5.1. Designing of Internet System Network Architecture
5.2. Choice of platforms
5.3. Server Concepts: WEB, Proxy, RADIUS, MAIL
5.4. Cookies
5.5. Load Balancing: Proxy Arrays
5.6. Server Setup and Configuration Guidelines
5.7. Security and System Administration Issues, Firewalls and Content Filtering

In the late 1960s, the US Department of Defense decides to make a large network from a multitude of small networks, all different, which begin to abound everywhere in North America. We had to find a way to these networks coexist and give them an outdoor visibility, the same for all users. Hence the name of InterNetwork (interline), abbreviated as Internet, data this network of networks.

The **Internet architecture** is based on a simple idea: ask all networks want to be part of carrying a single packet type, a specific format the IP protocol. In addition, this IP packet must carry an address defined with sufficient generality in order to identify each computer and terminals scattered throughout the world.

The user who wishes to make on this internetwork must store its data in IP packets that are delivered to the first network to cross. This first network encapsulates the IP packet in its own packet structure, the package A, which circulates in this form until an exit door, where it is decapsulated so as to retrieve the IP packet. The IP address is examined to locate, thanks to a routing algorithm, the next network to cross, and so on until arriving at the destination terminal.

Layer 1 - Devices and Their Functions

Defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between end systems. Some common examples are Ethernet segments and serial links like **Frame Relay** and **T1**.

Repeaters that provide signal amplification are also considered Layer 1 devices. Fiber Cabel, CAT6 cable

The **physical interface on the NIC** can also be considered part of Layer 1.

Layer 2 Devices and Their Functions

Defines how data is formatted for transmission and how access to the physical media is controlled. These devices also provide an interface between the Layer 2 device and the physical media. Some common examples are a **NIC** installed in a **host, bridge, or switch**.

Layer 3 Devices and Their Functions

Provides connectivity and path selection between two host systems that might be located on geographically separated networks. In the case of a host, this is the path between the data link layer and the upper layers of the NOS. In the case of a **router**, it is the actual path across the network.

Approach to Network Design

- necessity to account for all seven layers of the OSI model when creating a design for a network
- As well as accounting for that all important eighth layer, in other words the political factors that always have an effect on any technical decision
- Network design must be a complete process that matches business needs to the available technology to deliver a system that will maximize the organization

Network Design Steps

- Identifying Customer Needs/Goals
  - Analyzing Business Goals, Constraints and Technical Goals, Tradeoffs
  - Characterizing the Existing Network and Network Traffic
- Logical Network Design
  - Designing a Network Topology and Models for Addressing, Naming
  - Selecting Switching and Routing Protocols
  - Developing Network Security Strategies and Network Management Strategies
- Physical Network Design
  - Selecting Technologies and Devices for Campus Networks or Enterprise Networks
- Testing Optimizing Documenting
  - Testing the Network Design
  - Optimizing the Network Design
  - Documenting the Network Design

5.1. Designing of Internet System Network Architecture

Design considerations

- Budget

- Nature of applications
- Availability of expertise
- Fault tolerance in terms of applications, system and network access
- Ease of configuration
- Management

Small sized Network[SSN] (<80 users)

- Low budget for IT expense
- Little expertise in various technologies
- Mostly off the shelf applications
  - Low bandwidth consumption
- Mostly basic requirements, such as email, word processing, printing and file sharing
- One or two administrators
  - Responsible for every aspects of network (generalist)
  - Server management, backup tasks, connecting new devices, installation of workstations and troubleshooting PC problems

Requirements for SSN

- Low cost equipment
- Shared bandwidth for most users, switched for a selective few
- A central switch acting as a backbone
- Flat network design
- Little fault tolerance
- Minimal management required
- High growth provisioning of 20-50%

A sample firm

- Connect 50 users to a network
- Connect 10 printers to the network
- Connect the company's database and internal e-mail services to the network, hosted in a server
- Users require connectivity to the internet
- Several system require access to external email, the Web and FTP connectivity
- A future web site may be implemented

Connectivity design

- The aim is to have a design that is both cost effective and provisioned for future expansion
- There is a server room with all the connecting devices and servers
- The printers are fitted with built in Ethernet ports distributed in the building
- There are two groups of users, power users group and non power users
- Power group need to print a lot of documentation, take large documents from server or save presentation files into the server
- Non power users do more manual tasks such as answering phone calls
- They use the network mainly for reading emails and do some simple word processing
- They use low-end PCs

