Internet Technology (CSC-402)

Siddhanath Science Campus
Mahandranagar,Kanchanpur
Department of Computer Science & Information Technology
Tribhuvan University

Unit – I (Introduction)

Internet:

The Internet is a global system of interconnected computer networks that use the standard Internet protocol suite (often called TCP/IP, although not all applications use TCP) to serve billions of users worldwide. It is a network of networks that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic, wireless and optical networking technologies. The Internet carries an extensive range of information resources and services, such as the inter-linked hypertext documents of the World Wide Web (WWW) and the infrastructure to support email.

Internet is a short form of the technical term **internetwork**, the result of interconnecting computer networks with special gateways or routers. The Internet is also often referred to as the Net.

The terms Internet and World Wide Web are often used in everyday speech without much distinction. However, the Internet and the World Wide Web are not one and the same. The Internet establishes a global data communications system between computers. In contrast, the Web is one of the services communicated via the Internet. It is a collection of interconnected documents and other resources, linked by hyperlinks and URLs

Intranet:

An intranet is a computer network that uses Internet Protocol technology to share information, operational systems, or computing services within an organization. The term is used in contrast to internet, a network between organizations, and instead refers to a network within an organization. Sometimes, the term refers only to the organization's internal website, but may be a more extensive part of the organization's information technology infrastructure, and may be composed of multiple local area networks. The objective is to organise each individual's desktop with minimal cost, time and effort to be more productive, cost efficient, timely and competitive.

An intranet may host multiple private websites and constitute an important component and focal point of internal communication and collaboration. Any of the well known Internet protocols may be found in an intranet, such as HTTP (web services), SMTP (e-mail), and FTP (file transfer protocol). Internet technologies are often deployed to provide modern interfaces to legacy information systems hosting corporate data.

An intranet can be understood as a private analog of the Internet, or as a private extension of the Internet confined to an organization. The first intranet websites and home pages began to appear in organizations in 1996-1997. Although not officially noted, the term intranet first became common-place among early adopters, such as universities and technology corporations, in 1992.

Intranets are sometimes contrasted to extranets. While intranets are generally restricted to employees of the organization, extranets may also be accessed by customers, suppliers, or other approved parties. Extranets extend a private network onto the Internet with special provisions for authentication, authorization and accounting

VPN:

Internet Number:
An **Internet number** is a numerical identifier assigned to an Internet resource or used in the networking protocols of the Internet Protocol Suite. Examples include IP Addresses and Autonomous System (AS) numbers. Globally, Internet numbers are managed by the IANA.

An **Internet Protocol address** (**IP address**) is a numerical label assigned to each device (e.g., computer, printer) participating in a computer network that uses the Internet Protocol for communication.

Within the Internet, an **Autonomous System** (**AS**) is a collection of connected Internet Protocol (IP) routing prefixes under the control of one or more network operators that presents a common, clearly defined routing policy to the Internet. A group of networks and routers controlled by a single administrative authority is called an *autonomous system*(AS)

A unique ASN is allocated to each AS. ASN are important because the ASN uniquely identifies each network on the Internet.

IANA:
The **Internet Assigned Numbers Authority** (**IANA**) is the entity that oversees global IP address allocation, autonomous system number allocation, root zone management in the Domain Name System (DNS), media types, and other Internet Protocol-related symbols and numbers. IANA is a department operated by the Internet Corporation for Assigned Names and Numbers, also known as ICANN (Internet Corporation for Assigned Names and Numbers).

Prior to the establishment of ICANN for this purpose, IANA was administered primarily by Jon Postel at the Information Sciences Institute (ISI) of the University of Southern California (USC), under a contract USC/ISI had with the United States Department of Defense, until ICANN was created to assume the responsibility under a United States Department of Commerce contract.

IANA is broadly responsible for the allocation of globally unique names and numbers that are used in Internet protocols that are published as RFC documents. **Request for Comments** (**RFC**) **is a memorandum published by the Internet Engineering Task Force (IETF) describing methods, behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems.** These documents describe

methods, behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems. IANA also maintains a close liaison with the Internet Engineering Task Force (IETF) and RFC Editorial team in fulfilling this function.

In the case of the two major Internet namespaces, namely IP addresses and domain names, extra administrative policy and delegation to subordinate administrations is required because of the multi-layered distributed use of these resources.

IANA delegates allocations of IP address blocks to Regional Internet Registries (RIRs). Each RIR allocates addresses for a different area of the world. Collectively the RIRs have created the Number Resource Organization formed as a body to represent their collective interests and ensure that policy statements are coordinated globally.

The RIRs divide their allocated address pools into smaller blocks and delegate them in their respective operating regions to Internet service providers and other organizations. Since the introduction of the CIDR (**Classless Inter-Domain Routing**) system, IANA typically allocates address space in the size of /8 prefix blocks for IPv4 and /12 prefix blocks from the 2000::/3 IPv6 Block to requesting regional registries as needed.

Following are some of the tasks associated with IANA;

- IANA administers the data in the root name servers, which form the top of the hierarchical DNS tree. This task involves liaising with top-level domain operators, the root nameserver operators, and ICANN's policy making apparatus.
- IANA administers many parameters of IETF protocols. Examples include the names of Uniform Resource Identifier (URI) schemes and character encodings recommended for use on the Internet.

Internet Registry (IR):

An Internet Registry (IR) is an organization that is responsible for distributing IP address space to its members or customers and for registering those distributions.

RIR:

A **regional Internet registry** (**RIR**) is an organization that manages the allocation and registration of Internet number resources within a particular region of the world. Internet number resources include IP addresses and autonomous system (AS) numbers.

Regional Internet Registries (RIRs) are established and authorized by respective regional communities, and recognized by the IANA to serve and represent large geographical regions. The primary role of RIRs is to manage and distribute public Internet address space within their respective regions.

The Regional Internet Registry system evolved over time, eventually dividing the world into five RIRs:

- African Network Information Centre (AfriNIC) for Africa
- American Registry for Internet Numbers (ARIN) for the United States, Canada, several parts of the Caribbean region, and Antarctica
- Asia-Pacific Network Information Centre (APNIC) for Asia, Australia, New Zealand, and neighboring countries
- Latin America and Caribbean Network Information Centre (LACNIC) for Latin America and parts of the Caribbean region
- Réseaux IP Européens Network Coordination Centre (RIPE NCC) for Europe, Russia, the Middle East, and Central Asia

The IANA delegates Internet resources to the RIRs who, in turn, follow their regional policies to delegate resources to their customers, which include Internet service providers and end-user organizations.

NIR:

A **National Internet Registry** (or NIR) is an organization under the umbrella of a Regional Internet Registry with the task of coordinating IP address allocations and other Internet resource management functions at a national level within a country or economic unit.

NIRs operate primarily in the Asia Pacific region, under the authority of APNIC, the Regional Internet Registry for that region.

The following NIRs are currently operating in the APNIC region:

- APJII (Asosiasi Penyelenggara Jasa Internet Indonesia), Indonesian ISP Association
- CNNIC, China Internet Network Information Center
- JPNIC, Japan Network Information Center
- KRNIC, National Internet Development Agency of Korea
- SGNIC, Singapore Network Information Centre
- TWNIC, Taiwan Network Information Center
- VrNNIC, Vietnam Internet Network Information Center

The following NIRs are currently operating in the Latin-American (LACNIC) region:

- NIC Argentina
- NIC Bolivia
- NIC Chile
- NIC Mexico
- NIC Brazil

LIR:

A **local Internet registry** (**LIR**) is an organization that has been allocated a block of IP addresses by a regional Internet registry (RIR), and that assigns most parts of this block to its own customers. It primarily assigns address space to the users of the network services that it provides. LIRs are generally Internet Service Providers (ISPs), whose customers are primarily end users and possibly other ISPs. Membership in an RIR is required to become an LIR.

ISPs for internet number management:

An **Internet service provider** (**ISP**) is an organization that provides access to the Internet. Internet service providers can be either community-owned and non-profit, or privately owned and for-profit.

In 1990, Brookline, Massachusetts-based The World became the first commercial ISP.

ISPs can be of ;

- **Access providers:** SPs employ a range of technologies to enable consumers to connect to their network
- **Hosting ISPs:** Hosting ISPs routinely provide email, FTP, and web-hosting services. Other services include virtual machines, clouds, or entire physical servers where customers can run their own custom software
- **Transit ISPs:** Just as their customers pay them for Internet access, ISPs themselves pay upstream ISPs for Internet access. An upstream ISP usually has a larger network than the contracting ISP and/or is able to provide the contracting ISP with access to parts of the Internet the contracting ISP by itself has no access to.
- **Virtual ISPs:** A Virtual ISP (VISP) is an operation which purchases services from another ISP (sometimes called a "wholesale ISP") which allow the VISP's customers to access the Internet using services and infrastructure owned and operated by the wholesale ISP.
- **Free ISPs:** Free ISPs are Internet Service Providers (ISPs) which provide service free of charge. Many free ISPs display advertisements while the user is connected; like commercial television, in a sense they are selling the users' attention to the advertiser. Other free ISPs, often called freenets, are run on a nonprofit basis, usually with volunteer staff

Internet Domain:

A **domain name** is an identification string that defines a realm of administrative autonomy, authority, or control on the Internet. Domain names are formed by the rules and procedures of the Domain Name System (DNS).

Domain names are used in various networking contexts and application-specific naming and addressing purposes**. In general, a domain name represents an Internet Protocol (IP) resource, such as a personal computer used to access the Internet, a server computer hosting a web site, or the web site itself or any other service communicated via the Internet.** Domain names serve as humanly-memorable names for Internet participants, like computers, networks, and services. A domain name represents an Internet Protocol (IP) resource. Individual Internet host computers use domain names as host identifiers, or hostnames. **Hostnames are the leaf labels in the domain name system usually without further subordinate domain name space.** Hostnames appear as a component in Uniform Resource Locators (URLs) for Internet resources such as web sites

Domain names are organized in subordinate levels (subdomains) of the DNS root domain, which is nameless. The first-level set of domain names are the **top-level domains (TLDs)**, including the **generic top-level domains (gTLDs)**, such as the prominent domains .com, .net and .org, and the **country code top-level domains (ccTLDs)**. Below these top-level domains in the DNS hierarchy are the second-level and third-level domain names that are typically open for reservation by end-users who wish to connect local area networks to the Internet, create other publicly accessible Internet resources or run web sites. The registration of these domain names is usually administered by domain name registrars who sell their services to the public.

A **top-level domain** (TLD) is one of the domains at the highest level in the hierarchical Domain Name System of the Internet. The top-level domain names are installed in the root zone of the name space. **For all domains in lower levels, it is the last part of the domain name, that is, the last label of a fully qualified domain name. For example, in the domain name www.example.com, the top-level domain is .com (or .COM, as domain names are not case-sensitive).** Management of most top-level domains is done by ICANN.

A **generic top-level domain** (**gTLD**) is one of the categories of top-level domains (TLDs) maintained by the IANA for use in the Domain Name System of the Internet. The core group of generic top-level domains consists of the **com, info, net, and org domains**. In addition, the domains **biz, name, and pro** are also considered *generic*; however, these are designated as *restricted*, because registrations within them require proof of eligibility within the guidelines set for each.

Structure of domain name:

A domain name consists of one or more parts, technically called *labels* that are conventionally concatenated, and delimited by dots, such as *example.com*. The right-most label conveys the top-level domain; for example, the domain name *www.example.com*belongs to the top-level domain *com*.

The hierarchy of domains descends from the right to the left label in the name; each label to the left specifies a subdivision, or subdomain of the domain to the right. For example: the label *example* specifies a node *example.com* as a subdomain of the *com* domain, and *www* is a label to create *www.example.com*, a subdomain of *example.com*. **This tree of labels may consist of 127 levels. Each label may contain from 1 to 63 octets.** The empty label is reserved for the root node. The full domain name may not exceed a total length of 255 characters. In practice, some domain registries may have shorter limits.

A hostname is a domain name that has at least one associated IP address. For example, the domain names *www.example.com* and *example.com* are also hostnames, whereas the *com* domain is not. However, other top-level domains, particularly country code top-level domains, may indeed have an IP address, and if so, they are also hostnames.

Subdomain:

In the Domain Name System (DNS) hierarchy, a **subdomain** is a domain that is part of a larger domain. **The only domain that is not also a subdomain is the root domain. For example, *mail.jagdish.com* and *support.jagdish.com* are subdomains of the *jagdish.com* domain, which in turn is a subdomain of the com top-level domain (TLD).** A "subdomain" expresses relative dependence, not absolute dependence: for example, *jagdish.com* comprises a subdomain of the *.com* domain, and mail.jagdish.com comprises a subdomain of the domain *jagdish.com* .

*The practice of registering a domain only to turn around and sell it off to an interested party at a much higher price even has a name. It is called **cybersquatting**. Many companies that were slow off the mark when the Internet era began found their obvious domain names already taken when they tried to acquire them. In general, as long as no trademarks are being violated and no fraud is involved, it is first-come, first-served with names. Nevertheless, policies to resolve naming disputes are still being refined.*

Reserved Domain Names:

The domain names may be used for various specific purposes however, with the intention that these should not occur in production networks within the global domain name system:

- example: reserved for use in examples
- invalid: reserved for use in obviously invalid domain names
- localhost: reserved to avoid conflict with the traditional use of localhost as a hostname
- test: reserved for use in tests

Domain Name System:

The essence of DNS is the invention of a hierarchical, domain-based naming scheme and a distributed database system for implementing this naming scheme. **It is primarily used for mapping host names to IP addresses but can also be used for other purposes. A Domain Name Service resolves queries for these names into IP addresses for the purpose of locating computer services and devices worldwide.**

Internet name servers and a communication protocol implement the Domain Name System. **A DNS name server is a server that stores the DNS records for a domain name, such as address (A) records, name server (NS) records, and mail exchanger (MX) records; a DNS name server responds with answers to queries against its database**

The way DNS is used is as follows;

To map a name onto an IP address, an application program calls a library procedure called the **resolver**, passing it the name as a parameter. The resolver sends a query containing the name to a local DNS server, which looks up the name and returns a response containing the IP address to the resolver, which then returns it to the caller. The query and response messages are sent as UDP packets. Armed with the IP address, the program can then establish a TCP connection with the host or send it UDP packets.

Client Lookup:

Users generally do not communicate directly with a DNS resolver. Instead DNS resolution takes place transparently in applications such as web browsers, e-mail clients, and other Internet applications. When an application makes a request that requires a domain name lookup, such programs send a resolution request to the DNS resolver in the local operating system, which in turn handles the communications required.

The DNS resolver will almost invariably have a cache containing recent lookups. If the cache can provide the answer to the request, the resolver will return the value in the cache to the program that made the request. If the cache does not contain the answer, the resolver will send the request to one or more designated DNS servers. In the case of most home users, the Internet service provider to which the machine connects will usually supply this DNS server: such a user will either have configured that server's address manually or allowed DHCP to set it; however, where systems administrators have configured systems to use their own DNS servers, their DNS resolvers point to separately maintained nameservers of the organization. In any event, the name server thus queried will follow the process outlined above, until it either successfully finds a result or does not. It then returns its results to the DNS resolver; assuming it has found a result, the resolver duly caches that result for future use, and hands the result back to the software which initiated the request.

Reverse Lookup:

Computer networks use the Domain Name System to determine the IP address associated with a domain name. This process is also known as *forward* DNS resolution. *Reverse* DNS lookup is the inverse process, the resolution of an IP address to its designated domain name.

The reverse DNS database of the Internet is rooted in the *Address and Routing Parameter Area* (arpa) top-level domain of the Internet. IPv4 uses the in-addr.arpa domain and the ip6.arpa domain is delegated for IPv6. The process of reverse resolving an IP address uses the *pointer* DNS record type (PTR record).

A reverse lookup is a query of the DNS for domain names when the IP address is known. Multiple domain names may be associated with an IP address. The DNS stores IP addresses in the form of domain names as specially formatted names in pointer (PTR) records within the infrastructure top-level domain arpa. For IPv4, the domain is in-addr.arpa. For IPv6, the reverse lookup domain is ip6.arpa. The IP address is represented as a name in reverse-ordered octet representation for IPv4, and reverse-ordered nibble representation for IPv6.

When performing a reverse lookup, the DNS client converts the address into these formats, and then queries the name for a PTR record following the delegation chain as for any DNS query. For example, assume the IPv4 address 208.80.152.2 is assigned to Wikimedia. It is represented as a DNS name in reverse order like this: 2.152.80.208.in-addr.arpa. When the DNS resolver gets a PTR (reverse-lookup) request, it begins by querying the root servers (which point to ARIN's servers for the 208.in-addr.arpa zone). On ARIN's servers, 152.80.208.in-addr.arpa is assigned to Wikimedia, so the resolver sends another query to the Wikimedia nameserver for 2.152.80.208.in-addr.arpa, which results in an authoritative response.

Domain Name Registrar:

A **domain name registrar** is an organization or commercial entity that manages the reservation of Internet domain names. A domain name registrar must be accredited by ageneric top-level domain(gTLD) registry and/or a country code top-level domain(ccTLD) registry. The management is done in accordance with the guidelines of the designated domain name registries and to offer such services to the public.

Domain registration information is maintained by the domain name registries, which contract with domain registrars to provide registration services to the public. An end user selects a registrar to provide the registration service, and that registrar becomes the *designated registrar* for the domain chosen by the user.

Only the designated registrar may modify or delete information about domain names in a central registry database. It is not unusual for an end user to switch registrars, invoking a domain transfer process between the registrars involved, that is governed by specific domain name transfer policies.

Registration of a domain name establishes a set of Start of Authority (SOA) records in the DNS servers of the parent domain, indicating the IP address (or domain name) of DNS servers that are *authoritative* for the domain. This provides merely a reference for how to find the domain data – not the actual domain data.

Internet Backbone Networks:

Optical backbone:

Fiber-optic communication is a method of transmitting information from one place to another by sending pulses of light through an optical fiber. The light forms an electromagnetic carrier wave that is modulated to carry information.

The process of communicating using fiber-optics involves the following basic steps:

- creating the optical signal involving the use of a transmitter,
- relaying the signal along the fiber, ensuring that the signal does not become too distorted or weak,
- receiving the optical signal, and converting it into an electrical signal.

Modern fiber-optic communication systems generally include an **optical transmitter to convert an electrical signal into an optical signal to send into the optical fiber,** a cable containing bundles of multiple optical fibers that is routed through underground conduits and buildings, multiple kinds of amplifiers, and **an optical receiver to recover the signal as an electrical signal.** The information transmitted is typically digital information generated by computers, telephone systems, and cable television companies.

Optical fiber is used by many telecommunications companies to transmit telephone signals, Internet communication, and cable television signals. **Due to much lower attenuation and interference, optical fiber has large advantages over existing copper wire in long-distance and high-demand applications.**

Optical Fiber:

An optical fiber (or optical fibre) is a flexible, transparent fiber made of glass (silica) or plastic, slightly thicker than a human hair. It functions as a waveguide, or "light pipe". Optical fibers are widely used in fiber-optic communications, which permits transmission over longer distances and at higher bandwidths (data rates) than other forms of communication. Fibers are used instead of metal wires because signals travel along them with less loss and are also immune to electromagnetic interference. Fibers are also used for illumination, and are wrapped in bundles so that they may be used to carry images, thus allowing viewing in confined spaces. Specially-designed fibers are used for a variety of other applications, including sensors and fiber lasers.

Optical fibers typically include a transparent core surrounded by a transparent cladding material with a lower index of refraction. Light is kept in the core by total internal reflection. This causes the fiber to act as a waveguide. **Fibers that support many propagation paths or transverse modes are called multi-mode fibers (MMF), while those that only support a single mode are called single-mode fibers (SMF).** Multi-mode fibers generally have a wider core diameter, and are used for short-distance communication links and for applications where high power must be transmitted. Single-mode fibers are used for most communication links longer than 1,050 meters (3,440 ft).

Marine Cables:

The backbone of the nation's—and indeed, of the world's—information infrastructure is now preponderantly composed of fiber optic cables. A critical element of that backbone is the world's ever expanding network of submarine fiber optic cables.

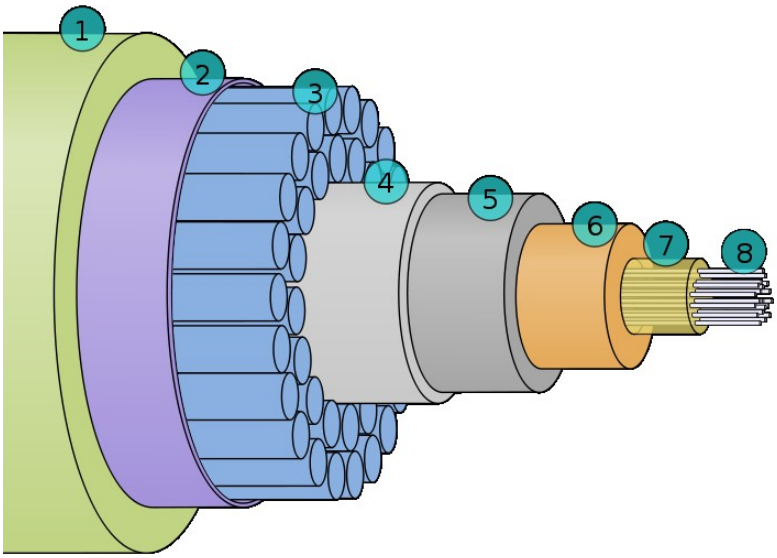
A submarine communications cable is a cable laid on the sea bed between land-based stations to carry telecommunication signals across stretches of ocean.

The first submarine communications cables carried telegraphy traffic. Subsequent generations of cables carried telephony traffic, then data communications traffic. Modern cables use only optical fiber technology to carry digital payloads, which carry telephone, Internet and private data traffic.

Modern cables are typically 69 millimetres (2.7 in) in diameter and weigh around 10 kilograms per metre (7 lb/ft), although thinner and lighter cables are used for deep-water sections.

As of 2006, overseas satellite links accounted for only 1 percent of international traffic, while the remainder was carried by undersea cable. The reliability of submarine cables is high, especially when (as noted above) multiple paths are available in the event of a cable break. **Also, the total carrying capacity of submarine cables is in the terabits per second, while satellites typically offer only megabits per second and display higher latency.** However, a typical multi-terabit, transoceanic submarine cable system costs several hundred million dollars to construct.

As a result of these cables' cost and usefulness, they are highly valued not only by the corporations building and operating them for profit, but also by national governments. For instance, the Australian government considers its submarine cable systems to be "vital to the national economy". Accordingly, the Australian Communications and Media Authority (ACMA) has created protection zones that restrict activities that could potentially damage cables linking Australia to the rest of the world. The ACMA also regulates all projects to install new submarine cables

1. 

Polyethylene
2. "Mylar" tape
3. Stranded metal (steel) wires
4. Aluminum water barrier
5. Polycarbonate
6. Copper or aluminum tube
7. Petroleum jelly
8. Optical fibers

Teleports:

Teleports are the ground-based side of the global satellite network – gateways that provide terrestrial networks with access to orbiting satellite transponders. But they are more than simple gateways. Teleports bridge incompatible systems and protocols, host and distribute content, and act as the hubs of broadband B2B networks. These companies range from small entrepreneurial operations with one to three facilities to large, publicly-traded companies with teleports in multiple geographic markets.
Dig out yourself ……

Satellite Links:

A satellite link is a communications subsystem that involves a link between a transmitting Earth station and a receiving Earth station via a communications satellite.

In 1962, the American telecommunications giant AT&T launched the world's first true communications satellite, called Telstar. Since then, countless communications satellites

have been placed into earth orbit, and the technology being applied to them is forever growing in sophistication.

Satellite communications are comprised of 2 main components:

The Satellite: The satellite itself is also known as the space segment, and is composed of three separate units, namely **the fuel system, the satellite and telemetry controls, and the transponder**. The transponder includes the receiving antenna to pick-up signals from the ground station, a broad band receiver, an input multiplexer, and a frequency converter which is used to reroute the received signals through a high powered amplifier for downlink. The primary role of a satellite is to reflect electronic signals. In the case of a telecom satellite, the primary task is to receive signals from a ground station and send them down to another ground station located a considerable distance away from the first. This relay action can be two-way, as in the case of a long distance phone call. Another use of the satellite is when, as is the case with television broadcasts, the ground station's uplink is then downlinked over a wide region, so that it may be received by many different customers possessing compatible equipment. Still another use for satellites is observation, wherein the satellite is equipped with cameras or various sensors, and it merely downlinks any information it picks up from its vantagepoint.

The Ground Station: This is the earth segment. The ground station's job is two-fold. In the case of an uplink, or transmitting station, terrestrial data in the form of baseband signals, is passed through a baseband processor, an up converter, a high powered amplifier, and through a parabolic dish antenna up to an orbiting satellite. In the case of a downlink, or receiving station, works in the reverse fashion as the uplink, ultimately converting signals received through the parabolic antenna to base band signal.

Satellites are able to provide communications in many instances where other forms of communications technology may not provide a feasible alternative.

Communications satellites provide a number of advantages;

Flexibility: Satellite systems are able to provide communications in a variety of ways without the need to install n/w fixed assets.

Mobility: Satellite communications are able to reach all areas of the globe dependent upon the type of satellite system in use, and the ground stations do not need to be in any one given location. For this reason, many ships use satellite communications.

Speedy deployment: Deployment of a satellite communications system can be very speedy. No ground infrastructure may be required as terrestrial lines, or wireless base stations are not needed. Therefore many remote areas, satellite communications systems provide an ideal solution.

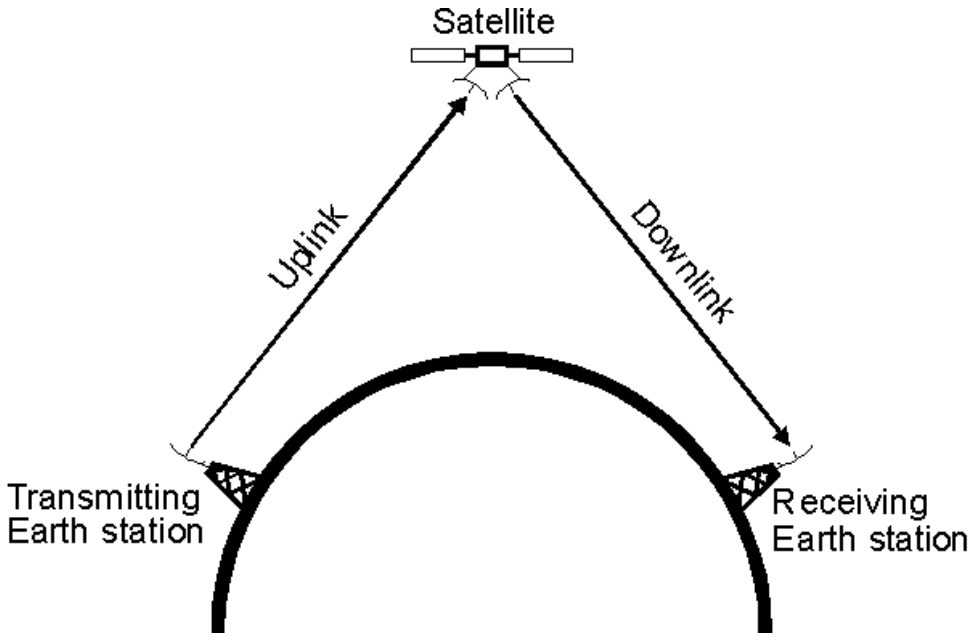
Coverage over the globe: Depending upon the type of satellite communications system and the orbits used, it is possible to provide complete global coverage. As a result, satellite communications systems are sued for providing communications capabilities in many remote areas where other technologies would not be viable..

When considering the use of a satellite some disadvantages also need to be taken into consideration.

Cost: Satellites are not cheap to build, place in orbit and then maintain. This means that the operational costs are high, and therefore the cost of renting or buying space on the satellite will also not be cheap.

Propagation delay: As distances are very much greater than those involved with terrestrial systems, propagation delay can be an issue, especially for satellites using geostationary orbits. Here the round trip from the ground to the satellite and back can be of the order of a quarter of a second.

Specialised satellite terminals required: Even though the operator will operate all the required infrastructure, the user will still need a specialised terminal that will communicate with the satellite. This is likely to be reasonably costly, and it will only be able to be used with one provider.



GEO VS LEO

A Low Earth Orbit or (LEO) is any Earth orbit below an altitude of 2,000 kilometers (1,200 mi). LEO satellites rotate the earth and currently deliver significant voice quality over the Geosynchronous (GEO) satellite systems. Globalstar and Iridium constellations

both use LEO satellites. GEO satellite systems orbit at an altitude of 35,000 kilometers (22,369 miles) above the earth's surface. GEO satellites never change location they move with the earth. GEO satellites are basically stationary over a certain region of the earth. They are best used for Television transmission and high-speed data transmission.

Terrestrial Link:

A communications line that travels on, near or below ground is terrestrial link.

UNIT – II (Internet Protocol Overview)

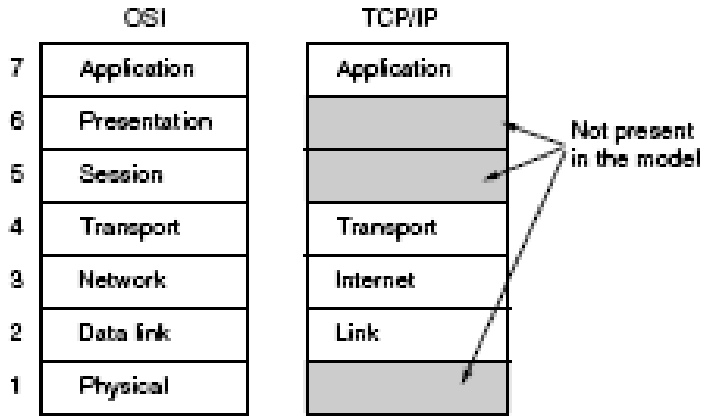
Internet Protocol Suite:

The **Internet protocol suite** is the set of communications protocols used for the Internet and similar networks, and generally the most popular protocol stack for wide area networks. It is commonly known as **TCP/IP**, because of its most important protocols: Transmission Control Protocol (TCP) and Internet Protocol (IP), which were the first networking protocols defined in this standard.

TCP/IP and the IP Layer overview:

The **Internet protocol suite** is the set of communications protocols used for the Internet and similar networks, and generally the most popular protocol stack for wide area networks. It is commonly known as **TCP/IP**, because of its most important protocols: Transmission Control Protocol (TCP) and Internet Protocol (IP), which were the first networking protocols defined in this standard

TCP/IP provides end-to-end connectivity specifying how data should be formatted, addressed, transmitted, routed and received at the destination. It has four abstraction layers, each with its own protocols



From lowest to highest, the layers are:

1. The link layer (commonly Ethernet) contains communication technologies for a local network.
2. The internet layer (IP) connects local networks, thus establishing internetworking.
3. The transport layer (TCP) handles host-to-host communication.
4. The application layer (for example HTTP) contains all protocols for specific data communications services on a process-to-process level (for example how a web browser communicates with a web server).

It loosely defines a four-layer model, with the layers having names, not numbers, as follows:

Application layer (process-to-process): This is the scope within which applications create user data and communicate this data to other processes or applications on another or the same host. The communications partners are often called *peers*. This is where the "higher level" protocols such as SMTP, FTP, SSH, HTTP, etc. operate.

The Application Layer combines a few services that the OSI separates into three layers. **These services are end user-related: authentication, data handling, and compression.** This is where email, Web browsers, telnet clients, and other Internet applications get their connections.

The application layer contains the higher-level protocols used by most applications for network communication. Examples of application layer protocols include the File Transfer Protocol (FTP) and the Simple Mail Transfer Protocol (SMTP). **Data coded according to application layer protocols are then encapsulated into one or (occasionally) more transport layer protocols (such as TCP or UDP), which in turn use lower layer protocols to effect actual data transfer. Since the IP stack defines no layers between the application and transport layers, the application layer must include any protocols that act like the OSI's presentation and session layer protocols.** This is usually done through libraries.

Transport layer (host-to-host): The transport layer constitutes the networking regime between two network hosts, either on the local network or on remote networks separated by routers. The transport layer provides a uniform networking interface that hides the actual topology (layout) of the underlying network connections. **This is where flow-control, error-correction, and connection protocols exist, such as TCP**. This layer deals with opening and maintaining connections between Internet hosts.

The transport layer establishes host-to-host connectivity, meaning it handles the details of data transmission that are independent of the structure of user data and the logistics of exchanging information for any particular specific purpose. **Its responsibility includes end-to-end message transfer independent of the underlying network, along with error control, segmentation, flow control, congestion control, and application addressing (port numbers). End to end message transmission or connecting applications at the transport layer can be categorized as connection-oriented, implemented in TCP, or connectionless, implemented in UDP.**

The layer simply establishes a basic data channel that an application uses in its task-specific data exchange. For this purpose the layer establishes the concept of the port, a numbered logical construct allocated specifically for each of the communication channels an application needs. For many types of services, these port numbers have been standardized so that client computers may address specific services of a server computer without the involvement of service announcements or directory services.

Internet layer (internetworking): The internet layer has the task of exchanging datagrams across network boundaries. **It is therefore also referred to as the layer that establishes internetworking, indeed, it defines and establishes the Internet. This layer defines the addressing and routing structures used for the TCP/IP protocol suite.** The primary protocol in this scope is the Internet Protocol, which defines **IP addresses**. Its function in routing is to transport datagrams to the next IP router that has the connectivity to a network closer to the final data destination.

The internet layer has the responsibility of sending packets across potentially multiple networks. Internetworking requires sending data from the source network to the destination network. This process is called routing.

In the Internet protocol suite, the Internet Protocol performs two basic functions:

- ***Host addressing and identification*: This is accomplished with a hierarchical addressing system (see IP address).**
- ***Packet routing*: This is the basic task of sending packets of data (datagrams) from source to destination by sending them to the next network node (router) closer to the final destination.**

The internet layer only provides an unreliable datagram transmission facility between hosts located on potentially different IP networks by forwarding the transport layer datagrams to an appropriate next-hop router for further relaying to its destination. With this functionality, the internet layer makes possible internetworking, the interworking of different IP networks, and it essentially establishes the Internet. The Internet Protocol is the principal component of the internet layer, and it defines two addressing systems to identify network hosts computers, and to locate them on the network. The schemes are IPV4 and IPV6.

