Chapter 8
Application Layer, Servers and Protocols

Hypertext Transfer Protocol (HTTP)
A standard Internet protocol that specifies the client/server interaction processes between Web browsers such as Mozilla Firefox and Web servers such as Apache. Its the network protocol used to deliver virtually all files and other data(collectively called resources) on the World-Wide-Web, whether they are HTML files, image files, query results or anything else. Usually HTTP takes place though TCP/IP Sockets.A Browser is an HTTP client because it sends requests to an HTTP server (Web Sever), which then sends response back to the client. The standard and default port for the HTTP servers to listen is 80, though they can use any port.

What are Resources?
HTTP is used to transmit resources not just files. A resource is some chunk of information that can be identified by a URL (its R in URL). The most common kind of resource is a file, but a resource may also be a dynamically generated query, the output of a CGI script, a document that is available in several languages or anything else.

The original Hypertext Transfer Protocol (HTTP) 1.0 protocol is a stateless protocol whereby a Web browser forms a connection with a Web server, downloads the appropriate file, and then terminates the connection. The browser usually requests a file using an HTTP GET method request on TCP port 80, which consists of a series of HTTP request headers that define the transaction method (GET, POST, HEAD, and so on) and indicates to the server the capabilities of the client. The server responds with a series of HTTP response headers that indicate whether the transaction is successful, the type of data being sent, the type of server, and finally the requested data.

IIS 4 supports a new version of this protocol called HTTP 1.1, which has new features that make it more efficient. These new features include the following:

- **Persistent connections:**
  An HTTP 1.1 server can keep TCP connections open after a file has been transferred, eliminating the need for a connection to be opened and closed each time a file is transferred, as is the case with HTTP 1.0.
- **Pipelining:**
  This is a process whereby an HTTP 1.1 client can send multiple Internet Protocol (IP) packets to the server without waiting for the server to respond to each packet.
- **Buffering:**
  This process allows several HTTP requests by the client to be buffered into a single packet and sent to the server, which results in faster transfer times because fewer and larger packets are used.
- **Host headers:**
  This feature enables an HTTP 1.1–compliant Web server to host multiple Web sites using a single IP address.
- **Http put and http delete commands:**
  These commands enable Web browsers to upload and delete files from Web servers using HTTP

HTTPS VS HTTP

As opposed to HTTP URLs that begin with "http://" and use port 80 by default, HTTPS URLs begin with "https://" and use port 443 by default. HTTP is unsecured and is subject to man-in-the-middle and eavesdropping attacks, which can let attackers gain access to website accounts and sensitive information. HTTPS is designed to withstand such attacks and is considered secure against such attacks . HTTP operates at the highest layer of the OSI Model, the Application layer; but the security protocol operates at a lower sub layer, encrypting an HTTP message prior to transmission and decrypting a message upon arrival. Strictly speaking, HTTPS is not a separate protocol, but refers to use of ordinary HTTP over an encrypted Secure Sockets Layer (SSL) or Transport Layer Security (TLS) connection. Everything in the HTTP message is encrypted, including the headers, and the request/response load.

DHCP (Dynamic Host Configuration Protocol)

A standard Internet protocol that enables the dynamic configuration of hosts on an Internet Protocol (IP) internetwork. Dynamic Host Configuration Protocol (DHCP) is an extension of the bootstrap protocol (BOOTP).

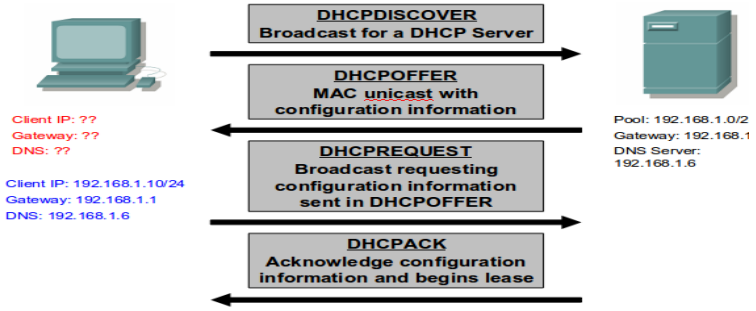
How It Works?