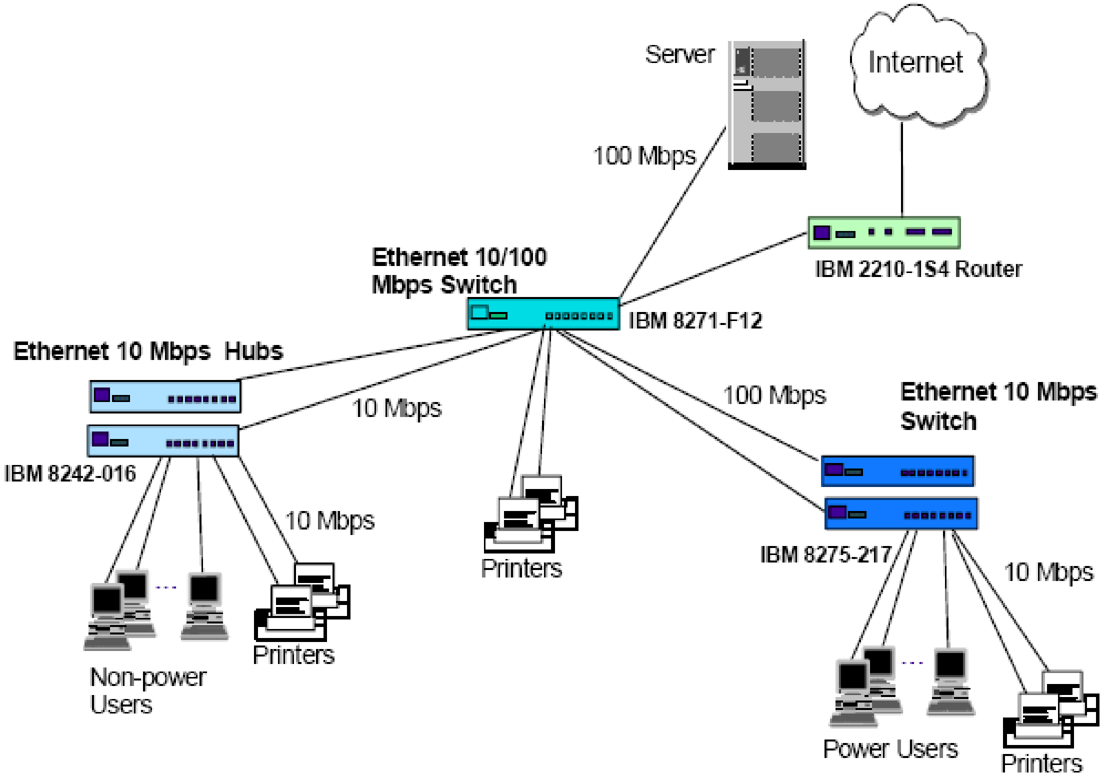


Fig. Physical Design

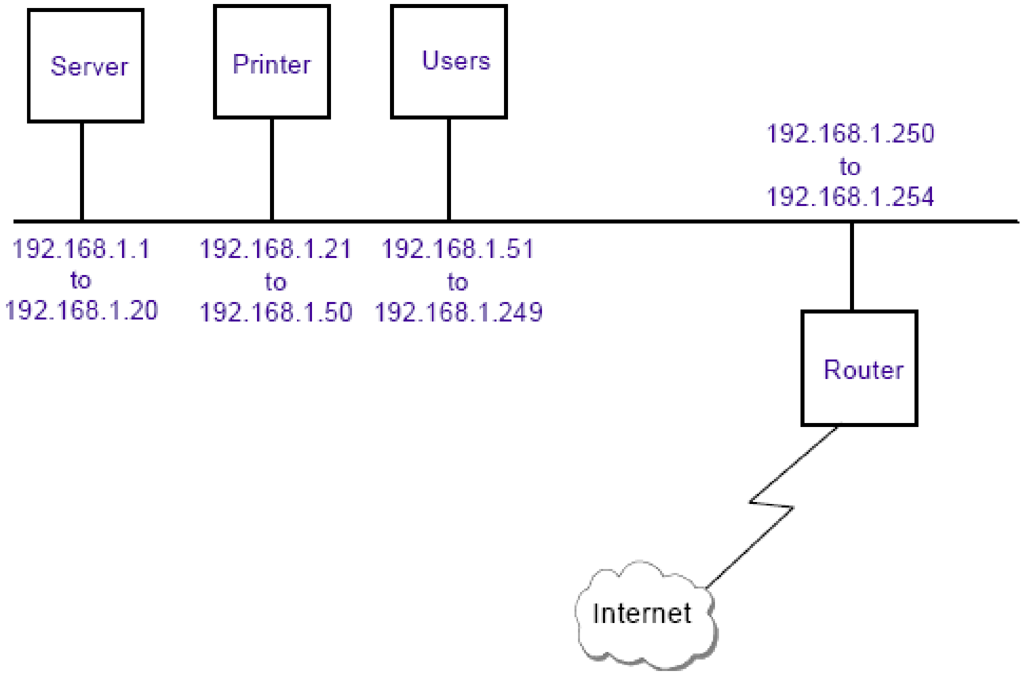


Fig. Logical Design

Addressing and Naming

- For this size of network a Class C address should be used. (::/64 IPv6 => sufficient addresses available)
- A private Class C address is used: 192.168.1.0 to 192.168.1.255 (FC00::/7 for V6 security perspectives)
- Dynamic or Static IP assignment?
  - It might be hard to maintain a DHCP server
  - Therefore for small sized network we may decide to use static IPs.
  - For large, IPv6 address auto-confuiguration
- How about a DNS server?
  - Again, setting and maintaining a DNS for this size of network may not be beneficial
  - Servers on ISP level if needed.

Connecting the network to the Internet

- In the design we used private IP addresses:
  - Computers can’t use Internet directly, there is a need for NAT functionality (require global Ipv6 not NAT)
  - There exists the advantage of security of network
- It is decided to use a router with built-in NAT functionality for Ipv4
- It is not cost effective to host email and Web service inside the organization however based on the size it may setup
- Therefore, such servers are outsourced to ISPs

Medium sized Network (<500 users)

- Fixed annual budget for IT expenditure
- MIS department taking care of the information system
- Develop own in-house applications
- Availability of one or a few dedicated network engineers
- Invest in server/host fault tolerance features
- May provide dial-in service to mobile workers

A sample firm

- Connecting 300 users to a network
- The company has a AS/400 host and 8 file servers
- There are 6 departments in the company, each with its own applications:
  - Marketing – mainly email with external customers, calendaring, word processing, presentation applications
  - Customer support – mainly handling customer queries, accessing the host for in-house developed applications
  - MIS – development of applications on AS/400
  - Human Resources – Mainly word processing
  - Engineering – make use of CAD/CAM workstations

Connectivity design

- Power users, such as the Engineering department, will have 100 Mbps switched connections to the desktop
- Because Marketing users deal with graphics presentation, they will be connected to the 10 Mbps switch in a ratio of 16 users to a switch.
- Since Customer Support and Human Resources users require fewer computing resources, they are connected to the 10 Mbps switch in a ratio of 24 to a switch.
- Except for the server in the Engineering department, all the servers are connected to the backbone switch at 100 Mbps. The engineering server is connected to the switch in the Engineering department at 100 Mbps.