Link layer: This layer defines the networking methods within the scope of the local network link on which hosts communicate without intervening routers. This layer describes the protocols used to describe the local network topology and the interfaces needed to affect transmission of Internet layer datagrams to next-neighbor hosts (the OSI data link layer).

This is the lowest component layer of the Internet protocols, as TCP/IP is designed to be hardware independent. As a result TCP/IP is able to be implemented on top of virtually any hardware networking technology.

The link layer is used to move packets between the Internet layer interfaces of two different hosts on the same link. The processes of transmitting and receiving packets on a given link can be controlled both in the software device driver for the network card, as well as on firmware or specialized chipsets. These will perform data

link functions such as adding a packet header to prepare it for transmission, then **actually transmit the frame over a physical medium.** The TCP/IP model includes specifications of translating the network addressing methods used in the Internet Protocol to data link addressing, such as Media Access Control (MAC), however all other aspects below that level are implicitly assumed to exist in the link layer, but are not explicitly defined.

Transmission Control Protocol (TCP):

Transmission Control Protocol (TCP) is a protocol that provides a reliable stream delivery and connection service to applications. TCP uses sequenced acknowledgment and is able to retransmit packets as needed.

The protocol corresponds to the transport layer of TCP/IP suite. TCP provides a communication service at an intermediate level between an application program and the Internet Protocol (IP). That is, when an application program desires to send a large chunk of data across the Internet using IP, instead of breaking the data into IP-sized pieces and issuing a series of IP requests, the software can issue a single request to TCP and let TCP handle the IP details. TCP is utilized extensively by many of the Internet's most popular applications, including the World Wide Web (WWW), E-mail, File Transfer Protocol, Secure Shell, peer-to-peer file sharing, and some streaming media applications.

Transmission Control Protocol accepts data from a data stream, segments it into chunks, and adds a TCP header creating a TCP segment. The TCP segment is then encapsulated into an Internet Protocol (IP) datagram. A TCP segment is "the packet of information that TCP uses to exchange data with its peers**. A TCP segment consists of a segment *header* and a *data* section. The TCP header contains 10 mandatory fields, and an optional extension field. The data section follows the header. Its contents are the payload data carried for the application**. The length of the data section is not specified in the TCP segment header.

TCP protocol operations may be divided into three phases. **Connections must be properly established in a multi-step handshake process (*connection establishment*) before entering the *data transfer* phase. After data transmission is completed, the *connection termination* closes established virtual circuits and releases all allocated resources.**

The TCP header appears as follows:

TCP Header

Offsets	Octet	0								1								2								3							
Octet	**Bit**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
0	**0**	Source port																Destination port															
4	**32**	Sequence number																															
8	**64**	Acknowledgment number (if ACK set)																															
12	**96**	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	**128**	Checksum																Urgent pointer (if URG set)														
20 ...	**160** **...**	Options (if Data Offset > 5, padded at the end with "0" bytes if necessary) ...																															

- **Source port (16 bits)** – identifies the sending port
- **Destination port (16 bits)** – identifies the receiving port
- **Sequence number (32 bits)** – has a dual role:

    - If the SYN flag is set (1), then this is the initial sequence number. The sequence number of the actual first data byte and the acknowledged number in the corresponding ACK are then this sequence number plus 1.
    - If the SYN flag is clear (0), then this is the accumulated sequence number of the first data byte of this packet for the current session.

- **Acknowledgment number (32 bits)** – if the ACK flag is set then the value of this field is the next sequence number that the receiver is expecting. This acknowledges receipt of all prior bytes (if any). The first ACK sent by each end acknowledges the other end's initial sequence number itself, but no data.
- **Data offset (4 bits)** – specifies the size of the TCP header in 32-bit words. The minimum size header is 5 words and the maximum is 15 words thus giving the minimum size of 20 bytes and maximum of 60 bytes, allowing for up to 40 bytes of options in the header. This field gets its name from the fact that it is also the offset from the start of the TCP segment to the actual data.
- **Reserved (3 bits)** – for future use and should be set to zero
- **Flags (9 bits) (i.e. Control bits)** – contains 9 1-bit flags

- NS (1 bit) – ECN-nonce concealment protection (added to header by RFC 3540).
- CWR (1 bit) – Congestion Window Reduced (CWR) flag is set by the sending host to indicate that it received a TCP segment with the ECE flag set and had responded in congestion control mechanism (added to header by RFC 3168).
- ECE (1 bit) – ECN-Echo indicates

- If the SYN flag is set (1), that the TCP peer is ECN capable.
- If the SYN flag is clear (0), that a packet with Congestion Experienced flag in IP header set is received during normal transmission (added to header by RFC 3168).

- URG (1 bit) – indicates that the Urgent pointer field is significant
- ACK (1 bit) – indicates that the Acknowledgment field is significant. All packets after the initial SYN packet sent by the client should have this flag set.
- PSH (1 bit) – Push function. Asks to push the buffered data to the receiving application.
- RST (1 bit) – Reset the connection
- SYN (1 bit) – Synchronize sequence numbers. Only the first packet sent from each end should have this flag set. Some other flags change meaning based on this flag, and some are only valid for when it is set, and others when it is clear.
- FIN (1 bit) – No more data from sender

- **Window size (16 bits)** – the size of the *receive window*, which specifies the number of bytes (beyond the sequence number in the acknowledgment field) that the sender of this segment is currently willing to receive
- **Checksum (16 bits)** – The 16-bit checksum field is used for error-checking of the header and data
- **Urgent pointer (16 bits)** – if the URG flag is set, then this 16-bit field is an offset from the sequence number indicating the last urgent data byte
- **Options (Variable 0-320 bits, divisible by 32)** – The length of this field is determined by the data offset field. **Options have up to three fields: Option-Kind (1 byte), Option-Length (1 byte), Option-Data (variable).** The Option-Kind field indicates the type of option, and is the only field that is not optional. Depending on what kind of option we are dealing with, the next two fields may be set: the Option-Length field indicates the total length of the option, and the Option-Data field contains the value of the option, if applicable. For example, an Option-Kind byte of 0x01 indicates that this is a No-Op option used only for padding, and does not have an Option-Length or Option-Data byte following it. An Option-Kind byte of 0 is the End Of Options option, and is also only one byte. An Option-Kind byte of 0x02

indicates that this is the Maximum Segment Size option, and will be followed by a byte specifying the length of the MSS field (should be 0x04). Note that this length is the total length of the given options field, including Option-Kind and Option-Length bytes. So while the MSS value is typically expressed in two bytes, the length of the field will be 4 bytes (+2 bytes of kind and length). In short, an MSS option field with a value of 0x05B4 will show up as (0x02 0x04 0x05B4) in the TCP options section.

- **Padding** – The TCP header padding is used to ensure that the TCP header ends and data begins on a 32 bit boundary. The padding is composed of zeros

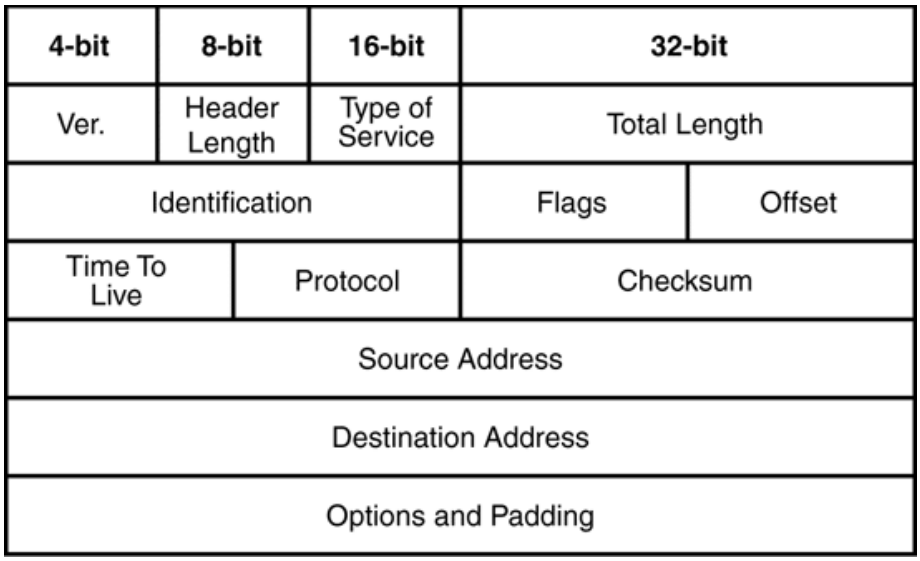
Internet Protocol (IP):

The **Internet Protocol** (**IP**) is the principal communications protocol used for relaying datagrams (also known as network packets) across an internetwork using the Internet Protocol Suite. Responsible for routing packets across network boundaries, it is the primary protocol that establishes the Internet. **IP manages how packets are delivered to and from servers and clients. IP defines datagram structures that encapsulate the data to be delivered. It also defines addressing methods that are used to label the datagram source and destination.**

The Internet Protocol is responsible for addressing hosts and routing datagrams (packets) from a source host to the destination host across one or more IP networks. For this purpose the Internet Protocol defines an addressing system that has two functions: identifying hosts and providing a logical location service. This is accomplished by defining standard datagrams and a standard addressing system.

Each datagram has two components, a header and a payload. The IP header is tagged with the source IP address, destination IP address, and other meta-data needed to route and deliver the datagram. The payload is the data to be transported. This process of nesting data payloads in a packet with a header is called encapsulation.

The IP header appears as follows:



Each field contains information about the IP packet that it carries.

Version Number

The version number indicates the version of IP that is in use for this packet. IP version 4 (Ipv4) is currently in widespread use.

Header Length

The header length indicates the overall length of the header. The receiving machine then knows when to stop reading the header and start reading data.

Type of Service

Mostly unused, the Type of Service field indicates the importance of the packet in a numerical value. Higher numbers result in prioritized handling.

Total Length

Total length shows the total length of the packet in bytes. The total packet length cannot exceed 65,535 bytes or it will be deemed corrupt by the receiver.

Identification

If there is more than one packet (an invariable inevitability), the identification field has an identifier that identifies its place in line, as it were. Fragmented packets retain their original ID number.

Flags

The first flag, if set, is ignored. If the DF (Do Not Fragment) flag is set, under no circumstances can the packet be fragmented. If the MF (More Fragments) bit is turned on (1), there are packet fragments to come, the last of which is set to off (0).

Offset

If the Flag field returns a 1 (on), the Offset field contains the location of the missing piece(s) indicated by a numerical offset based on the total length of the packet.

Time To Live (TTL)

Typically 15 to 30 seconds, TTL indicates the length of time that a packet is allowed to remain in transit. If a packet is discarded or lost in transit, an indicator is sent back to the sending computer that the loss occurred. The sending machine then has the option of resending that packet.

Protocol

The protocol field holds a numerical value indicating the handling protocol in use for this packet.

Checksum

The checksum value acts as a validation checksum for the header.

Source Address

The source address field indicates the address of the sending machine.

Destination Address

The destination address field indicates the address of the destination machine.

Options and Padding

The Options field is optional. If used, **it contains codes that indicate the use of security, strict or loose source routing, routing records, and timestamping.** If no options are used, the field is called *padded* and contains a 1. Padding is used to force a byte value that is rounded. Following indicates the bit counts for the options available;

Table: Bit Counts

Class	Number	Option
0	0	End of option list
0	2	Military security
0	3	Loose source routing

Class	Number	Option
0	7	Routing record*
0	9	Strict source routing
2	4	Timestamping

**This option adds fields.*

IP Address:

An **Internet Protocol address** (**IP address**) is a numerical label assigned to each device (e.g., computer, printer) participating in a computer network that uses the Internet Protocol for communication. An IP address serves two principal functions: host or network interface identification and location addressing. Its role has been characterized as follows: "*A name indicates what we seek. An address indicates where it is. A route indicates how to get there.*"

The designers of the Internet Protocol defined an IP address as a 32-bit number and this system, known as Internet Protocol Version 4 (IPv4), is still in use today. However, due to the enormous growth of the Internet and the predicted depletion of available addresses, a new version of IP (IPv6), using 128 bits for the address, was developed in 1995. IPv6 was standardized as RFC 2460 in 1998, and its deployment has been ongoing since the mid-2000s.

IP addresses are binary numbers, but they are usually stored in text files and displayed in human-readable notations, such as 172.16.254.1 (for IPv4), and 2001:db8:0:1234:0:567:8:1 (for IPv6). Two versions of the Internet Protocol (IP) are in use: IP Version 4 and IP Version 6. Each version defines an IP address differently. Because of its prevalence, the generic term *IP address* typically still refers to the addresses defined by IPv4. The gap in version sequence between IPv4 and IPv6 resulted from the assignment of number 5 to the experimental Internet Stream Protocol in 1979, which however was never referred to as IPv5.

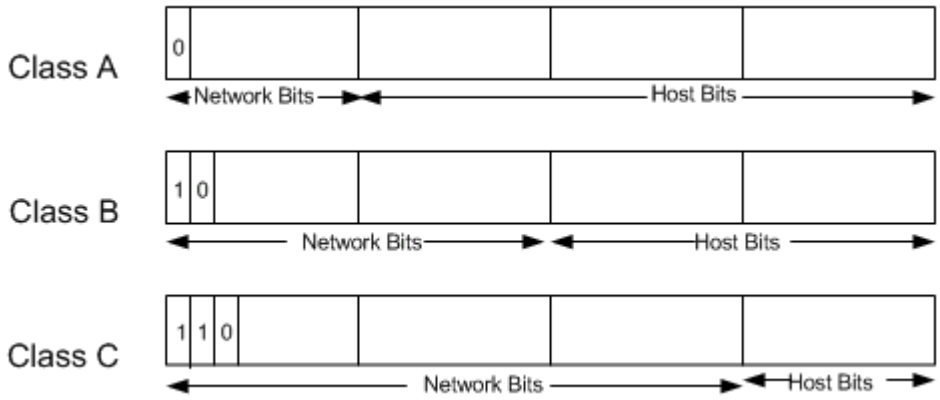
IPV4:

The IP address space is divided into different network classes. The system defined five classes, Class A, B, C, D, and E. **The Classes A, B, and C had different bit lengths for the new network identification. The rest of an address was used as previously to**

identify a host within a network, which meant that each network class had a different capacity to address hosts. Class D was allocated for multicast addressing and Class E was reserved for future applications.

Class A networks are intended mainly for use with a few very large networks, because they provide only 8 bits for the network address field. Class B networks allocate 16 bits, and Class C networks allocate 24 bits for the network address field. Class C networks only provide 8 bits for the host field, however, so the number of hosts per network may be a limiting factor. In all three cases, the left most bit(s) indicate the network class. IP addresses are written in dotted decimal format; for example, 34.0.0.1.

Address Formats for Class A, B, and C IP Networks



In IPv4 an address consists of 32 bits which limits the address space to 4294967296 (2^{32}) possible unique addresses. IPv4 reserves some addresses for special purposes such as private networks (~18 million addresses) or multicast addresses (~270 million addresses).

IPv4 addresses are canonically represented in dot-decimal notation, which consists of four decimal numbers, each ranging from 0 to 255, separated by dots, e.g., 172.16.254.1. Each part represents a group of 8 bits (octet) of the address. In some cases of technical writing, IPv4 addresses may be presented in various hexadecimal, octal, or binary representations.

IPV6:

An **Internet Protocol Version 6 address** (**IPv6 address**) is a numerical label that is used to identify a network interface of a computer or other network node participating in anIPv6-enabled computer network. IPv6 implements a new addressing system that allows for far more addresses to be assigned than with IPv4.

IPv6 does not implement interoperability features with IPv4, but essentially creates a parallel, independent network. Exchanging traffic between the two networks requires special translator gateways, but this is not generally required, since most computer

operating systems and software implement both protocols for transparent access to both networks, either natively or using a tunneling protocol like 6to4, 6in4, or Teredo.

IPv6 is the successor to the Internet's first addressing infrastructure, Internet Protocol version 4 (IPv4). In contrast to IPv4, which defined an IP address as a 32-bit value, IPv6 addresses have a size of 128 bits. Therefore, IPv6 has a vastly enlarged address space compared to IPv4.

IPv6 addresses, as usually represented in human-readable notation, consist of eight groups of four hexadecimal digits separated by colons, for example;

2001:0db8:85a3:0042:0000:8a2e:0370:7334.

IPv6 addresses are classified by the primary addressing and routing methodologies common in networking: unicast addressing, anycast addressing, and multicast addressing.

- **A unicast address** identifies a single network interface. The Internet Protocol delivers packets sent to a unicast address to that specific interface.
- **An anycast address** is assigned to a group of interfaces, usually belonging to different nodes. A packet sent to an anycast address is delivered to just one of the member interfaces, typically the *nearest* host, according to the routing protocol's definition of distance. Anycast addresses cannot be identified easily, they have the same format as unicast addresses, and differ only by their presence in the network at multiple points. Almost any unicast address can be employed as an anycast address.
- **A multicast address** is also used by multiple hosts, which acquire the multicast address destination by participating in the multicast distribution protocol among the network routers. A packet that is sent to a multicast address is delivered to all interfaces that have joined the corresponding multicast group.

An IPv6 address consists of 128 bits. Addresses are classified into various types for applications in the major addressing and routing methodologies: unicast, multicast, and anycast networking. In each of these, various address formats are recognized by logically dividing the 128 address bits into bit groups and establishing rules for associating the values of these bit groups with special addressing features.

Unicast and anycast address format

Unicast and anycast addresses are typically composed of two logical parts: a 64-bit network prefix used for routing, and a 64-bit interface identifier used to identify a host's network interface.

General unicast address format (routing prefix size varies)			
bits	48 (or more)	16 (or fewer)	64

field	*routing prefix*	*subnet id*	*interface identifier*

The *network prefix* (the *routing prefix* combined with the *subnet id*) is contained in the most significant 64 bits of the address. The size of the routing prefix may vary; a larger prefix size means a smaller subnet id size. The bits of the *subnet id(entifier)* field are available to the network administrator to define subnets within the given network. The 64-bit *interface identifier* is either automatically generated from the interface's MAC address using the modified EUI-64 format, obtained from a DHCPv6 server, automatically established randomly, or assigned manually.

A link-local address is also based on the interface identifier, but uses a different format for the network prefix.

Link-local address format			
bits	10	54	64
field	*prefix*	*zeroes*	*interface identifier*

The *prefix* field contains the binary value 1111111010. The 54 zeroes that follow make the total network prefix the same for all link-local addresses, rendering them non-routable.

Multicast address format

Multicast addresses are formed according to several specific formatting rules, depending on the application.

General multicast address format				
bits	8	4	4	112
field	*prefix*	*flg*	*sc*	*group ID*

The *prefix* holds the binary value 11111111 for any multicast address.

Currently, 3 of the 4 flag bits in the *flg* field are defined; the most-significant flag bit is reserved for future use.

Multicast address flags			
bit	flag	Meaning when 0	Meaning when 1
8	*reserved*	*reserved*	*reserved*
9	R (Rendezvous)	Rendezvous point not	Rendezvous point embedded

		embedded	
10	P (Prefix)	Without prefix information	Address based on network prefix
11	T (Transient)	Well-known multicast address	Dynamically assigned multicast address

The 4-bit scope field (*sc*) is used to indicate where the address is valid and unique.

There are special multicast addresses, like Solicited Node.

Solicited-Node multicast address format						
bits	8	4	4	79	9	24
field	*prefix*	*flg*	*sc*	*zeroes*	*ones*	*unicast address*

The *sc(ope)* field holds the binary value 0010 (link-local). Solicited-node multicast addresses are computed as a function of a node's unicast or anycast addresses. A solicited-node multicast address is created by copying the last 24 bits of a unicast or anycast address to the last 24 bits of the multicast address.

Unicast-prefix-based multicast address format								
bits	8	4	4	4	4	8	64	32
field	*prefix*	*flg*	*sc*	*res*	*riid*	*plen*	*network prefix*	*group ID*

Link-scoped multicast addresses use a comparable format.

IPv4 and IPv6 Header Structure:

IPV4 Header Structure:

The IPv4 packet header consists of 14 fields, of which 13 are required. The 14th field is optional (red background in table) and aptly named: options. The fields in the header are packed with the most significant byte first (big endian), and for the diagram and discussion, the most significant bits are considered to come first (MSB 0 bit numbering). The most significant bit is numbered 0, so the version field is actually found in the four most significant bits of the first byte, for example.

bit offset	0–3	4–7	8–13	14-15	16–18	19–31
0	Version	Internet Header Length	Differentiated Services Code Point	Explicit Congestion Notification	Total Length	

32	Identification		Flags	Fragment Offset
64	Time to Live	Protocol	Header checksum	
96	Source IP Address			
128	Destination IP Address			
160	Options (if Header Length > 5)			

Version : The first header field in an IP packet is the four-bit version field. For IPv4, this has a value of 4 (hence the name IPv4).

Internet Header Length (IHL): The second field (4 bits) is the Internet Header Length (IHL), which is the number of 32-bit words in the header. Since an IPv4 header may contain a variable number of options, this field specifies the size of the header (this also coincides with the offset to the data). The minimum value for this field is 5 (RFC 791), which is a length of 5×32 = 160 bits = 20 bytes. Being a 4-bit value, the maximum length is 15 words (15×32 bits) or 480 bits = 60 bytes.

Differentiated Services Code Point (DSCP): Originally defined as the Type of Service field, this field is now defined by RFC 2474 for Differentiated services (DiffServ). New technologies are emerging that require real-time data streaming and therefore make use of the DSCP field. An example is Voice over IP (VoIP), which is used for interactive data voice exchange.

Explicit Congestion Notification (ECN): This field is defined in RFC 3168 and allows end-to-end notification of network congestion without dropping packets. ECN is an optional feature that is only used when both endpoints support it and are willing to use it. It is only effective when supported by the underlying network.

Total Length: This 16-bit field defines the entire packet (fragment) size, including header and data, in bytes. The minimum-length packet is 20 bytes (20-byte header + 0 bytes data) and the maximum is 65,535 bytes — the maximum value of a 16-bit word. The largest datagram that any host is required to be able to reassemble is 576 bytes, but most modern hosts handle much larger packets. Sometimes subnetworks impose further restrictions on the packet size, in which case datagrams must be fragmented. Fragmentation is handled in either the host or router in IPv4.

Identification: This field is an identification field and is primarily used for uniquely identifying fragments of an original IP datagram. Some experimental work has suggested using the ID field for other purposes, such as for adding packet-tracing information to help trace datagrams with spoofed source addresses.

Flags; A three-bit field follows and is used to control or identify fragments. They are (in order, from high order to low order):

- bit 0: Reserved; must be zero.
- bit 1: Don't Fragment (DF)
- bit 2: More Fragments (MF)

If the DF flag is set, and fragmentation is required to route the packet, then the packet is dropped. This can be used when sending packets to a host that does not have sufficient resources to handle fragmentation. It can also be used for Path MTU Discovery, either automatically by the host IP software, or manually using diagnostic tools such as ping or traceroute.

For unfragmented packets, the MF flag is cleared. For fragmented packets, all fragments except the last have the MF flag set. The last fragment has a non-zero Fragment Offset field, differentiating it from an unfragmented packet.

Fragment Offset: The fragment offset field, measured in units of eight-byte blocks, is 13 bits long and specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP datagram. The first fragment has an offset of zero. This allows a maximum offset of $(2^{13} – 1) \times 8 = 65{,}528$ bytes, which would exceed the maximum IP packet length of 65,535 bytes with the header length included (65,528 + 20 = 65,548 bytes).

Time To Live (TTL): An eight-bit time to live field helps prevent datagrams from persisting (e.g. going in circles) on an internet. This field limits a datagram's lifetime. It is specified in seconds, but time intervals less than 1 second are rounded up to 1. In practice, the field has become a hop count—when the datagram arrives at a router, the router decrements the TTL field by one. When the TTL field hits zero, the router discards the packet and typically sends a ICMP Time Exceeded message to the sender.

The program traceroute uses these ICMP Time Exceeded messages to print the routers used by packets to go from the source to the destination.

Protocol: This field defines the protocol used in the data portion of the IP datagram. The Internet Assigned Numbers Authority maintains a list of IP protocol numbers which was originally defined in RFC 790.

Header Checksum: The 16-bit checksum field is used for error-checking of the header. When a packet arrives at a router, the router calculates the checksum of the header and compares it to the checksum field. If the values do not match, the router discards the

packet. Errors in the data field must be handled by the encapsulated protocol. Both UDP and TCP have checksum fields.

When a packet arrives at a router, the router decreases the TTL field. Consequently, the router must calculate a new checksum. RFC 1071 defines the checksum calculation:

The checksum field is the 16-bit one's complement of the one's complement sum of all 16-bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

For example, consider Hex 450000304422400080060000 8c7c19acae241e2b (20 bytes IP header):

Step 1.

4500 + 0030 + 4422 + 4000 + 8006 + 0000 + 8c7c + 19ac + ae24 + 1e2b = 2BBCF (16-bit sum)

Step 2.

2 + BBCF = BBD1 = 1011101111010001 (1's complement 16-bit sum)

Step 3.

~BBD1 = 0100010000101110 = 442E (1's complement of 1's complement 16-bit sum)

To validate a header's checksum the same algorithm may be used - the checksum of a header which contains a correct checksum field is a word containing all zeros (value 0):

2BBCF + 442E = 2FFFD. 2 + FFFD = FFFF. the 1'S of FFFF = 0.

Source address: This field is the IPv4 address of the sender of the packet. Note that this address may be changed in transit by a network address translation device.

Destination address: This field is the IPv4 address of the receiver of the packet. As with the source address, this may be changed in transit by a network address translation device.

Options: The options field is not often used. Note that the value in the IHL field must include enough extra 32-bit words to hold all the options (plus any padding needed to ensure that the header contains an integral number of 32-bit words). The list of options may be terminated with an EOL (End of Options List, 0x00) option; this is only necessary if the end of the options would not otherwise coincide with the end of the header.

IPV6 Header Structure:

IPv6 specifies a new packet format, designed to minimize packet header processing by routers. Because the headers of IPv4 packets and IPv6 packets are significantly different, the two protocols are not interoperable. However, in most respects, IPv6 is a conservative extension of IPv4. Most transport and application-layer protocols need little or no change to operate over IPv6; exceptions are application protocols that embed internet-layer addresses, such as FTP and NTPv3, where the new address format may cause conflicts with existing protocol syntax.

The fixed header of an IPv6 packet consists of its first 40 octets (320 bits). It has the following format:

Fixed header format

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version				*Traffic Class*								*Flow Label*																			
Payload Length																*Next Header*								*Hop Limit*						
Source Address																															
Destination Address																															

***Version* (4 bits):** The constant 6 (bit sequence 0110).

***Traffic Class* (8 bits):** The bits of this field hold two values. The 6 most-significant bits are used for DSCP, which is used to classify packets. The remaining two bits are used for ECN; priority values subdivide into ranges: traffic where the source provides congestion control and non-congestion control traffic.

***Flow Label* (20 bits):** Originally created for giving real-time applications special service. Flow Label specifications and minimum requirements are described, and first uses of this field are emerging.

***Payload Length* (16 bits):** The size of the payload in octets, including any extension headers. The length is set to zero when a *Hop-by-Hop* extension header carries a Jumbo Payload option.

***Next Header* (8 bits):** Specifies the type of the next header. This field usually specifies the transport layer protocol used by a packet's payload. When extension headers are present

in the packet this field indicates which extension header follows. The values are shared with those used for the IPv4 protocol field, as both fields have the same function

***Hop Limit* (8 bits):** Replaces the time to live field of IPv4. This value is decremented by one at each intermediate node visited by the packet. When the counter reaches 0 the packet is discarded.

***Source Address* (128 bits):** The IPv6 address of the sending node.

***Destination Address* (128 bits):** The IPv6 address of the destination node(s).

Internet RFCs:

In computer network engineering, a **Request for Comments** (**RFC**) is a memorandum published by the Internet Engineering Task Force (IETF) describing methods, behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems. It began in 1969 as a set of working notes about ARPAnet research and development.

Memos in the **Requests for Comments (RFC)** document series contain technical and organizational notes about the Internet. They cover many aspects of computer networking, including protocols, procedures, programs, and concepts, as well as meeting notes, opinions, and sometimes humor.

RFCs are numbered (roughly) consecutively, and these numbers provide a single unique label space for all RFCs. RFCs are published online through a number of repositories, and there is an online index of RFCs.

The most common meaning for the word *standard* on the Internet is probably 'current (i.e. non-obsoleted) RFC'. This isn't quite as rigorous a concept as it may sound. ***RFC***s are *an enumerated series of documents issued by the IETF*, varying greatly in their nature and status. Some of them are often called standards - this may apply even to RFCs which explicitly state that they do not define a standard of any kind! - but according to the official terminology, only a few of them have been designated as ***Internet standards***. An Internet standard has, in addition to an RFC number, an STD number, which does not change even if the RFC number is changed; for example, the IP protocol is defined by STD 5 which is currently RFC 791.

RFC Categories

Each RFC has a "category" or "status" designation. The possible categories are:

- STANDARD, DRAFT STANDARD, PROPOSED STANDARD

These are *standards-track* documents, official specifications of the Internet protocol suite defined by the Internet Engineering Task Force (IETF) and its steering group the IESG.

- BEST CURRENT PRACTICE

  These are official guidelines and recommendations, but not standards, from the IETF.

- INFORMATIONAL, EXPERIMENTAL

  These non-standards documents may originate in the IETF or may be independent submissions.

- HISTORIC

  These are former standards that have been actively deprecated.

Unit – 3 (Protocols and Client/Server Applications)

SMTP (Simple Mail Transfer Protocol) :

Simple Mail Transfer Protocol (**SMTP**) is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks. SMTP uses TCP port 25. SMTP is a connection-oriented, text-based protocol in which a mail sender communicates with a mail receiver by issuing command strings and supplying necessary data over a reliable ordered data stream channel, typically a Transmission Control Protocol (TCP) connection. An *SMTP session* consists of commands originated by an SMTP client (the initiating agent, sender, or transmitter) and corresponding responses from the SMTP server (the listening agent, or receiver) so that the session is opened, and session parameters are exchanged. A

session may include zero or more SMTP transactions. An *SMTP transaction* consists of three command/reply sequences. They are:

1. **MAIL** command, to establish the return address, a.k.a. Return-Path, 5321.From, mfrom, or envelope sender. This is the address for bounce messages.
2. **RCPT** command, to establish a recipient of this message. This command can be issued multiple times, one for each recipient. These addresses are also part of the envelope.
3. **DATA** to send the *message text*. This is the content of the message, as opposed to its envelope. It consists of a *message header* and a *message body* separated by an empty line. DATA is actually a group of commands, and the server replies twice: once to the *DATA command* proper, to acknowledge that it is ready to receive the text, and the second time after the end-of-data sequence, to either accept or reject the entire message.

SMTP is a delivery protocol 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. 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.

The basic SMTP works well, but it is limited in several respects. **It does not include authentication**. This is quite useful for sending spam. **Another limitation is that SMTP transfers ASCII messages, not binary data.** This is why the base64 MIME content transfer encoding was needed. However, with that encoding the mail transmission uses bandwidth inefficiently, which is an issue for large messages. **A third limitation is that SMTP sends messages in the clear. It has no encryption to provide a measure of privacy against prying eyes.** To allow these and many other problems related to message processing to be addressed, SMTP was revised to have an extension mechanism. This mechanism is a mandatory part of the RFC 5321 standard. The use of SMTP with extensions is called **ESMTP** (**Extended SMTP**).

POP (Post Office Protocol):

The **Post Office Protocol** (**POP**) is an application-layer Internet standard protocol used by local e-mail clients to retrieve mail from a remote server over a TCP/IP connection. POP and IMAP (Internet Message Access Protocol) are the two most prevalent Internet standard protocols for e-mail retrieval. Virtually all modern e-mail clients and servers support both.

The POP protocol has been developed through several versions, with version 3 (POP3) being the current standard. Most webmail service providers such as Hotmail, Gmail and Yahoo! Mail also provide IMAP and POP3 service.

Like it seems everything on the internet, mail retrieval is a client-server application. The Post Office Protocol defines how your email client should talk to the POP server. The POP is a very simple protocol. This makes it easy to implement, has earned the Post Office Protocol widespread adoption and makes it very robust, but it also means the Post Office Protocol provides only basic functionality.

Things that can be done via the POP include:

- Retrieve mail from an ISP and delete it on the server.
- Retrieve mail from an ISP but not delete it on the server.
- Ask whether new mail has arrived but not retrieve it.
- Peek at a few lines of a message to see whether it is worth retrieving.

Usually the POP server listens to port 110 for incoming connections. Upon connection from a POP client (your email program) it will hopefully respond with *+OK pop.philo.org ready* or something similar. The *+OK* indicates that everything is â€" OK. Its negative equivalent is *-ERR*, which means something has gone wrong. Maybe your email client has already shown you one of these negative server replies.

Clients that leave mail on servers generally use the UIDL (Unique ID Listing) command to get the current association of message-numbers to message identified by its unique identifier. The unique identifier is arbitrary, and might be repeated if the mailbox contains identical messages. In contrast, IMAP uses a 32-bit unique identifier (UID) that is assigned to messages in ascending (although not necessarily consecutive) order as they are received. When retrieving new messages, an IMAP client requests the UIDs greater than the highest UID among all previously-retrieved messages, whereas a POP client must fetch the entire UIDL map. For large mailboxes, this can require significant processing

An informal proposal had been outlined for a *"POP4"* specification, complete with a working server implementation. This *"POP4"* proposal added basic folder management, multipart message support, as well as message flag management, allowing for a light protocol which supports some popular IMAP features which POP3 currently lacks.

IMAP (Internet Message Access Protocol):

IMAP stands for Internet Message Access Protocol. It is a method of accessing electronic mail or bulletin board messages that are kept on a (possibly shared) mail server. In other words, it permits a "client" email program to access remote message stores as if they were local. **For example, email stored on an IMAP server can be manipulated from a desktop computer at home, a workstation at the office, and a notebook computer while traveling, without the need to transfer messages or files back and forth between these computers.**

IMAP's ability to access messages (both new and saved) from more than one computer has become extremely important as reliance on electronic messaging and use of multiple computers increase, but this functionality cannot be taken for granted: the widely used Post Office Protocol (POP) works best when one has only a single computer, since it was designed to support "offline" message access, wherein messages are downloaded and then deleted from the mail server. This mode of access is not compatible with access from multiple computers since it tends to sprinkle messages across all of the computers used for mail access. Thus, unless all of those machines share a common file system, the offline mode of access that POP was designed to support effectively ties the user to one computer for message storage and manipulation.