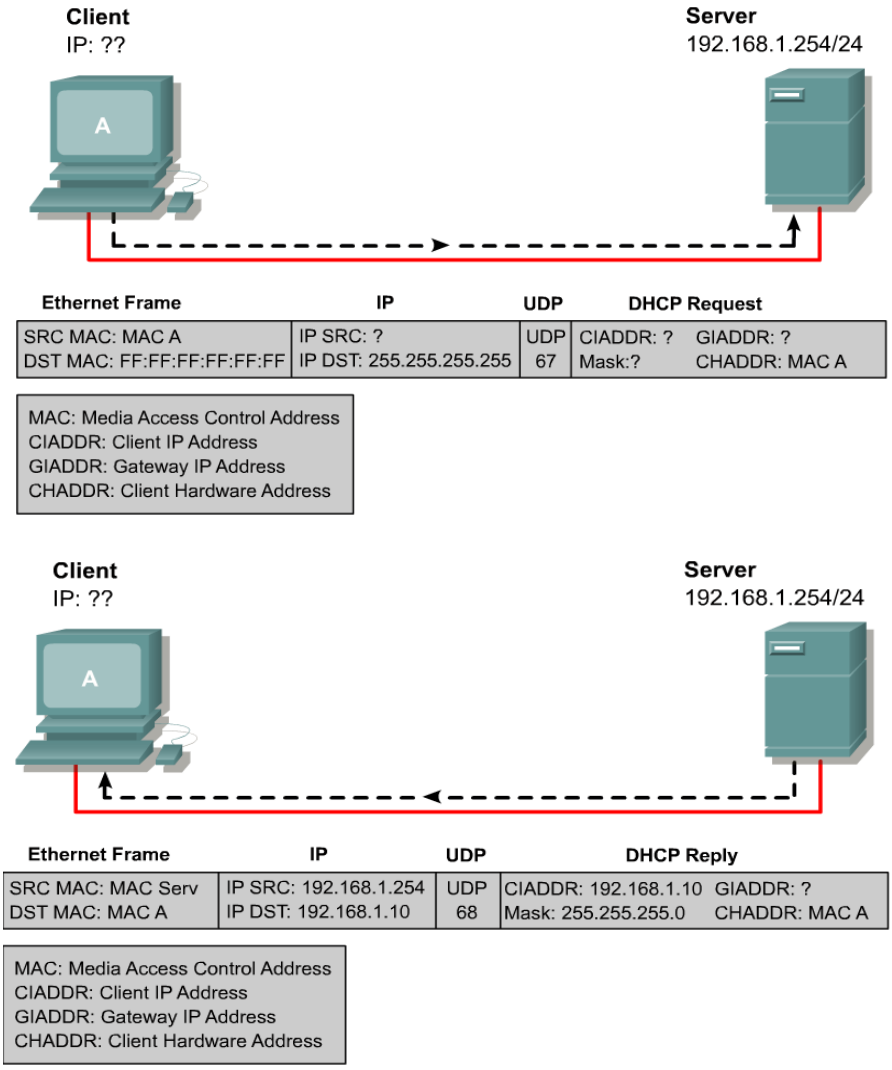
DHCP is a client-server protocol that uses DHCP servers and DHCP clients. A DHCP server is a machine that runs a service that can lease out IP addresses and other TCP/IP information to any client that requests them. For example, on Linux System example Ubuntu you can install the DHCP Server service to perform this function. The DHCP server typically has a pool of IP addresses that it is allowed to distribute to clients, and these clients lease an IP address from the pool for a specific period of time, usually several days. Once the lease is ready to expire, the client contacts the server to arrange for renewal.

DHCP clients are client machines that run special DHCP client software enabling them to communicate with DHCP servers. All versions of Linux and Windows include DHCP client software, which is installed when the TCP/IP protocol stack is installed on the machine.

DHCP clients obtain a DHCP lease for an IP address, a subnet mask, and various DHCP options from DHCP servers in a four-step process:

1. DHCPDISCOVER:
   The client broadcasts a request for a DHCP server.
2. DHCPOFFER:
   DHCP servers on the network offer an address to the client.
3. DHCPREQUEST:
   The client broadcasts a request to lease an address from one of the offering DHCP servers.
4. DHCPACK:
   The DHCP server that the client responds to acknowledges the client, assigns it any configured DHCP options, and updates its DHCP database. The client then initializes and binds its TCP/IP protocol stack and can begin network communication.





Domain Name System (DNS):

IP address are tough for human to remember and impossible to guess. Domain Name System are usually used to translate a hostname or Domain name (eg. nec.edu.np) into an IP address (eg. 202.37.94.177). Domain name comprise a hierarchy so that names are unique, yet easy to remember.