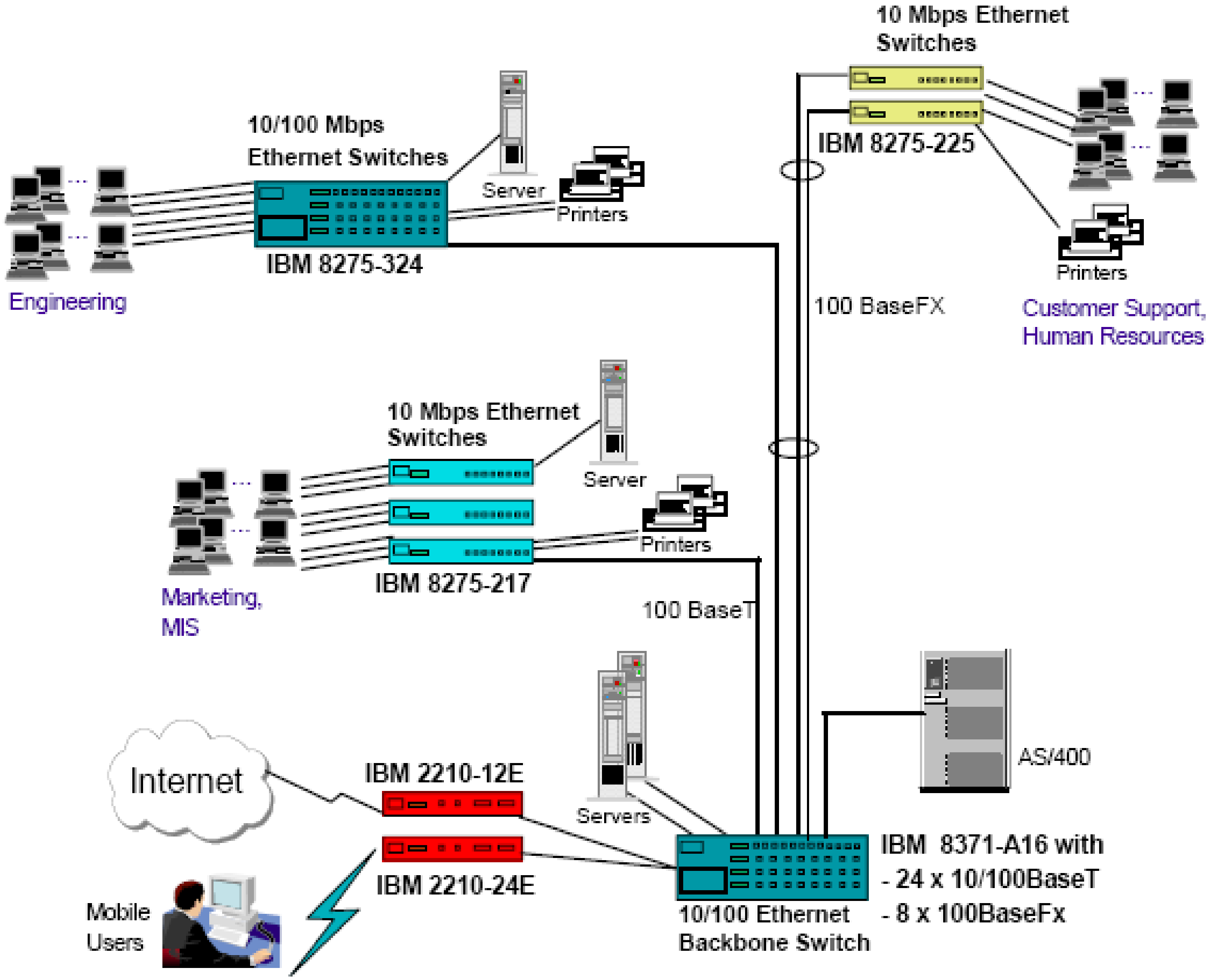


Fig. Physical Design

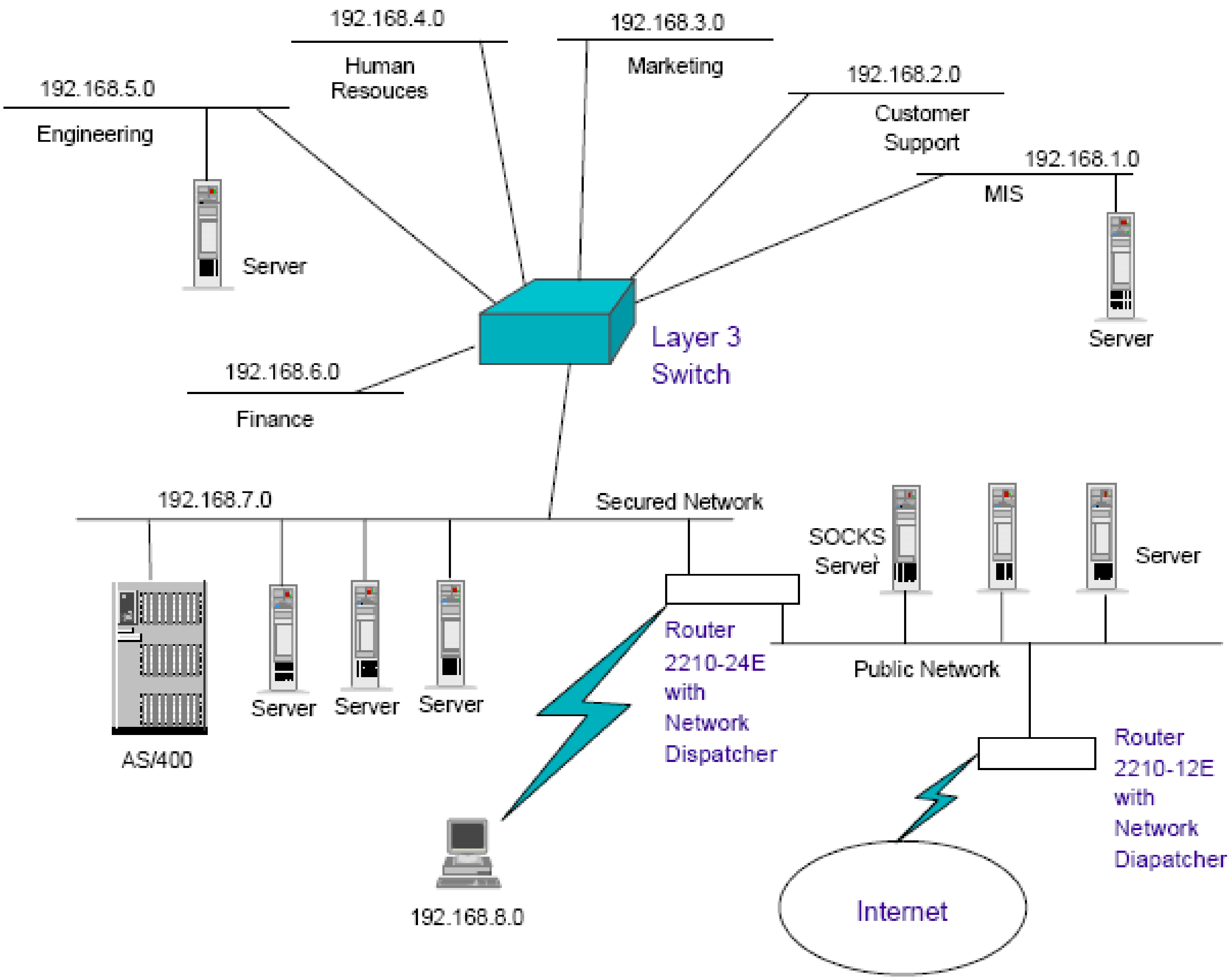


Fig. Logical Network Design

Remote access

- Provide dial-in users/ADSL Users service
- Provide concurrent dial-in connections
- A dial-back service will be implemented. That is, a remote user initiates a call to the router and triggers the router to dial back to the user.
- Remote users have to authenticate themselves through a login ID and a password.