Key goals for IMAP include:

1. Be fully compatible with Internet messaging standards, e.g. MIME.
2. Allow message access and management from more than one computer.
3. Allow access without reliance on less efficient file access protocols.
4. Provide support for "online", "offline", and "disconnected" access modes
5. Support for concurrent access to shared mailboxes
6. Client software needs no knowledge about the server's file store format.

IMAP supports both on-line and off-line modes of operation. E-mail clients using IMAP generally leave messages on the server until the user explicitly deletes them. This and other characteristics of IMAP operation allow multiple clients to manage the same mailbox. Most e-mail *clients* support IMAP in addition to POP to retrieve messages; however, fewer e-mail *services* support IMAP. IMAP offers access to the mail storage. Clients may store local copies of the messages, but these are considered to be a temporary cache.

To use IMAP, the mail server runs an IMAP server that listens to port 143. The user agent runs an IMAP client. Incoming e-mail messages are sent to an e-mail server that stores messages in the recipient's e-mail box. **The user retrieves the messages with an e-mail client that uses one of a number of e-mail retrieval protocols. Some clients and servers preferentially use vendor-specific, proprietary protocols, but most support the Internet standard protocols, SMTP for sending e-mail and POP and IMAP for retrieving e-mail, allowing interoperability with other servers and clients.**

When using POP, clients typically connect to the e-mail server briefly, only as long as it takes to download new messages. When using IMAP4, clients often stay connected as long as the user interface is active and download message content on demand. For users with many or large messages, this IMAP4 usage pattern can result in faster response times.

The POP protocol requires the currently connected client to be the only client connected to the mailbox. In contrast, the IMAP protocol specifically allows simultaneous access by multiple clients and provides mechanisms for clients to detect changes made to the mailbox by other, concurrently connected, clients.

PGP (Pretty Good Privacy)

Pretty Good Privacy (PGP) is a data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication. PGP is often used for signing, encrypting and decrypting texts, e-mails, files, directories and whole disk partitions to increase the security of e-mail communications.

PGP is a public key encryption package to protect e-mail and data files. It lets you communicate securely with people you've never met, with no secure channels needed for prior exchange of keys. It's well featured and fast, with sophisticated key management, digital signatures, data compression, and good economic design. **The actual operation of PGP is based on five services: authentication, confidentiality, compression, e-mail compatibility, and segmentation.**

- PGP provides authentication via a digital signature scheme.
- PGP provides confidentiality by encrypting messages before transmission using RSA schemes.
- PGP compresses the message after applying the signature and before encryption. The idea is to save space.
- PGP encrypts a message together with the signature (if not sent separately) resulting into a stream of arbitrary 8-bit octets. But since many e-mail systems permit only use of blocks consisting of ASCII text, PGP accommodates this by converting the raw 8-bit binary streams into streams of printable ASCII characters using a radix-64 conversion scheme. On receipt, the block is converted back from radix-64 format to binary.
- To accommodate e-mail size restrictions, PGP automatically segments email messages that are too long. However, the segmentation is done after all the housekeeping is done on the message, just before transmitting it. So the session key and signature appear only once at the beginning of the first segment

transmitted. At receipt, the receiving PGP strips off all e-mail headers and reassemble the original mail.

HTTP

HTTP is the protocol to exchange or transfer hypertext. ***H****yper****T****ext* ***T****ransfer* ***P****rotocol*, the underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. For example, when you enter a URL in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page. HTTP is the framework for how browsers will display and use file formats. When you enter in a URL with HTTP at the beginning, you are requesting a web page which can contain other elements (such as pictures) and links to other resources. HTTP utilizes TCP port 80 by default, though other ports such as 8080 can alternatively be used.

The current version of HTTP in widespread use - HTTP version 1.1 - was developed to address some of the performance limitations of the original version - HTTP 1.0. HTTP 1.1 is documented in RFC 2068.

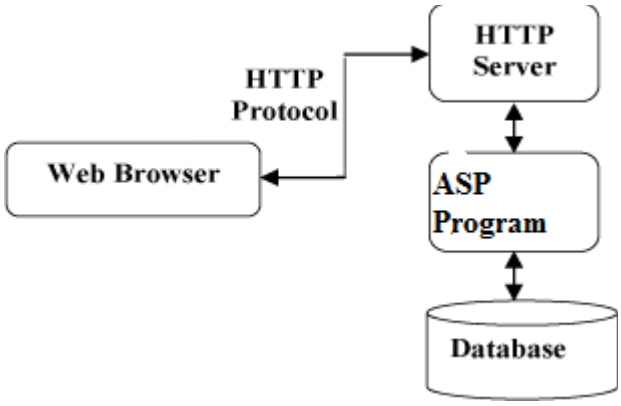
There are three important things about HTTP of which you should be aware:

- **HTTP is connectionless:** After a request is made, the client disconnects from the server and waits for a response. The server must re-establish the connection after it processes the request.

- **HTTP is media independent:** Any type of data can be sent by HTTP as long as both the client and server know how to handle the data content. How content is handled is determined by the MIME specification.

- **HTTP is stateless:** This is a direct result of HTTP's being connectionless. The server and client are aware of each other only during a request. Afterwards, each forgets the other. For this reason neither the client nor the browser can retain information between different requests across the web pages.

Following diagram shows where HTTP Protocol fits in communication;



HTTP is an application layer network protocol built on top of TCP. HTTP clients (such as Web browsers) and servers communicate via HTTP request and response messages. **The three main HTTP message types are GET, POST, and HEAD.**

An HTTP session is a sequence of network request-response transactions**. An HTTP client initiates a request by establishing a Transmission Control Protocol (TCP) connection to a particular port on a server (typically port 80). An HTTP server listening on that port waits for a client's request message. Upon receiving the request, the server sends back a status line, such as "HTTP/1.1 200 OK", and a message of its own**. The body of this message is typically the requested resource, although an error message or other information may also be returned

HTTP Request Methods:

HTTP defines nine methods (sometimes referred to as "verbs") indicating the desired action to be performed on the identified **resource**. What this resource represents, whether pre-existing data or data that is generated dynamically, depends on the implementation of the server. Often, the resource corresponds to a file or the output of an executable residing on the server.

HEAD: Asks for the response identical to the one that would correspond to a GET request, but without the response body. This is useful for retrieving meta-information written in response headers, without having to transport the entire content.

GET: Requests a representation of the specified resource. Requests using GET should only retrieve data and should have no other effect. (This is also true of some other HTTP methods.) The W3C has published guidance principles on this distinction, saying, "Web application design should be informed by the above principles, but also by the relevant limitations."

POST: Submits data to be processed (e.g., from an HTML form) to the identified resource. The data is included in the body of the request. This may result in the creation of a new resource or the updates of existing resources or both.

PUT: Uploads a representation of the specified resource.

DELETE: Deletes the specified resource.

TRACE: Echoes back the received request, so that a client can see what (if any) changes or additions have been made by intermediate servers.

OPTIONS: Returns the HTTP methods that the server supports for specified URL. This can be used to check the functionality of a web server by requesting '*' instead of a specific resource.

CONNECT: Converts the request connection to a transparent TCP/IP tunnel, usually to facilitate SSL-encrypted communication (HTTPS) through an unencrypted HTTP proxy.

PATCH: Is used to apply partial modifications to a resource.

HTTP servers are required to implement at least the GET and HEAD methods[13] and, whenever possible, also the OPTIONS method.

FTP:

File Transfer Protocol (FTP) lives up to its name and provides a method for transferring files over a network from one computer to another. More generally, it provides for some simple file management on the contents of a remote computer. It is an old protocol and is used less than it was before the World Wide Web came along. Today, its primary use is uploading files to a Web site. It can also be used for downloading from the Web but, more often than not, downloading is done via HTTP. Sites that have a lot of downloading (software sites, for example) will often have an FTP server to handle the traffic. If FTP is involved, the URL will have *ftp:* at the front.

The File Transfer Protocol is used to send files from one system to another under user commands. Both text and binary files are accommodated and the protocol provides features for controlling user access. When a user wishes to engage in File transfer, FTP sets up a TCP connection to the target system for the exchange of control messages. These allow used ID and password to be transmitted and allow the user to specify the file and file

action desired. Once file transfer is approved, a second TCP connection is set up for data transfer. The file is transferred over the data connection, without the overhead of headers, or control information at the application level. When the transfer is complete, the control connection is used to signal the completion and to accept new file transfer commands.

FTP can be run in active or passive mode, which determines how the data connection is established. In active mode, the client sends the server the IP address and port number on which the client will listen, and the server initiates the TCP connection. at the condition when the client is behind a firewall and unable to accept incoming TCP connections, passive mode may be used. In this mode the client sends a PASV command to the server and receives an IP address and port number in return. The client uses these to open the data connection to the server. Data transfer can be done in any of three modes:

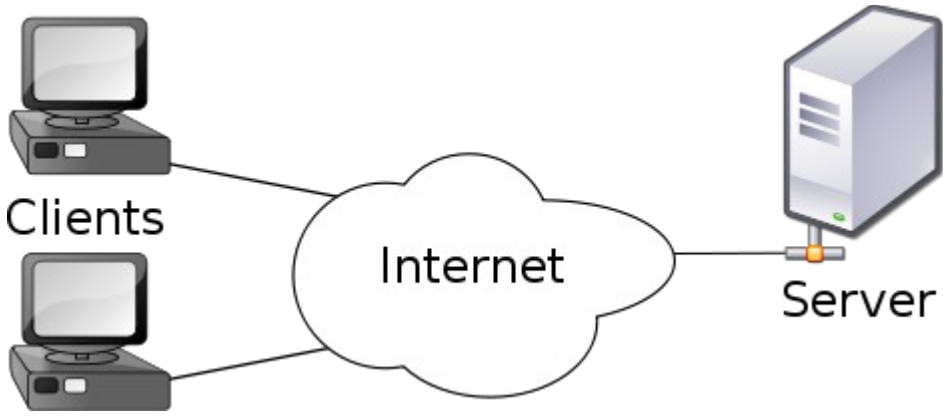
- Stream mode: Data is sent as a continuous stream, relieving FTP from doing any processing. Rather, all processing is left up to TCP. No End-of-file indicator is needed, unless the data is divided into records.
- Block mode: FTP breaks the data into several blocks (block header, byte count, and data field) and then passes it on to TCP.
- Compressed mode: Data is compressed using a single algorithm (usually run-length encoding).

Client/Server Model:

The **client–server model** is a computing model that acts as distributed application which partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server machine is a host that is running one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

Client/Server Architecture:

Client server network architecture consists of two kinds of computers: clients and servers. Clients are the computers that that do not share any of its resources but requests data and other services from the server computers and server computers provide services to the client computers by responding to client computers requests. Normally servers are powerful computers and clients are less powerful personal computers. Web servers are included as part of a larger package of internet and intranet related programs for serving e-mail, downloading requests for FTP files and building and publishing web pages.



Advantages

- The client/ server architecture reduces network traffic by providing a query response to the user rather than transferring total files.
- The client/ server model improves multi-user updating through a graphical user interface (GUI) front end to the shared database.
- Easy to implement security policies, since the data are stored in central location
- Simplified network administration

Disadvantages

- Failure of the server causes whole network to be collapsed
- Expensive than P2P, Dedicated powerful servers are needed
- Extra effort are needed for administering and managing the server.

Client/Server architecture can be of different model based on the number of layers it holds. Some of them are;

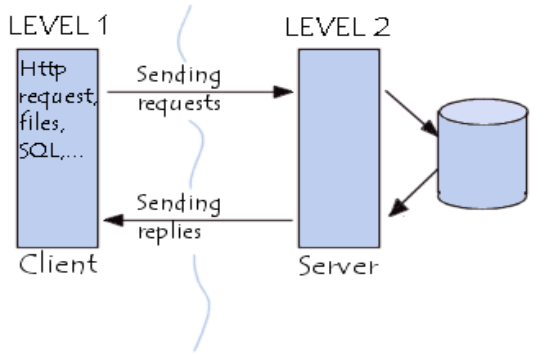
- **2-Tier Architecture**

  2-tier architecture is used to describe client/server systems where the client requests resources and the server responds directly to the request, using its own resources. This means that the server does not call on another application in order to provide part of the service. It runs the client processes separately from the server processes, usually on a different computer:

  – The client processes provide an interface for the customer, and gather and present data usually on the customer's computer. This part of the application is the presentation layer

  – The server processes provide an interface with the data store of the business. This part of the application is the data layer

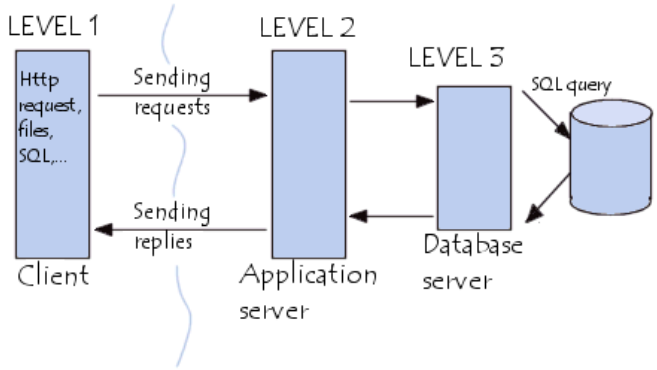
  – The business logic that validates data, monitors security and permissions, and performs other business rules can be housed on either the client or the server, or split between the two.
    - Fundamental units of work required to complete the business process
    - Business rules can be automated by an application program.



- **3-Tier Architecture**

In 3-tier architecture, there is an intermediary level, meaning the architecture is generally split up between:

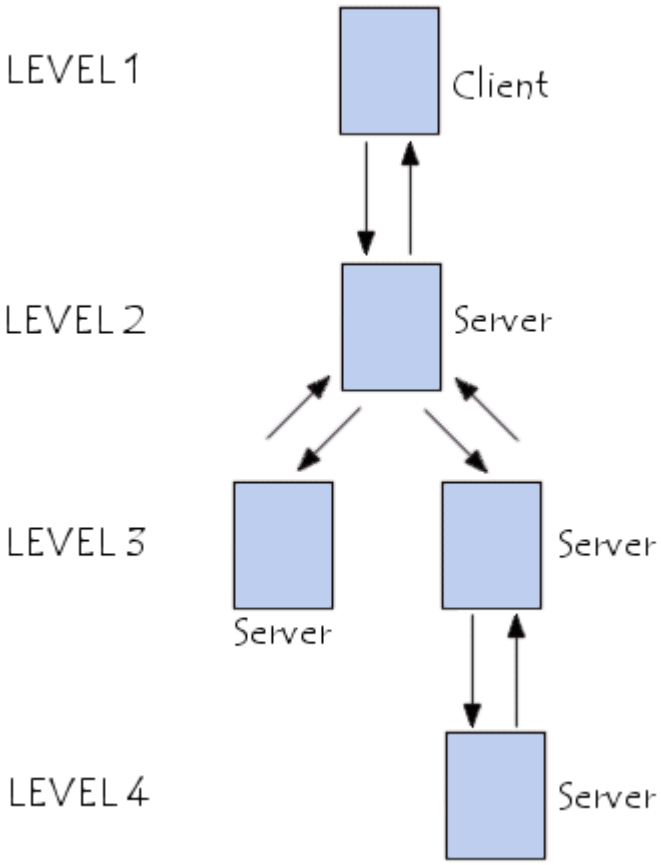
- A client, i.e. the computer, which requests the resources, equipped with a user interface (usually a web browser) for presentation purposes
- The application server (also called **middleware**), whose task it is to provide the requested resources, but by calling on another server
- The data server, which provides the application server with the data it requires



- **N-Tier Architecture (multi-tier)**

N-tier architecture (with N more than 3) is really 3 tier architectures in which the middle tier is split up into new tiers. The application tier is broken down into separate parts. What these parts are differs from system to system. The following picture shows it:

The primary advantage of N-tier architectures is that they make load balancing possible. Since the application logic is distributed between several servers, processing can then be more evenly distributed among those servers. N-tiered architectures are also more easily scalable, since only servers experiencing high demand, such as the application server, need be upgraded. The primary disadvantage of N-tier architectures is that it is also more difficult to program and test an N-tier architecture due to its increased complexity.



Advantages of Multi-Tier Client/Server architectures include:

- Changes to the user interface or to the application logic are largely independent from one another, allowing the application to evolve easily to meet new requirements.

- Network bottlenecks are minimized because the application layer does not transmit extra data to the client, only what is needed to handle a task.

- The client is insulated from database and network operations. The client can access data easily and quickly without having to know where data is or how many servers are on the system.

- Database connections can be 'pooled' and thus shared by several users, which greatly reduces the cost associated with per-user licensing.

- The organization has database independence because the data layer is written using standard SQL which is platform independent. The enterprise is not tied to vendor-specific stored procedures.

- The application layer can be written in standard third or fourth generation languages, such as ASP, PHP with which the organization's in-house programmers are experienced.

What kind of systems can benefit?

Generally, any Client/Server system can be implemented in an 'N-Tier' architecture, where application logic is partitioned among various servers. This application partitioning creates an integrated information infrastructure which enables consistent, secure, and global access to critical data. A significant reduction in network traffic, which leads to faster network communications, greater reliability, and greater overall performance is also made possible in a 'N-Tier' Client/Server architecture.

Universal Internet Browsing: (Discussed in class)

Multiple Protocol Support:

Complex data communication systems no longer utilize a single protocol to handle all transmission tasks. Instead, they require a set of protocols, called a protocol suite. The reason for using multiple protocols is to make them less complicated; this simplifies dealing with problems that arise when machines communicate over a network. such problems include the following;

- **Hardware failure:** when a router or host hardware fails, the protocol software needs to detect the failure and recover from it.

- **Network congestion:** the protocol needs to detect when the network capacity has been exceeded and arrange a way to handle the congestion.

- **Packet delay or loss:** the protocol software needs to adapt to long delay in order not to lose packets that were significantly delayed.

- **Data corruption:** the protocol software needs to detect and recover from transmission errors and corruption due to transmission impairments of hardware failures

- **Data duplication or sequence error:** networks that offer multiple routes may deliver data out of sequence or deliver duplicates of packets. The protocol software needs to reorder packets and remove duplicates.

It is difficult or undesirable to write a single protocol that will handle everything in the network, particularly since many networks are heterogeneous. Hence multiple protocols are needed in network communications. There has been lots of protocols adapted till the date.

Unit – 4 (HTTP and the Web Services)

HTTP:

Refer Unit 3.

Web Servers and Web Access:

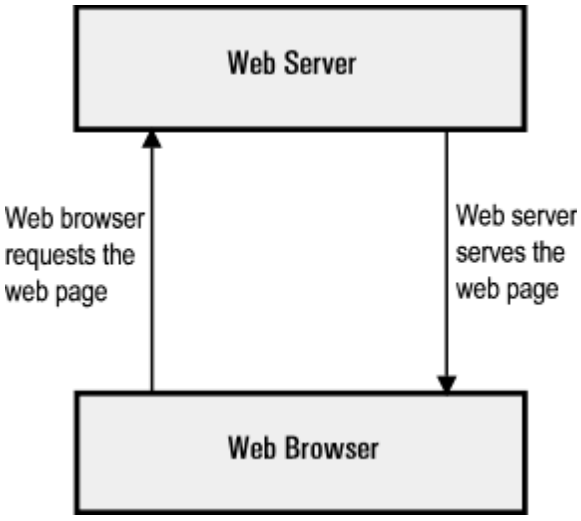
The primary function of a web server is to deliver web pages on the request to clients using the Hypertext Transfer Protocol (HTTP). This means delivery of HTML documents and any additional content that may be included by a document, such as images, style sheets and scripts.

A user agent, commonly a web browser or web crawler, initiates communication by making a request for a specific resource using HTTP and the server responds with the content of that resource or an error message if unable to do so. The resource is typically a real file on the server's secondary memory, but this is not necessarily the case and depends on how the web server is implemented.

While the primary function is to serve content, a full implementation of HTTP also includes ways of receiving content from clients. This feature is used for submitting web forms, including uploading of files.

Web servers are not always used for serving the World Wide Web. They can also be found embedded in devices such as printers, routers, webcams and serving only a local network. The web server may then be used as a part of a system for monitoring and/or administering the device in question. This usually means that no additional software has to be installed on the client computer; since only a web browser is required (which now is included with most operating systems).

Whenever you view a web page on the internet, you are requesting that page from a web server. When you type a URL into your browser (for example, http://www.bhattag.com/internet_technology/index.cfm"), your browser requests the page from the web server and the web server sends the page back:



Basically to get service from the server:

1. **Your web browser first needs to know which IP address the website, say, "www.jagdish.com" resolves to. If it doesn't already have this information stored in it's cache, it requests the information from one or more DNS servers (via the internet). The DNS server tells the browser which IP address the website is located at. Note that the IP address was assigned when the website was first created on the web server.**
2. **Now that the web browser knows which IP address the website is located at, it can request the full URL from the web server.**
3. The web server responds by sending back the requested page. If the page doesn't exist (or another error occurs), it will send back the appropriate error message.
4. Your web browser receives the page and renders it as required.

When referring to web browsers and web servers in this manner, we usually refer to them as a *client* (web browser) and a *server* (web server).

There are many Web server software applications, including public domain software from NCSA and Apache, and commercial packages from Microsoft, Netscape and others.

Universal Naming with URI:

In the field of computer networking, a **URI scheme** is the top level of the uniform resource identifier (URI) naming structure. **All URIs and absolute URI references are formed with a scheme name, followed by a colon character (":"), and the remainder of the URI called *the scheme-specific part*.** The syntax and semantics of the scheme-specific part are left largely to the specifications governing individual schemes, subject to certain constraints such as reserved characters and how to "escape" them.

URI schemes are frequently and incorrectly referred to as "protocols", or specifically as **URI protocols** or **URL protocols**, since most were originally designed to be used with a particular protocol, and often have the same name. The http scheme, for instance, is

generally used for interacting with Web resources using HyperText Transfer Protocol. Today, URIs with that scheme are also used for other purposes,

Every URI is defined as consisting of four parts, as follows:

<schema name>:<hierarchical part?[?<query>][#<fragment>]

The **scheme name** consists of a sequence of characters beginning with a letter and followed by any combination of letters, digits, plus ("+"), period ("."), or hyphen ("-"). Although schemes are case-insensitive, the canonical form is lowercase and documents that specify schemes must do so with lowercase letters. It is followed by a colon (":").

The **hierarchical part** of the URI is intended to hold identification information hierarchical in nature. Usually this part begins with a double forward slash ("//"), followed by an ***authority*** **part and an optional** ***path*****.**

- The **authority** part holds an optional user-information part, terminated with "@" (e.g. username:password@); a hostname (e.g., domain name or IP address); and an optional port number, preceded by a colon ":".
- The **path** part is a sequence of segments (conceptually similar to directories, though not necessarily representing them) separated by a forward slash ("/"). Historically, each segment was specified to contain parameters separated from it using a semicolon (";"), though this was rarely used in practice and current specifications allow but no longer specify such semantics.

The **query** is an optional part, separated by a question mark ("?"), that contains additional identification information that is not hierarchical in nature. The query string syntax is not generically defined, but it is commonly organized as a sequence of <key>=<value> pairs, with the pairs separated by a semicolon or an ampersand. For example:

The **fragment** is an optional part separated from the front parts by a hash ("#"). It holds additional identifying information that provides direction to a secondary resource, e.g., a section heading (in an article) identified by the remainder of the URI. When the primary resource is an HTML document, the **fragment** is often an id attribute of a specific element and web browsers will make sure this element is visible.

Example

mailto:username@example.com?subject=Topic

ftp:// jagdish@ftp.example.org

Universal Naming Convention:

Universal Naning Convention (UNC) is the way to represent the path to a directory on a networked computer. It is of the form:

\\<name of network computer>\directory

i.e \\home\downloads

Although the UNC address looks similar to a URL the two are completely different. URLs use forward slashes rather than the backslash and the initial forward slashes are preceded by the transfer protocol (ftp, http) and a colon.

WWW Technology: HTML, DHTML, WML, XML:

HTML:

HTML stands for **hypertext markup language**. It is not a programming language. A markup language specifies the *layout and style* of a document. A markup language consists of a set of **markup tags**. HTML uses markup tags to describe web pages. HTML tags are keywords surrounded by **angle brackets** like <html>. Most HTML tags normally come in pairs like and . The first tag is called the **start tag** (or **opening tag**) and the second tag is called the **end tag** (or **closing tag**). HTML documents describe Web pages. HTML documents contain HTML tags and plain text. HTML documents are also called Web pages. A web browser read HTML documents and displays them as Web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. A simple HTML document is given below:

```
<html>
    <head>
        <title>This is my first web page</title>
    </head>
    <body>
        <h1>My first heading</h1>
        <p>My first paragraph</p>
    </body>
</html>
```

Save this page with **.html** or **.htm** extension. However, it is good practice to use **.htm** extension.

HTML Elements

HTML documents are defined by HTML elements. An HTML element is everything from the start tag to the end tag. For example, My first paragraph . An HTML element consists of start tag, end tag, and element content. The element content is everything between the start tag and end tag. Empty elements are closed in the start tag. Most HTML elements can have attributes. For example, **src** attribute of **img** tag.

HTML Attributes

Attributes provide additional information about HTML elements. Attributes are always specified in the start tag. Attributes come in name/value pair like name = "value". For example, HTML links are defined with **<a>** tag and the link address is provided as an attribute **href** like **cdcsit**.
Note: Always quote attribute values and use lowercase attributes.

HTML Headings

HTML headings are defined with the <h1> to <h6> tags. <h1> displays largest text and <h6> smallest. For example, <h1>My first heading</h1>.

HTML Paragraphs

HTML paragraphs are defined with tag. For example, My first paragraph .

HTML Rules (Lines)

We use <hr /> tag to create horizontal line.

HTML Comments

We use comments to make our HTML code more readable and understandable. Comments are ignored by the browser and are not displayed. Comments are written between <!-- and -->. For example, <!-- This is a comment -->.

HTML Line Breaks

If you want a new line (line break) without starting a new paragraph, use <br /> tag.

HTML Formatting Tags

We use different tags for formatting output. For example, <b> is used for bold and <i> is used for italic text. Some other tags are <big>, <small>, <sup>, <sub> etc.

HTML Styles

It is a new HTML attribute. It introduces CSS to HTML. The purpose of style attribute is to provide a common way to style all HTML elements. For example, <body style =

“background-color:yellow”>, <p style = “font-family:courier new; color:red; font-size:20px”>, <h1 style = “text-align:center”> etc.

(Have a review from Web technology !!!)

DHTML:

DHTML stands for **D**ynamic **HTML**. **DHTML is the art of combining HTML, JavaScript, DOM, and CSS** and is NOT a language or a web standard. According to the World Wide Web Consortium (W3C): ***"Dynamic HTML is a term used by some vendors to describe the combination of HTML, style sheets and scripts that allows documents to be animated."***

The W3C HTML 4 standard has rich support for dynamic content:

- HTML supports JavaScript
- HTML supports the Document Object Model (**DOM**)
- HTML supports HTML Events
- HTML supports Cascading Style Sheets (**CSS**)

DHTML is about using these features, to create dynamic and interactive web pages. DHTML allows authors to add effects to their pages without the overhead of server-side programs or complicated sets of controls to achieve special effects.For example, DHTML allows the page author to:

- Animate text and images in their document, independently moving each element from any starting point to any ending point, following a predetermined path or one chosen by the user.
- Embed a ticker that automatically refreshes its content with the latest news, stock quotes, or other data.
- Use a form to capture user input, and then process and respond to that data without having to send data back to the server.

- Include rollover buttons or drop-down menus.

The **term "DHTML" has fallen out of use in recent years as it was associated with practices and conventions that tended to not work well between various web browsers.** DHTML may now be referred to as **unobtrusive JavaScript coding** (DOM Scripting), in an effort to place an emphasis on agreed-upon best practices while allowing similar effects in an accessible, standards-compliant way. **The unobtrusive JavaScript includes separation of of functionality (the "behavior layer") from a Web page's structure/content and presentation.** Best practices to avoid the problems of traditional JavaScript programming. In traditional javascript we write as;

<input type="text" name="date" onchange="validateDate()" />

The unobtrusive solution is to register the necessary event handlers programmatically, rather than inline. Rather than adding the onchange attribute explicitly as above, the relevant element(s) are simply identified, for example by class, id or some other means in the markup:
<input type="text" name="date" id="date" />

A script that runs when the page is first loaded into the browser can then look for the relevant element(s) and set them up accordingly:

```
window.onload = function() {
   document.getElementById('date').onchange = validateDate;
};
```

Few examples;

1.

```
<!DOCTYPE html>
<html>
<body>

<h1 onclick="this.style.color='red'">Click Me!</h1>

</body>
</html>
```

2.

```
<!DOCTYPE html>
<html>
<head>
```

```
<script type="text/javascript">
function mymessage()
{
alert("This message was triggered from the onload event");
}
</script>
</head>

<body onload="mymessage()">
</body>

</html>
```

WML:

Wireless Markup Language (WML), based on XML, is a markup language intended for devices that implement the Wireless Application Protocol (WAP) specification, such as mobile phones. It provides navigational support, data input, hyperlinks, text and image presentation, and forms, much like HTML (HyperText Markup Language). It preceded the use of other markup languages now used with WAP, such as HTML itself, and XHTML (which are gaining in popularity as processing power in mobile devices increases).

WML Decks and Cards:

A main difference between HTML and WML is that the basic unit of navigation in HTML is a page, while that in WML is a card. A WML file can contain multiple cards and they form a deck.

When a WML page is accessed from a mobile phone, all the cards in the page are downloaded from the WAP server. So if the user goes to another card of the same deck, the mobile browser does not have to send any requests to the server since the file that contains the deck is already stored in the wireless device.

You can put links, text, images, input fields, option boxes and many other elements in a card.

A WML document is known as a “deck”. Data in the deck is structured into one or more “cards” (pages) – each of which represents a single interaction with the user.

A WML program is typically divided into two parts: the document prolog and the body. Consider the following code:

Following is the basic structure of a WML program:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

    <card id="one" title="First Card">
    <p>
    This is the first card in the deck

    </p>
    </card>

    <card id="two" title="Second Card">
    <p>
    Ths is the second card in the deck
    </p>
    </card>

</wml>
```

WML Document Prolog:

The first line of this text says that this is an XML document and the version is 1.0. The second line selects the document type and gives the URL of the document type definition (DTD). The DTD referenced is defined in WAP 1.2, but this header changes with the versions of the WML. The header must be copied exactly so that the tool kits automatically generate this prolog.

The prolog components are not WML elements and they should not be closed, i.e. you should not give them an end tag or finish them with />.

WML Document Body:

The body is enclosed within a <wml> </wml> tag pair. The body of a WML document can consist of one or more of the following:

- Deck
- Card
- Content to be shown
- Navigation instructions

Unlike HTML 4.01 Transitional, text cannot be enclosed directly in the <card>...</card> tag pair. So you need to put a content inside ... as shown above.

WBXML:

WAP Binary XML (**WBXML**) is a binary representation of XML. It was developed by the WAP Forum and is now maintained by the Open Mobile Alliance as a standard to allow XML documents to be transmitted in a compact manner over mobile networks and proposed as an addition to the World Wide Web Consortium's Wireless Application Protocolfamily of standards. The MIME media type application/vnd.wap.wbxml has been defined for documents that use WBXML.

WBXML is used by a number of mobile phones. Usage includes SyncML (Synchronization markup language) for transmitting address book and calendar data, Wireless Markup Language

WML Script:

WMLScript is the dialect of **JavaScript** used for WML pages and is part of the Wireless Application Protocol (WAP). WMLScript is a client-side scripting language and is very similar to JavaScript. Just like JavaScript WMLScript is used for tasks such as user input validation, generation of error message and other Dialog boxes etc.

WMLScript is based on **ECMAScript** (European Computer Manufacturers Association Script), which is JavaScript's standardized version. Thus the syntax of WMLScript is very similar to JavaScript but not fully compatible. A major difference between JavaScript and WMLScript is that JavaScript code can be embedded in the HTML markup, whereas WMLScript code is always placed in a file separate from the WML markup. URLs are used to refer to the actual WMLScript code in the WML document.

The following "Hello World" WMLScript example shows you how a WMLScript file typically looks like and demonstrates how to call WMLScript code in a WML document.

(helloWorldEg1.wml)

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">

<wml>
```

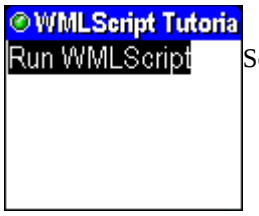
```
  <card id="card1" title="WMLScript Tutorial">
    <p>
      <a href="helloWorldEg1.wmls#helloWorld()">Run WMLScript</a><br/>
      $(message)
    </p>
  </card>
</wml>
```

Here is the file that contains the WMLScript code:

(helloWorldEg1.wmls)

```
extern function helloWorld()
{
  WMLBrowser.setVar("message", "Hello World. Welcome to our WMLScript tutorial.");
  WMLBrowser.refresh();
}
```

Open the *helloWorldEg1.wml* file in a mobile phone browser and you can see something like this:



Sony Ericsson T68i

Nokia Mobile Browser 4.0

If you select the "Run WMLScript" link, the WMLScript function *helloWorld()* is executed and the line "Hello World. Welcome to our WMLScript tutorial." will appear in the mobile phone browser.

Sony Ericsson T68i

Nokia Mobile Browser 4.0

In the above example, the WMLScript code is not embedded in the WML markup and they are kept in separate files. This is the rule of WMLScript and you need to follow this when programming mobile Internet browsing applications.

There is only one function, *helloWorld()*, in the WMLScript file. The *extern* keyword is used to specify that the *helloWorld()* function is allowed to be called from outside the WMLScript file*helloWorldEg1.wmls*. The *extern* keyword is necessary here since we want to call the function from the WML file *helloWorldEg1.wml*.

Inside the *helloWorld()* function, we use two functions of the WMLBrowser standard library, *setVar()* and*refresh()*. The *setVar()* function is used to assign a value to a WML variable. We use the WMLScript code:

WMLBrowser.setVar("message", "Hello World. Welcome to our WMLScript tutorial.");

- to assign the value "Hello World. Welcome to our WMLScript tutorial." to a WML variable named*message*.

The *refresh()* function is used to instruct the WAP browser to refresh the current WML card. In the*helloWorld()* function, after we have assigned a value to the *message* variable, we make use of the line:

WMLBrowser.refresh();

- to refresh the WML card so that the change made to the *message* variable is shown on the screen of the mobile device.