DNS makes its possible to refer to the Internet protocol (IP) based system (hosts) by human friendly names (domain names). Name resolution is that act of determining the IP address of a given hostname. The benefits of DNS are two folds. First Domain Name can be logical and easily remembered. Secondly, should an IP address for a host change, the domain name can still resolve transparently to the users or application. DNS name resolution is a critical Internet service. Many network services require functional name service for correct operation.

Domain names are separated by dots with the topmost element on the right. Each element may be up to 63 characters long; the entire name may be at most 255 characters long. Letters, numbers or dashes may be used in an element.

Domain Name Space:

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127.

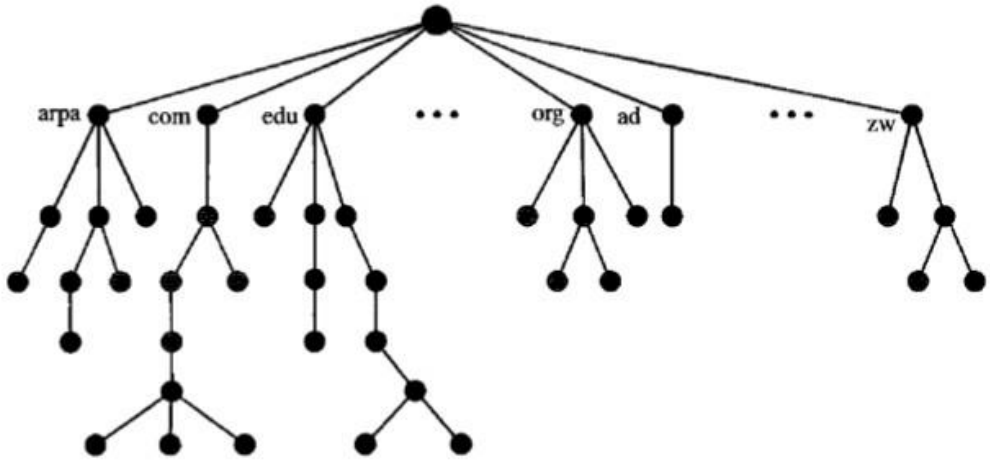


Fig: Domain Name Space

Domain Name

Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root. The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing. Figure shows some domain names.

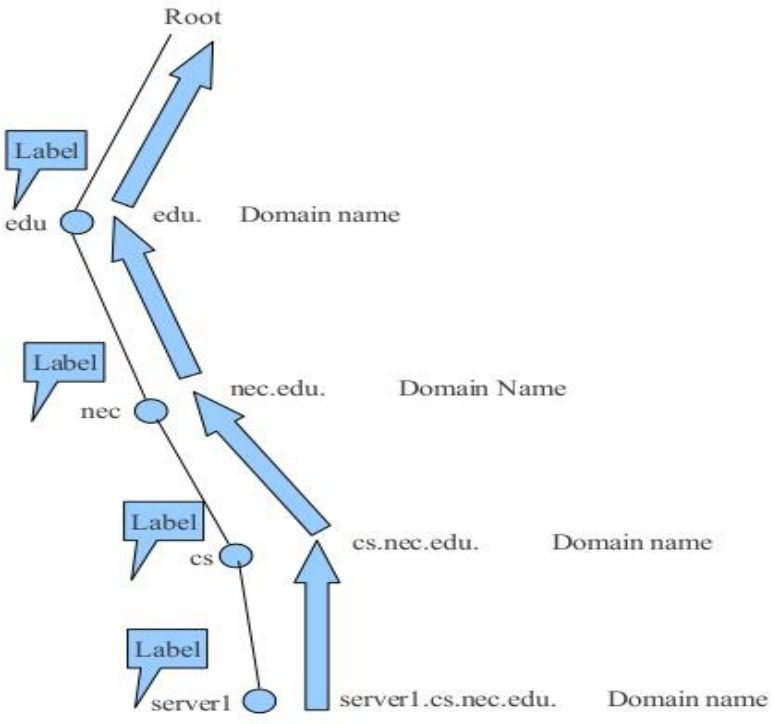


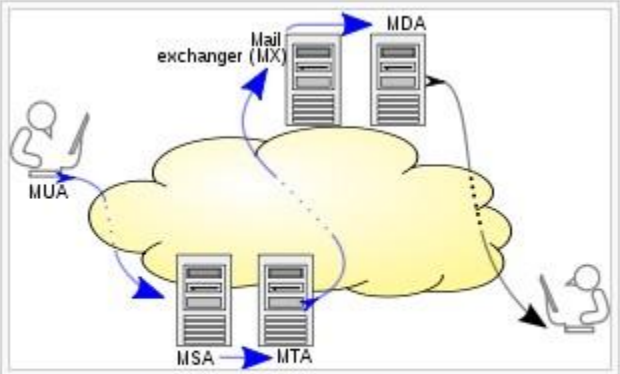
Fig: Domain Name and Labels

Simple Mail Transfer Protocol (SMTP)

One of the most popular network services, email is supported by TCP/IP protocol SMTP. It provides system for sending message to other computers and provide a mail exchange between users. SMTP supports:

- Sending a message to one or more recipients.
- Sending message that includes texts, voice, video or graphics.
- Sending message to users on the network outside the Internet.