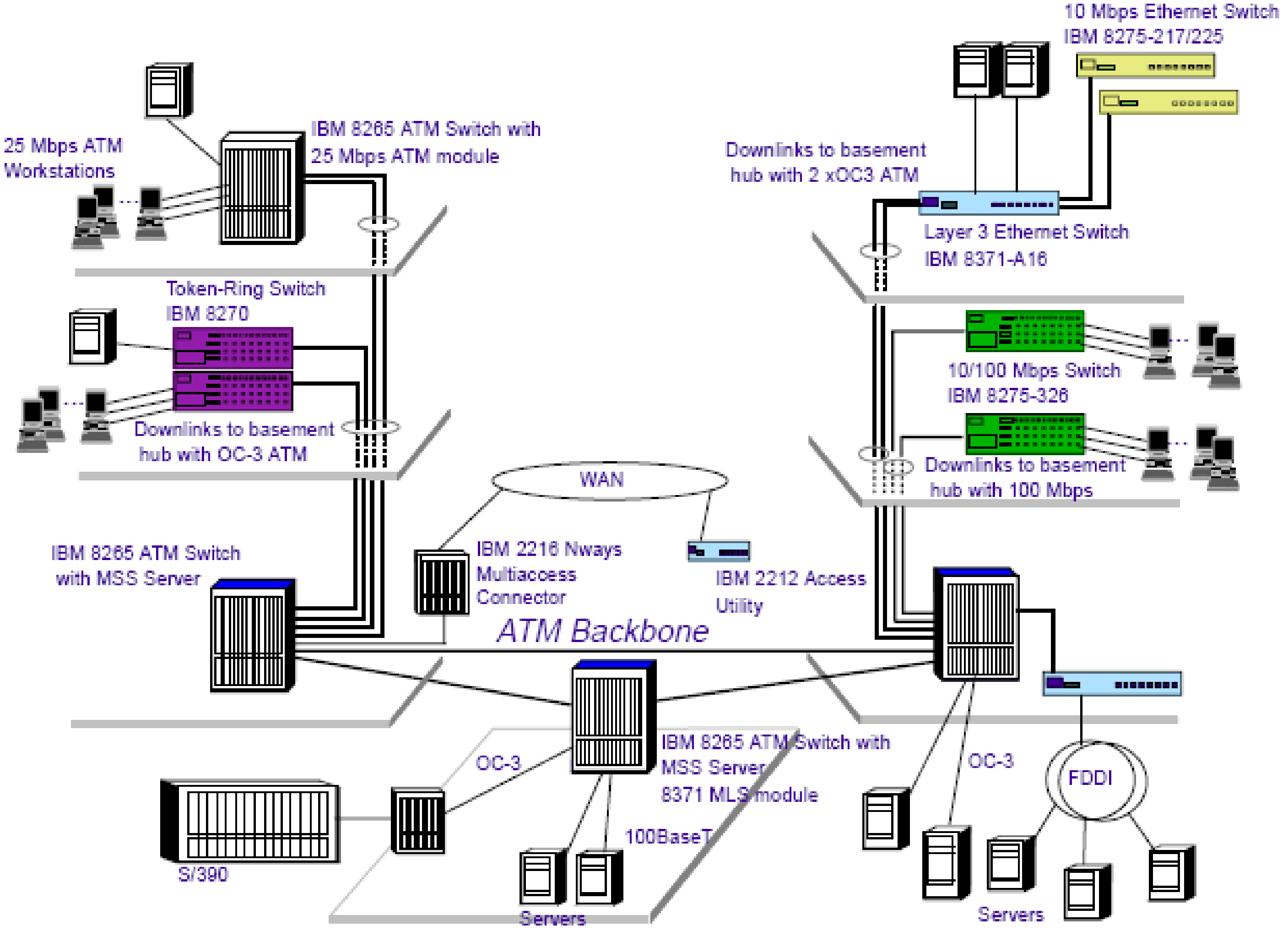
Addressing and Naming

- There is a requirement for set of public addresses to be obtained from the organization's ISP. These would be for the organizational firewall, the services server hosting FTP, HTTP and e-mail services, the primary DNS server.
- All these servers should have their IP addresses assigned statically.
- Organizational domain name must be registered
- To reduce WAN traffic, the primary DNS server may be placed on the ISP site.