To call the WMLScript function *helloWorld()* in the WML document, we use the URL below:

helloWorldEg1.wmls#helloWorld()

helloWorldEg1.wmls is the file that contains the WMLScript code and *helloWorld()* is the function to call.

XML:

XML is designed to transport and store data. XML stands for EXtensible Markup Language and is much like HTML. XML was designed to carry data, not to display data. XML tags are not predefined. You must define your own tags. XML is designed to be self-descriptive. **Extensible Markup Language** (**XML**) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

XML is not a replacement for HTML. XML and HTML were designed with different goals:

- XML was designed to transport and store data, with focus on what data is
- HTML was designed to display data, with focus on how data looks

HTML is about displaying information, while XML is about carrying information.

Maybe it is a little hard to understand, but XML does not DO anything. XML was created to structure, store, and transport information. The following example is a note to Tulsi, from Giri, stored as XML:

<note>
<to>Tulsi</to>
<from>Giri</from>
<heading>Reminder</heading>
<body>Don't forget to bunk web tech class at Patan!</body>
</note>

The note above is quite self descriptive. It has sender and receiver information, it also has a heading and a message body. But still, this XML document does not DO anything. It is just information wrapped in tags. Someone must write a piece of software to send, receive or display it.

The tags in the example above (like <to> and <from>) are not defined in any XML standard. These tags are "invented" by the author of the XML document. That is because the XML language has no predefined tags. However, the tags used in HTML are predefined. HTML documents can only use tags defined in the HTML standard (like <p>, <h1>, etc.). In contrast, XML allows the author to define his/her own tags and his/her own document structure. The XML processor can not tell us which elements and attributes are valid. As a result we need to define the XML markup we are using. To do this, we need to define the markup language's grammar. There are numerous "tools" that can be used to build an XML language – some relatively simple, some much more complex. They include DTD (Document Type Definition), RELAX, TREX, RELAX NG, XML Schema, Schmatron, etc.

The design goals for XML are:

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs which process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.

XML Usages

XML is used in many aspects of web development, often to simplify data storage and sharing.

XML Separates Data from HTML: If you need to display dynamic data in your HTML document, it will take a lot of work to edit the HTML each time the data changes. With XML, data can be stored in separate XML files. This way you can concentrate on using HTML for layout and display, and be sure that changes in the underlying data will not require any changes to the HTML. With a few lines of JavaScript code, you can read an external XML file and update the data content of your web page.

XML Simplifies Data Sharing: In the real world, computer systems and databases contain data in incompatible formats. XML data is stored in plain text format. This provides a software- and hardware-independent way of storing data. This makes it much easier to create data that can be shared by different applications.

XML Simplifies Data Transport: One of the most time-consuming challenges for developers is to exchange data between incompatible systems over the Internet. Exchanging data as XML greatly reduces this complexity, since the data can be read by different incompatible applications.

XML Simplifies Platform Changes: Upgrading to new systems (hardware or software platforms), is always time consuming. Large amounts of data must be converted and incompatible data is often lost. XML data is stored in text format. This makes it easier to expand or upgrade to new operating systems, new applications, or new browsers, without losing data.

XML Makes Your Data More Available: Different applications can access your data, not only in HTML pages, but also from XML data sources. With XML, your data can be available to all kinds of "reading machines" (Handheld computers, voice machines, news feeds, etc), and make it more available for blind people, or people with other disabilities.

XML Used to Create New Internet Languages: A lot of new Internet languages are created with XML. Here are some examples:

- XHTML
- WSDL (Web Services Description Language) for describing available web services
- WAP and WML (Wireless Markup Language) as markup languages for handheld devices
- RSS (Really Simple Syndication / Rich Site Summary) languages for news feeds
- RDF (Resource Description Framework), a family of w3c spec, and OWL (Web Ontology Language) for describing resources and ontology
- SMIL (Synchronized Multimedia Integration Language) for describing multimedia for the web

XML Tree

XML documents form a tree structure that starts at "the root" and branches to "the leaves". XML documents use a self-describing and simple syntax:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
        <to>Tulsi</to>
        <from>Giri</from>
         <heading>Reminder</heading>
        <body>Don't forget to bunk the web tech class at Patan!</body>
</note>
```

The first line is the XML declaration. It defines the XML version (1.0) and the encoding used (ISO-8859-1 = Latin-1/West European character set). The next line describes the **root element** of the document (like saying: "this document is a note"):

<note>

The next 4 lines describe 4 **child elements** of the root (to, from, heading, and body):

<to>Tulsi</to>
<from>Giri</from>
<heading>Reminder</heading>
<body>Don't forget to bunk the web tech class at Patan!</body>

And finally the last line defines the end of the root element:

</note>

You can assume, from this example, that the XML document contains a note to Tulsi from Giri.

Thus, XML documents must contain a **root element**. This element is "the parent" of all other elements. The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree. All elements can have sub elements (child elements):

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

The terms parent, child, and sibling are used to describe the relationships between elements. Parent elements have children. Children on the same level are called siblings (brothers or sisters). All elements can have text content and attributes (just like in HTML).

XML Syntax Rules

The syntax rules of XML are very simple and logical. The rules are easy to learn, and easy to use.

1. **All XML Elements Must Have a Closing Tag**. In HTML, some elements may not have to have a closing tag, like;

   ```
   <p>This is a paragraph.
   <br>
   ```

   In XML, it is illegal to omit the closing tag. All elements must have a closing tag:

   ```
   <p>This is a paragraph.</p>
   <br />
   <hello> This is hello </hello>
   ```

2. **XML tags are case sensitive**. The tag <Letter> is different from the tag <letter>. Opening and closing tags must be written with the same case:

   ```
   <Message>This is incorrect</message>
   <message>This is correct</message>
   ```

3. **XML Elements Must be Properly Nested**. In HTML, you might see improperly nested elements:

   ```
   <b><i>This text is bold and italic</b></i>
   ```

   In XML, all elements must be properly nested within each other:

   ```
   <b><i>This text is bold and italic</i></b>
   ```

4. **XML Documents Must Have a Root Element**. XML documents must contain one element that is the **parent** of all other elements. This element is called the **root** element.

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

5. **XML Attribute Values Must be Quoted.** XML elements can have attributes in name/value pairs just like in HTML. In XML, the attribute values must always be quoted. Study the two XML documents below. The first one is incorrect, the second is correct:

```
<note date=06/01/2012>
  <to>Tulsi</to>
  <from>Giri</from>
</note>

<note date="06/01/2012">
  <to>Tulsi</to>
  <from>Giri</from>
</note>
```

   The error in the first document is that the date attribute in the note element is not quoted.

6. **Entity Reference.** Some characters have a special meaning in XML. If you place a character like "<" inside an XML element, it will generate an error because the parser interprets it as the start of a new element. This will generate an XML error:

```
<message>if salary < 1000 then</message>
```

   To avoid this error, replace the "<" character with an **entity reference**:

```
<message>if salary &lt; 1000 then</message>
```

   There are 5 predefined entity references in XML:

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

7. **Comments in XML.** The syntax for writing comments in XML is similar to that of HTML.

```
<!-- This is a comment -->
```

8. White-space is preserved in XML**. HTML truncates multiple white-space characters to one single white-space:**

```
HTML: Hello        Tulsi
Output: Hello Tulsi
```

With XML, the white-space in a document is not truncated.

XML Elements

An XML document contains XML Elements. An XML element is everything from (including) the element's start tag to (including) the element's end tag.

An element can contain:

- other elements
- text
- attributes
- or a mix of all of the above...

Consider an example;

```
<bookstore>

 <book category="CHILDREN">
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
 </book>

 <book category="WEB">
  <title>Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
 </book>

</bookstore>
```

In the example above, <bookstore> and <book> have **element contents**, because they contain other elements. <book> also has an **attribute** (category="CHILDREN"). <title>, <author>, <year>, and <price> have **text content** because they contain text.

XML Elements are Extensible

XML elements can be extended to carry more information. Look at the following XML example:

```
<note>
<to>Tulsi</to>
<from>Giri</from>
<body>Don't forget to bunk the web tech class at Patan!</body>
</note>
```

Let's imagine that we created an application that extracted the <to>, <from>, and <body> elements from the XML document to produce this output:

MESSAGE

To: Tulsi
From: Giri

Don't forget to bunk the web tech class at Patan!

Suppose the XML document has been modified by adding some extra information to it like:

```
<note>
<date>2012-01-06</date>
<to>Tulsi</to>
<from>Giri</from>
<heading>Reminder</heading>
<body>Don't forget to bunk thee web tech class at Patan!</body>
</note>
```

Should the application break or crash?

No. The application should still be able to find the <to>, <from>, and <body> elements in the XML document and produce the same output. Thus, one of the beauties of XML, is that it can be extended without breaking applications.

(Have a review from Web Tech !!!)

WYSWIG Authoring Tool:

The somehow cryptic abbreviation WYSIWYG stands for “What You See Is What You Get”. In such editors you edit not directly the source code of your documents, but its **presentation** as it (hopefully) will appear in the final document. So instead of writing blocks of code manually (as you e.g. would do it in Word or Latex), you manipulate with design components using an editor window. This means that you view something very similar to the end result while the document or image is being created. WYSIWYG (What You See Is What You Get) code generators offer speed and ease of use. Many of these editors do not require any knowledge of the programming languages generated by the software.

Some of these editors store pages in a proprietary format and then export them as HTML (possibly along with other formats); the user would continue to maintain the website by working with the files in the proprietary format and re-exporting them. Other, generally simpler WYSIWYG editors are designed to work directly with HTML files. Exported files tend to be larger than hand-coded pages (those produced with a text-based HTML editor or a plain text editor). WYSIWYG generators tend to be better than word processors at producing highly graphical and interactive pages. Some of the WYSIWYG tools are

- ASP.NET Web Matrix
- Adobe Dreamweaver (formerly Macromedia Dreamweaver)
- Amaya
- Microsoft Visual Studio
- Microsoft Visual Web Developer Express

A given HTML document will have an inconsistent appearance on various platforms and computers for several reasons:

Different browsers and applications will render the same markup differently: The same page may display slightly differently in Internet Explorer and Firefox on a high-resolution screen, but it will look very different in the perfectly valid text-only Lynx browser. It needs to be rendered differently again on a PDA, an internet-enabled television and on a mobile phone. Usability in a speech or braille browser, or via a screen-reader working with a conventional browser, will place demands on entirely different aspects of the underlying HTML. Printing the page, via different browsers and different printers onto various paper sizes, around the world, places other demands. With the correct use of modern HTML and CSS there is no longer any need to provide 'Printable page' links and then have to maintain two versions of the whole site. Nor is there any excuse for pages not fitting the user's preferred paper size and orientation, or wasting ink printing solid background colours unnecessarily, or wasting paper reproducing navigation panels that will be entirely useless once printed out.

Browsers and computer graphics systems have a range of user settings: Resolution, font size, colour, contrast etc can all be adjusted at the user's discretion, and many modern browsers allow even more user control over page appearance. All an author can do is suggest an appearance.

Web browsers, like all computer software, have bugs: They may not conform to current standards. It is hopeless to try to design Web pages around all of the common browsers' current bugs: each time a new version of each browser comes out, a significant proportion of the World Wide Web would need re-coding to suit the new bugs and the new fixes. It is generally considered much wiser to design to standards, staying away from 'bleeding edge' features until they settle down, and then wait for the browser developers to catch up to your pages, rather than the other way round.[9] For instance, no one can argue that CSS is still 'cutting edge' as there is now widespread support available in common browsers for all the major features,[10] even if many WYSIWYG and other editors have not yet entirely caught up.[11]

A single visual style can represent multiple semantic meanings: Semantic meaning, derived from the underlying structure of the HTML document, is important for search engines and also for various accessibility tools. On paper we can tell from context and experience whether bold text represents a title, or emphasis, or something else. But it is very difficult to convey this distinction in a WYSIWYG editor. Simply making a piece of text bold in a WYSIWYG editor is not sufficient to tell the editor *why* the text is bold - what the boldness represents semantically. What you see may be what most visitors get, but it is not guaranteed to be what *everyone* gets.

WYSIWYM (what you see is what you mean) is an alternative paradigm to WYSIWYG, in which the focus is on the semantic structure of the document rather than on the presentation. These editors produce more logically structured markup than is typical of WYSIWYG editors, while retaining the advantage in ease of use over hand-coding using a text editor. WYMeditor is such editor.

Helper applications:

CGI: Abbreviation of ***C****ommon* ***G****ateway* ***I****nterface*, a specification for transferring information between a World Wide Web server and a CGI program. A CGI program is any program designed to accept and return data that conforms to the CGI specification. The program could be written in any programming language, including C, Perl, Java, or Visual Basic. CGI programs are the most common way for Web servers to interact dynamically with users. Many HTML pages that contain forms, for example, use a CGI program to process the form's data once it's submitted. Another increasingly common way to provide dynamic feedback for Web users is to include scripts or programs that run on the user's machine rather than the Web server. These programs can be Java applets, Java scripts, or ActiveX controls. These technologies are known collectively as *client-side* solutions, while the use of CGI is a *server-side* solution because the processing occurs on the Web server.

One problem with CGI is that each time a CGI script is executed, a new process is started. For busy Web sites, this can slow down the server noticeably. A more efficient solution, but

one that it is also more difficult to implement, is to use the server's API, such as ISAPI or NSAPI. Another increasingly popular solution is to use Java servlets.

CGI is the Common Gateway Interface, a standard for programs to interface with information servers such as HTTP (web) servers. CGI allows the HTTP server to run an executable program or script in response to a user request, and generate output on the fly. This allows web developers to create dynamic and interactive web pages. Perl is a very common language for CGI programming as it is largely platform independent and the language's features make it very easy to write powerful applications.

It is important to remember that CGI is not a language in itself. CGI is merely a type of program which can be written in any language.

Perl:

Perl is a high-level, general-purpose, interpreted, dynamic programming language. Though Perl is not officially an acronym, there are various backronyms in usage, such as: *Practical Extraction and Reporting Language*. Perl was originally developed by Larry Wall in 1987 as a general-purpose Unix scripting language to make report processing easier. Since then, it has undergone many changes and revisions and become widely popular amongst programmers.

Larry Wall continues to oversee development of the core language, and its upcoming version, Perl 6. Perl borrows features from other programming languages including C, shell scripting (sh), AWK, and sed. The language provides powerful text processing facilities without the arbitrary data length limits of many contemporary Unix tools, facilitating easy manipulation of text files. Perl gained widespread popularity in the late 1990s as a CGI scripting language, in part due to its parsing abilities.

In addition to CGI, Perl is used for graphics programming, system administration, network programming, finance, bioinformatics, and other applications. Its major features include support for multiple programming paradigms (procedural, object-oriented, and functional styles), reference counting memory management (without a cycle-detecting garbage collector), built-in support for text processing, and a large collection of third-party modules.

Simple Perl Statements

```
print "Hello, world!\n"
say "Hello, world!"

$n = '3 apples';
$m = '2 oranges';
print $n + $m;
```

PHP: Will provide separate handout for PHP.

Note: *Java, JavaScript, Asp, .Net Appications* (Self Study… !!!)

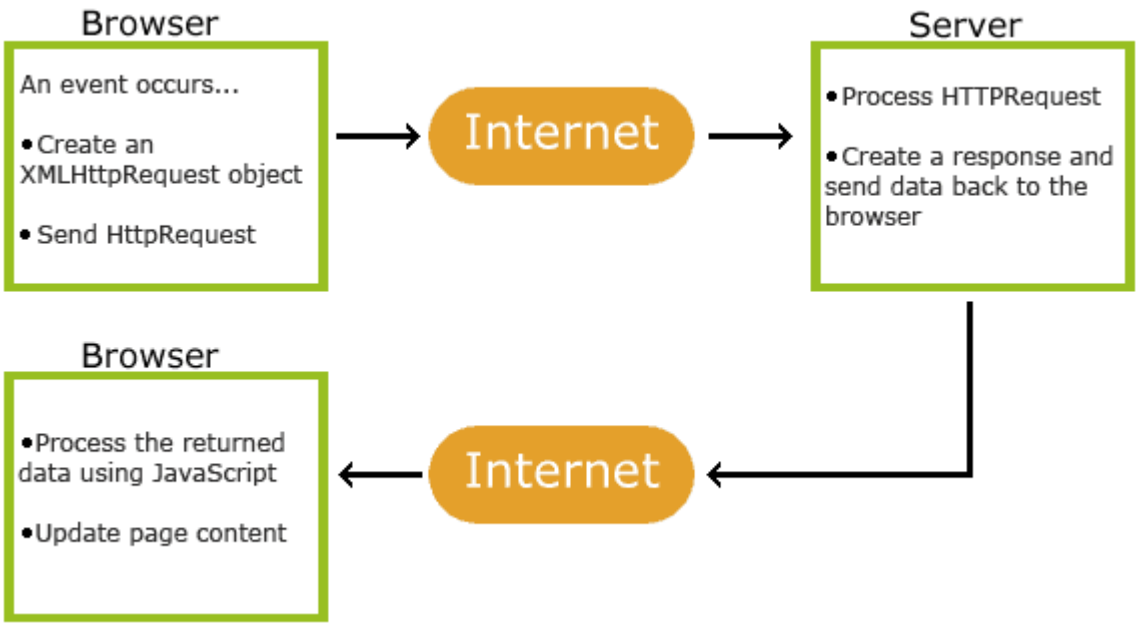
Introduction to Ajax:

Ajax (**Asynchronous JavaScript and XML**) is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Data can be retrieved using the XMLHttpRequest object. Despite the name, the use of XML is not required (JSON is often used instead, **JSON** (or **JavaScript Object Notation**, is a text-based open standard designed for human-readable data interchange. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for many languages), and the requests do not need to be asynchronous.

Ajax is not a single technology, but a group of technologies. HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display, and to allow the user to interact with the information presented. JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

n the 1990s, most web sites were based on complete HTML pages; each user action required that the page be re-loaded from the server (or a new page loaded). This process is inefficient, as reflected by the user experience: all page content disappears then reappears, etc. Each time a page is reloaded due to a partial change, all of the content must be re-sent instead of only the changed information. This can place additional load on the server and use excessive bandwidth. Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.

How AJAX Works



AJAX is based on internet standards, and uses a combination of:

- XMLHttpRequest object (to exchange data asynchronously with a server)
- JavaScript/DOM (to display/interact with the information)
- CSS (to style the data)
- XML (often used as the format for transferring data)

AJAX applications are browser and platform-independent. The keystone of AJAX is the XMLHttpRequest object.

Steps of AJAX Operation

1. A client event occurs
2. An XMLHttpRequest object is created
3. The XMLHttpRequest object is configured
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the callback() function and processes the result.
7. The HTML DOM is updated

The XMLHttpRequest Object

All modern browsers support the XMLHttpRequest object (IE5 and IE6 use an ActiveXObject). The XMLHttpRequest object is used to exchange data with a server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page. All modern browsers (IE7+, Firefox, Chrome, Safari, and Opera) have a built-in XMLHttpRequest object.

Syntax for creating an XMLHttpRequest object:

***variable*=new XMLHttpRequest();**

Old versions of Internet Explorer (IE5 and IE6) uses an ActiveX Object:

***variable*=new ActiveXObject("Microsoft.XMLHTTP");**

To handle all modern browsers, including IE5 and IE6, check if the browser supports the XMLHttpRequest object. If it does, create an XMLHttpRequest object, if not, create an ActiveXObjec

The XMLHttpRequest object is used to exchange data with a server.

Send a Request to a Server:

To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object:

```
xmlhttp.open("GET","ajax_info.txt",true);
xmlhttp.send();
```

Method	Description
open(*method*,*url*,*async*)	Specifies the type of request, the URL, and if the request should be handled asynchronously or not. *method*: the type of request: GET or POST *url*: the location of the file on the server *async*: true (asynchronous) or false (synchronous)
send(*string*)	Sends the request off to the server. *string*: Only used for POST requests

GET or POST?

GET is simpler and faster than POST, and can be used in most cases. However, always use POST requests when:

- A cached file is not an option (update a file or database on the server)

- Sending a large amount of data to the server (POST has no size limitations)
- Sending user input (which can contain unknown characters), POST is more robust and secure than GET

GET Requests

A simple GET request:

If you want to send information with the GET method, add the information to the URL:

```
xmlhttp.open("GET","demo_get2.asp?fname=Henry&lname=Ford",true);
xmlhttp.send();
```

The url - A File On a Server

The url parameter of the open() method, is an address to a file on a server:

```
xmlhttp.open("GET","ajax_test.asp",true);
```

The file can be any kind of file, like .txt and .xml, or server scripting files like .asp and .php (which can perform actions on the server before sending the response back).

Asynchronous - True or False?

AJAX stands for Asynchronous JavaScript and XML, and for the XMLHttpRequest object to behave as AJAX, the async parameter of the open() method has to be set to true:

```
xmlhttp.open("GET","ajax_test.asp",true);
```

Sending asynchronous requests is a huge improvement for web developers. Many of the tasks performed on the server are very time consuming. Before AJAX, this operation could cause the application to hang or stop.

With AJAX, the JavaScript does not have to wait for the server response, but can instead:

- execute other scripts while waiting for server response
- deal with the response when the response ready

Async=true

When using async=true, specify a function to execute when the response is ready in the onreadystatechange event:

Example

```
xmlhttp.onreadystatechange=function()
  {
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
    document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
    }
  }
xmlhttp.open("GET","ajax_info.txt",true);
xmlhttp.send();
```

Async=false

To use async=false, change the third parameter in the open() method to false:

```
xmlhttp.open("GET","ajax_info.txt",false);
```

Using async=false is not recommended, but for a few small requests this can be ok.

Remember that the JavaScript will NOT continue to execute, until the server response is ready. If the server is busy or slow, the application will hang or stop.

Note: When you use async=false, do NOT write an onreadystatechange function - just put the code after the send() statement:

Example

```
xmlhttp.open("GET","ajax_info.txt",false);
xmlhttp.send();
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
```

Server Response

To get the response from a server, use the responseText or responseXML property of the XMLHttpRequest object.

Property	Description
responseText	get the response data as a string
responseXML	get the response data as XML data

The responseText Property

If the response from the server is not XML, use the responseText property. The responseText property returns the response as a string, and you can use it accordingly:

Example

```
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
```

The responseXML Property

If the response from the server is XML, and you want to parse it as an XML object, use the responseXML property:

Request the file cd_catalog.xml and parse the response:

```
xmlDoc=xmlhttp.responseXML;
txt="";
x=xmlDoc.getElementsByTagName("ARTIST");
for (i=0;i<x.length;i++)
  {
  txt=txt + x[i].childNodes[0].nodeValue + "<br />";
  }
document.getElementById("myDiv").innerHTML=txt;
```

The onreadystatechange event

When a request to a server is sent, we want to perform some actions based on the response. The onreadystatechange event is triggered every time the readyState changes. The readyState property holds the status of the XMLHttpRequest. Three important properties of the XMLHttpRequest object:

Property	Description
onreadystatechange	Stores a function (or the name of a function) to be called automatically each time the readyState property changes
readyState	Holds the status of the XMLHttpRequest. Changes from 0 to 4: 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready

status	200: "OK" 404: Page not found

In the onreadystatechange event, we specify what will happen when the server response is ready to be processed.

When readyState is 4 and status is 200, the response is ready:

Example

```
xmlhttp.onreadystatechange=function()
  {
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
    document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
    }
  }
```

Note: The onreadystatechange event is triggered four times, one time for each change in readyState.

Using a Callback Function:

A callback function is a function passed as a parameter to another function. If you have more than one AJAX task on your website, you should create ONE standard function for creating the XMLHttpRequest object, and call this for each AJAX task.

The function call should contain the URL and what to do on onreadystatechange (which is probably different for each call):

Example

```
function myFunction()
{
loadXMLDoc("ajax_info.txt",function()
  {
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
    document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
    }
  });
}
```

Ajax framework:

In web application development, an **Ajax framework** is a framework which leverages Ajax, a collection of technologies for building dynamic web pages on the client side. While data is read from and sent to the server by JavaScript requests, frameworks may include server-side or client-side components to process the client's requests.

Some of the frameworks are JavaScript compilers, for generating JavaScript and Ajax that runs in the web browser client; some are pure JavaScript libraries; others are server-side frameworks that typically utilize JavaScript libraries. For examples; jquery Moo Tools etc.

Some Examples

```
<!DOCTYPE html>
<html>
<head>
        <script type="text/javascript">
        function loadXMLDoc()
        {
        var xmlhttp;
        if (window.XMLHttpRequest)
          {// code for IE7+, Firefox, Chrome, Opera, Safari
          xmlhttp=new XMLHttpRequest();
          }
        else
          {// code for IE6, IE5
          xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
          }
        xmlhttp.onreadystatechange=function()
          {
          if (xmlhttp.readyState==4 && xmlhttp.status==200)
            {
            document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
            }
          }
        xmlhttp.open("GET","ajax_info.txt",true);
        xmlhttp.send();
        }
        </script>
</head>
<body>
        <div id="myDiv"><h2>Let AJAX change this text</h2></div>
        <button type="button" onclick="loadXMLDoc()">Change Content</button>
</body>
</html>
```

One more example;

It consist of a textbox. When a user types a character in the input field above, the function "showHint()" is executed. The function is triggered by the "onkeyup" event:

```
function showHint(str)
{
var xmlhttp;
if (str.length==0)
  {
  document.getElementById("txtHint").innerHTML="";
  return;
  }
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.onreadystatechange=function()
  {
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
    document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
    }
  }
xmlhttp.open("GET","gethint.asp?q="+str,true);
xmlhttp.send();
}
```

If the input field is empty (str.length==0), the function clears the content of the txtHint placeholder and exits the function.

If the input field is not empty, the showHint() function executes the following:

- Create an XMLHttpRequest object
- Create the function to be executed when the server response is ready
- Send the request off to a file on the server
- Notice that a parameter (q) is added to the URL (with the content of the input field)

The page on the server called by the JavaScript above is an ASP file called "gethint.asp". The file is as;

```
<%
response.expires=-1
dim a(10)
'Fill up array with names
```

```
a(1)="Anna"
a(2)="Brittany"
a(3)="Cinderella"
a(4)="Diana"
a(5)="Eva"
a(6)="Fiona"
a(7)="Gunda"
a(8)="Hege"
a(9)="Inga"
a(10)="Johanna"

q=ucase(request.querystring("q"))       'get the q parameter from URL

if len(q)>0 then                        'lookup all hints from array if length of q>0
  hint=""
  for i=1 to 10
    if q=ucase(mid(a(i),1,len(q))) then
      if hint="" then
        hint=a(i)
      else
        hint=hint & " , " & a(i)
      end if
    end if
  next
end if