SMTP supports sending of email only It cannot pull messages from a remote server on demand. Other protocols, such as the Post Office Protocol (POP) and the Internet Message Access Protocol (IMAP) are specifically designed for retrieving messages and managing mail boxes. However, SMTP has a feature to initiate mail queue processing on a remote server so that the requesting system may receive any messages destined for it (cf. Remote Message Queue Starting). POP and IMAP are preferred protocols when a user's personal computer is only intermittently powered up, or Internet connectivity is only transient and hosts cannot receive message during off-line periods.



Blue arrows can be implemented using SMTP variations.

The overall flow for message creation, mail transport, and delivery may be illustrated as shown.

Email is submitted by a mail client (MUA, mail user agent) to a mail server (MSA, mail submission agent) using SMTP on TCP port 587. Most mailbox providers still allow submission on traditional port 25. From there, the MSA delivers the mail to its mail transfer agent (MTA, mail transfer agent). Often, these two agents are just different instances of the same software launched with different options on the same machine. Local processing can be done either on a single machine, or split among various appliances; in the former case, involved processes can share files; in the latter case, SMTP is used to transfer the message internally, with each host configured to use the next appliance as a smart host. Each process is an MTA in its own right; that is, an SMTP server.

The boundary MTA has to locate the target host. It uses the Domain name system (DNS) to look up the mail exchanger record (MX record) for the recipient's domain (the part of the address on the right of @). The returned MX record contains the name of the target host. The MTA next looks up the A record for that name in order to get the IP address and connect to such host as an SMTP client.

Once the MX target accepts the incoming message, it hands it to a mail delivery agent (MDA) for local mail delivery. An MDA is able to save messages in the relevant mailbox

format. Again, mail reception can be done using many computers or just one —the picture displays two nearby boxes in either case. An MDA may deliver messages directly to storage, or forward them over a network using SMTP, or any other means, including the Local Mail Transfer Protocol (LMTP), a derivative of SMTP designed for this purpose.

Once delivered to the local mail server, the mail is stored for batch retrieval by authenticated mail clients (MUAs). Mail is retrieved by end-user applications, called email clients, using Internet Message Access Protocol (IMAP), a protocol that both facilitates access to mail and manages stored mail, or the Post Office Protocol (POP) which typically uses the traditional m box mail file format or a proprietary system such as Microsoft Exchange/Outlook or Lotus Notes/Domino. Webmail clients may use either method, but the retrieval protocol is often not a formal standard.

IMAP (Internet Mail Access Protocol)
An Internet standard protocol for storing and retrieving messages from Simple Mail Transfer Protocol (SMTP) hosts. Internet Mail Access Protocol version provides functions similar to Post Office Protocol version 3 (POP3), with additional features as described in this entry.

How It Works?
SMTP provides the underlying message transport mechanism for sending e-mail over the Internet, but it does not provide any facility for storing and retrieving messages. SMTP hosts must be continuously connected to one another, but most users do not have a dedicated connection to the Internet.
IMAP4 provides mechanisms for storing messages received by SMTP in a receptacle called a mailbox. An IMAP4 server stores messages received by each user until the user connects to download and read them using an IMAP4 client such as Evolution or Microsoft Outlook Express.

IMAP4 includes a number of features that are not supported by POP3. Specifically, IMAP4 allows users to

- Access multiple folders, including public folders
- Create hierarchies of folders for storing messages
- Leave messages on the server after reading them so that they can access the messages again from another location
- Search a mailbox for a specific message to download
- Flag messages as read
- Selectively download portions of messages or attachments only
- Review the headers of messages before downloading them

To retrieve a message from an IMAP4 server, an IMAP4 client first establishes a Transmission Control Protocol (TCP) session using TCP port 143. The client then identifies itself to the server and issues a series of IMAP4 commands:

- LIST:
  Retrieves a list of folders in the client's mailbox
- SELECT:
  Selects a particular folder to access its messages
- FETCH:
  Retrieves individual messages

- LOGOUT:
  Ends the IMAP4 session

Post Office Protocol version 3 (POP3)

An Internet standard protocol for storing and retrieving messages from Simple Mail Transfer Protocol (SMTP) hosts.