Large size network (>500 users)

- Internetwork of networks, with a mix of technologies such as Ethernet, token-ring, FDDI and ATM.
- Involves multiprotocol such as TCP/IP, IPX, SNA or NetBIOS.
- Fault tolerance features for mission-critical applications, such as hardware redundancies, network path redundancies and extensive investment on backup services.
- Fairly large MIS department to take care of the information system

- In-house application development teams that constantly look at the deployment of new Internet technologies such as Java and multimedia applications.
- Availability of experts in areas such as system management, network infrastructure and management.
- Substantial amount of company's annual budget is spent on IT investment.
- All Necessary Servers at the Department Network Control Room
- Network Segmentation based on the types of users and security. VLAN
- Intra-site VPN service.
- Appropriate Security System: DMZ to avoid external attack.
- Backup and mirror server management.
- Workstation Control: appropriate antivirus.
- Prioritize traffic management system.
- In-house Training and Education



***Principles for Designing Network Architecture/ Factors for well design networks**

- ***Simplicity***: In terms of network architecture *the most critical aspect is that of simplicity of structure*. Simplicity is a key principle in so far as it effectively imposes the *minimum of constraints,* and allows each client of the service to readily interface their infrastructure and service environment into a *national environment*, and *allows the national network the capability to adopt* to change technologies and changing service requirements that may be imposed by the client base in the future.
- ***Functional Capability/Suitability***: the architecture should meet the *basic client service objectives without imposing additional qualifications or constraints.*
- ***Affordability***: Affordability is perhaps implied within any such architecture, but it is explicitly stated here simply to note that any network architecture which is not affordable *within available resources* will never be implemented.
- ***Implementable today*** : *Technical feasibility* is also a principle which is effectively implied within any architecture, but again it perhaps worth explicitly noting within the set of architectural principles that if a network architecture relies on *technologies which cannot be purchased and deployed today,* then the architecture cannot be used as the basis of subsequent implementation engineering, and accordingly such an architecture specification is functionally irrelevant for any other purpose than a vision statement of potential future service objectives.

- ***Designed to meet actual end client requirements*** : Networks are service structures, and the *architecture of a network should accordingly be designed to meet actual end client needs, rather than impose additional constraints and conditions on the client base*. This implies that a network should provide service to the end user application services and protocols which are being deployed by the user base, rather than implement a service environment which forces clients to deploy new services and protocols.

- ***Uses (and develops) local expertise*** : Critically within the area of public national network infrastructure provision, it is also highly desirable that any such program uses, and fosters the *further development of national expertise and skills within the adopted service technology domain*. A "black box" approach to this area results in a service operation which has significant negative impact on issues of quality, integrity and future viability of the service.

- ***Where feasible uses locally available components*** : To assist in this development of *local capability and expertise the network* should be able to use locally available components and services wherever feasible.

-***Connectivity and Security :*** Network connectivity today means *more than Ethernet cables and wireless access points*. People today are more connected while mobile than ever before and many of them want access to company email and data while they are out of the office. Balancing those needs while maintaining security is a challenge that needs to be addressed in the design phase of any network. This includes where data is stored, either in-house or offsite with *cloud-based solutions,* what types of information should be accessible, who should be able to access it, and which types of devices should be included. *Firewalls and access servers need to be secure* without slowing down operations.

-***Redundancy :*** Redundancy means *having backup devices in place for any mission-critical components in the network*. Even small organizations should consider using two servers. Two identical servers, for example, can be configured with fail-safes so that one will take over if the other fails or requires maintenance. A good rule of thumb is to have redundant components and services in place for any part of a network that cannot be down for more than an hour.

-***Standardization :*** Standardization of the *hardware and software used in a network is important for ensuring the network runs smoothly*. It also *reduces costs associated with maintenance*, updates and repairs. Conducting a full audit of the current computer systems, software and peripherals will help to determine which should be standardized.

-***Disaster Recovery :*** A detailed *disaster recovery plan should be a part of any network design.* This includes, but is not limited to, provisions for back-up power and what procedures should be followed if the network or server crashes. It should also include when data is backed up, how it is backed up and where copies of the data are stored. In most cases, *important data should be backed up daily*. Many organizations do a full weekly backup, with daily incremental backups that copy any files that have been modified since the last weekly backup. Backup files should be stored in a secure location off-site in the event of a building disaster, such as a fire.

Unmanaged switches are *basic plug-and-play switches with no remote configuration, management, or monitoring options, although many can be locally monitored and configured via LED indicators and DIP switches*. These inexpensive switches are typically used in small networks or to add temporary workgroups to larger networks.

Managed switches *support Simple Network Management Protocol (SNMP) via embedded agents and have a command line interface (CLI) that can be accessed via serial console, Telnet, and Secure Shell.* These switches can often be configured and managed as groups. More recent managed switches may also support a Web interface for management through a Web browser.

Most managed switches offer you features like:

- *View the bridging table to see which MAC addresses are associated with a given port*
- *View error statistics for each port*
- *View packet transmit / receive statistics for each port*
- *Set duplex / speed negotiation (or lack thereof) on a per-port basis*
- *View power-over-Ethernet status and current draw for each port (if applicable)*

Hierarchical Network Design

In networking, a hierarchical design is used to group devices into multiple networks. The networks are organized in a layered approach.

The hierarchical design model has three basic layers:

■ ***Core layer***: Connects distribution layer devices. The core layer design *enables the efficient, high-speed transfer of data between one section of the network and another.* The core layer of the hierarchical design is the high-speed backbone of the internetwork.

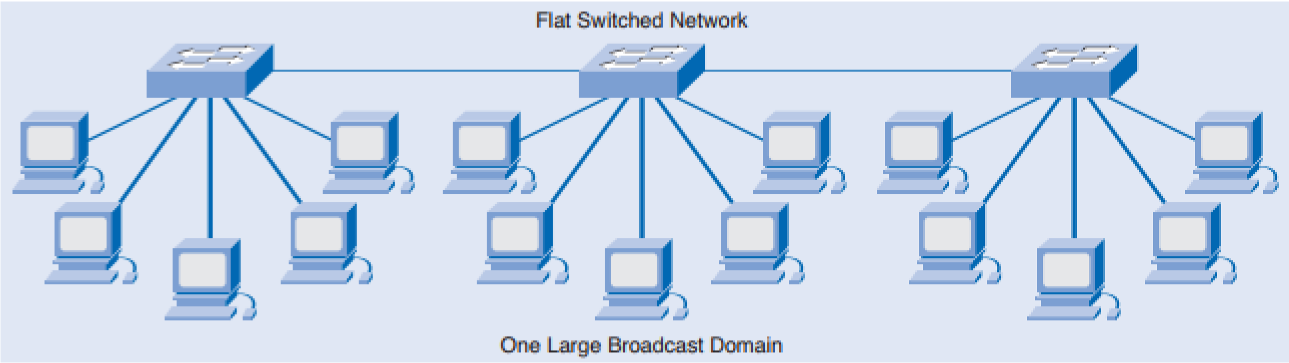
The core layer is *critical for interconnectivity between distribution layer devices*, so it is important for the core to be highly available and redundant. The core area can also connect to Internet resources. The core aggregates the traffic from all the distribution layer devices, so it must be capable of forwarding large amounts of data quickly. The primary design goals at the core layer are as follows:

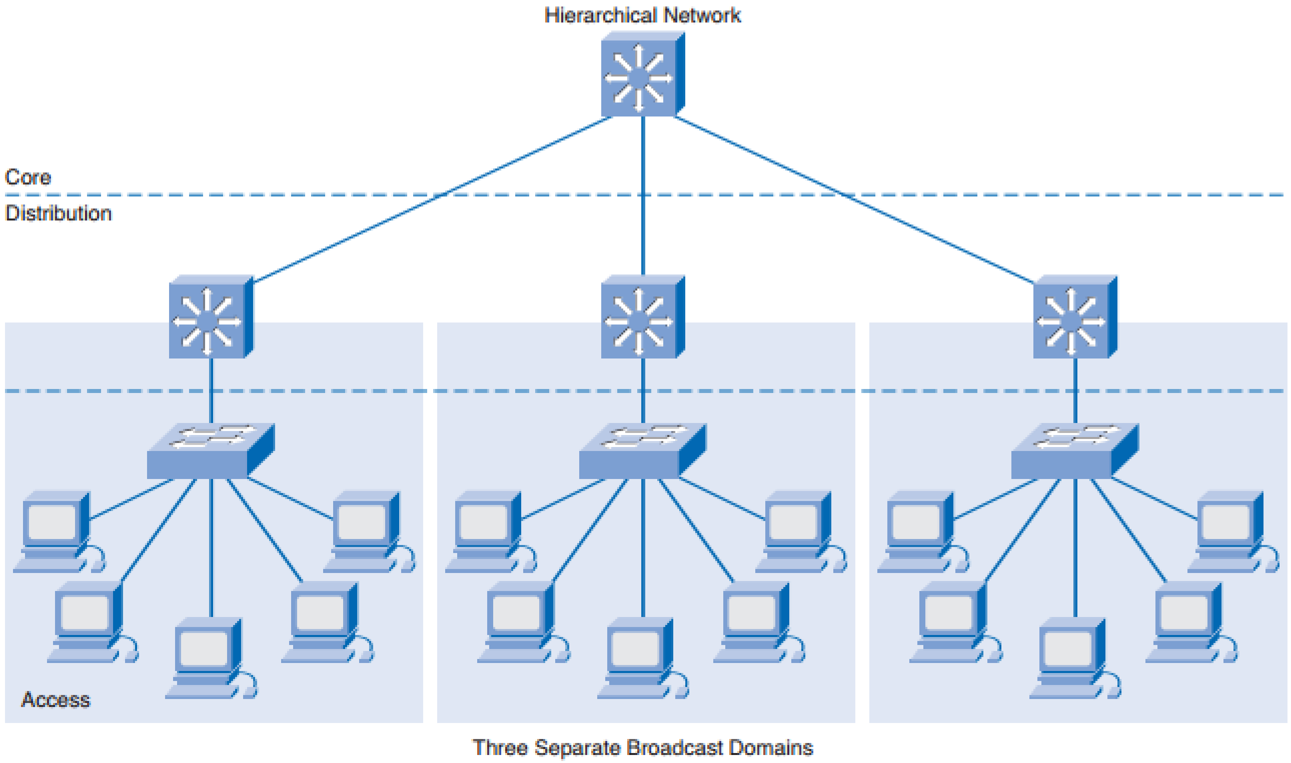
- *Provide 100% uptime.*
- *Maximize throughput.*
- *Facilitate network growth.*

Technologies used at the core layer include the following:

- *Routers or multilayer switches that combine routing and switching in the same device*
- *Redundancy and load balancing*
- *High-speed and aggregate links*

- *Routing protocols that scale well and converge quickly, such as Enhanced Interior Gateway Routing Protocol (EIGRP) and Open Shortest Path First (OSPF) Protocol*





■ ***Distribution layer:*** The distribution layer *aggregates the data received from the access layer switches before it is transmitted to the core layer for routing to its final destination*. The distribution layer controls the flow of network traffic using policies and defines broadcast domains by performing routing functions between virtual LANs (VLANs) defined at the access layer.

VLANs allow you to segment the traffic on a switch into separate subnetworks. For example, in a university you might separate traffic according to faculty, students, and guests.

Distribution layer switches are typically high-performance devices that *have high availability and redundancy to ensure reliability*.

■ ***Access layer:*** The access layer *interfaces with end devices, such as PCs, printers, and IP phones, to provide access to the rest of the network*. The access layer can include routers, switches, bridges, hubs, and wireless access points. The main purpose of the access layer is to provide a means of connecting devices to the network and controlling which devices are allowed to communicate on the network.

Benefits of a Hierarchical Network

(i)Scalability : Hierarchical networks *scale very well*. The modularity of the design allows you to *replicate design elements as the network grows.* Because each instance of the module is consistent, expansion is easy to plan and implement. *For example, if your design model consists of two distribution layer switches for every 10 access layer switches, you can continue to add access layer switches until you have 10 access layer switches cross-connected to the two distribution layer switches before you need to add additional distribution layer switches to the network topology.*

(ii)Redundancy : As a network grows, *availability becomes more important*. You can dramatically increase availability through easy redundant implementations with hierarchical networks. *Access layer switches are connected to two different distribution layer switches to ensure path redundancy. If one of the distribution layer switches fails, the access layer switch can switch to the other distribution layer switch. Additionally, distribution layer switches are connected to two or more core layer switches to ensure path availability if a core switch fails.*

(iii)Performance : Communication *performance is enhanced by avoiding the transmission of data through low performing, intermediary switches. Data is sent through aggregated switch port links from the access layer to the distribution layer at near wire speed in most cases*. The distribution layer then uses its high-performance switching capabilities to forward the traffic up to the core, where it is routed to its final destination. Because the core and distribution layers perform their operations at very high speeds, no contention for network bandwidth occurs. As a result, properly designed hierarchical networks can achieve near wire speed between all devices.