          'Output "no suggestion" if no hint were found or output the correct values
if hint="" then
  response.write("no suggestion")
else
  response.write(hint)
end if
%>
```

Browser as a rendering engine:

A web browser engine, (sometimes called layout engine or rendering engine), is a software component that takes marked upcontent (such as HTML, XML, image files, etc.) and formatting information (such as CSS, XSL, etc.) and displays the formatted content on the screen. It "paints" on the content area of a window, which is displayed on a monitor or a printer. **A web browser engine is typically embedded in web browsers, e-mail clients, on-line help systems or other applications that require the displaying (and editing) of web content. Engines may wait for all data to be received before rendering a page, or may begin rendering before all data is received.** This can result in

pages changing as more data is received, such as images being filled in or a flash of unstyled content if rendering begins before formatting information is received.

The browser's main components are:

1. The user interface - this includes the address bar, back/forward button, bookmarking menu etc. Every part of the browser display except the main window where you see the requested page.

2. **The browser engine - the interface for querying and manipulating the rendering engine.**

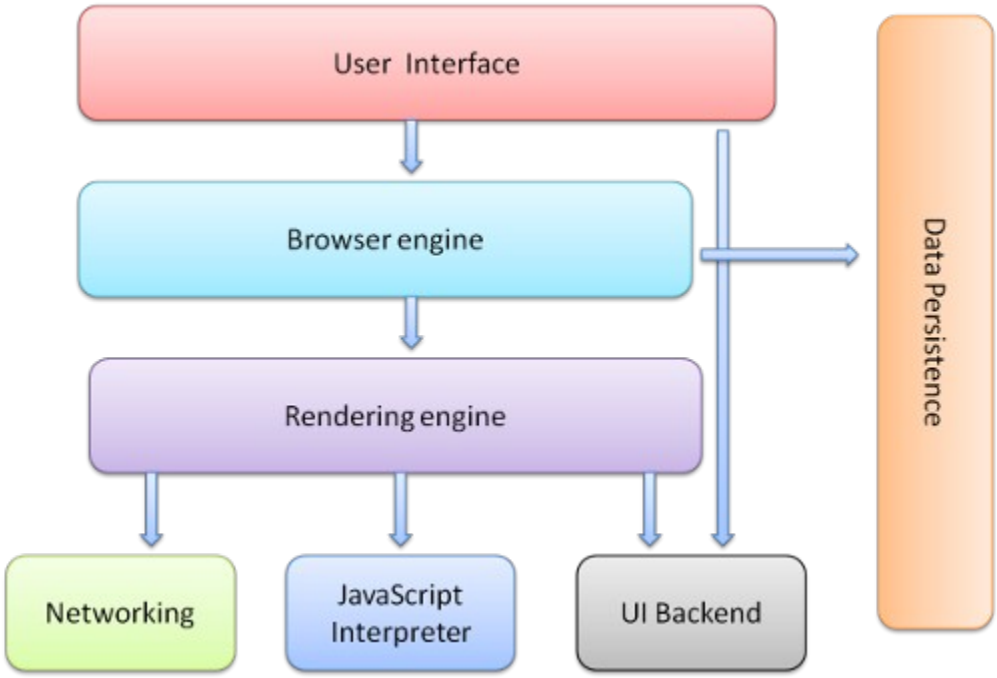
3. **The rendering engine - responsible for displaying the requested content. For example if the requested content is HTML, it is responsible for parsing the HTML and CSS and displaying the parsed content on the screen.**

4. Networking - used for network calls, like HTTP requests. It has platform independent interface and underneath implementations for each platform.

5. UI backend - used for drawing basic widgets like combo boxes and windows. It exposes a generic interface that is not platform specific. Underneath it uses the operating system user interface methods.

6. JavaScript interpreter. Used to parse and execute the JavaScript code.

7. Data storage. This is a persistence layer. The browser needs to save all sorts of data on the hard disk, for examples, cookies. The new HTML specification (HTML5) defines 'web database' which is a complete (although light) database in the browser.



Browser main components.

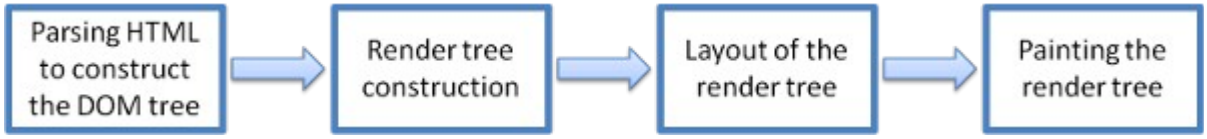
It is important to note that Chrome, unlike most browsers, holds multiple instances of the rendering engine - one for each tab, Each tab is a separate process.

The rendering engine:

The responsibility of the rendering engine, rendering, that is display of the requested contents on the browser screen. By default the rendering engine can display HTML and XML documents and images.

Our reference browsers - Firefox, Chrome and Safari are built upon two rendering engines. Firefox uses Gecko - a "home made" Mozilla rendering engine. Both Safari and Chrome use Webkit. Webkit is an open source rendering engine which started as an engine for the Linux platform and was modified by Apple to support Mac and Windows.

The rendering engine will start getting the contents of the requested document from the networking layer. This will usually be done in 8K chunks. After that this is the basic flow of the rendering engine:

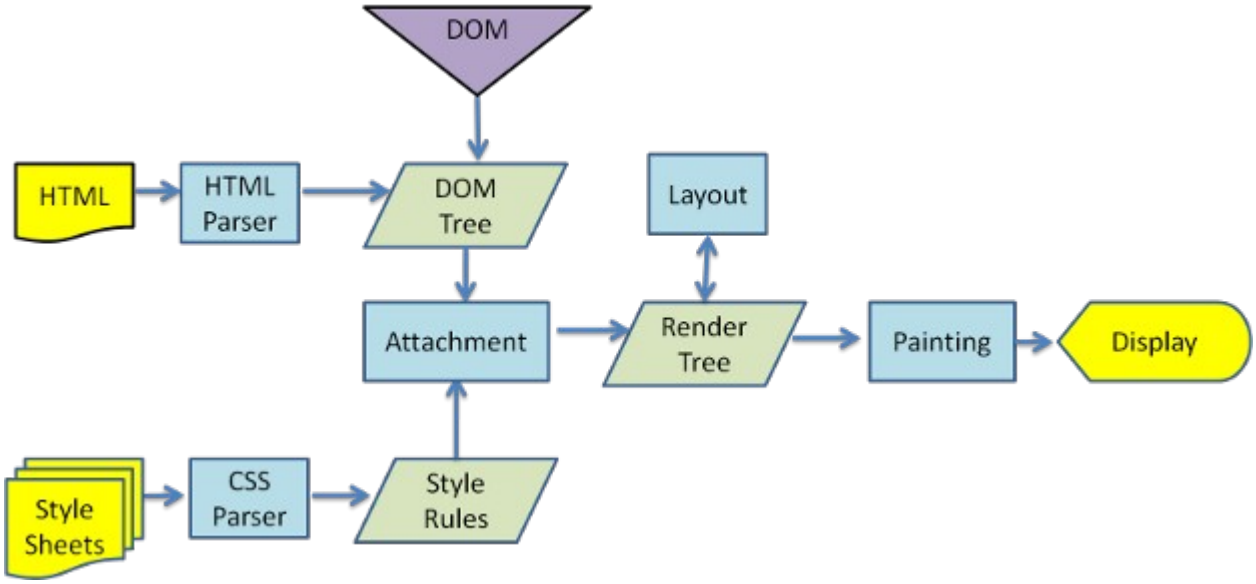


Rendering engine basic flow.

The rendering engine will start parsing the HTML document and turn the tags to DOM nodes in a tree called the "content tree". It will parse the style data, both in external CSS files and in style elements. The styling information together with visual instructions in the HTML will be used to create another tree - the render tree.

The render tree contains rectangles with visual attributes like color and dimensions. The rectangles are in the right order to be displayed on the screen. After the construction of the render tree it goes through a "layout" process. This means giving each node the exact coordinates where it should appear on the screen. The next stage is painting - the render tree will be traversed and each node will be painted using the UI backend layer.

It's important to understand that this is a gradual process. For better user experience, the rendering engine will try to display contents on the screen as soon as possible. It will not wait until all HTML is parsed before starting to build and layout the render tree. Parts of the content will be parsed and displayed, while the process continues with the rest of the contents that keeps coming from the network.



Render tree construction

While the DOM tree is being constructed, the browser constructs another tree, the render tree. This tree is of visual elements in the order in which they will be displayed. It is the visual representation of the document. The purpose of this tree is to enable painting the contents in their correct order.

Firefox calls the elements in the render tree "frames". Webkit uses the term renderer or render object. A renderer knows how to layout and paint itself and it's children.

Unit 5: (Designing Internet Systems and Servers)

Designing of Internet System Network Architecture:

The term ”network architecture” is commonly used to describe a set of abstract principles for the technical design of protocols and mechanisms for computer communication. Network architecture represents a set of deliberate choices out of many design alternatives, where these choices are informed by an understanding of the

requirements. In turn, the architecture provides a guide for the many technical decisions required to standardize network protocols and algorithms. The purpose of the architecture is provide coherence and consistency to these decisions and to ensure that the requirements are met.

Network architecture is a set of high-level design principles that guides the technical design of the network, especially the engineering of its protocols and algorithms. To flesh out this simple definition, we have examples of the constituents of the architecture and how it is applied. A network architecture must typically specify:

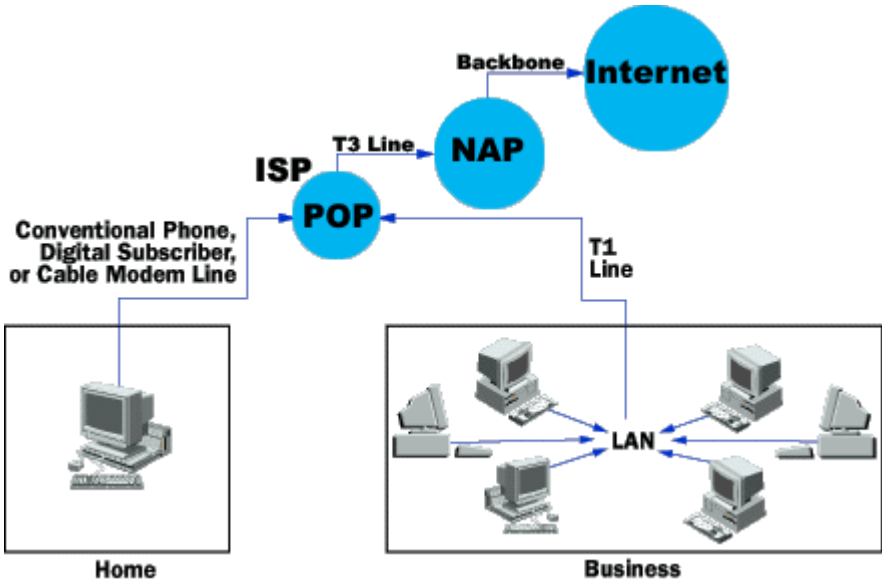
- Where and how state is maintained and how it is removed.
- What entities are named
- How naming, addressing, and routing functions inter-relate and how they are performed.
- How communication functions are modularized, e.g., into “layers” to form a “protocol stack”.
- How network resources are divided between flows and how end-systems react to this division, i.e., fairness and congestion control.
- Where security boundaries are drawn and how they are enforced.
- How management boundaries are drawn and selectively pierced.
- How differing QoS is requested and achieved.

As an example, the following list is a brief summary of the requirements underlying the original Internet architecture. This list is ordered with the most important requirements first;

- **Internetworking:** existing networks must be interconnected.
- **Robustness:** Internet communication must continue despite loss of networks or routers.
- **Heterogeneity:** The Internet architecture must accommodate a variety of network
- **Distributed management:** The Internet architecture must permit distributed management of its resources
- **Cost:** The Internet architecture must be cost effective.
- **Ease of Attachment:** The Internet architecture must permit host attachment with a low level of effort.
- **Accountability:** The resources used in the internet architecture must be accountable.

Components of Internet Network Architecture:

Internet system architecture is defined as the arrangement of different types of parts of computer or the network hardware to configure or setup the internet technology is known as internet network architecture. Different types of devices or the hardware is required to setup up the internet network architecture. It can operate with the both networks such as wired or either wirelessly.



There are lots of components that are involved in maintaining the architecture of the internet technology. Some important parts that are used to configure the networking of the internet technology are as follows

Satellite: A major part of the internet network architecture is the satellite. Satellite plays a vital role in catching and distributing the signals over the network and the users use the internet network to search different types of information at any time.

Network Adapters: There are different types of network adapters that are used to configure or setup the internet technology on your operating system. First install the network adapters in the system then install its software for the sake of its proper working or compatibility. **Some common network adapters that are used for access of the internet are LAN cards or modems etc.**

Routers: As we know that this technology is also operates wirelessly so some components that are used to configure the internet network technology wireless router plays an important role and it is also the main part of the architecture. **It is defined as the device that is used to transmit data from one place to another in the form of packets that are called as data packets is known as router.** These data packets are also called data gram.

Access Points: A special type of routing device that is used to transmit the data between wired and wireless networking device is called as AP. **It is often connected with the help of wired devices such as Ethernet.** It only transmits or transfers the data between wireless internet technology and wired internet network technology by using infra structure mode

of network. One access point can only support a small group of networks and works more efficiently. It is operated less than hundred feet. It is denoted by AP.

Clients: Any kind of device such as personal computers, Note books, or any kind of mobile devices which are inter linked with wireless network area referred as a client of internet network architecture.

Bridges: A special type of connectors which is used to establish connections between wired network devices such as Ethernet and different wireless networks such as wireless LAN. It is called as bridge. It acts as a point of control in internet network architecture.

Building Blocks of Internet Architecture

Data Formatting: In the Internet, all types of digital information are encapsulated in packets with a standard format defined by the IP protocol. At a minimum, a host needs to be able to send and receive packets in that format in order to be connected to the Internet. It further includes issues like packet encapsulation, IP header formats, packet fragmentation and reassembly.

Addressing: In internet, defining addressing schemes is next important aspect. The process portion of the address definition, called port, has been standardized as part of both TCP and UDP header formats, and the network and host portions of the address definition have been combined into one 32-bit value, called IP address, which should be globally unique. Further includes defining DNS, DHCP, Subnetting , NAT.

Dynamic Routing: Routing in the context of the Internet is about maintaining consistent forwarding tables at the routers, in accordance with the network's store and forward communication paradigm. In the early days of the Internet, routing was not a major issue because of the small number of networks connected to it. Routing within a network was often done with a private protocol and routes between networks were static and manually set up. Later, as the size of the Internet grew, dynamic routing became necessary as the topology of the network constantly changed.

The forwarding is done by matching subnet prefixes. A typical forwarding table at a router, often referred to as the Forwarding Information Base (FIB) for the router, contains entries (i.e., routes) in the format of: <network prefix>, <next hop>, <metric>. <network prefix> is the subnet prefix of the destination network for this route, <next hop> the interface to use as part of this route to reach the destination, and <metric> a measure of goodness of this route. To forward a packet, the router first looks up its FIB with the packet's destination address and looks for subnet prefix matches. When the packet's destination address matches multiple routes in the FIB, the router chooses the route with the longest prefix match, i.e., with the most matching network bits. If there are more than one longest prefix matches, the router uses the <metric> field to break the tie. After determining a route, the router forwards the packet to the output port as defined by that route's <next hop>.

Resource Allocation: Resource allocation did not receive serious consideration in the original design of the Internet architecture because of the datagram service principle. However, as the reach of the Internet extends and the access speed increases, latency or loss sensitive applications such as video phone start to be deployed. These applications require the network to provide some minimum level of performance guarantee with respect to throughput, packet delay, packet loss rate, etc. A new catch phrase, quality of services (QoS), has since been coined by the networking community to refer to the level of performance guarantee a computer network provides. The various approaches to enforce this include traffic engineering, Differentiated services model, Integrated services model, Multiprotocol level switching.

Security: Today, with the Internet becoming an open infrastructure for ecommerce and E-government, security is one of the most pressing issues faced by the Internet community. Several security mechanisms such as firewall, virtual private network, transport layer security, secure email, and public key infrastructure (PKI), have been added to the Internet architecture with some level of successes.

Choice of Platforms

Software Platforms for servers:

Every website needs a reliable web server to be hosted on, so that it can be accessed via internet users. Today, in web hosting market there are many types of web servers available running on different platform to select.

There are at least three categories of server software platforms you need to consider:

- Choose a network computing operating system that fits the size, needs and resources of your business. A **networking operating system** (**NOS**), also referred to as the Dialoguer, is the software that runs on a server and enables the server to manage data, users, groups, security, applications, and other networking functions. The network operating system is designed to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks. The most popular network operating systems are Microsoft Windows Server 2003, Microsoft Windows Server 2008,UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

- Pick a file server platform that's reliable and secure to protect your company's data.

- Use web server platform software that can handle the amount of traffic you'll get and that has the functionality you want.

The most popular platforms and web servers are:

- **UNIX and Linux running Apache web server**
- **Window NT/2000 running Internet Information Server (IIS)**

How do you choose your web server platform?

If your website is purely made up of static web pages (i.e. HTML files), then any web hosting platform will work fine for you. However, if your website allows dynamic content, you will most likely need to run specific server-side functionality such as CGI scripts, JSP, ASP, SSI or PHP. In this case, UNIX platform web hosting will be ideal for your requirement.

On the other hand, if you need to use specific applications that require Windows to run such as ASP, .Net, MS Access, Microsoft SQL server or Cold Fusion, then you will need to find a web hosting providers that support Microsoft's Windows NT platform. Otherwise, all other server-side functionalities such as PHP, Perl and MySQL can be supported by UNIX platform.

While common programs such as Perl, PHP, Flash etc run on both UNIX and Window platform. Many other free open source software programs are available only for UNIX than for Windows. As a result, UNIX hosting is less expensive than Window hosting. So, if hosting cost is a big concern to you, then you should consider UNIX or Linux hosting.

Software Development Platforms/ Frameworks

- **IIS: Internet information service is** a web server application and set of feature extension modules created by Microsoft for use with Microsoft Windows

- **AMP Platforms**
  - **Lamp: LAMP** is an acronym for a solution stack of free, open source software, referring to the first letters of Linux (operating system), Apache HTTP Server, MySQL (database software) and originally Perl (but now sometimes PHP or Python), principal components to build a viable general purpose web server. The software combination has become popular because it is free of cost, open-source, and therefore easily adaptable, and because of the ubiquity of its components which are bundled with most current Linux distributions. When used together, they form a solution stack of technologies that support application servers.
  - **Wamp: WAMP**s are packages of independently-created programs installed on computers that use a Microsoft Windows operating system. WAMP is an acronym formed from the initials of the operating system Microsoft Windows and the principal components of the package: Apache, MySQL and one of PHP, Perl or Python. Apache is a web server. MySQL is an open-source database. PHP is a scripting language that can manipulate information held in a database and generate

web pages dynamically each time content is requested by a browser. Other programs may also be included in a package, such as phpMyAdmin which provides a graphical user interface for the MySQL database manager, or the alternative scripting languages Python or Perl.

- **MAMP:** The acronym **MAMP** refers to a set of free software programs commonly used together to run dynamic web sites on servers running the Apple Macintosh operating system, Mac OS X:
    - **M**ac OS X, the operating system;
    - **A**pache, the Web server;
    - **M**ySQL, the database management system (or database server);
    - **P** for **P**HP, **P**erl, or **P**ython, all programming languages used for web development.

Any open source Web platform made up of these software programs and built upon Mac OS X is a MAMP.

- **Cross Platform**

- **XAMPP:**

- **Zend Server Community**

Hardware Platform for servers:

Hardware requirements for servers vary, depending on the server application. Absolute CPU speed is not usually as critical to a server as it is to a desktop machine . Servers' duties to provide service to many users over a network lead to different requirements such as fast network connections and high I/O throughput. Since servers are usually accessed over a network, they may run in headless mode without a monitor or input device. Processes that are not needed for the server's function are not used. Many servers do not have a graphical user interface (GUI) as it is unnecessary and consumes resources that could be allocated elsewhere. Similarly, audio and USB interfaces may be omitted.

To increase reliability, most of the servers use memory with error detection and correction, redundant disks, redundant power supplies and so on. Such components are also frequently hot swappable, allowing technicians to replace them on the running server without shutting it down.

Beside server computer the important hardware resources to establish a successful client/server model include; gateways, routers, network bridges, switches, hubs, and repeaters

Server Concepts: WEB, Proxy, RADIUS, MAIL:

Web Server:

Web server can refer to either the hardware (the computer) or the software (the computer application) that helps to deliver Web content that can be accessed through the Internet. The most common use of web servers is to host websites, but there are other uses such as gaming, data storage or running enterprise applications. The primary function of a web server is to deliver web pages on the request to clients using the Hypertext Transfer Protocol (HTTP). This means delivery of HTML documents and any additional content that may be included by a document, such as images, style sheets and scripts.

A user agent, commonly a web browser or web crawler, initiates communication by making a request for a specific resource using HTTP and the server responds with the content of that resource or an error message if unable to do so. The resource is typically a real file on the server's secondary memory, but this is not necessarily the case and depends on how the web server is implemented. While the primary function is to serve content, a full implementation of HTTP also includes ways of receiving content from clients. This feature is used for submitting web forms, including uploading of files.

Many generic web servers also support server-side scripting using Active Server Pages (ASP), PHP, or other scripting languages. This means that the behavior of the web server can be scripted in separate files, while the actual server software remains unchanged. Usually, this function is used to create HTML documents dynamically ("on-the-fly") as opposed to returning static documents. The former is primarily used for retrieving and/or modifying information from databases. The latter is typically much faster and more easily cached. Web servers are not always used for serving the World Wide Web. They can also be found embedded in devices such as printers, routers, webcams and serving only a local network. The web server may then be used as a part of a system for monitoring and/or administering the device in question. This usually means that no additional software has to be installed on the client computer, since only a web browser is required (which now is included with most operating systems).

A web server (program) has defined load limits, because it can handle only a limited number of concurrent client connections (usually between 2 and 80,000, by default between 500 and 1,000) per IP address (and TCP port) and it can serve only a certain maximum number of requests per second depending on its own settings, the HTTP request type, whether the content is static or dynamic, whether the content is cached, and the hardware and software limitations of the OS of the computer on which the web server runs.

When a web server is near to or over its limits, it becomes unresponsive.

At any time web servers can be overloaded because of:

- Too much legitimate web traffic. Thousands or even millions of clients connecting to the web site in a short interval, e.g., Slashdot effect; the term "Slashdot effect" refers to phenomenon of a website becoming virtually unreachable because too many people are hitting it after the site was mentioned in an interesting article on the popular Slashdot news service.
- Distributed Denial of Service attacks. A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer or network resource unavailable to its intended users;
- Computer worms that sometimes cause abnormal traffic because of millions of infected computers (not coordinated among them);
- XSS viruses can cause high traffic because of millions of infected browsers and/or web servers;
- Internet bots. Traffic not filtered/limited on large web sites with very few resources (bandwidth, etc.);
- Internet (network) slowdowns, so that client requests are served more slowly and the number of connections increases so much that server limits are reached;
- Web servers (computers) partial unavailability. This can happen because of required or urgent maintenance or upgrade, hardware or software failures, back-end (e.g., database) failures, etc.; in these cases the remaining web servers get too much traffic and become overloaded.

Proxy Server:

In computer networks, a proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server. The proxy server evaluates the request as a way to simplify and control their complexity. Today, most proxies are web proxies, facilitating access to content on the World Wide Web.

In an enterprise that uses the Internet, a proxy server is a server that acts as an intermediary between a workstation user and the Internet so that the enterprise can ensure security, administrative control, and caching service. A proxy server is associated with or part of a gateway server that separates the enterprise network from the outside network and a firewall server that protects the enterprise network from outside intrusion.

A proxy server receives a request for an Internet service (such as a Web page request) from a user. If it passes filtering requirements, the proxy server, assuming it is also a cache server, looks in its local cache of previously downloaded Web pages. If it finds the page, it returns it to the user without needing to forward the request to the Internet. If the page is not in the cache, the proxy server, acting as a client on behalf of the user, uses one of its own IP addresses to request the page from the server out on the Internet. When the page is returned, the proxy server relates it to the original request and forwards it on to the user.

A proxy server has a variety of potential purposes, including:

- To keep machines behind it anonymous, mainly for security.

- To speed up access to resources (using caching). Web proxies are commonly used to cache web pages from a web server.
- To apply access policy to network services or content, e.g. to block undesired sites.
- To access sites prohibited or filtered by your ISP or institution.
- To log / audit usage, i.e. to provide company employee Internet usage reporting.
- To bypass security / parental controls.
- To circumvent Internet filtering to access content otherwise blocked by governments.
- To scan transmitted content for malware before delivery.
- To scan outbound content, e.g., for data loss prevention.
- To allow a web site to make web requests to externally hosted resources (e.g. images, music files, etc.) when cross-domain restrictions prohibit the web site from linking directly to the outside domains.

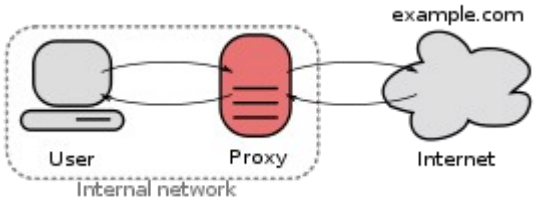
Proxy server can be placed in the user's local computer or at various points between the user and the destination servers on the Internet.

A proxy server that passes requests and responses unmodified is usually called a gateway or sometimes tunneling proxy.

A forward proxy is an Internet-facing proxy used to retrieve from a wide range of sources (in most cases anywhere on the Internet).

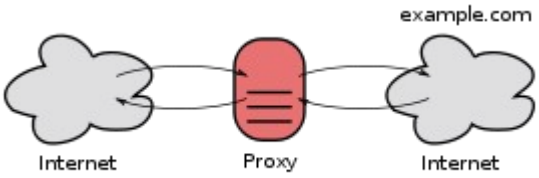
A reverse proxy is (usually) an Internet-facing proxy used as a front-end to control and protect access to a server on a private network, commonly also performing tasks such as load-balancing, authentication, decryption or caching.

Forward proxies



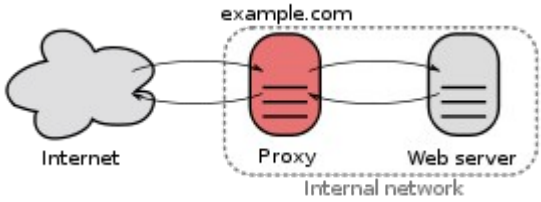
A forward proxies are those taking requests from an internal network and forwarding them to the Internet. Forward proxies are proxies where the client server names the target server to connect to. Forward proxies are able to retrieve from a wide range of sources (in most cases anywhere on the Internet).
The terms "forward proxy" and "forwarding proxy" are a general description of behavior (forwarding traffic) and thus ambiguous. Except for Reverse proxy, the types of proxies described in this article are more specialized sub-types of the general forward proxy concept.

Open proxies



An open proxy is one forwarding requests from and to anywhere on the Internet. An open proxy is a forwarding proxy server that is accessible by any Internet user. An anonymous open proxy allows users to conceal their IP address while browsing the Web or using other Internet services. There are varying degrees of anonymity however, as well as a number of methods of 'tricking' the client into revealing itself regardless of the proxy being used.

Reverse proxies



A reverse proxy is one 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. A reverse proxy (or surrogate) 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 web server.

Reverse proxies are installed in the neighborhood of one or more web servers. All traffic coming from the Internet and with a destination of one of the neighborhood's web servers goes through the proxy server. The use of "reverse" originates in its counterpart "forward proxy" since the reverse proxy sits closer to the web server and serves only a restricted set of websites.

There are several reasons for installing reverse proxy servers:

- **Encryption / SSL acceleration:** when secure web sites are created, the SSL encryption is often not done by the web server itself, but by a reverse proxy that is equipped with SSL acceleration hardware. Furthermore, a host can provide a single "SSL proxy" to provide SSL encryption for an arbitrary number of hosts; removing the need for a separate SSL Server Certificate for each host, with the downside that all hosts behind the SSL proxy have to share a common DNS name or IP address for SSL connections.
- **Load balancing:** the reverse proxy can distribute the load to several web servers, each web server serving its own application area. In such a case, the reverse proxy may need to rewrite the URLs in each web page (translation from externally known URLs to the internal locations).
- **Serve/cache static content:** A reverse proxy can offload the web servers by caching static content like pictures and other static graphical content.

- **Compression:** the proxy server can optimize and compress the content to speed up the load time.
- **Spoon feeding:** reduces resource usage caused by slow clients on the web servers by caching the content the web server sent and slowly "spoon feeding" it to the client. This especially benefits dynamically generated pages.
- **Security:** the proxy server is an additional layer of defense and can protect against some OS and WebServer specific attacks. However, it does not provide any protection to attacks against the web application or service itself, which is generally considered the larger threat.
- **Extranet Publishing:** a reverse proxy server facing the Internet can be used to communicate to a firewalled server internal to an organization, providing extranet access to some functions while keeping the servers behind the firewalls. If used in this way, security measures should be considered to protect the rest of your infrastructure in case this server is compromised, as its web application is exposed to attack from the Internet.

Use of Proxy servers:

Filtering:
A content-filtering web proxy server provides administrative control over the content that may be relayed in one or both directions through the proxy. It is commonly used in both commercial and non-commercial organizations (especially schools) to ensure that Internet usage conforms to acceptable use policy. In some cases users can circumvent the proxy, since there are services designed to proxy information from a filtered website through a non filtered site to allow it through the user's proxy

A content filtering proxy will often support user authentication, to control web access. It also usually produces logs, either to give detailed information about the URLs accessed by specific users, or to monitor bandwidth usage statistics

Some common methods used for content filtering include: URL or DNS blacklists, URL regex filtering, MIME filtering, or content keyword filtering. Some products have been known to employ content analysis techniques to look for traits commonly used by certain types of content providers.

Requests made to the open internet must first pass through an outbound proxy filter. The web-filtering company provides a database of URL patterns (regular expressions) with associated content attributes. This database is updated weekly by site-wide subscription, much like a virus filter subscription. The administrator instructs the web filter to ban broad classes of content (such as sports, pornography, online shopping, gambling, or social networking). Requests that match a banned URL pattern are rejected immediately.

Caching:
A caching proxy server accelerates service requests by retrieving content saved from a previous request made by the same client or even other clients. Caching proxies keep local copies of frequently requested resources, allowing large organizations to significantly

reduce their upstream bandwidth usage and costs, while significantly increasing performance. Most ISPs and large businesses have a caching proxy. Caching proxies were the first kind of proxy server. Another important use of the proxy server is to reduce the hardware cost. An organization may have many systems on the same network or under control of a single server, prohibiting the possibility of an individual connection to the Internet for each system. In such a case, the individual systems can be connected to one proxy server, and the proxy server connected to the main server.

DNS Proxy:
A DNS proxy server takes DNS queries from a (usually local) network and forwards them to an Internet Domain Name Server. It may also cache DNS records.

Gateways to private networks:
Proxy servers can perform a role similar to a network switch in linking two networks.

Accessing services anonymously:
An anonymous proxy server, sometimes called a web proxy, generally attempts to anonymize web surfing. There are different varieties of anonymizers. The destination server (the server that ultimately satisfies the web request) receives requests from the anonymizing proxy server, and thus does not receive information about the end user's address. However, the requests are not anonymous to the anonymizing proxy server, and so a degree of trust is present between the proxy server and the user. Many of them are funded through a continued advertising link to the user.

RADIUS:

Remote Authentication Dial In User Service (RADIUS) is a networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for computers to connect and use a network service.

Because of the broad support and the ubiquitous nature of the RADIUS protocol, it is often used by ISPs and enterprises to manage access to the Internet or internal networks, wireless networks, and integrated e-mail services. These networks may incorporate modems, DSL, access points, VPNs, network ports, web servers, etc

RADIUS serves three functions:

- **to authenticate users or devices before granting them access to a network,**
- **to authorize those users or devices for certain network services and**
- **to account for usage of those services.**

RADIUS servers use the AAA concept to manage network access in the following two-step process, also known as an "AAA transaction". AAA stands for “authentication, authorization and accounting”.

Authentication and authorization: The user or machine sends a request to a Remote Access Server (RAS) to gain access to a particular network resource using access credentials. The credentials are passed to the RAS device via the link-layer protocol - for example, Point-to-Point Protocol (PPP) in the case of many dialup or DSL providers or posted in an HTTPS secure web form. In turn, the RAS sends a RADIUS Access Request message to the RADIUS server, requesting authorization to grant access via the RADIUS protocol.

This request includes access credentials, typically in the form of username and password or security certificate provided by the user. Additionally, the request may contain other information which the RAS knows about the user, such as its network address or phone number, and information regarding the user's physical point of attachment to the RAS.
The RADIUS server checks that the information is correct using authentication schemes like PAP, CHAP or EAP. The user's proof of identification is verified, along with, optionally, other information related to the request, such as the user's network address or phone number, account status and specific network service access privileges. Historically, RADIUS servers checked the user's information against a locally stored flat file database. Modern RADIUS servers can do this, or can refer to external sources-commonly SQL, Kerberos, LDAP, or Active Directory servers—to verify the user's credentials.

The RADIUS server then returns one of three responses to the RAS ;

1. Access Reject
2. Access Challenge
3. Access Accept.

Access Reject - The user is unconditionally denied access to all requested network resources. Reasons may include failure to provide proof of identification or an unknown or inactive user account.

Access Challenge - Requests additional information from the user such as a secondary password, PIN, token or card. Access Challenge is also used in more complex authentication dialogs where a secure tunnel is established between the user machine and the Radius Server in a way that the access credentials are hidden from the RAS.

Access Accept - The user is granted access. Once the user is authenticated, the RADIUS server will often check that the user is authorized to use the network service requested. A given user may be allowed to use a company's wireless network, but not its VPN service, for example. Again, this information may be stored locally on the RADIUS server, or may be looked up in an external source like LDAP or Active Directory.

Each of these three RADIUS responses may include a Reply-Message attribute which may give a reason for the rejection, the prompt for the challenge, or a welcome message for the accept. The text in the attribute can be passed on to the user in a return web page.

Authorization attributes are conveyed to the RAS stipulating terms of access to be granted.

For example: the following authorization attributes may be included in an Access-Accept.

- The specific IP address to be assigned to the user
- The address pool from which the user's IP should be chosen
- The maximum length that the user may remain connected

Accounting: When network access is granted to the user by the NAS, an Accounting Start (a RADIUS Accounting Request packet containing an Acct-Status-Type attribute with the value "start") is sent by the NAS to the RADIUS server to signal the start of the user's network access. "Start" records typically contain the user's identification, network address, point of attachment and a unique session identifier.

Periodically, Interim Update records (a RADIUS Accounting Request packet containing an Acct-Status-Type attribute with the value "interim-update") may be sent by the NAS to the RADIUS server, to update it on the status of an active session. "Interim" records typically convey the current session duration and information on current data usage.

Finally, when the user's network access is closed, the NAS issues a final Accounting Stop record (a RADIUS Accounting Request packet containing an Acct-Status-Type attribute with the value "stop") to the RADIUS server, providing information on the final usage in terms of time, packets transferred, data transferred, reason for disconnect and other information related to the user's network access.

Typically, the client sends Accounting-Request packets until it receives an Accounting-Response acknowledgement, using some retry interval. The primary purpose of this data is that the user can be billed accordingly; the data is also commonly used for statistical purposes and for general network monitoring.

RADIUS is commonly used to facilitate roaming between ISPs, for example:

- by companies which provide a single global set of credentials that are usable on many public networks;
- by independent, but collaborating, institutions issuing their own credentials to their own users, that allow a visitor from one to another to be authenticated by their home institution,

Mail Server:

A mail server is a computer that serves as an electronic post office for email. Mail exchanged across networks is passed between mail servers that run specially designed software. This software is built around agreed-upon, standardized protocols for handling mail messages and the graphics they might contain.

Mail servers can be broken down into two main categories: outgoing mail servers and incoming mail servers. Outgoing mail servers are known as SMTP, or Simple Mail Transfer Protocol, servers. Incoming mail servers come in two main varieties. POP3, or Post Office Protocol, version 3, servers are best known for storing sent and received messages on PCs' local hard drives. IMAP, or Internet Message Access Protocol, servers always store copies of messages on servers. Most POP3 servers can store messages on servers, too, which is a lot more convenient.

There are several different components to the email system. They work together to move, deliver or retrieve your email.

Mail Transfer Agent (MTA): The MTA does a great deal of the hard work in moving mail around as it is responsible to move the mail from the local MTA to the destination MTA on the Internet. The Mail Transfer Agent works closely with DNS in making this all happen. The MTA uses a specific language SMTP to transfer mail on port 25, which is a standard. Several examples of MTAs are Sendmail, Postfix, and QMAIL.

Mail Delivery Agent (MDA): The MDA will receive the mail destined for the local network from the MTA and then will make this mail available for the user. The MDA will use POP3 on port 110 or IMAP on port 143 to make this available to users. Examples of MDA are Dovecot or Cyrus-IMAP.

Mail User Agent (MUA): The MUA is the client program that the end user uses to retrieve and view email. Users are able to view web based email with a browser but will user tools like Outlook, Thunderbird, Mutt, or Evolution to download mail to the local machine. When you send or read your email, the only part you see is the MUA, which is a fancy way of saying email client. But, there’s a lot more than that involved. To send an email, you’ll first sit down at your computer and fire up Thunderbird, or whichever other email client that you’re using. When you compose the message and click on the send button, the MUA will send it to the MTA. (The MTA could either be on the corporate network or at your ISP.) This MTA will send the message to successive MTA’s until it gets to the MTA that serves the email recipient. This MTA will then send the message to a Post Office Protocol (POP)/Internet Mail Application Protocol (IMAP) server. This server will store the email until the recipient accesses it with her MUA. Of course, this same process could take place within a corporate network instead of across the Internet.

The Process of Sending an Email:
Now that you know the basics about incoming and outgoing mail servers, it will be easier to understand the role that they play in the emailing process. The basic steps of this process are outlined below for your convenience.

Step #1: After composing a message and hitting send, your email client - whether it's Outlook Express or Gmail - connects to your domain's SMTP server. This server can be named many things; a standard example would be smtp.example.com.

Step #2: Your email client communicates with the SMTP server, giving it your email address, the recipient's email address, the message body and any attachments.

Step #3: The SMTP server processes the recipient's email address - especially its domain. If the domain name is the same as the sender's, the message is routed directly over to the domain's POP3 or IMAP server - no routing between servers is needed. If the domain is different, though, the SMTP server will have to communicate with the other domain's server.

Step #4: In order to find the recipient's server, the sender's SMTP server has to communicate with the DNS, or Domain Name Server. The DNS takes the recipient's email domain name and translates it into an IP address. The sender's SMTP server cannot route an email properly with a domain name alone; an IP address is a unique number that is assigned to every computer that is connected to the Internet. By knowing this information, an outgoing mail server can perform its work more efficiently.