How It Works?
SMTP provides the underlying transport mechanism for sending e-mail messages over the Internet, but it does not provide any facility for storing messages and retrieving them. SMTP hosts must be continuously connected to one another, but most users do not have a dedicated connection to the Internet.

Post Office Protocol version 3 (POP3) provides mechanisms for storing messages sent to each user and received by SMTP in a receptacle called a mailbox. A POP3 server stores messages for each user until the user connects to download and read them using a POP3client such as Microsoft Outlook 98, Microsoft Outlook Express, or Microsoft Mail and News.

To retrieve a message from a POP3 server, a POP3 client establishes a Transmission Control Protocol (TCP) session using TCP port 110, identifies itself to the server, and then issues a series of POP3 commands:

- stat:
  Asks the server for the number of messages waiting to be retrieved
- list:
  Determines the size of each message to be retrieved
- retr:
  Retrieves individual messages
- Quit:
  Ends the POP3 session

After a POP3 client reads a message in its mailbox on a POP3 server, the message is deleted. Primarily because of this, POP3 is being supplanted by Internet Mail Access Protocol version 4 (IMAP4), which offers better support for mobile users. POP3 is supported by Microsoft Exchange Server.

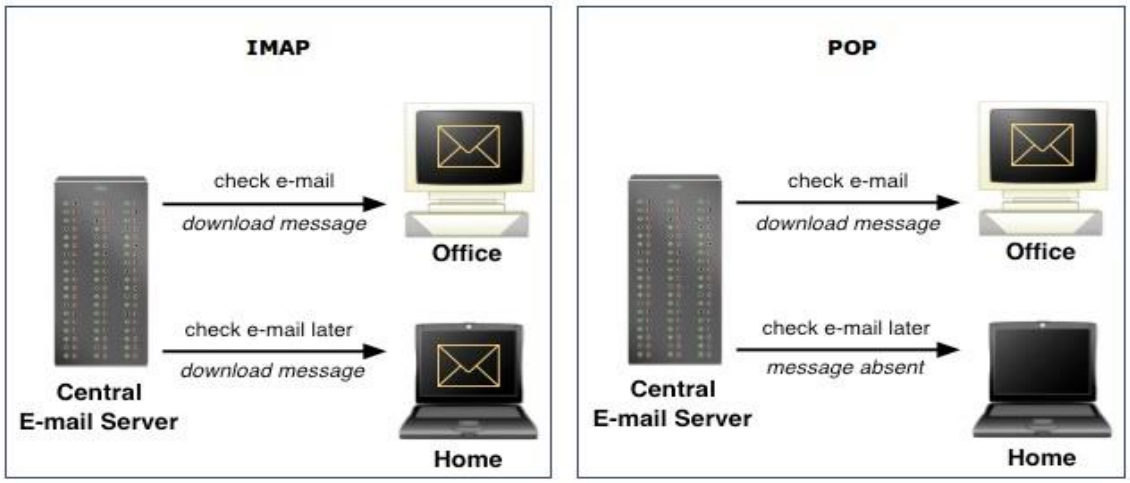
IMAP VS POP:

What's the difference?
The main difference, as far as we are concerned here, is the way in which IMAP or POP controls your e-mail inbox.

When you use IMAP you are accessing your inbox on the mail server. IMAP does not actually move messages onto your computer. You can think of an e-mail program using IMAP as a window to your messages on the server. Although the messages appear on your computer while you work with them, they remain on the central mail server.

POP does the opposite. Instead of just showing you what is in your inbox on the U's mail server, it checks the server for new messages, downloads all the new messages in your inbox onto your computer, and then deletes them from the server. This means that every time you

use POP to view your new messages; they are no longer on the central mail server. Figure illustrates these concepts



IMAP makes it easier to view mail from home, work, and other locations

Because IMAP leaves all of your messages on the central mail server, you can view these messages from any location with Internet access. This means the U of M e-mail inbox you view from home will be the same one you see at work.

Since POP downloads new messages to your computer and removes them from the server, you will not be able to see those new messages on another computer when you check your inbox. Those messages exist only on the computer that downloaded them using POP.

