of text called a selector string.

Key Concept: The Gopher Protocol is a distributed document search and retrieval protocol that was developed at the University of Minnesota in the late 1980s. Resources are stored on Gopher servers, which organize information using a hierarchical directory structure. Gopher clients access servers to retrieve directory listings of available resources, which are presented to the user as a menu from which an item may be selected for retrieval. Gopher's chief advantage is simplicity and ease of use, but it lacks flexibility in presentation and the ability to effectively

present graphics and multimedia. For this reason, despite Gopher predating the World Wide Web, the Web has almost entirely replaced it, and Gopher is now a niche protocol.

Telnet Protocol

Telnet was designed for direct access: to let a user access a remote machine and use it as if he or she were connected to it locally.

At first glance, it may be surprising that Telnet took so long to develop, because in theory, it should be a very simple protocol to define: all it needs to do is send keystrokes and program output over the network like any other protocol.

- **The Network Virtual Terminal (NVT):** Telnet defines a standardized, fictional terminal called the Network Virtual Terminal (NVT) that is used for universal communication by all devices. A Telnet client takes input from a user and translates it from its native form to the NVT format to send to a Telnet server running on a remote computer; the server translates from NVT to whatever representation the computer being accessed requires.
- **Options and Option Negotiation:** Having Telnet clients and servers act as NVTs avoids incompatibilities between devices, but does so by stripping all terminal-specific functionality to provide a common base representation that is understood by everyone.
- **Symmetric Operation:** While Telnet is a client/server protocol, it is specifically designed to not make assumptions about the nature of the client and server software. Once a Telnet session is established, they can each send and receive data as equals. They can also each initiate the negotiation of options. This makes the protocol extremely flexible, and has led to its use in a variety of places, as we will discuss in a moment.

Telnet is used for the interactive communication of data and commands between client and server over a prolonged period of time, and is thus strongly based on the concept of a session. For this reason, Telnet runs over the connection-oriented Transmission Control Protocol (TCP). Telnet servers listen for connections on well-known TCP port number 23.

Key Concept: Telnet is a client/server protocol that uses TCP to establish a session between a user terminal and a remote host. The Telnet client software takes input from the user and sends it to the server, which feeds it to the host machine's operating system. The Telnet server takes output from the host and sends it to the client to display to the user. While Telnet is most often used to implement remote login capability, there is no concept specifically pertaining to logins in the protocol, which is general enough to allow it to be used for a variety of functions.