Step #5: Now that the SMTP server has the recipient's IP address, it can connect to its SMTP server. This isn't usually done directly, though; instead, the message is routed along a series of unrelated SMTP servers until it arrives at its destination.

Step #6: The recipient's SMTP server scans the incoming message. If it recognizes the domain and the user name, it forwards the message along to the domain's POP3 or IMAP server. From there, it is placed in a sendmail queue until the recipient's email client allows it to be downloaded. At that point, the message can be read by the recipient.

Cookies:

A cookie, also known as an HTTP cookie, web cookie, or browser cookie, is usually a small piece of data sent from a website and stored in a user's web browser while a user is browsing a website. When the user browses the same website in the future, the data stored in the cookie can be retrieved by the website to notify the website of the user's previous activity. Cookies were designed to be a reliable mechanism for websites to remember the state of the website or activity the user had taken in the past. This can include clicking particular buttons, logging in, or a record of which pages were visited by the user even months or years ago.

Perhaps most importantly, authentication cookies are the most common method used by web servers to know whether the user is logged in or not, and which account they are logged in under. Without such a mechanism, the site would not know whether to send a page containing sensitive information, or require the user to authenticate himself by logging-in. The security of an authentication cookie generally depends on the security of the issuing website and the user's web browser. If not implemented correctly, a cookie's data can be intercepted by a hacker to gain unapproved access to the user's data and possibly to the originating website.

Cookies are arbitrary pieces of data chosen by the Web server and sent to the browser. The browser returns them unchanged to the server, introducing a state (memory of previous events) into otherwise stateless HTTP transactions. Without cookies, each retrieval of a Web page or component of a Web page is an isolated event, mostly unrelated to all other views of the pages of the same site. Other than being set by a web server, cookies can also be set by a script in a language such as JavaScript, if supported and enabled by the Web browser.

The cookies consist of several values;

1. **The name and value** are encoded into the cookie and represent the state. The interpretation is that the name has an associated value.
2. **The expires field** indicates when the cookie is valid. Expired cookies are discarded; they are not to be given out. If this field is not present, the cookie will be deleted at the end of the session.
3. **The domain** states the domain for which the cookie is intended. It consists of the last n fields of the domain name of a server. For example, domain=.adv.com specifies that the cookie is to be sent to any requesting server in the adv.com domain. A domain field must have at least one embedded "." in it. There is no requirement that a cookie be sent from a host in the domain. This can be used to track certain types of accesses, as discussed below.
4. **The path** further restricts the dissemination of the cookie. When a Web server requests a cookie, it provides a domain. Cookies that match that domain may be sent to the server. If the server specifies a path, the path must be the leading substring of the path specified in the cookie.
5. If **the secure field** is set, the cookie will be sent only over secured connections (that is, to "https" or "http" over SSL).

Cookies can contain authentication information, both user-related and host-related. Using cookies for authentication treats them as tokens supplied by the browser to validate (or state and validate) an identity. Depending on the sensitivity of the interactions with the server, protecting the confidentiality of these cookies may be critical.

Types of cookie:

Session cookie: A user's session cookie for a website exists only while the user is reading and navigating the website. When an expiry date or validity interval is not set at cookie creation time, a session cookie is created. Web browsers normally delete session cookies when the user exits the browser.

Persistent cookie: A persistent cookie will outlast user sessions. If a persistent cookie has its Max-Age set to 1 year, then, within the year, the initial value set in that cookie would be sent back to the server every time the user visited the server. This could be used to record a vital piece of information such as how the user initially came to this website. For this reason persistent cookies are also called tracking cookies.

Secure cookie: A secure cookie has the secure attribute enabled and is only used via HTTPS, ensuring that the cookie is always encrypted when transmitting from client to server. This makes the cookie less likely to be exposed to cookie theft via eavesdropping.

HttpOnly cookie: The HttpOnly cookie is supported by most modern browsers. On a supported browser, an HttpOnly session cookie will be used only when transmitting HTTP (or HTTPS) requests, thus restricting access from other, non-HTTP APIs (such as JavaScript). This restriction mitigates but does not eliminate the threat of session cookie theft via cross-site scripting (XSS). This feature applies only to session-management cookies, and not other browser cookies.

Third-party cookie: First-party cookies are cookies set with the same domain (or its subdomain) in your browser's address bar. Third-party cookies are cookies being set with different domains from the one shown on the address bar (i.e. the web pages on that domain may feature content from a third-party domain - e.g. an advertisement run by www.advexample.com showing advert banners). (Privacy setting options in most modern browsers allow you to block third-party tracking cookies).

For example: Suppose a user visits www.example1.com, which sets a cookie with the domain ad.foxytracking.com. When the user later visits www.example2.com, another cookie is set with the domain ad.foxytracking.com. Eventually, both of these cookies will be sent to the advertiser when loading their ads or visiting their website. The advertiser can then use these cookies to build up a browsing history of the user across all the websites this advertiser has footprints on.

Supercookie: A "supercookie" is a cookie with a public suffix domain, like .com, .co.uk or k12.ca.us. Most browsers, by default, allow first-party cookies—a cookie with domain to be the same or sub-domain of the requesting host. For example, a user visiting www.example.com can have a cookie set with domain www.example.com or .example.com, but not .com. A supercookie with domain .com would be blocked by browsers; otherwise, a malicious website, like attacker.com, could set a supercookie with domain .com and potentially disrupt or impersonate legitimate user requests to example.com. The Public Suffix List is a cross-vendor initiative to provide an accurate list of domain name suffixes changing. Older versions of browsers may not have the most up-to-date list, and will therefore be vulnerable to certain supercookies.

The term "supercookies" is sometimes used for tracking technologies that do not rely on HTTP cookies. Two such "supercookie" mechanisms were found on Microsoft websites: cookie syncing that respawned MUID cookies, and ETag cookies

Zombie cookie: A zombie cookie is any cookie that is automatically recreated after a user has deleted it. This is accomplished by a script storing the content of the cookie in some other locations, such as the local storage available to Flash content, HTML5 storages and other client side mechanisms, and then recreating the cookie from backup stores when the cookie's absence is detected.

Uses of Cookies:

Session Management:
Cookies may be used to maintain data related to the user during navigation, possibly across multiple visits. Cookies were introduced to provide a way to implement a "shopping cart" (or "shopping basket"), a virtual device into which users can store items they want to purchase as they navigate throughout the site.

Shopping basket applications today usually store the list of basket contents in a database on the server side, rather than storing basket items in the cookie itself. A web server typically sends a cookie containing a unique session identifier. The web browser will send back that session identifier with each subsequent request and shopping basket items are stored associated with a unique session identifier.

Allowing users to log in to a website is a frequent use of cookies. Typically the web server will first send a cookie containing a unique session identifier. Users then submit their credentials and the web application authenticates the session and allows the user access to services.

Personalization:
Cookies may be used to remember the information about the user who has visited a website in order to show relevant content in the future. For example a web server may send a cookie containing the username last used to log in to a website so that it may be filled in for future visits.

Many websites use cookies for personalization based on users' preferences. Users select their preferences by entering them in a web form and submitting the form to the server. The server encodes the preferences in a cookie and sends the cookie back to the browser. This way, every time the user accesses a page, the server is also sent the cookie where the preferences are stored, and can personalize the page according to the user preferences. For example, the Wikipedia website allows authenticated users to choose the webpageskin they like best; the Google search engine once allowed users (even non-registered ones) to decide how many search results per page they want to see.

Tracking:
Tracking cookies may be used to track internet users' web browsing. This can also be done in part by using the IP address of the computer requesting the page or the referrer field of the HTTP request header, but cookies allow for greater precision.

This can be demonstrated as follows:
If the user requests a page of the site, but the request contains no cookie, the server presumes that this is the first page visited by the user; the server creates a random string and sends it as a cookie back to the browser together with the requested page;

From this point on, the cookie will be automatically sent by the browser to the server every time a new page from the site is requested; the server sends the page as usual, but also stores the URL of the requested page, the date/time of the request, and the cookie in a log file.

By analyzing the log file collected in the process, it is then possible to find out which pages the user has visited, and in what sequence.

Load Balancing: Proxy Arrays

Most commonly, the term load balancing refers to distributing incoming HTTP requests across Web servers in a server farm, to avoid overloading any one server. Because load balancing distributes the requests based on the actual load at each server, it is excellent for ensuring availability and defending against denial of service attacks.

Load balancing is a computer networking methodology to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy. The load balancing service is usually provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server.

Why is load balancing of servers needed?

Load balancing is especially important for networks where it's difficult to predict the number of requests that will be issued to a server. Busy Web sites typically employ two or more Web servers in a load balancing scheme. If one server starts to get swamped, requests are forwarded to another server with more capacity. Load balancing can also refer to the communications channels themselves.

If there is only one web server responding to all the incoming HTTP requests for your website, the capacity of the web server may not be able to handle high volumes of incoming traffic once the website becomes popular. The website's pages will load slowly as some of the users will have to wait until the web server is free to process their requests. The increase in traffic and connections to your website can lead to a point where upgrading the server hardware will no longer be cost effective.

In order to achieve web server scalability, more servers need to be added to distribute the load among the group of servers, which is also known as a *server cluster*. The load distribution among these servers is known as load balancing. Load balancing applies to all types of servers (application server, database server), however, we will be devoting this section for load balancing of web servers (HTTP server) only.

Application

One of the most common applications of load balancing is to provide a single Internet service from multiple servers, sometimes known as a server farm. Commonly, load-balanced systems include popular web sites, large Internet Relay Chat networks, high-bandwidth File Transfer Protocol sites, Network News Transfer Protocol (NNTP) servers and Domain Name System (DNS) servers. Lately, some load balancers evolved to support databases; these are called database load balancers.

For Internet services, the load balancer is usually a software program that is listening on the port where external clients connect to access services. The load balancer forwards requests to one of the "backend" servers, which usually replies to the load balancer. This allows the load balancer to reply to the client without the client ever knowing about the internal separation of functions. It also prevents clients from contacting backend servers directly, which may have security benefits by hiding the structure of the internal network and preventing attacks on the kernel's network stack or unrelated services running on other ports.

How?

When multiple web servers are present in a server group, the HTTP traffic needs to be evenly distributed among the servers. In the process, these servers must appear as one web server to the web client, for example an internet browser. The load balancing mechanism used for spreading HTTP requests is known **as *IP Spraying*.** The equipment used for IP spraying is also called the 'load dispatcher' or 'network dispatcher' or simply, the 'load balancer'. In this case, the IP sprayer intercepts each HTTP request, and redirects them to a server in the server cluster. Depending on the type of sprayer involved, the architecture can provide scalability, load balancing and fail-over requirements.

Proxy Server Arrays are way to handle loads in internet Proxy Server provides a feature called proxy arrays**.** An array is a peer-to-peer configuration of proxy servers instead of a hierarchy. **A proxy array is a solution whereby one or multiple proxy servers operate as a single cache for client requests**. Each Proxy Server that belongs to the proxy array performs the following functions:

- Maintains membership information for the proxy array. This information provides input on which array members are available and which array members are unavailable. A client therefore has to only query one array member because each member of the proxy array maintains information on all other array members.

- Uses a hash algorithm to perform routing decisions. The hash algorithm uses the following factors:
  List of current available servers.
  URL of client request.

Load factor.

Arrays have the following benefits:

- Users and application sessions are *load-balanced* across the array. To scale to more users, simply add more SGD servers to the array. See Load Balancing for more details.
- With more than one server, there is no single point of failure. You can decommission a server temporarily with the minimum of disruption to your users.

- Configuration information, including all the objects in your organizational hierarchy, is replicated to all array members. All array members have access to all information.

An array contains the following:

- **One primary server.** This server is the authoritative source for global SGD information, and maintains the definitive copy of the organizational hierarchy, called the local repository.
- **One or more secondary servers.** The primary server replicates information to these servers.

Load Balancing Approaches

Load balancing of servers by an IP sprayer can be implemented in different ways. These methods of load balancing can be set up in the load balancer based on available load balancing types. There are various algorithms used to distribute the load among the available servers.

Random Allocation

In a random allocation, the HTTP requests are assigned to any server picked randomly among the group of servers. In such a case, one of the servers may be assigned many more requests to process, while the other servers are sitting idle. However, on average, each server gets its share of the load due to the random selection.

***Pros*:** Simple to implement.

***Cons*:** Can lead to overloading of one server while under-utilization of others.

Round-Robin Allocation

In a round-robin algorithm, the IP sprayer assigns the requests to a list of the servers on a rotating basis. The first request is allocated to a server picked randomly from the group, so that if more than one IP sprayer is involved, not all the first requests go to the same server.

For the subsequent requests, the IP sprayer follows the circular order to redirect the request. Once a server is assigned a request, the server is moved to the end of the list. This keeps the servers equally assigned.

***Pros*:** Better than random allocation because the requests are equally divided among the available servers in an orderly fashion.

***Cons*:** Round robin algorithm is not enough for load balancing based on processing overhead required and if the server specifications are not identical to each other in the server group.

Weighted Round-Robin Allocation

Weighted Round-Robin is an advanced version of the round-robin that eliminates the deficiencies of the plain round robin algorithm. In case of a weighted round-robin, one can assign a weight to each server in the group so that if one server is capable of handling twice as much load as the other, the powerful server gets a weight of 2. In such cases, the IP sprayer will assign two requests to the powerful server for each request assigned to the weaker one.

***Pros*:** Takes care of the capacity of the servers in the group.

***Cons*:** Does not consider the advanced load balancing requirements such as processing times for each individual request.

Dynamic Round Robin (Called **Dynamic Ratio** on the BIG-IP)

It is similar to Weighted Round Robin, however, weights are based on continuous monitoring of the servers and are therefore continually changing. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time. This *Application Delivery Controller* method is *rarely* available in a simple load balancer.

The configuration of a load balancing software or hardware should be decided on the particular requirement. For example, if the website wants to load balance servers for static HTML pages or light database driven dynamic webpages, round robin will be sufficient. However, if some of the requests take longer than the others to process, then advanced load balancing algorithms are used. The load balancer should be able to provide intelligent monitoring to distribute the load, directing them to the servers that are capable of handling them better than the others in the cluster of server.

Server Setup and Configuration Guidelines: *Needs to be finalized*

Security and System Administration Issues:

Security issues:

An organization's servers provide a wide variety of services to internal and external users, and many servers also store or process sensitive information for the organization. Some of the most common types of servers are Web, email, database, infrastructure management, and file servers.

A *server* is a host that provides one or more services for other hosts over a network as a primary function. For example, a file server provides file sharing services so that users can access, modify, store, and delete files. Another example is a database server that provides database services for Web applications on Web servers. The Web servers, in turn, provide Web content services to users' Web browsers. There are many other types of servers, such as application, authentication, directory services, email, infrastructure management, logging, name/address resolution services (e.g., Domain Name Server [DNS]), print, and remote access

Servers are frequently targeted by attackers because of the value of their data and services. For example, a server might contain personally identifiable information that could be used to perform identity theft. The following are examples of common security threats to servers:

- Malicious entities may exploit software bugs in the server or its underlying operating system to gain unauthorized access to the server.

- Denial of service (DoS) attacks may be directed to the server or its supporting network infrastructure, denying or hindering valid users from making use of its services.

- Sensitive information on the server may be read by unauthorized individuals or changed in an unauthorized manner.

- Sensitive information transmitted unencrypted or weakly encrypted between the server and the client may be intercepted.

- Malicious entities may gain unauthorized access to resources elsewhere in the organization's network via a successful attack on the server.

- Malicious entities may attack other entities after compromising a server. These attacks can be launched directly (e.g., from the compromised host against an external server) or indirectly (e.g., placing malicious content on the compromised server that attempts to exploit vulnerabilities in the clients of users accessing the server).

The classic model for information security defines three objectives of security: maintaining confidentiality, integrity, and availability. *Confidentiality* refers to protecting information from being accessed by unauthorized parties. *Integrity* refers to ensuring the authenticity of information—that information is not altered, and that the source of the information is

genuine. *Availability* means that information is accessible by authorized users. Each objective addresses a different aspect of providing protection for information.

When addressing server security issues, it is an excellent idea to keep in mind the following general information security principles:

- **Simplicity**—Security mechanisms (and information systems in general) should be as simple as possible. Complexity is at the root of many security issues.

- **Fail-Safe**—If a failure occurs, the system should fail in a secure manner, i.e., security controls and settings remain in effect and are enforced. It is usually better to lose functionality rather than security.

- **Complete Mediation**—Rather than providing direct access to information, mediators that enforce access policy should be employed. Common examples of mediators include file system permissions, proxies, firewalls, and mail gateways.

- **Open Design**—System security should not depend on the secrecy of the implementation or its components.

- **Separation of Privilege**—Functions, to the degree possible, should be separate and provide as much granularity as possible. The concept can apply to both systems and operators and users. In the case of systems, functions such as read, edit, write, and execute should be separate. In the case of system operators and users, roles should be as separate as possible. For example, if resources allow, the role of system administrator should be separate from that of the database administrator.

- **Least Privilege**—This principle dictates that each task, process, or user is granted the minimum rights required to perform its job. By applying this principle consistently, if a task, process, or user is compromised, the scope of damage is constrained to the limited resources available to the compromised entity.

- **Psychological Acceptability**—Users should understand the necessity of security. This can be provided through training and education. In addition, the security mechanisms in place should present users with sensible options that give them the usability they require on a daily basis. If users find the security mechanisms too cumbersome, they may devise ways to work around or compromise them. The objective is not to weaken security so it is understandable and acceptable, but to train and educate users and to design security mechanisms and policies that are usable and effective.

- **Least Common Mechanism**—When providing a feature for the system, it is best to have a single process or service gain some function without granting that same function to other parts of the system. The ability for the Web server process to access a back-end database, for instance, should not also enable other applications on the system to access the back-end database.

- **Defense-in-Depth**—Organizations should understand that a single security mechanism is generally insufficient. Security mechanisms (defenses) need to be layered so that compromise of a single security mechanism is insufficient to compromise a host or network. No “silver bullet” exists for information system security.

- **Work Factor**—Organizations should understand what it would take to break the system or network’s security features. The amount of work necessary for an attacker to break the system or network should exceed the value that the attacker would gain from a successful compromise.

- **Compromise Recording**—Records and logs should be maintained so that if a compromise does occur, evidence of the attack is available to the organization. This information can assist in securing the network and host after the compromise and aid in identifying the methods and exploits used by the attacker. This information can be used to better secure the host or network in the future. In addition, these records and logs can assist organizations in identifying and prosecuting attackers.

Administration issues

The Server administration includes designing, installing, administering, and optimizing company servers and related components to achieve high performance of the various business applications supported by tuning the servers as necessary. This includes ensuring the availability of client/server applications, configuring all new implementations, and developing processes and procedures for ongoing management of the server environment. Where applicable, the Server Administrator will assist in overseeing the physical security, integrity, and safety of the data center/server farm.

Server Administration is handled by the administrators. Server administrators are system architects responsible for the overall design, implementation, and maintenance of a server. Network administrators are responsible for the overall design, implementation, and maintenance of a network.

The Vital activities include handling and analyzing log files, performing regular server backups, recovering from server compromises, testing server security regularly, and performing remote administration securely.

Logging

Logging is a cornerstone of a sound security posture. Capturing the correct data in the logs and then monitoring those logs closely is vital.35 Network and system logs are important, especially system logs in the case of encrypted communications, where network monitoring is less effective. Server software can provide additional log data relevant to server-specific events.

Reviewing logs is mundane and reactive, and many server administrators devote their time to performing duties that they consider more important or urgent. However, log files are often the only record of suspicious behavior. Enabling the mechanisms to log information allows the logs to be used to detect failed and successful intrusion attempts and to initiate alert mechanisms when further investigation is needed. Procedures and tools need to be in place to process and analyze the log files and to review alert notifications.
Server logs provide—

- Alerts to suspicious activities that require further investigation
- Tracking of an attacker's activities
- Assistance in the recovery of the server
- Assistance in post-event investigation
- Required information for legal proceedings.

Server Backup Procedures

One of the most important functions of a server administrator is to maintain the integrity of the data on the server. This is important because servers are often some of the most exposed and vital hosts on an organization's network. The server administrator needs to perform backups of the server on a regular basis for several reasons. A server could fail as a result of a malicious or unintentional act or a hardware or software failure. In addition, Federal agencies and many other organizations are governed by regulations on the backup and archiving of server data. Server data should also be backed up regularly for legal and financial reasons.

Security Testing Servers

Periodic security testing of servers is critical. Without periodic testing, there is no assurance that current protective measures are working or that the security patch applied by the server administrator is functioning as advertised. Although a variety of security testing techniques exists, vulnerability scanning is the most common. Vulnerability scanning assists a server administrator in identifying vulnerabilities and verifying whether the existing security measures are effective. Penetration testing is also used, but it is used less frequently and usually only as part of an overall penetration test of the organization's network

Authorization and Authentication

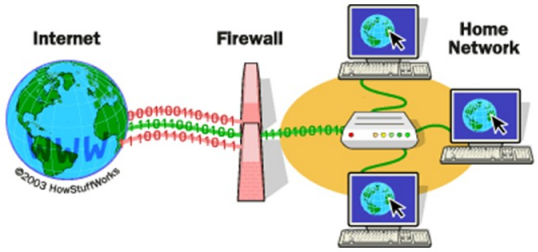
Vulnerability Scanning

Vulnerability scanners are automated tools that are used to identify vulnerabilities and misconfigurations of hosts. Many vulnerability scanners also provide information about mitigating discovered vulnerabilities. Vulnerability scanners attempt to identify vulnerabilities in the hosts scanned. Vulnerability scanners can help identify out-of-date software versions, missing patches, or system upgrades, and they can validate compliance with or deviations from the organization's security policy. To accomplish this, vulnerability scanners identify OSs, server software, and other major software applications running on hosts and match them with known vulnerabilities in their vulnerability databases.

Firewalls:

Firewall is hardware device or software applications that act as filters between a company's private network and the internet. It protects networked computers from intentional hostile intrusion that could compromise confidentiality or result in data corruption or denial of service by enforcing an access control policy between two networks.

The main purpose of a firewall system is to control access to or from a protected network (i.e., a site). It implements a network access policy by forcing connections to pass through the firewall, where they can be examined and evaluated. A firewall system can be a router, a personal computer, a host, or a collection of hosts, set up specifically to shield a site or subnet from protocols and services that can be abused from hosts outside the subnet. A firewall system is usually located at a higher level gateway, such as a site's connection to the Internet, however firewall systems can be located at lower-level gateways to provide protection for some smaller collection of hosts or subnets. The main function of a firewall is to centralize access control. A firewall serves as the gatekeeper between the untrusted Internet and the more trusted internal networks. The earliest firewalls were simply routers.



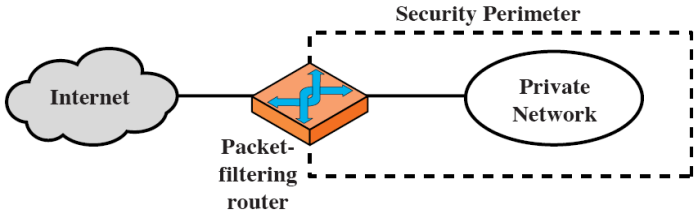
Firewalls provide several types of protection:

- They can block unwanted traffic.
- They can direct incoming traffic to more trustworthy internal systems.
- They hide vulnerable systems, which can't easily be secured from the Internet.
- They can log traffic to and from the private network.
- They can hide information like system names, network topology, network device types, and internal user ID's from the Internet.
- They can provide more robust authentication than standard applications might be able to do.

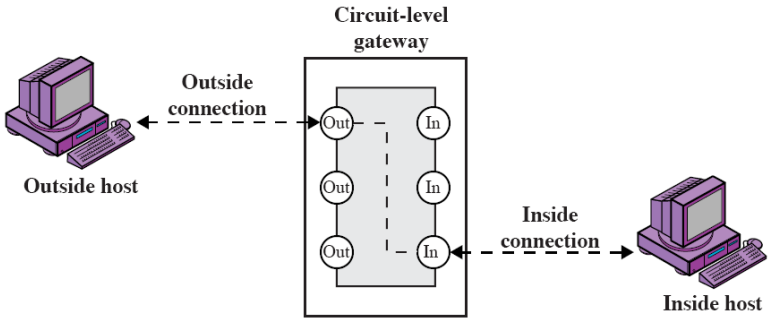
What does a firewall do?

A firewall examines all traffic routed between the two networks to see if it meets certain criteria. If it does, it is routed between the networks, otherwise it is stopped. It can be used to log all attempts to enter the private network and trigger alarms when hostile or unauthorized entry is attempted. Firewalls can filter packets based on their source and destination addresses and port numbers. This is known as address filtering. Firewalls can also filter specific types of network traffic. This is also known as protocol filtering because the decision to forward or reject traffic is dependent upon the protocol used, for example HTTP, ftp or telnet. Firewalls can also filter traffic by packet attribute or state.
There are different types of firewalls available in today's world some of them are:

Packet Filters: Packet filtering firewalls work at the network layer (OSI model), or the IP layer (TCP/IP). In this each packet is compared to a set of criteria before it is forwarded. Depending on the packet and the criteria, the firewall can drop, forward the packet or send a message to the originator. Rules can be source and destination IP address, source and destination port number and protocol used. The advantages of packet filtering firewalls is their low cost and low impact on network performance

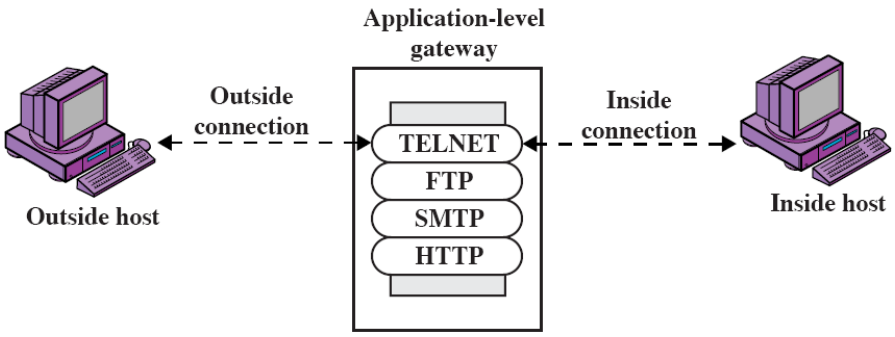


Circuit Level Gateways: It work at the session layer (OSI model), or the TCP layer (TCP/IP). They monitor TCP handshaking between packets to determine whether a requested session is legitimate. Information passed to a remote computer through a circuit level gateway appears to have originated from the gateway. This is useful for hiding information about protected networks. Circuit level gateways are relatively inexpensive and have the advantage of hiding information about the private network they protect. On the other hand, they do not filter individual packets.



Application Gateways: Application level gateways, also called proxies, are similar to circuit-level gateways except that they are application specific. They can filter packets at the application layer of the OSI model. Incoming or outgoing packets cannot access

services for which there is no proxy. In plain terms, an application level gateway that is configured to be a web proxy acts as the server to the internal network and client to the external network. Because they examine packets at application layer, they can filter application specific commands such as http: post and get, etc. Application level gateways can also be used to log user activity and logins. They offer a high level of security, but have a significant impact on network performance.



Stateful Multilayer Inspection Firewall: It combines the aspects of the other three types of firewalls. They filter packets at the network layer, determine whether session packets are legitimate and evaluate contents of packets at the application layer. They rely on algorithms to recognize and process application layer data instead of running application specific proxies. Stateful multilayer inspection firewalls offer a high level of security, good performance and transparency to end users. They are expensive however, and due to their complexity are potentially less secure than simpler types of firewalls if not administered by highly competent personnel.

Content Filtering:

Content filtering is the technique whereby content is blocked or allowed based on analysis of its content, rather than its source or other criteria. It is most widely used on the internet to filter email and web access.

Content filtering on Web commonly is named Web filtering.

Content filtering is commonly used by organizations such as offices and schools to prevent computer users from viewing inappropriate web sites or content, or as a pre-emptive security measure to prevent access of known malware hosts. **Filtering rules are typically set by a central IT department and may be implemented via software on individual computers or at a central point on the network such as the proxy server or internet router.** Depending on the sophistication of the system used, it may be possible for different computer users to have different levels of internet access.

On the Internet, content filtering (also known as *information filtering*) is the use of a program to screen and exclude from access or availability Web pages or e-mail that is deemed objectionable. Content filtering is used by corporations as part of

Internet firewall computers and also by home computer owners, especially by parents to screen the content their children have access to from a computer.

Content filtering usually works by specifying character strings that, if matched, indicate undesirable content that is to be screened out. Content is typically screened for pornographic content and sometimes also for violence- or hate-oriented content. Critics of content filtering programs point out that it is not difficult to unintentionally exclude desirable content.

Content filtering and the products that offer this service can be divided into Web filtering, the screening of Web sites or pages, and e-mail filtering, the screening of e-mail for spam or other objectionable content.

Filters can be implemented in many different ways: by a software program on a personal computer via network infrastructure such as proxy servers that provide Internet access.

Browser based filters: Browser based content filtering solution is the most lightweight solution to do the content filtering, and is implemented via a third party browser extension.

Client-side filters: This type of filter is installed as software on each computer where filtering is required. This filter can typically be managed, disabled or uninstalled by anyone who has administrator-level privileges on the system.

Content-limited (or filtered) ISPs: Content-limited (or filtered) ISPs are Internet service providers that offer access to only a set portion of Internet content on an opt-in or a mandatory basis. Anyone who subscribes to this type of service is subject to restrictions. The type of filters can be used to implement government, regulatory or parental control over subscribers.

Network-based filtering: This type of filter is implemented at the transport layer as a transparent proxy, or at the application layer as a web proxy. Filtering software may include data loss prevention functionality to filter outbound as well as inbound information. All users are subject to the access policy defined by the institution. The filtering can be customized, so a school district's high school library can have a different filtering profile than the district's junior high school library.

Search-engine filters: Many search engines, such as Google and Alta Vista offer users the option of turning on a safety filter. When this safety filter is activated, it filters out the inappropriate links from all of the search results. If one knows the actual URL of a website that features sexual explicit or 18 + content, they have the ability to access it without using a search engine. Engines like Lycos, Yahoo, and Bing offer kid-oriented versions of their engines that permit only children friendly websites.

There are four options for content filtering: Client based software, server based software, stand-alone appliances (hardware) and managed service.

- **Client based software** - Client based software is installed on the workstation that is used to surf the Internet. This software is in addition to web browsers which offer a certain level of content filtering. Software requires separate installation and maintenance for each workstation. Client based software may be used in environments with a limited number of users.
- **Server based software** - Server based software is installed on a host server such as a web server, proxy server, or firewall and is maintained using administrator software. Server based software allows for the central control of an entire network or single subnet with a single installation. Because of cost concerns and the required network infrastructure, server-based software is typically used in mid to large-sized environments.
- **Stand-alone appliance** - Stand-alone appliances are hardware devices that can be installed on a network you wish to filter and monitor. These are generally higher performance solutions because they use dedicated hardware and are optimized for filtering. Appliance solutions can be scaled for deployment in small to large environments.
- **Managed Service** - This solution involves outsourcing a portion or all of a company's security infrastructure including content filtering. Third-party managed security solutions have been deployed in all environments.

Some Internet Service Providers (ISP's) provide content filtering as part of a business grade broadband offering.

Content Filtering Types

- **From Address**
  1. From specific addresses
  2. From specific domains
  3. From trusted senders

- **Contains Specific Words or Phrases**
  4. Subject
  5. Body Text
  6. Subject or Body Text
  7. From Address
  8. To Address
  9. Email Headers
  10. Anywhere in Message

- **To Address**
  11. To Specific Addresses
  12. To Specific Domains
  13. Only to Me
  14. My Address in To Field

15. My Address not in To Field
16. My Address in To or CC Field

- **Attachments**
  17. Has any Attachment
  18. Specific Filenames
  19. Specific Extensions
  20. Over Specific Size

- **Other**
  21. Flagged as High Priority
  22. Flagged as Normal Priority
  23. Flagged as Low Priority
  24. Message is Automated (no return address)
  25. Message under Size
  26. Message over Size
  27. Received in Date Range
  28. Sent through a Specific Server (by IP)
  29. Spam Probability

Email Filtering:

Email filtering is the processing of email to organize it according to specified criteria. Most often this refers to the automatic processing of incoming messages, but the term also applies to the intervention of human intelligence in addition to anti-spam techniques, and to outgoing emails as well as those being received.

Email filtering softwares are given emails as inputs. For its output, it might pass the message through unchanged for delivery to the user's mailbox, redirect the message for delivery elsewhere, or even throw the message away. Some mail filters are able to edit messages during processing.

Mail filters can operate on inbound and outbound email traffic. Inbound email filtering involves scanning messages from the Internet addressed to users protected by the filtering system or for lawful interception. Outbound email filtering involves the reverse - scanning email messages from local users before any potentially harmful messages can be delivered to others on the Internet. One method of outbound email filtering that is commonly used by Internet service providers is transparent **SMTP proxying**, in which email traffic is intercepted and filtered via a transparent proxy within the network. Outbound filtering can also take place in an email server. Many corporations employ data leak prevention technology in their outbound mail servers to prevent the leakage of sensitive information via email.

SMTP Proxy:

SMTP Proxies are commonly used to process and filter inbound and outbound email traffic. SMTP proxy can be used to control email messages and email content. The proxy scans SMTP messages for a number of filtered parameters, and compares them against the rules in the proxy configuration.