(iv)Security: Security is improved and easier to manage. *Access layer switches can be configured with various port security options that provide control over which devices are allowed to connect to the network*. You may apply access control policies that define which communication protocols are deployed on your network and where they are permitted to go. *For example, if you want to limit the use of HTTP to a specific user community connected at the access layer, you could apply a policy that blocks HTTP traffic at the distribution layer. Restricting traffic based on higher layer protocols, such as IP and HTTP, requires that your switches are able to process policies at that layer*.

(v)Manageability : Manageability is *relatively simple on a hierarchical network*. *Each layer of the hierarchical design performs specific functions that are consistent throughout that layer. Therefore, if you need to change the functionality of an access layer switch, you could repeat that change across all access layer switches in the network because they presumably perform the same functions at their layer.*

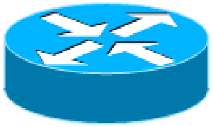
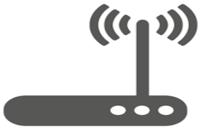
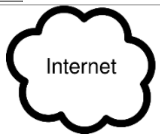
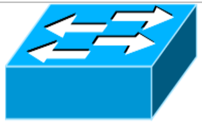
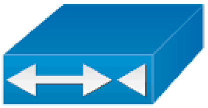
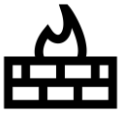
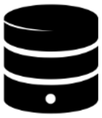
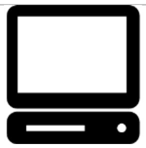
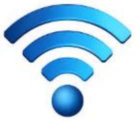
(vi)Maintainability : Because *hierarchical networks are modular in nature and scale very easily, they are easy to maintain*. With other network topology designs, maintainability becomes increasingly complicated as the network grows. Also, in some network design models, there is a finite limit to how large the network can grow before it becomes too complicated and expensive to maintain. In the hierarchical design model, switch functions are defined at each layer, making the selection of the correct switch easier. Adding switches to one layer does not necessarily mean there will not be a bottleneck or other limitation at another layer.

How to Create a Network Diagram

There are many different ways to create a network diagram. While they can be created using pen and paper or a white board, a diagramming tool designed for this purpose is a much more efficient and effective approach.

Here are some tips to consider when creating a network diagram:

- **Choose a network***: Decide which network will be illustrated*. The diagram could focus on a personal computer, or on an entire company network. Once a focus has been chosen, set limits on what outside connections will be included so that the diagram remains concise.
- **Add relevant equipment***: Begin by placing any involved computers, servers, and other components* on the page. Use visual representations and add the names of the components for clarity.
- **Add any other important components:** *Add other important components such as internet connections and firewalls.* Once again, use visual representations and add text descriptions as needed.
- **Label***: Label each of the items on the page to make it easy for anyone to understand* what they're looking at. Alternatively, number the items and attach a legend with descriptions to keep the diagram less cluttered.
- **Draw Connecting Lines:** *Use lines with directional arrows* to show how each component is related and connected to another.

 Router	 Wireless Router		 Switch	 Cable modem
 Firewall	 Server	 PC	 WiFi	

Example of BoQ

- with necessary network resources required in quantity for the complete networking

SN	Item	Description	Unit	Qty.
1	Cat6 UTP cable box	Category 6 UTP cables shall extend between the work area location and its associated telecommunications closet and consist of 4 pair, 23 AWG, UTP	Nos.	3
2	Core Switch	Switch should be mountable on 19" standard rack.	Nos.	2
3	Router	Cisco® 2900 Series Integrated Services (*to use video-conferencing and virtualization services and transport other kinds of rich media over a wide area network (WAN))* Routers	Nos.	5

5.2. Choice of platforms

(i)Software Platforms for servers:

Every website requires a reliable web server to be hosted on; therefore, it can be accessed via internet users. Nowadays in web hosting market there are different types of web servers that are available running on various platform to select.

There are at least three types of server software platforms we need to consider:

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

- And then pick a file server platform which is reliable and that is secure to protect our company's data.
- Use web server platform software which can handle the amount of traffic we will get and that has the functionality we want.

The most popular platforms and web servers are listed below:

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

(ii)Hardware Platform for servers:

Hardware requirements for servers differ which is depending on the server application. *Absolute CPU speed is not generally as critical to a server as it is to a desktop machine*. *Server's duty is to provide service to many users over a network that lead to various requirements such as fast network connections and high Input/output throughput.* Since servers are generally accessed over a network this may run in headless mode without a monitor or input device. Processes which aren't required for the server's function aren't used. *Many servers don't have a graphical user interface(GUI) because it is unnecessary and it consumes resources which could be allocated elsewhere. Likewise, audio and USB interfaces can be omitted.*

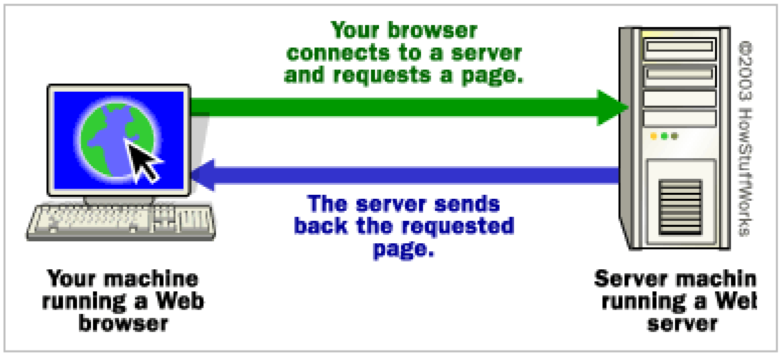
Most of the servers use memory with error detection and correction to increase reliability, redundant disks and redundant power supplies and so on. The important hardware resources to establish a successful client/server model include gateways, routers, network bridges, switches, hubs, and repeaters.