However, if you use IMAP and create e-mail folders on the server, these folders are accessible from anywhere you read your e-mail using IMAP. If you use POP and create e-mail folders, they are stored locally, and you cannot access these folders from anywhere except the computer on which you created them.

POP can create problems if you alternate between it and IMAP. There is an option in many POP e-mail programs to leave copies of the messages on the server, but this option has complications. When you leave copies of the messages on the server, then access your e-mail using Webmail or another IMAP e-mail client, the POP client
may create duplicate messages next time it accesses the inbox; you will see each of the messages more than once, and you will have to clean out (delete) the unwanted ones.

Virtual Private Network (VPN)

The Internet is a worldwide, publicly accessible IP network. Due to its vast global proliferation, it has become a viable method of interconnecting remote sites. However, the fact that it is a public infrastructure has deterred most enterprises from adopting it as a viable remote access method for branch and SOHO sites.
A virtual private network (VPN) is a concept that describes how to create a private network over a public network infrastructure while maintaining confidentiality and security. VPNs use cryptographic tunneling protocols to provide sender authentication, message integrity, and confidentiality by protecting against packet sniffing. VPNs can be implemented at Layers 2, 3, and 4 of the Open Systems Interconnection (OSI) model. Figure illustrates a typical VPN topology. Components required to establish a VPN include:

- An existing network with servers and workstations
- Connection to the Internet
- VPN gateways (i.e., routers, PIX, ASA, VPN concentrators) that act as endpoints to establish, manage, and control VPN connections
- Software to create and manage tunnels

The key to VPN technology is security. VPNs secure data by encapsulating the data, encrypting the data, or both encapsulating the data and then encrypting it:

- Encapsulation is also referred to as tunneling because encapsulation transmits data transparently from network to network through a shared network infrastructure.
- Encryption codes data into a different format. Decryption decodes encrypted data into the data's original unencrypted format.

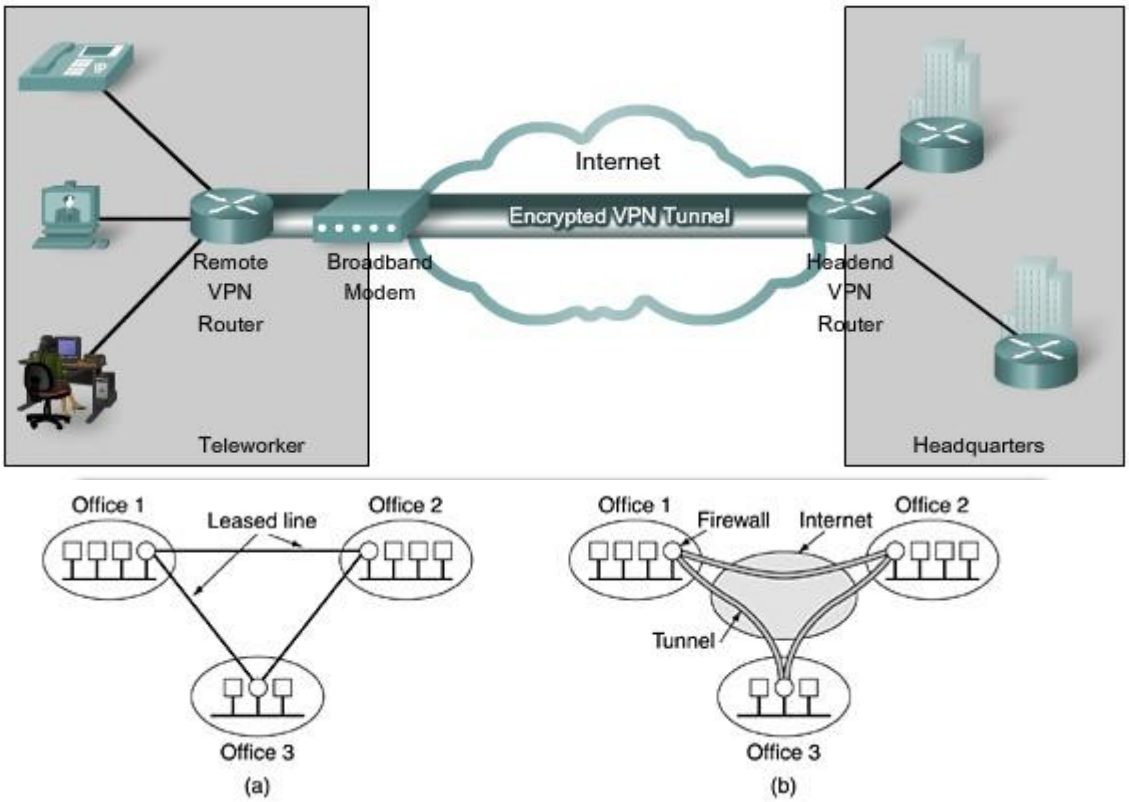


Fig:(a) A leased-line private network. (b) A virtual private network.