With an SMTP proxy filter you can:

- Adjust timeout, maximum email size, and line length limit to make sure the SMTP proxy does not use too many network resources and can prevent some types of attacks.
- Customize the deny message that users see when an email they try to receive is blocked.
- Filter content embedded in email with MIME types and name patterns.
- Limit the email addresses that email can be addressed to and automatically block email from specific senders.

SMTP proxies are specialized Mail Transfer Agents (MTAs) that, similar to other types of proxy servers, pass SMTP sessions through to other MTAs without using the store-and-forward approach of a typical MTA. When an SMTP proxy receives a connection, it initiates another SMTP session to a destination MTA. Any errors or status information from the destination MTA will be passed back to the sending MTA through the proxy

SMTP proxies come in a few fundamental flavors:

- **Synchronous** - each SMTP client connection causes the proxy to establish a single connection with a downstream mail server.
- **Multiplexing** - the proxy establishes downstream connections to the mail server only as needed, and by intelligently juggling a pool of SMTP connections; this juggling protects the downstream mail server from excessive connection concurrency
- **Transparent** - the proxy is inserted into the network between clients and servers, masquerading itself in such a way that the client and server believe they are talking directly to each other, even though there is a proxy in the middle.

Unit 6
Internet and Intranet Systems Development

Intranet:

Intranet is the generic term for a collection of private computer networks within an organization. An intranet uses network technologies as a tool to facilitate communication between people or work groups to improve the data sharing capability and overall knowledge base of an organization's employees.

Intranets utilize standard network hardware and software technologies like Ethernet, WiFi, TCP/IP, Web browsers and Web servers. An organization's intranet typically includes Internet access but is firewalled so that its computers cannot be reached directly from the outside.

Intranets are generally used for four types of applications:

1) Communication and collaboration:
- send and receive e-mail, faxes, voice mail, and paging
- discussion rooms and chat rooms
- audio and video conferencing
- virtual team meetings and project collaboration
- online company discussions as events (e.g., IBM Jams)
- inhouse blogs

2) Web publishing
- develop and publish hyperlinked multi-media documents such as:
- policy manuals, company newsletters
- product catalogs
- technical drawings
- training material
- telephone directories

3) Business operations and management
- order processing
- inventory control
- production setup and control
- management information systems
- database access

4) Intranet portal management
- centrally administer all network functions including servers, clients, security, directories, and traffic

- give users access to a variety of internal and external business tools/applications
- integrate different technologies
- conduct regular user research to identify and confirm strategy (random sample surveys, usability testing, focus groups, in-depth interviews with wireframes, etc.)

Benefits of intranet:

- **Workforce productivity**: Intranets can help users to locate and view information faster and use applications relevant to their roles and responsibilities. With the help of a web browser interface, users can access data held in any database the organization wants to make available, anytime and — subject to security provisions — from anywhere within the company workstations, increasing employees' ability to perform their jobs faster, more accurately, and with confidence that they have the right information. It also helps to improve the services provided to the users.

- **Time**: Intranets allow organizations to distribute information to employees on an *as-needed* basis; Employees may link to relevant information at their convenience, rather than being distracted indiscriminately by electronic mail.

- **Communication**: Intranets can serve as powerful tools for communication within an organization, vertically and horizontally. From a communications standpoint, intranets are useful to communicate strategic initiatives that have a global reach throughout the organization. The type of information that can easily be conveyed is the purpose of the initiative and what the initiative is aiming to achieve, who is driving the initiative, results achieved to date, and who to speak to for more information. By providing this information on the intranet, staff have the opportunity to keep up-to-date with the strategic focus of the organization. Some examples of communication would be chat, email, and or blogs.

- **Web publishing** allows cumbersome corporate knowledge to be maintained and easily accessed throughout the company using hypermedia and Web technologies. Examples include: employee manuals, benefits documents, company policies, business standards, news feeds, and even training, can be accessed using common Internet standards (Acrobat files, Flash files, CGI applications). Because each business unit can update the online copy of a document, the most recent version is usually available to employees using the intranet.

- **Business operations and management**: Intranets are also being used as a platform for developing and deploying applications to support business operations and decisions across the internetworked enterprise.

- **Cost-effective**: Users can view information and data via web-browser rather than maintaining physical documents such as procedure manuals, internal phone list and requisition forms. This can potentially save the business money on printing, duplicating documents, and the environment as well as document maintenance overhead.

- **Enhance collaboration**: Information is easily accessible by all authorised users, which enables teamwork.

- **Cross-platform capability**: Standards-compliant web browsers are available for Windows, Mac, and UNIX.

- **Built for one audience**: Many companies dictate computer specifications which, in turn, may allow Intranet developers to write applications that only have to work on one browser (no cross-browser compatibility issues). Being able to specifically address your "viewer" is a great advantage.

- **Promote common corporate culture**: Every user has the ability to view the same information within the Intranet.

- **Immediate updates**: When dealing with the public in any capacity, laws, specifications, and parameters can change. Intranets make it possible to provide your audience with "live" changes so they are kept up-to-date, which can limit a company's liability.

- **Supports a distributed computing architecture**: The intranet can also be linked to a company’s management information system, for example a time keeping system.

Drawbacks of intranet

- it is an evolving technology that requires upgrades and could have software incompatibility problems
- security features can be inadequate
- inadequate system performance management and poor user support
- may not scale up adequately
- maintaining content can be time consuming
- some employees may not have PCs at their desks
- The aims of the organization in developing an intranet may not align with user needs

Protocols used in intranet systems:

An intranet uses the same concepts and technologies as the World Wide Web and Internet. This includes web browsers and servers running on the internet protocol suite and using Internet protocols such as FTP, TCP/IP, Simple Mail Transfer Protocol (SMTP) and so on.
Note: we have discussed all of these protocols already in previous units!!!

Intranet Network Infrastructure:

A network infrastructure is an interconnected group of computer systems linked by the various parts of a telecommunications architecture. **Specifically, this infrastructure refers to the organization of its various parts and their configuration — from individual networked computers to routers, cables, wireless access points, switches, backbones, network protocols, and network access methodologies.** Infrastructures can be either *open* or *closed*, such as the open architecture of the Internet or the closed architecture of a private *intranet*. They can operate over wired or wireless network connections, or a combination of both.

The simplest form of network infrastructure typically consists of one or more computers, a network or Internet connection, and a *hub* to both link the computers to the network connection and tie the various systems to each other. The hub merely links the computers, but does not limit data flow to or from any one system. To control or limit access between systems and regulate information flow, a switch replaces the hub to create network protocols that define how the systems communicate with each other. To allow the network created by these systems to communicate to others, via the network connection, requires a router, which bridges the networks and basically provides a common language for data exchange, according to the rules of each network.

Why Is the Network Infrastructure Important to Your Intranet?

An intranet is made up of two parts: **the applications (software / protocols) and the network infrastructure on which the applications run**. Applications— the visible part of an intranet —provide the functionality to improve productivity and lower costs. A wide spectrum of Internet/intranet applications is available from many vendors. **The network infrastructure includes the hardware—network interface cards (NICs), hubs, routers, switches, and servers—over which the applications run.** All network hardware is not the same, and an intranet is only as usable, reliable, and cost-effective as the hardware on which it runs. Crucial considerations in choosing appropriate hardware include:

- Bandwidth availability
- Reliability
- Value, in terms of both initial cost and ease of use and management
- Scalability, to ensure that present and future needs can be met

So as a part of network infrastructure, go through the above highlighted portions. I think you have studied those in data communication as well.

Intranet Implementation Guidelines:

When planning an intranet, there are a number of questions to be considered. These questions will set the tone for how you go about developing your intranet, help you establish guidelines.

1. What is your business case for building the intranet?
2. Who can publish to the intranet?
3. What types of content can be published?

Content Design, Development, Publishing and Management:

Content is a substance, and information on the site should be relevant to the site and should target the area of the public that the website is concerned with.

Content Management:

Content management, or **CM**, is the set of processes and technologies that support the collection, managing, and publishing of information in any form or medium. In recent times this information is typically referred to as content or, to be precise, digital content. Digital content may take the form of text (such as electronic documents), multimedia files (such as audio or video files), or any other file type that follows a content lifecycle requiring management. A critical aspect of content management is the ability to manage versions of content as it evolves

Content management is an inherently collaborative process. It often consists of the following basic roles and responsibilities:

- Creator - responsible for creating and editing content.
- Editor - responsible for tuning the content message and the style of delivery, including translation and localization.
- Publisher - responsible for releasing the content for use.
- Administrator - responsible for managing access permissions to folders and files, usually accomplished by assigning access rights to user groups or roles. Admins may also assist and support users in various ways.
- Consumer, viewer or guest- the person who reads or otherwise takes in content after it is published or shared.

A content management system is a set of automated processes that may support the following features:

- Import and creation of documents and multimedia material.
- Identification of all key users and their roles.
- The ability to assign roles and responsibilities to different instances of content categories or types.
- Definition of workflow tasks often coupled with messaging so that content managers are alerted to changes in content.
- The ability to track and manage multiple versions of a single instance of content.
- The ability to publish the content to a repository to support access to the content. Increasingly, the repository is an inherent part of the system, and incorporates enterprise search and retrieval.

Intranet Design with Open source Tools: DRUPAL, JUMLA:

Druple:

Drupal content management system or Drupal CMS is an open source modular framework and Content Management System written in PHP that can be used to manage your website or blog from an online interface. Drupal is used as a "back end" system for many different types of websites; ranging from a small personal blog to large corporate sites. It allows an individual or a community of users to easily publish, manage and organize a wide variety of content on a website.

Joomla:

Joomla CMS is a web application that makes it easy for any person to build a website. A website created with custom Joomla design allows the user to take control of their website. The beauty of Joomla is that the designers can leverage the existing framework and user interface to deliver applications to the end users in a familiar, powerful environment. This process saves time as well as cuts the budget down.

Tunneling Protocols:

A tunneling protocol is the one utilized by computer networks in cases where the network protocol or the delivery protocol encapsulates an unsuited payload protocol at a peer level or lower than it. The protocol is termed as such because this appears as if it makes its way through the various types of packets. It is sometimes recognized with the name “encapsulation protocol” but this label is very vague for the reason that there are other network protocols which are also designed to perform the process of encapsulation.

Tunneling protocol is widely used in transmitting large amounts of protocols through the typical networks. In addition, it may serve as a medium for transferring virtual private networks (VPNs) that are already encrypted.

This protocol comes as an advantage since tunneling may be employed in transporting a payload over the mismatched delivery-network. Tunneling protocol is also helpful when it comes to presentation of a safe passageway over a suspicious-looking network.

In common cases, tunneling may differ with some other forms of layered protocol including TCP/IP and OSI. There are times when a delivery protocol functions at a more advanced level in the model compared to that of a payload protocol. Rarely, however, does both the delivery and payload protocol work at similar level.

Wrapping of protocols is a product of the mechanism performed by the conventional layered protocols. This works in line with the other models such as the OSI model and TCP/IP model, which does not belong in the category of protocols that carry out tunneling. There are different procedures that may be employed by these tunneling protocols so as to do its job successfully. One of which is the utilization of data encryption for the purpose of transferring a vulnerable payload protocol through a public network, in which the most common type is the Internet. Lastly, this process solely offers the functionality of the VPN.

Different types of VPN tunneling:

Two types of tunneling include voluntary and compulsory.

- **Voluntary VPN tunneling:** In this particular tunneling type, the VPN client sets up the connection. At first, the client establishes a connection with the network provider or the ISP. Later on, utilizing this live connection, it creates a tunnel to a particular VPN server.

- **Compulsory VPN tunneling:** The carrier network provider is responsible for managing the set up for VPN connection in this type of tunneling. It is quicker than its voluntary counterpart and can be established in just a single step as compared to the two-step process of the other one. This network device is known with varied other names as well such as Network Access Server (NAS), VPN Front End Processor (FEP) and Point of Presence Server (POS).

Example of VPN Tunneling:

The following steps illustrate the principles of a VPN client-server interaction in simple terms;

Assume a remote host with public IP address 1.2.3.4 wishes to connect to a server found inside a company network. The server has internal address 192.168.1.10 and is not reachable publicly. Before the client can reach this server, it needs to go through a VPN server / firewall device that has public IP address 5.6.7.8 and an internal address of 192.168.1.1. All data between the client and the server will need to be kept confidential; hence a secure VPN is used.

1. The VPN client connects to a VPN server via an external network interface.
2. The VPN server assigns an IP address to the VPN client from the VPN server's subnet. The client gets internal IP address 192.168.1.50, for example, and creates a virtual network interface through which it will send encrypted packets to the other tunnel endpoint (the device at the other end of the tunnel). (This interface also gets the address 192.168.1.50.)
3. When the VPN client wishes to communicate with the company server, it prepares a packet addressed to 192.168.1.10, encrypts it and encapsulates it in an outer VPN packet, say an IPSec packet. This packet is then sent to the VPN server at IP address 5.6.7.8 over the public Internet. The inner packet is encrypted so that even if someone intercepts the packet over the Internet, they cannot get any information from it. They can see that the remote host is communicating with a server/firewall, but none of the contents of the communication. The inner encrypted packet has source address 192.168.1.50 and destination address 192.168.1.10. The outer packet has source address 1.2.3.4 and destination address 5.6.7.8.
4. When the packet reaches the VPN server from the Internet, the VPN server unencapsulates the inner packet, decrypts it, finds the destination address to be 192.168.1.10, and forwards it to the intended server at 192.168.1.10.
5. After some time, the VPN server receives a reply packet from 192.168.1.10, intended for 192.168.1.50. The VPN server consults its routing table, and sees this packet is intended for a remote host that must go through VPN.
6. The VPN server encrypts this reply packet, encapsulates it in a VPN packet and sends it out over the Internet. The inner encrypted packet has source address 192.168.1.10 and destination address 192.168.1.50. The outer VPN packet has source address 5.6.7.8 and destination address 1.2.3.4.
7. The remote host receives the packet. The VPN client unencapsulates the inner packet, decrypts it, and passes it to the appropriate software at upper layers.

Overall, it is as if the remote computer and company server are on the same 192.168.1.0/24 network.

Tunneling protocols for VPN:

Five prominent tunneling protocols are readily used to establish successful VPN connection that includes **PPTP VPN**, **L2TP VPN**, **IPSec**, **SSH VPN** and **SSTP VPN**. Let us discuss them in brief.

- **Point-to-Point Tunneling Protocol (PPTP):** It is among the most widely preferred tunneling protocols and is available as a built in facility in almost all the windows OS versions. It utilizes a control channel over TCP to encapsulate PPP data packets. It itself does not provide authentication or encryption features but is dependent on

the Point-to-Point Protocol (PPP). Still, it is the best to provide high security level and remote access during a VPN connection.

- **Layer 2 Tunneling Protocol** (**L2TP**): It is also a capable tunneling protocol that supports VPN connection. Like PPTP, it also does not offer confidentiality and encryption on its own but depends on an encryption protocol for the same that it leverages to assure privacy within the tunnel. It has been developed out of the combination of L2F and PPTP taking their best features and exists at the data link layer in the OSI model, same as PPTP.

- **IP Security (IPSec):** It is better known as an assemblage of varied protocols instead of being a single one. When combines with PPTP or L2TP, it provides accomplished encryption solutions and secures the data transfer within a VPN tunnel. It exists at the Layer 3, i.e. Network Layer of the OSI model.

- **Secure Shell (SSH):** This is a new protocol as compared all the rest ones and seeks assistance of an encrypted channel to transfer the unencrypted data via a secure network efficiently. In locations where VPN is blocked, SSH somehow manages to hide the identity of users and prevents their IP address from being blocked.

- **Secure Socket Tunneling Protocol:** This is yet another effective protocol that makes way for secure data transfer from network server to a remote terminal and vice versa, thereby bypassing all the firewalls and web proxies coming in its way. To accomplish such a successful data transaction, it utilizes HTTPs protocol and is very useful at places where PPTP or L2TP/IPSec cease to perform as per expected.

Unit 7

Internet and Intranet Systems Development

Email:

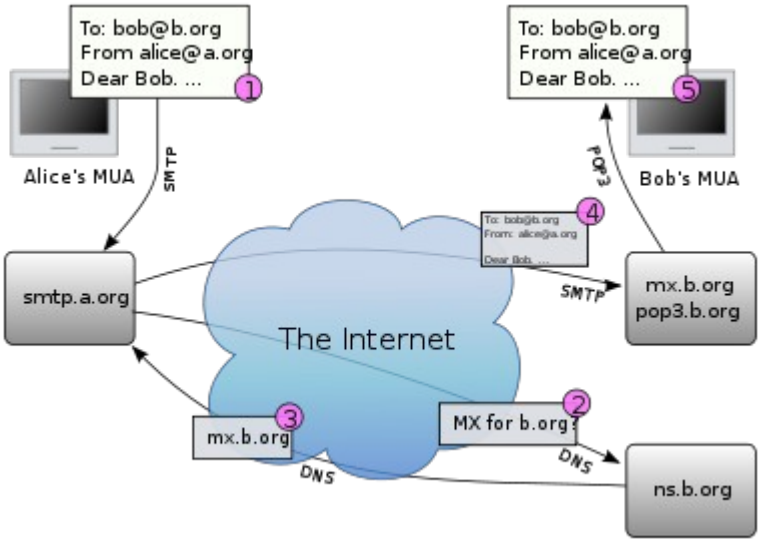
Electronic mail, also known as **email** or **e-mail**, is a method of exchanging digital messages from an author to one or more recipients. Modern email operates across the Internet or other computer networks. Some early email systems required that the author and the recipient both be online at the same time, in common with instant messaging. Today's email systems are based on a store-and-forward model. Email servers accept, forward, deliver and store messages. Neither the users nor their computers are required to be online simultaneously; they need connect only briefly, typically to an email server, for as long as it takes to send or receive messages.

An Internet email message consists of three components, the message *envelope*, the message *header*, and the message *body*. The message header contains control information, including, minimally, an originator's email address and one or more recipient addresses. Usually descriptive information is also added, such as a subject header field and a message submission date/time stamp.

Network-based email was initially exchanged on the ARPANET in extensions to the File Transfer Protocol (FTP), but is now carried by the Simple Mail Transfer Protocol (SMTP). In the process of transporting email messages between systems, SMTP communicates delivery parameters using a message *envelope* separate from the message (header and body) itself.

Operation Overview:

Following it shows a typical sequence of events that takes place when Alice composes a message using her mail user agent (MUA). She enters the email address of her correspondent, and hits the "send" button.



1. Her MUA formats the message in email format and uses the Submission Protocol (a profile of the Simple Mail Transfer Protocol (SMTP),) to send the message to the local mail submission agent (MSA), in this case smtp.a.org, run by Alice's internet service provider (ISP).

2. The MSA looks at the destination address provided in the SMTP protocol (not from the message header), in this case bob@b.org. An Internet email address is a string of the formlocalpart@exampledomain. The part before the @ sign is the *local part* of the address, often the username of the recipient, and the part after the @ sign is a domain name or a fully qualified domain name. The MSA resolves a domain name to determine the fully qualified domain name of the mail exchange server in the Domain Name System (DNS).

3. The DNS server for the b.org domain, ns.b.org, responds with any MX records listing the mail exchange servers for that domain, in this case mx.b.org, a message transfer agent (MTA) server run by Bob's ISP. A **mail exchanger record** (**MX record**) is a type of resource record in the Domain Name System that specifies a mail server responsible for accepting email messages on behalf of a recipient's domain, and a preference value used to prioritize mail delivery if multiple mail servers are available.

4. smtp.a.org sends the message to mx.b.org using SMTP.

This server may need to forward the message to other MTAs before the message reaches the final message delivery agent (MDA).

1. The MDA delivers it to the mailbox of the user bob.
2. Bob presses the "get mail" button in his MUA, which picks up the message using either the Post Office Protocol (POP3) or the Internet Message Access Protocol (IMAP4).

Message Format:

The Internet email message format is now defined with multi-media content attachments collectively called *Multipurpose Internet Mail Extensions* or *MIME*. Internet email messages consist of two major sections:

Header – Structured into fields such as From, To, CC, Subject, Date, and other information about the email.

Body – The basic content, as unstructured text; sometimes contains a signature block at the end. This is exactly the same as the body of a regular letter.

The header is separated from the body by a blank line.

Message Header:

Each message has exactly one header, which is structured into fields. Each field has a name and a value. Email header fields can be multi-line, and each line should be at most 78 characters long and in no event more than 998 characters long.

The message header must include at least the following fields:

- ***From*:** The email address, and optionally the name of the author(s). In many email clients not changeable except through changing account settings.
- ***Date*:** The local time and date when the message was written. Like the *From:* field, many email clients fill this in automatically when sending. The recipient's client may then display the time in the format and time zone local to him/her.

The message header should include at least the following fields:

- ***Message-ID*:** Also an automatically generated field; used to prevent multiple delivery and for reference in In-Reply-To:
- ***In-Reply-To*:** Message-ID of the message that this is a reply to. Used to link related messages together. This field only applies for reply messages.

Common header fields for email include:

- ***To*:** The email address(es), and optionally name(s) of the message's recipient(s). Indicates primary recipients (multiple allowed), for secondary recipients (Cc: and Bcc:)
- ***Subject*:** A brief summary of the topic of the message. Certain abbreviations are commonly used in the subject, including "RE:" and "FW:".
- ***Bcc*:** Blind Carbon Copy; addresses added to the SMTP delivery list but not (usually) listed in the message data, remaining invisible to other recipients.
- ***Cc*:** Carbon copy; Many email clients will mark email in your inbox differently depending on whether you are in the To: or Cc: list.
- **Content-Type:** Information about how the message is to be displayed, usually a MIME type.
- ***Precedence*:** commonly with values "bulk", "junk", or "list"; used to indicate that automated "vacation" or "out of office" responses should not be returned for this mail, e.g. to prevent vacation notices from being sent to all other subscribers of a mailing list. Sendmail uses this header to affect prioritization of queued email, with "Precedence: special-delivery" messages delivered sooner. With modern high-bandwidth networks delivery priority is less of an issue than it once was. Microsoft Exchange respects a fine-grained automatic response suppression mechanism, the X-Auto-Response-Suppress header.

- ***References*:** Message-ID of the message that this is a reply to, and the message-id of the message the previous reply was a reply to, etc.
- ***Reply-To*:** Address that should be used to reply to the message.
- ***Sender*:** Address of the actual sender acting on behalf of the author listed in the From: field (secretary, list manager, etc.).
- ***Archived-At*:** A direct link to the archived form of an individual email message

Message Body:

Most modern graphic email clients allow the use of either plain text or HTML for the message body at the option of the user. HTML email messages often include an automatically generated plain text copy as well, for compatibility reasons.

Advantages of HTML include the ability to include in-line links and images, set apart previous messages in block quotes, wrap naturally on any display, use emphasis such as underlines and italics, and change font styles.

Client/server applications for accessing mail

Messages are exchanged between hosts using the Simple Mail Transfer Protocol with software programs called mail transfer agents (MTAs); and delivered to a mail store by programs called mail delivery agents (MDAs, also sometimes called local delivery agents, LDAs). Users can retrieve their messages from servers using standard protocols such as POP or IMAP.

Mail can be stored on the client, on the server side, or in both places. Standard formats for mailboxes include Maildir and mbox. Several prominent email clients use their own proprietary format and require conversion software to transfer email between them. Server-side storage is often in a proprietary format but since access is through a standard protocol such as IMAP, moving email from one server to another can be done with any MUA supporting the protocol.

Accepting a message obliges an MTA to deliver it, and when a message cannot be delivered, that MTA must send a bounce message back to the sender, indicating the problem.

File name extensions:

Upon reception of email messages, email client applications save messages in operating system files in the file system. Some clients save individual messages as separate files, while others use various database formats, often proprietary, for collective storage. A historical standard of storage is the *mbox* format. The specific format used is often indicated by special filename extensions:

eml: Used by many email clients including Microsoft Outlook Express, Windows Mail and Mozilla Thunderbird. The files are plain text in MIME format, containing the email header as well as the message contents and attachments in one or more of several formats.

emlx: Used by Apple Mail.

msg: Used by Microsoft Office Outlook and OfficeLogic Groupware.

mbx: Used by Opera Mail, KMail, and Apple Mail based on the mbox format.

Types:

Web-Based Email (Webmail): This is the type of email that most users are familiar with. Many free email providers host their serves as web-based email. (e.g.: Hotmail, Yahoo, Gmail, AOL). This allows users to log into the email account by the help of a Internet browser to send and receive their email. Its main disadvantage is the need to be connected to the internet while using it. There exist also other software tools to integrate parts of the webmail functionality into the OS (e.g. creating messages directly from third party applications via MAPI).

POP3 Email Services: POP3 is the acronym for Post Office Protocol 3. It is a leading email account type on the Internet. In a POP3 email account, your email messages are downloaded to your computer and then they are deleted from the mail server. It is difficult to save and view your messages on multiple computers. Also, the messages you send from the computer are not copied to the Sent Items folder on the computers. The messages are deleted from the server to make room for more incoming messages. POP supports simple download-and-delete requirements for access to remote mailboxes (termed maildrop in the POP RFC's). Although most POP clients have an option to leave messages on the server after downloading a copy of them, most e-mail clients using POP3 simply connect, retrieve all messages, store them on the user's computer as new messages, delete them from the server, and then disconnect. Other protocols, notably IMAP, (Internet Message Access Protocol) provide more complete and complex remote access to typical mailbox operations. Many e-mail clients support POP as well as IMAP to retrieve messages; however, fewer Internet Service Providers (ISPs) support IMAP.

IMAP Email Servers: IMAP refers to Internet Message Access Protocol. It is an alternate to the POP3 email. With an Internet Message Protocol (IMAP) account, you have access to mail folders on the mail server and you can use any computer to read your messages wherever you are. It shows the headers of your messages, the sender and it is subject and choose to download only those messages you need to read. Usually mail is saved on the mail server, therefore it is safer and it is backed up on the email server.

MAPI Email Servers: Messaging Application Programming Interface (MAPI) is a messaging architecture and a Component Object Model based API for Microsoft Windows.

MIME: Multipurpose Internet Mail Extensions (MIME):

It is an Internet standard that extends the format of email to support:

- Text in character sets other than ASCII
- Non-text attachments
- Message bodies with multiple parts
- Header information in non-ASCII character sets

The basic Internet email transmission protocol, SMTP, supports only 7-bit ASCII characters. MIME defines mechanisms for sending other kinds of information in email. These include text in languages other than English using character encodings other than ASCII, and 8-bit binary content such as files containing images, sounds, movies, and computer programs. Mapping messages into and out of MIME format is typically done automatically by an email client or by mail servers when sending or receiving Internet (SMTP/MIME) email.

Gopher:

The **Gopher protocol** is a TCP/IP application layer protocol designed for distributing, searching, and retrieving documents over the Internet. Strongly oriented towards a menu-document design, the Gopher protocol presented an attractive alternative to the World Wide Web in its early stages, but ultimately failed to achieve popularity.

Gopher is a protocol system, which in advance of the World Wide Web, allowed server based text files to be hierarchically organised and easily viewed by end users who accessed the server using Gopher applications on remote computers. Initially Gopher browsers could only display text-based files before developments such as HyperGopher, which were able to handle simple graphic formats though they were never used on a widespread basis

as by this time the World Wide Web and its Hypertext Transfer Protocol (HTTP) were gaining in popularity, and had similar and more extensive functions.

Its central goals were,

- A file-like hierarchical arrangement that would be familiar to users.
- A simple syntax.
- A system that can be created quickly and inexpensively.
- Extending the file system metaphor, such as searches.

Gopher uses a server-client protocol to access and manage the files. Gopher clients use the TCP port 70 to connect to the Gopher servers. The server sends a list of files available, with each line having a standard code that identifies the type of file. It uses 0 for files, 1 for directories and 7 for search services. Other code parts include the selector string, which is the part that needs to be sent back to the server in order to get the requested resource, the server port name and the port number.

Gopher faced some serious competition from the Hypertext Transfer Protocol, or HTTP, since the linking could be done from directly within the documents instead of having to be done from the servers. HTTP also overtakes Gopher when it comes to displaying information, because unlike Gopher which only uses text, the HTTP protocol can present the information in any shape and with pictures.

The main advantage of Gopher is its simplicity. It assures compatibility between platforms, and the bandwidth usage is minimal. The fact that linking needed to be done directly by the servers was considered by some a cleaner approach.

The Gopher protocol is now obsolete and it's not used anymore. Due to its limitations, it lost the competition against the Internet. All files in the system were converted to be compatible with the HTTP protocol, and most of the Gopher servers are offline. The final hit for the Gopher was delivered in 2002 when a security breach related to Gopher was found in Internet Explorer, and Microsoft decided to retract its support for the Gopher protocol.

Multimedia and Digital Video/Audio Broadcasting:

Digital Video Broadcasting (**DVB**) is a suite of internationally accepted open standards for digital television. DVB systems distribute data using a variety of approaches, including:

- Satellite: DVB-S, DVB-S2 and DVB-SH
- DVB-SMATV for distribution via SMATV
- Cable: DVB-C, DVB-C2
- Terrestrial television: DVB-T, DVB-T2
- Digital terrestrial television for handhelds: DVB-H, DVB-SH
- Microwave: using DTT (DVB-MT), the MMDS (DVB-MC), and/or MVDS standards (DVB-MS)

These standards define the physical layer and data link layer of the distribution system. Devices interact with the physical layer via a synchronous parallel interface (SPI), synchronous serial interface (SSI), or asynchronous serial interface (ASI). All data is transmitted in MPEG transport streams with some additional constraints (DVB-MPEG).

Digital Video Broadcasting (DVB) is a set of standards that define digital broadcasting using existing satellite, cable, and terrestrial infrastructures. In the early 1990s, European broadcasters, consumer equipment manufacturers, and regulatory bodies formed the European Launching Group (ELG) to discuss introducing digital television (DTV) throughout Europe.

The main forms of DVB are summarised below:

DVB STANDARD	MEANING	DESCRIPTION
DVB-C	Cable	The standard for delivery of video service via cable networks.
DVB-H	Handheld	DVB services to handheld devices, e.g. mobile phones, etc
DVB-RSC	Return satellite channel	Satellite DVB services with a return channel for interactivity.
DVB-S	Satellite services	DVB standard for delivery of television / video from a satellite.
DVB-SH	Satellite handheld	Delivery of DVB services from a satellite to handheld devices
DVB-S2	Satellite second generation	The second generation of DVB satellite broadcasting.
DVB-T	Terrestrial	The standard for Digital Terrestrial Television Broadcasting.

Digital Audio Broadcasting:

Traditionally radio programmes were broadcast on different frequencies via FM and AM, and the radio had to be tuned into each frequency, as needed. This used up a comparatively large amount of spectrum for a relatively small number of stations, limiting listening choice. DAB is a digital radio broadcasting system that through the application of multiplexing and compression combines multiple audio streams onto a relatively narrow band centered on a single broadcast frequency called a DAB ensemble.

Digital audio broadcasting (DAB), also known as digital radio and high-definition radio, is audio broadcasting in which analog audio is converted into a digital signal and transmitted on an assigned channel in the AM or (more usually) FM frequency range. DAB is said to offer compact disc (CD)- quality audio on the FM (frequency modulation) broadcast band and to offer FM-quality audio on the AM (amplitude modulation) broadcast band. The technology was first deployed in the United Kingdom in 1995, and has become common throughout Europe.

It is an entirely new system for broadcasting and receiving radio stations. As the name indicates signals are broadcast in a digital format to enable CD quality to be achieved.

Within an overall target bit rate for the DAB ensemble, individual stations can be allocated different bit rates. The number of channels within a DAB ensemble can be increased by lowering average bit rates, but at the expense of the quality of streams. Error correction under the DAB standard makes the signal more robust but reduces the total bit rate available for streams.

Advantages of DAB over analogue systems:

- **Improved features for users:** DAB radios automatically tune to all the available stations, offering a list for the user to select from.
- **More stations:** DAB is not more bandwidth efficient than analogue measured in programmes per MHz of a specific transmitter
- **Reception quality:** The DAB standard integrates features to reduce the negative consequences of multipath fading and signarl noise, which afflict existing analogue systems.
- **Less pirate interference:** The specialized nature and cost of DAB broadcasting equipment provide barriers to pirate radio stations broadcasting on DAB.
- **Variable bandwidth**
- **Transmission costs:** It is common belief that DAB is more expensive to transmit than FM. It is true that DAB uses higher frequencies than FM and therefore there is a need to compensate with more transmitters, higher radiated powers, or a combination, to achieve the same coverage. A DAB network is also more expensive than an FM network. However, the last couple of years has seen significant improvement in power efficiency for DAB-transmitters.

Internet Relay Chat (IRC):

Internet Relay Chat is a method to broadcast and receive live, synchronous, messages. **Internet Relay Chat** (**IRC**) is a protocol for real-time Internet text messaging (chat) or synchronous conferencing. It is mainly designed for group communication in discussion forums, called *channels*, but also allows one-to-one communication via private message as well as chat and data transfer, including file sharing.

To join an IRC discussion, you need an *IRC client* and Internet access. The IRC client is a program that runs on your computer and sends and receives messages to and from an IRC *server*. The IRC server, in turn, is responsible for making sure that all messages are broadcast to everyone participating in a discussion. There can be many discussions going on at once; each one is assigned a unique *channel*.

IRC (Internet Relay Chat) is a real-time multi-user messaging system. Users connect to an IRC server and join one or multiple channels or enter into one-on-one chats with individual users. Users type in messages (up to a few hundred characters long, I'm not sure what the actual limits are) and send them to the channel they are in. Other people in that channel then receive the messages that everyone else has sent. The server is a central point of contact for the channel and serves as a relay for the messages from each user, thus the name. Additionally, servers can be chained together, relaying their traffic back and forth. Typically, an IRC client will render the conversations in a channel as an upward scrolling list of messages in

chronological order, with each message on one or several lines (depending on length) and prefixed with some username identifier and perhaps other information (such as a time stamp) depending on the individual configuration of the client.

Components of IRC:

Servers: The server forms the backbone of IRC as it is the only component of the protocol which is able to link all the other components together: it provides a point to which clients may connect to talk to each other, and a point for other servers to connect to. The server is also responsible for providing the basic services defined by the IRC protocol.

Clients: A client is anything connecting to a server that is not another server. There are two types of clients which both serve a different purpose.

User Clients: User clients are generally programs providing a text based interface that is used to communicate interactively via IRC. This particular type of clients is often referred as "users".