Basically, a VPN is a private network that uses a public network (usually the Internet) to connect remote sites or users together. Instead of using a dedicated, real-world connection such as leased line, a VPN uses "virtual" connections routed through the Internet from the company's private network to the remote site or employee.

A well-designed VPN can greatly benefit a company. For example, it can:

- Extend geographic connectivity
- improve security
- Reduce operational costs versus traditional WAN
- Reduce transit time and transportation costs for remote users
- Improve productivity
- Simplify network topology
- Provide global networking opportunities
- Provide telecommuter support
- Provide broadband networking compatibility

- Provide faster ROI (return on investment) than traditional WAN

What features are needed in a well-designed VPN? It should incorporate:

- Security
- Reliability
- Scalability
- Network management
- Policy management

IPSEC

IPsec provides a mechanism for secure data transmission over IP networks, ensuring confidentiality, integrity, and authenticity of data communications over unprotected networks such as the Internet . IPsec encompasses a suite of protocols and is not bound to any specific encryption or authentication algorithms, key generation technique, or security association (SA). IPsec provides the rules while existing algorithms provide the encryption, authentication, key management, and so on. IPsec acts at the network layer, protecting and authenticating IP packets between IPsec devices (peers), such as Cisco PIX Firewalls, Adaptive Security Appliances (ASA), Cisco routers, the Cisco Secure VPN Client, and other IPsec-compliant products.

IPsec provides the following essential security functions:

- **Data confidentiality:** IPsec ensures confidentiality by using encryption. Data encryption prevents third parties from reading the data, especially data that is transmitted over public networks or wireless networks. The IPsec sender can encrypt packets before transmitting the packets across a network and prevent anyone from hearing or viewing the communication (eavesdropping).
- **Data integrity:** IPsec ensures that data arrives unchanged at the destination; that is, that the data is not manipulated at any point along the communication path. IPsec ensures data integrity by using hashes.
- **Data origin authentication:** The IPsec receiver can authenticate the source of the IPsec packets. Authentication ensures that the connection is actually made with the desired communication partner.
- **Anti-replay:** Anti-replay protection verifies that each packet is unique, not duplicated. IPsec packets are protected by comparing the sequence number of the received packets and a sliding window on the destination host, or security gateway. A packet whose sequence number is before the sliding window is considered late, or a duplicate. Late and duplicate packets are dropped.

Proxy server

A computer that can act on the behalf of other computers to request content from the Internet or an intranet. Proxy Server is placed between a user's machine and the Internet. It can act as a firewall to provide protection and as a cache area to speed up Web page display. A firewall mechanism that replaces the IP address of a host on the internal (protected) network with its own IP address for all traffic passing through it. A software agent that acts on behalf of a user, typical proxies accept a connection from a user, make a decision as to whether or not the user or client IP address is permitted to use the proxy, perhaps does additional authentication, and then completes a connection on behalf of the user to a remote destination.

Proxy servers have two main purposes:

- **Improve Performance:** Proxy servers can dramatically improve performance for groups of users. This is because it saves the results of all requests for a certain amount of time. proxy server is often on the same network as the user, this is a much faster operation. Real proxy servers support hundreds or thousands of users.
- **Filter Requests:** Proxy servers can also be used to filter requests.

Types of Proxy:

1. **Forward Proxy:**
   Forward proxies are proxies where the client server names the target server to connect to. Forward proxies are able to retrieve from a wide range of sources (in most cases anywhere on the Internet).The terms "forward proxy" and "forwarding proxy" are a general description of behavior (forwarding traffic) and thus ambiguous. Except for Reverse proxy

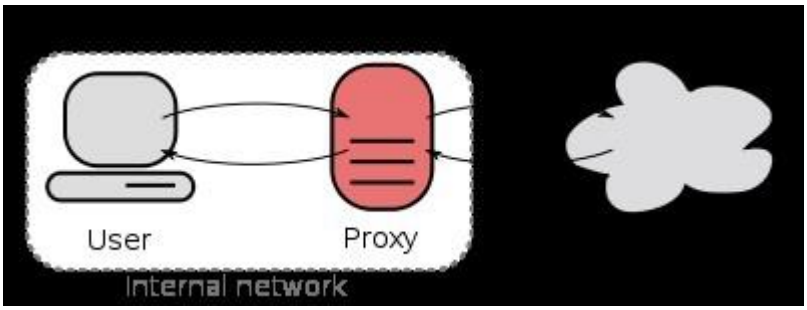


Fig:A forward proxy taking requests from an internal network and forwarding them to the Internet

2. **Open Proxy:**
   An open proxy is a forward proxy server that is accessible by any Internet user. Gordon Lyon estimates there are "hundreds of thousands" of open proxies on the Internet.[4] An anonymous open proxy allows users to conceal their IP address while browsing the Web or using other Internet services.