Service Clients: Unlike users, service clients are not intended to be used manually nor for talking. They have a more limited access to the chat functions of the protocol, while optionally having access to more private data from the servers. Services are typically automatons used to provide some kind of service (not necessarily related to IRC itself) to users. An example is a service collecting statistics about the origin of users connected on the IRC network.

Architecture of IRC:

An IRC network is defined by a group of servers connected to each other. A single server forms the simplest IRC network. The only network configuration allowed for IRC servers is that of a spanning tree where each server acts as a central node for the rest of the network it sees.

The IRC protocol provides no mean for two clients to directly communicate. All communication between clients is relayed by the server(s).

IRC Protocol Services:

This section describes the services offered by the IRC protocol. The combinations of these services allow real-time conferencing.

Client Locator: To be able to exchange messages, two clients must be able to locate each other. Upon connecting to a server, a client registers using a label which is then used by other servers and clients to know where the client is located. Servers are responsible for keeping track of all the labels being used.

Message Relaying: The IRC protocol provides no mean for two clients to directly communicate. All communication between clients is relayed by the server(s).

Channel Hosting and Management: A channel is a named group of one or more users which will all receive messages addressed to that channel. A channel is characterized by its name and current members, it also has a set of properties which can be manipulated by (some of) its members. Channels provide a mean for a message to be sent to several clients. Servers host channels, providing the necessary message multiplexing. Servers are also responsible for managing channels by keeping track of the channel members. The exact role of servers is defined in "Internet Relay Chat: Channel Management".

Broadband Communication:

The term ***broadband*** refers to a telecommunications signal or device of greater bandwidth, in some sense, than another standard or usual signal or device (and the broader the band, the greater the capacity for traffic).

XDSL:

XDSL Refers collectively to all types of ***d****igital* ***s****ubscriber* ***l****ines*, the two main categories being ADSL and SDSL. Two other types of xDSL technologies are *High-data-rate DSL (HDSL)* and *Very high DSL (VDSL)*.

DSL technologies use sophisticated modulation schemes to pack data onto copper wires. They are sometimes referred to as last-mile technologies because they are used only for connections from a telephone switching station to a home or office, not between switching stations.

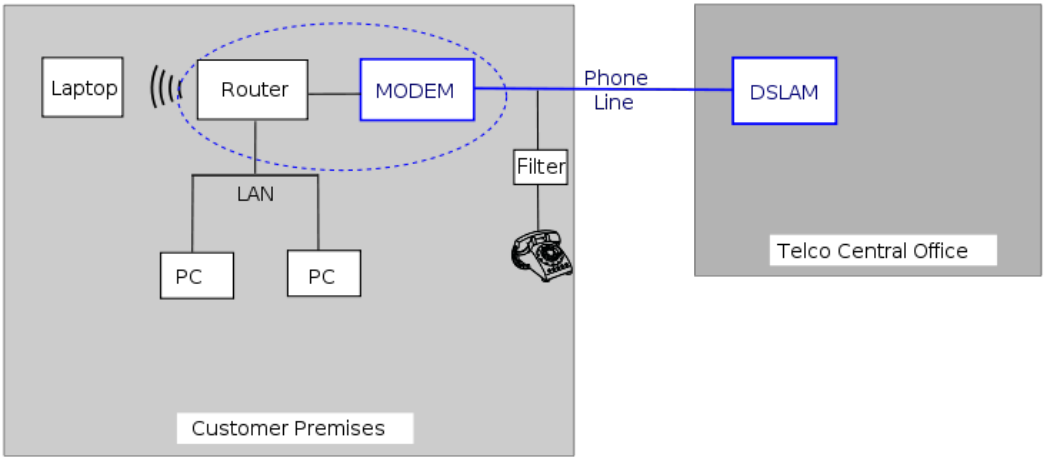
Digital subscriber line (**DSL**, originally **digital subscriber loop**) is a family of technologies that provide internet access by transmitting digital data over the wires of a local telephone network. In telecommunications marketing, the term DSL is widely understood to mean asymmetric digital subscriber line (ADSL), the most commonly installed DSL technology. DSL service is delivered simultaneously with wired telephone service on the same telephone line. This is possible because DSL uses higher frequency bands for data separated by filtering. On the customer premises, a DSL filter on each outlet removes the high frequency interference, to enable simultaneous use of the telephone and data.

The data bit rate of consumer DSL services typically ranges from 256 kbit/s to 40 Mbit/s in the direction to the customer (downstream), depending on DSL technology, line conditions, and service-level implementation. In ADSL, the data throughput in the upstream direction, (the direction to the service provider) is lower, hence the designation of *asymmetric* service. In symmetric digital subscriber line (SDSL) services, the downstream and upstream data rates are equal.

xDSL is similar to ISDN(***i****ntegrated* ***s****ervices* ***d****igital* ***n****etwork*) inasmuch as both operate over existing copper telephone lines (***p****lain* ***o****ld* ***t****elephone* ***s****ervice*) and both require the short runs to

a central telephone office (usually less than 20,000 feet). However, xDSL offers much higher speeds - up to 32 Mbps for upstream traffic, and from 32 Kbps to over 1 Mbps for downstream traffic.

Fig: DSL Connection



Benefits

- **High-speed data service:** DSL typically >10x faster than 56-kbps analog modem
- **Always on connection:** No need to "dial-up"
- **Uses existing copper wires:** Co-exists w/ POTS service
- **Reasonably priced today and getting cheaper**

Applications

- **High speed Internet access**
- **Multimedia, Long distance learning, gaming**
- **Video on Demand**
- **VPN**
- **VoDSL**

ADSL:

Asymmetric Digital Subscriber Line (**ADSL**) is a type of digital subscriber line technology, a data communications technology that enables faster data transmission over copper telephone lines than a conventional voiceband modem can provide. It does this by utilizing frequencies that are not used by a voice telephone call. A splitter, or DSL filter, allows a single telephone connection to be used for both ADSL service and voice calls at the same time. ADSL can generally only be distributed over short distances from the telephone exchange (the last mile), typically less than 4 kilometres (2 mi), but has been known to exceed 8 kilometres (5 mi) if the originally laid wire gauge allows for further distribution.

At the telephone exchange the line generally terminates at a digital subscriber line access multiplexer (DSLAM) where another frequency splitter separates the voice band signal for the conventional phone network. Data carried by the ADSL are typically routed over the telephone company's data network and eventually reach a conventional Internet Protocol network.

SDSL:

It is the Short for **Symmetric Digital Subscriber Line**, a technology that allows more data to be sent over existing copper telephone lines (POTS). SDSL supports data rates up to 3 Mbps. SDSL works by sending digital pulses in the high-frequency area of telephone wires and can not operate simultaneously with voice connections over the same wires. SDSL requires a special SDSL modem. SDSL is called *symmetric* because it supports the same data rates for upstream and downstream traffic.

ADSL Vs. SDSL

ADSL (Asymmetric Digital Subscriber Line) and SDSL (Symmetric Subscriber Digital Subscriber Line) are the two major groups when it comes to broadband internet connections. The most major difference between these two groups is in how much bandwidth they allocate to the user. Since SDSL is symmetric. It offers equal download and upload speeds to the user while, although the download speed is also very high for ADSL, the upload speed can be significantly slower.

The most major advantage that ADSL has is the ability to have the DSL and a telephone unit on the same two wires and they can be used simultaneously. This is because ADSL does not take up the entire bandwidth. SDSL uses the whole bandwidth and leaves no room for a telephone unit. The bandwidth for the telephone is allocated to the upload speed and explains the much higher bandwidth despite using the same two wires.

Cable Internet:

In telecommunications, **cable Internet access**, often shortened to **cable Internet** or simply **cable**, is a form of broadband Internet access that uses the cable television infrastructure. Like digital subscriber line and fiber to the premises services, cable Internet access provides network edge connectivity (last mile access) from the Internet service provider to an end user. It is integrated into the cable television infrastructure analogously to DSL which uses the existing telephone network. Cable TV networks and telecommunications networks are the two predominant forms of residential Internet access. Recently, both have seen increased competition from fiber deployments, wireless, and mobile networks.

Broadband cable Internet access requires a cable modem at the customer's premises and a cable modem termination system at a cable operator facility, typically a cable television headend. The two are connected via coaxial cable or a Hybrid Fiber Coaxial (HFC) plant. While access networks are sometimes referred to as *last-mile* technologies, cable Internet systems can typically operate where the distance between the modem and the termination system is up to 100 miles (160 km). If the HFC network is large, the cable modem termination system can be grouped into hubs for efficient management.

Downstream, the direction toward the user, bit rates can be as much as 400Mbit/s for business connections, and 100Mbit/s for residential service in some countries. Upstream traffic, originating at the user, ranges from 384kbit/s to more than 20Mbit/s. One downstream channel can handle hundreds of cable modems. As the system grows, the cable modem termination system (CMTS) can be upgraded with more downstream and upstream ports, and grouped into hubs CMTS for efficient management.

VOIP:

Voice over IP (**VoIP**, or **voice over Internet Protocol**) commonly refers to the communication protocols, technologies, methodologies, and transmission techniques involved in the delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet. Other terms commonly associated with VoIP are *IP telephony*, *Internet telephony*, *voice over broadband* (VoBB), *broadband telephony*, *IP communications*, and *broadband phone*.

Internet telephony refers to communications services —voice, fax, SMS, and/or voice-messaging applications— that are transported via the Internet, rather than the public switched telephone network (PSTN). The steps involved in originating a VoIP telephone call are signaling and media channel setup, digitization of the analog voice signal, encoding, packetization, and transmission as Internet Protocol (IP) packets over a packet-switched network. On the receiving side, similar steps (usually in the reverse order) such as reception of the IP packets, decoding of the packets and digital-to-analog conversion reproduce the original voice stream

Early providers of voice over IP services offered business models (and technical solutions) that mirrored the architecture of the legacy telephone network. Second generation providers, such as Skype have built closed networks for private user bases, offering the

benefit of free calls and convenience, while denying their users the ability to call out to other networks. This has severely limited the ability of users to mix-and-match third-party hardware and software. Third generation providers, such as Google Talk have adopted the concept of Federated VoIP - which is a complete departure from the architecture of the legacy networks. These solutions typically allow arbitrary and dynamic interconnection between any two domains on the Internet whenever a user wishes to place a call.

VoIP systems employ session control protocols to control the set-up and tear-down of calls as well as audio codecs which encode speech allowing transmission over an IP network as digital audio via an audio stream. The choice of codec varies between different implementations of VoIP depending on application requirements and network bandwidth; some implementations rely on narrowband and compressed speech, while others support high fidelity stereo codecs.

. Examples of the network protocols used to implement VoIP include:

- H.323
- Media Gateway Control Protocol (MGCP)
- Session Initiation Protocol (SIP)
- Real-time Transport Protocol (RTP)
- Session Description Protocol (SDP)

There are several advantages to using voice over IP. The biggest single advantage VoIP has over standard telephone systems is cost. In addition, international calls using VoIP are usually very inexpensive. One other advantage, which will become much more pronounced as VoIP use climbs, calls between VoIP users are usually free. Using services such as TrueVoIP, subscribers can call one another at no cost to either party.

How Does VoIP Work?

VoIP phone service (Voice over IP; also known as digital phone service, digital telephony, or broadband phone) replaces your phone line with a high-speed Internet connection. It's that simple.

While traditional telephone service compresses your voice into a frequency on a wire, VoIP compresses the sound of your voice into packets of data. In milliseconds, these data packets are sent over the Internet. When the data reaches the final destination, it is converted back to sound. If use VoIP to call someone on the traditional phone network (the "PSTN" or Public Switched Telephone Network), the VoIP call is converted to sound once it reaches the network and the call is routed normally. The difference is that you've paid a lot less for that call.

GSM VOIP (GOIP):-

It bridges the GSM services and the IP networks. The use of GSM Terminals together with VoIP Networks has significantly increased in the last years and continues to rise. GSM/VoIP Network configurations permit service providers to bypass the local Wired Public Switched Telephone Network (PSTN) in communicating with a GSM mobile handset. Thus, the excessive rates charged by telephone companies are avoided.

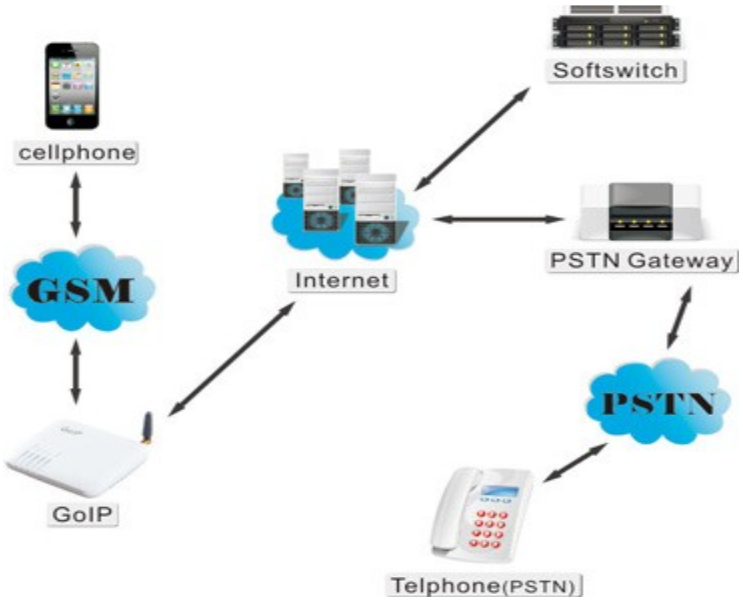
A VoIP GSM Gateway enables direct routing between IP, digital, analog and GSM networks. With these devices (fixed cellular terminals) companies can significantly reduce the money they spend on telephony, especially the money they spend on calls from IP to GSM. The core idea behind cost saving with VoIP GSM Gateways is Least Cost Routing (LCR).

Through least cost routing the gateways select the most cost-effective telephone connection. They check the number which is dialed as well as rate information which is stored in an internal routing table. Because several SIM cards and GSM modules are integrated within the VOIP GSM Gateway it is able to make relatively cheaper GSM to GSM calls instead of expensive IP to GSM calls.

VoIP GSM Gateways are IP-based systems that enable users to make cost effective calls from an IP phone to a GSM network through LCR (Least Cost Routing). The several SIM cards and GSM modules are integrated within the VOIP GSM Gateway. Often used in business and considered a method of significant savings over traditional telephony, VoIP GSM Gateways enables connections between IP, digital, analog and GSM networks.

The VOIP GSM Gateway is generally useful for the business enterprises or the users with the IP Telephone systems using VOIP enabled through SIP or H.323 protocol at the multiple locations who want to share the GSM Gateway connections. The VOIP GSM Gateway enables the VOIP users to make calls to the mobile phone networks at the reduced rates without needing to convert the call to ISDN, PSTN or any other traditional switching mode to use the traditional GSM Gateway equipment. The VoIP GSM Gateway normally enables both inbound and outbound VoIP and Cellular calls through one compact box.

Examples of some of the well known VoIP GSM Gateways suppliers include Hypermedia System, 2N Telekomunikace, Eurodesig BG, NovaTec, PorTech etc



IP Interconnection:

Interconnection links networks so as to enable the customers of one operator to establish and maintain communications with the customers of another operator.

IP Interconnection is defined as that which comprises the physical and logical interconnection of carrier networks required to initiate, terminate and/or exchange Managed VoIP services traffic and associated features and functions. Standards for this type of interconnection invoke the well-known, structured approach to computer-to-computer communications known as the ISO Open Systems Interconnection Reference Model (the "OSI Model") as a means to explain individualized and composite functionality.

In order to understand IP Interconnection, one must first understand the concept of constructing a transmission unit (packet) by combining logical functions conducted by all layers, in a progressive manner, for transmission on one or more, physical interconnection link(s) between the parties' networks. Each layer of the receiving party's network then deconstructs the transmission unit, interpreting directives for it and passing the remainder upward to the higher levels for further processing.

IP Interconnection, then, is much more than just the physical continuity between two networks. It comprises the functional support for ALL layers of the OSI reference model according to standardized protocols and rules drafted to support the services in question, in this case, voice/video telephony-over-IP services traffic and associated features and functions.

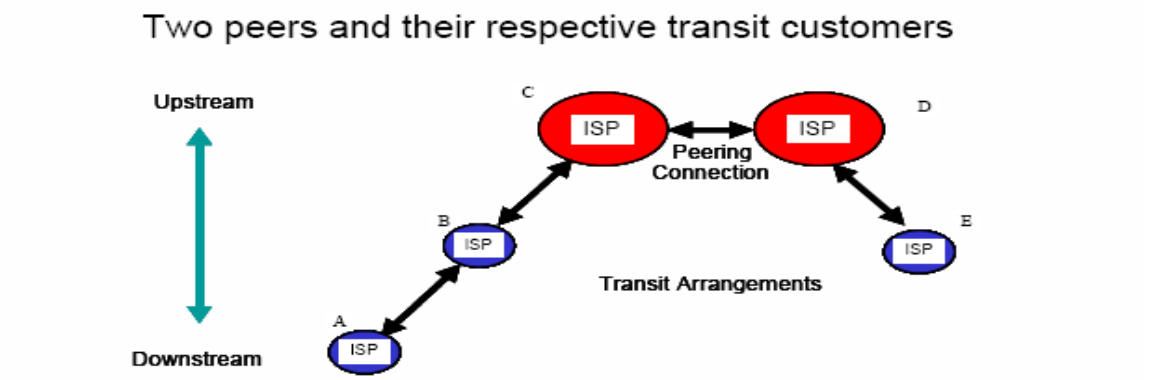
IP Interconnection will provide a significant reduction in the number of interconnection ports and facilities required to comprise ubiquitous interconnection.

The majority of connections between two IP-based networks today take place according to one of two basic interconnection models: **transit and peering.** Most readers are familiar with transit services, although not necessarily by that name: If you subscribe to an Internet access service, for example a DSL or cable broadband service, you are using transit. A transit service provider connects you, the transit customer, to the Internet for a fee. The service that a large Internet Service Provider (ISP) provides to a large business enterprise, or to a smaller or less well connected ISP, is on a larger scale but is not fundamentally different from your individual transit service.

Transit is an inherently asymmetric model. The transit provider (the ISP) carries traffic for the transit customer, but the transit customer is not under any obligation to carry traffic for the transit provider. The transit customer pays the provider, not the other way around.

With peering, ISPs exchange traffic for their respective customers (and for customers of their respective customers), but not for third parties. Peering is a substantially symmetric form of network interconnection.

Peering is an agreement between ISPs to carry traffic for each other and for their respective customers. Peering does not include the obligation to carry traffic to third parties. Peering is usually a bilateral business and technical arrangement, where two providers agree to accept traffic from one another, and from one another's customers (and thus from their customers' customers). Transit is an agreement where an ISP agrees to carry traffic on behalf of another ISP or end user. Transit is usually a bilateral business and technical arrangement, where one provider (the transit provider) agrees to carry traffic to third parties on behalf of another provider or an end user (the customer).



Parties	*Interconnection Arrangement*	*Typical Nature of Agreement*	*Typical Commercial Arrangements*
A – B	Transit	Bilateral	Payment reflects capacity, and may reflect volume of traffic or near-peak traffic level.
B – C	Transit	Bilateral	
E – D	Transit	Bilateral	
C – D	Peering	Bilateral	Often done without payment

Data Centers:

A **data center** or **computer centre** (also **datacenter**) is a facility used to house computer systems and associated components, such as telecommunications and storage systems. It generally includes redundant or backup power supplies, redundant data communications connections, environmental controls (e.g., air conditioning, fire suppression) and security devices.

IT operations are a crucial aspect of most organizational operations. One of the main concerns is **business continuity**; companies rely on their information systems to run their operations. If a system becomes unavailable, company operations may be impaired or stopped completely. It is necessary to provide a reliable infrastructure for IT operations, in order to minimize any chance of disruption. Information security is also a concern, and for this reason a data center has to offer a secure environment which minimizes the chances of a security breach. A data center must therefore keep high standards for assuring the integrity and functionality of its hosted computer environment. This is accomplished through redundancy of both fiber optic cables and power, which includes emergency backup power generation.

The "lights-out" data center, also known as a darkened or a dark data center, is a data center that, ideally, has all but eliminated the need for direct access by personnel, except under extraordinary circumstances. Because of the lack of need for staff to enter the data center, it can be operated without lighting. All of the devices are accessed and managed by remote systems, with automation programs used to perform unattended operations. In addition to the energy savings, reduction in staffing costs and the ability to locate the site further from population centers, implementing a lights-out data center reduces the threat of malicious attacks upon the infrastructure.

There is a trend to modernize data centers in order to take advantage of the performance and energy efficiency increases of newer IT equipment and capabilities, such as cloud computing. This process is also known as data center transformation

Three main elements of datacenters consist of:

1. hardware (servers, switches, routers, modems, and network utilities),
2. utilities required for environmental controls (including temperature control, humidity, Closed Circuit Cameras, and fire alarms),
3. Information security (such as using hardware and software and security procedures and the appropriate network design for this purpose to materialize)

Presently, the companies and the organizations use one of the three following methods to save their information and applications:

1. Use foreign datacenter services
2. Keep servers within their organizations
3. Use domestic datacenter services

In general, data centers can be broken down into three types- Internet **Data center (IDC)**, **Storage area network (SAN)** and **Enterprise data center (EDC)**.

- An Internet data center (IDC) is a facility that provides data and Internet services for other companies

- A Storage Area Network (SAN) is a network of interconnected storage devices and data servers usually located within an enterprise data center or as an off-site facility offering leased storage space.

- An Enterprise data center (EDC) is the central processing facility for an enterprise's computer network

Besides providing computing and storage resources on demand, another important aspect/role of Data Centre is to provide **data protection.** Therefore, Data Centers need to have strong **state-of-the-art backup and recovery and vaulting solutions** in place.

Data Warehousing:

Packet Clearing House:

Packet Clearing House is a non-profit research institute that supports operations and analysis in the areas of Internet traffic exchange, routing economics, and global network development. It has since grown to become a leading proponent of neutral independent network interconnection and provider of route-servers at major exchange points worldwide. PCH provides equipment, training, data, and operational support to organizations and individual researchers seeking to improve the quality, robustness, and accessibility of the Internet.

PCH works closely with the United States Telecommunications Training Institute (USTTI) to offer courses on telecommunications regulation, Internet infrastructure construction and management, domain name system management, and Internet security coordination, three times yearly in Washington D.C., in addition to the eighty to one hundred workshops PCH teaches on-location throughout the world each year.

Unified Messaging:

Unified Messaging is the integration of different electronic messaging and communications media (e-mail, SMS, Fax, voicemail, video messaging, etc.) technologies into a single interface, accessible from a variety of different devices. While traditional communications systems delivered messages into several different types of stores such as voicemail systems, e-mail servers, and stand-alone fax machines, with Unified Messaging all types of messages are stored in one system. Voicemail messages, for

example, can be delivered directly into the user's inbox and played either through a headset or the computer's speaker. This simplifies the user's experience (only one place to check for messages) and can offer new options for workflow such as appending notes or documents to forwarded voicemails.

Unified Messaging is an indistinct term that can refer to the typical definition of simple inclusion of incoming faxes and voice-mail in one's email inbox, all the way to dictating a message into a cell phone and the intelligent delivery of that message to the intended recipient in a variety of possible formats like text email, fax, or voice recording.

Unified messaging provides a number of benefits for users to manage their businesses with accessible, interfaced electronic communication systems, such as e-mail, voice, messenger services;

- **A single inbox.** Unified messaging can deliver all types of messaging and communication to a single inbox. The single inbox is easier for administrators to maintain, and provides flexibility for users to manage and interact with all of their communications.
- **Efficient communication.** Users can communicate more efficiently by having access to all communications at one time and being free to share, forward, or manage them in the way that's most convenient or effective for the given communication.
- **Cost savings.** Merging streamlines the communications administration and consolidates the infrastructure onto fewer physical servers, saving money for the enterprise.
- **Access from anywhere.** Unified messaging provides alternative methods of accessing communications. By merging e-mail, voice, and other communications, users can get voice messages in e-mail, have e-mail dictated over the phone, or access communications via the Web.

Fundamentals of e-commerce:

Traditional commerce:

- Mass-marketing and sales force-driven
- Difficulty to search for the best price and quality
- Information asymmetry – any disparity in relevant market information among parties in transaction

E-commerce:

In its broadest definition, e-commerce is digitally enabled commercial transactions between and among organizations and individuals, where digitally enabled means, for the most part, transactions that occur over the Internet and World Wide Web.

Commercial transactions involve the exchange of value (e.g. money) across organizational or individual boundaries in return for products and services. E-Commerce is a modern business methodology that addresses the needs of organizations, merchants, and consumers to cut costs while improving the quality of goods and service and increasing the speed of service delivery.

Ecommerce can be defined from different dimensions as;

- From a **communications perspective**, EC is the delivery of information, product/services, or payments over telephone lines, computer networks, or any other electronic means.

- From a **business process perspective**, EC is the application of technology toward the automation of business transactions and work flow.

- From a **service perspective**, EC is a tool that addresses the desire of firms, consumers, and management to cut service costs while improving the quality of goods and increasing the speed of service delivery.

- From an **online perspective**, EC provides the capability of buying and selling products and information on the Internet and other online services.

- From a **collaboration perspective**, EC is the facilitator for inter- and intra-organizational collaboration.

- From a **community perspective**, EC provides place for community members, to learn, transact and collaborate.

E-commerce Vs. E-business

E-business is the use of Internet and digital technology to execute all the business processes in the enterprise. E-business includes e-commerce as well as processes for the internal management of the firm and for coordination with suppliers and other business partners. E-business includes digital enabling of transactions and processes within a firm, involving information systems under the control of the firm.

For example, a company’s online inventory control mechanisms are a component of e-business and online selling of company product is e-commerce.

Types of ecommerce:

Classified by the nature of market relationship – who is selling to whom

1. **Business-to-Consumer (B2C):**

- Businesses sell products or services to individual customers
- Example:

  Walmart.com sells merchandise to consumers through its Web site

2. **Business-to-Business (B2B)**

- Businesses sell products or services to other businesses
- Types include inter-business exchanges, e-distributors, B2B service providers, matchmakers and infomediaries.
- Examples:

  Grainger.com sells industrial supplies to large and small businesses through its Web site,
  Intel sells products to other business rather than customers.

3. **Consumer-to-Consumer (C2C)**

- Participants in an online marketplace can buy and sell goods with each other
- Example:

  Consumers and businesses trade with each other on eBay.com

Classified by technology used;

1. **Peer-to-Peer (P2P)**

- Use of peer-to-peer technology, which enables Internet users to share files and computer resources directly without having to go through a central Web server, in e-commerce
- Examples: BitTorrent and eDonkey.

2. **Mobile commerce (M-commerce)**

- Use of digital wireless devices to enable transactions on the Web

Unique features of ecommerce

The features the set e-commerce Technology apart from others used in traditional commerce are:

1. **Ubiquity** – internet/web technology is available everywhere: at work, home and elsewhere via mobile devices.
   - ✓ Marketplace extended beyond traditional boundaries

- ✓ "Marketspace" is created, available 24/7/365
- ✓ Customer convenience increased, costs reduced.
- ✓ Ubiquity reduces *transaction cost* – the cost of participating in a market

2. **Global Reach** – the technology reaches across national boundaries, around the learth.
   - ✓ The potential market size is roughly equal to the size of the world's online population
   - ✓ The total number of users or customers an e-commerce business can obtain is a measure of its reach

3. **Universal standards** – there is one set of technology standards, namely internet standards that is shared by all nations around the world.
   - ✓ Promotes technology adoption
   - ✓ Reduces costs of adoption
   - ✓ Greatly lower *market entry cost* for merchants
   - ✓ Reduce *search cost* for consumers

4. **Richness** – Refers to the complexity and content of a message
   - ✓ Video, audio, and text messages are integrated into a single marketing message
   - ✓ The Internet has the potential for offering considerably more information richness than traditional media like printing press, radio, and television because it is interactive and can adjust the message to individual users

5. **Information Density -** Internet and Web vastly increase the total amount and quality of information available to all market participants
   - ✓ Information processing, storage and communication costs drop dramatically.
   - ✓ Accuracy and timeliness improve greatly.
   - ✓ Information becomes plentiful, cheap and accurate.

6. **Interactivity** – the technology allows active user involvement.
   - ✓ Enable two-way communication between merchant and consumer
   - ✓ Traditional televisions cannot ask viewers any questions or enter into conversations, and it cannot request that customer information be entered into a form
   - ✓ Interactivity allows an online merchant to engage a consumer in ways similar to a face-to-face experience on a global scale where consumers engage in dynamic dialog

7. **Social Technology** – the technology allows the persons to create communities of their own interest.
   - ✓ The Internet and e-commerce technologies have evolved to be much more social by allowing users to create and share content in the form of text, videos, music, or photos with a worldwide community.

- ✓ Using these forms of communication, users are able to create new social networks and strengthen existing ones

8. **Personalization/Customization** – the technology reaches allows personalized messages to be delivered to individuals as well as groups.
   - ✓ E-commerce technologies permit personalization by targeting of marketing message to specific individuals by adjusting the message to a person's name, interests, and past purchases
   - ✓ The technology also permits by changing the delivered product or service based on user's preferences or prior behavior
   - ✓ Potential customer reach extended.

Ecommerce Challenges:

Although using e-commerce offers organization a wealth of new opportunities and ways of doing business, it also presents managers with a number of serious challenges as given below

- **Unproven Business Models** - Many Internet business models are new and largely unproven to prove enduring sources of profit

- **Business Process Change Requirements** - Web-enabled business processes for e-commerce and e-business requires far-reaching organizational change

- **Channel Conflicts** - Using the Web for online sales and marketing may create channel conflicts with the firm's traditional channels

- **Legal Issues** - laws governing electronic commerce are still being written

- **Trust, Security, and Privacy** - Electronic commerce does not provide trust among buyers, sellers, and other partners involved in online transactions

Facebook: The New Face of E-Commerce ☺

- Do you use Facebook, and if so, how often? What has the experience been like?

- Have you purchased anything based on an advertisement on Facebook or by using a link provided by a friend? Or have you used Facebook Marketplace Feauture?

- Are you concerned about the privacy of the information you have posted on Facebook?

Ecommerce trends 2010- 2012

- Social networking continues to grow

- Social e-commerce platform emerges

- Mobile computing begins to rival PC
- Explosive growth in online video viewing
- Group Purchasing
- Continued privacy and security concerns

Concept of Grid and Cloud Computing:

Grid Computing:

grid computing is a computer network in which each computer's resources are shared with every other computer in the system. Processing power, memory and data storage are all community resources that authorized users can tap into and leverage for specific tasks. A grid computing system can be as simple as a collection of similar computers running on the same operating system or as complex as inter-networked systems comprised of every computer platform you can think of.

Grid computing is the federation of computer resources from multiple administrative domains to reach a common goal. The **grid** can be thought of as a distributed system with non-interactive workloads that involve a large number of files. What distinguishes grid computing from conventional high performance computing systems such as cluster computing is that grids tend to be more loosely coupled, heterogeneous, and geographically dispersed . Although a single grid can be dedicated to a particular application, commonly a grid is used for a variety of purposes. Grids are often constructed with general-purpose grid middleware software libraries.

Grid size varies a considerable amount. Grids are a form of distributed computing whereby a **"super virtual computer"** is composed of many networked loosely coupledcomputers acting together to perform large tasks. For certain applications, "distributed" or "grid" computing, can be seen as a special type of parallel computing that relies on complete computers (with onboard CPUs, storage, power supplies, network interfaces, etc.) connected to a network (private, public or the Internet) by a conventional network interface, such as Ethernet. This is in contrast to the traditional notion of a supercomputer, which has many processors connected by a local high-speed computer bus.

Grid computing combines computers from multiple administrative domains to reach **a common goal**, **to solve a single task**, and may then disappear just as quickly.

One of the main strategies of grid computing is to use middleware to divide and apportion pieces of a program among several computers, sometimes up to many thousands. Grid computing involves computation in a distributed fashion, which may also involve the aggregation of large-scale cluster computing-based systems.

The size of a grid may vary from small—confined to a network of computer workstations within a corporation, for example—to large, public collaborations across many companies and networks. "The notion of a confined grid may also be known as an intra-nodes cooperation whilst the notion of a larger, wider grid may thus refer to an inter-nodes cooperation".

Cloud Computing:

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation.

There are many types of public cloud computing: Infrastructure as a service (IaaS), Platform as a service (PaaS), Software as a service (SaaS), Storage as a service (STaaS), Security as a service (SECaaS), Data as a service (DaaS), Test environment as a service (TEaaS), Desktop as a service (DaaS), API as a service (APIaaS).

Characteristics:

Cloud computing exhibits the following key characteristics:

- **Agility** improves with users' ability to re-provision technological infrastructure resources.
- **Application programming interface** (API) accessibility to software that enables machines to interact with cloud software in the same way the user interface facilitates interaction between humans and computers.
- **Cost** is claimed to be reduced and in a public cloud delivery model capital expenditure is converted to operational expenditure.[26] This is purported to lower barriers to entry, as infrastructure is typically provided by a third-party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained with usage-based options and fewer IT skills are required for implementation (in-house)
- **Device and location independence**[29] enable users to access systems using a web browser regardless of their location or what device they are using (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect from anywhere.
- **Virtualization** technology allows servers and storage devices to be shared and utilization be increased. Applications can be easily migrated from one physical server to another.

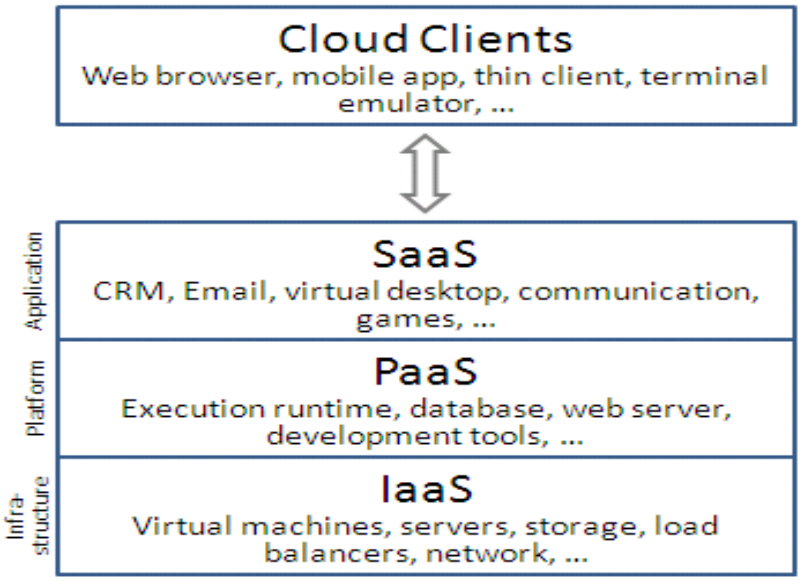
- **Multitenancy** enables sharing of resources and costs across a large pool of users thus allowing for:
    - **Centralization** of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
    - **Peak-load capacity** increases (users need not engineer for highest possible load-levels)
    - **Utilisation and efficiency**
- **Reliability** is improved if multiple redundant sites are used, which makes well-designed cloud computing suitable for business continuity and disaster recovery.
- **Scalability and elasticity** via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis near real-time, without users having to engineer for peak loads.
- **Performance** is monitored, and consistent and loosely coupled architectures are constructed using web services as the system interface.
- **Security** could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels.
- **Maintenance** of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.

Cloud Clients:

Users access cloud computing using networked client devices, such as desktop computers, laptops, tablets and smartphones. Some of these devices - *cloud clients* - rely on cloud computing for all or a majority of their applications so as to be essentially useless without it. Examples are thin clients and the browser-based Chromebook. Many cloud applications do not require specific software on the client and instead use a web browser to interact with the cloud application. With Ajax and HTML5 these Web user interfaces can achieve a similar or even better look and feel as native applications. Some cloud applications, however, support specific client software dedicated to these applications (e.g., virtual desktop clients and most email clients). Some legacy applications (line of business applications that until now have been prevalent in thin client Windows computing) are delivered via a screen-sharing technology.

Cloud Service Models:

Cloud computing providers offer their services according to three fundamental models: Infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) where IaaS is the most basic and each higher model abstracts from the details of the lower models.



Iaas:

In this most basic cloud service model, cloud providers offer computers, as physical or more often as virtual machines, and other resources. The virtual machines are run as guests by a hypervisor, such as Xen or KVM. Management of pools of hypervisors by the cloud operational support system leads to the ability to scale to support a large number of virtual machines. Other resources in IaaS clouds include images in a virtual machine image library, raw (block) and file-based storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. IaaS cloud providers supply these resources on demand from their large pools installed in data centers. For wide areaconnectivity, the Internet can be used or—in carrier clouds -- dedicated virtual private networks can be configured.

To deploy their applications, cloud users then install operating system images on the machines as well as their application software. In this model, it is the cloud user who is responsible for patching and maintaining the operating systems and application software. Cloud providers typically bill IaaS services on a utility computing basis, that is, cost will reflect the amount of resources allocated and consumed.

IaaS refers not to a machine that does all the work, but simply to a facility given to businesses that offers users the leverage of extra storage space in servers and data centers.

Examples of IaaS include: Amazon CloudFormation (and underlying services such as Amazon EC2), Rackspace Cloud, Terremark and Google Compute Engine.

Paas:

In the PaaS model, cloud providers deliver a computing platform typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware

and software layers. With some PaaS offers, the underlying computer and storage resources scale automatically to match application demand such that cloud user does not have to allocate resources manually.

Examples of PaaS include: Amazon Elastic Beanstalk, Heroku, EngineYard, Mendix, Google App Engine, Microsoft Azure and OrangeScape.

Saas:

In this model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. The cloud users do not manage the cloud infrastructure and platform on which the application is running. This eliminates the need to install and run the application on the cloud user's own computers simplifying maintenance and support. What makes a cloud application different from other applications is its elasticity. This can be achieved by cloning tasks onto multiple virtual machines at run-time to meet the changing work demand. Load balancers distribute the work over the set of virtual machines. This process is inconspicuous to the cloud user who sees only a single access point. To accommodate a large number of cloud users, cloud applications can be multitenant, that is, any machine serves more than one cloud user organization. It is common to refer to special types of cloud based application software with a similar naming convention: desktop as a service, business process as a service, test environment as a service, communication as a service.

The pricing model for SaaS applications is typically a monthly or yearly flat fee per user, so price is scalable and adjustable if users are added or removed at any point.

Examples of SaaS include: Google Apps, innkeypos, Quickbooks Online, Limelight Video Platform, Salesforce.com and Microsoft Office 365.