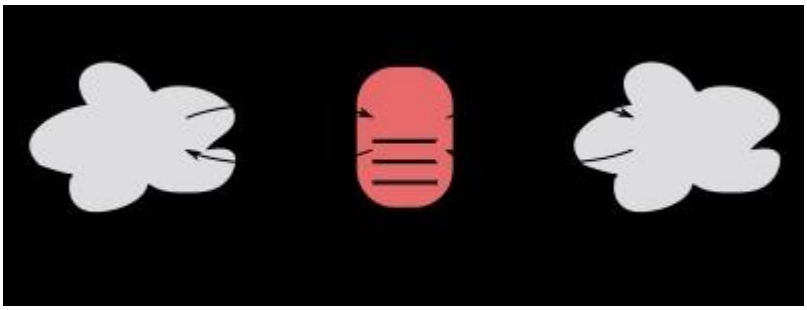
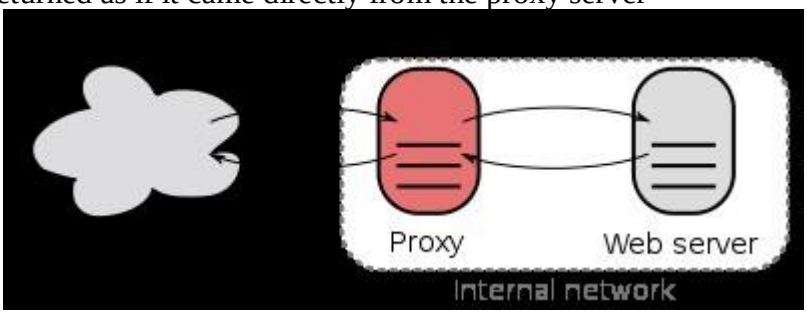


Fig: An open proxy forwarding requests from and to anywhere on the Internet.

3. **Reverse Proxy:**
   A reverse proxy is a proxy server that appears to clients to be an ordinary server. Requests are forwarded to one or more origin servers which handle the request. The

response is returned as if it came directly from the proxy server



ù

Fig: A reverse proxy taking requests from the Internet and forwarding them to servers in an internal network. Those making requests connect to the proxy and may not be aware of the internal network.

File Transfer Protocol (FTP)

An Internet standard application-level TCP/IP protocol that can be used for transferring files between hosts on a TCP/IP internetwork.

How It Works?

File Transfer Protocol (FTP) is one of the earliest Internet protocols, and is still used for uploading and downloading files between clients and servers. An FTP client is an application that can issue FTP commands to an FTP server, while an FTP server is a service or daemon running on a server that responds to FTP commands from a client. FTP commands can be used to change directories, change transfer modes between binary and ASCII, upload files, and download files. FTP uses Transmission Control Protocol (TCP) for reliable network communication by establishing a session before initiating data transfer. TCP port number 21 on the FTP server listens for connection attempts from an FTP client and is used as a control port for establishing a connection between the client and server, for allowing the client to send an FTP command to the server, and for returning the server's response to the command. Once a control connection has been established, the server opens port number 20 to form a new connection with the client for transferring the actual data during uploads and downloads.

While transferring Data over the network, two modes can be used:

1. Ascii Mode
2. Binary Mode

The two types differ from the way they send the data. When a file is sent using an ASCII-type transfer, the individual letters, numbers and characters are sent. The receiving machine saves these in a text file in the appropriate format (for example, a Unix machine saves it in a Unix format, a Macintosh saves it in a Mac format). Hence if an ASCII transfer is used it can be assumed plain text is sent, which is stored by the receiving computer in its own format.

Sending a file in binary mode is different. The sending machine sends each file bit for bit and as such the recipient stores the bit-stream as it receives it.

By default, most FTP clients use ASCII mode. Some clients, nevertheless are more clever and try to determine the required transfer-mode by inspecting the file's contents.