5.3 Server Concepts : WEB, Proxy, RADIUS, MAIL

*** Web Server** : *Web servers are computers that deliver (serves up) Web pages. Every Web server has an IP address and possibly a domain name. For example, if you enter the URL*http://www.hcoe.edu.np/index.html *in your browser, this sends a request to the Web server whose domain name is* www.hcoe.edu.np. *The server then fetches the page named* index.html *and sends it to your browser.*

Any computer can be turned into a Web server by installing server software and connecting the machine to the Internet. There are many Web server software applications, including public domain software and commercial packages.

When you clicked on the link for this page, or typed in its URL (**uniform resource locator**), what happened behind the scenes to bring this page onto your screen?


Your browser connects to a server and requests a page.
The server sends back the requested page.
Your machine running a Web browser
Server machin running a Wel server
©2003 HowStuffWorks

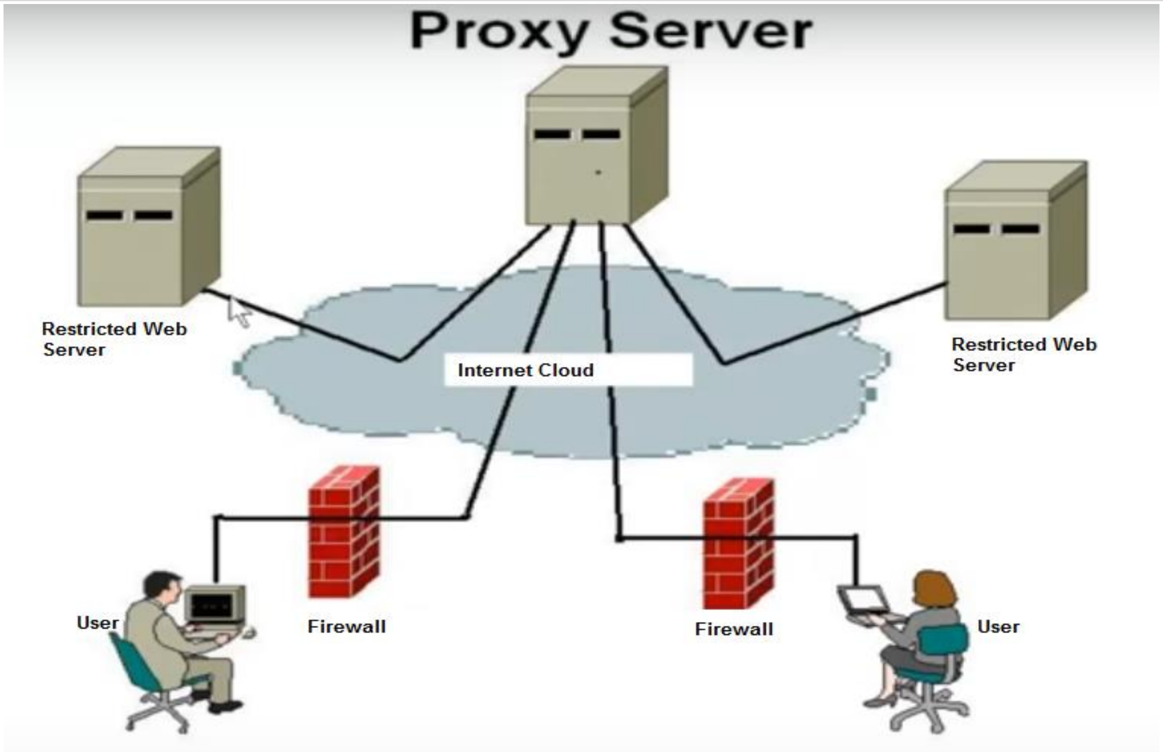
The browser broke the URL into three parts:

- *The protocol ("http")*
- *The server name ("www.hcoe.edu.np")*
- *The file name ("index.htm")*

- The browser communicated with *a name server to translate the server name "www.hcoe.edu.np" into an **IP Address***, which it uses to connect to the server machine. The browser then formed a connection to the server at that *IP address on port 80*. (We'll discuss ports later in this article.)
- Following the HTTP protocol, the browser sent *a GET request to the server, asking for the file "http://www.hcoe.edu.np/index.htm*
- The server then sent the HTML text for the Web page to the browser. (Cookies may also be sent from server to browser in the header for the page.) The browser read the HTML tags and formatted the page onto your screen.
- If you've never explored this process before, that's a lot of new vocabulary. To understand this whole process in detail, you need to learn about IP addresses, ports, protocols... The following sections will lead you through a complete explanation.

***Proxy Server (Web Caches)**

A proxy server is a dedicated computer or a software system running on a computer that acts as an intermediary between an endpoint device, such as a computer, and another server from which is used to filter or cache requests made by the client. The proxy server may exist in the same machine as a firewall server or it may be on a separate server, *which forwards requests through the firewall.* Provide a single point of access and control



A proxy server is an intermediary device between a client and a server which handles transaction between the two, without ever exposing them to each other.
A **Caching Server** is a sub-type of Proxy Servers which stores the content being fetched by it from the WAN locally to make it available to other computers without having to reach out to the WAN.
A good example of Proxy-Caching Server is Squid, which can function as a standalone proxy or a combined proxy-cache server.

Here's a simple example of how proxy servers work:

- When a proxy server receives a request for an Internet resource (such as a Web page), it looks in its local cache of previously 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.*
- *Proxy servers are used for both legal and illegal purposes.* In the "enterprise, a proxy server" is used to facilitate security, administrative control or caching services, among other purposes. In a "personal computing context, proxy servers" are used to enable user privacy and anonymous surfing. Proxy servers can also be used for the *opposite purpose:* To monitor traffic and undermine user privacy.
- To the user, the proxy server is invisible; all Internet requests and returned responses appear to be directly with the addressed Internet server. (The proxy is not actually invisible; its IP address has to be specified as a configuration option to the browser or other protocol program.)
- *Users can access web proxies online or configure web browsers to constantly use a proxy server*. Browser settings include automatically detected and manual options for HTTP, SSL, FTP, and SOCKS proxies. Proxy servers may serve many users or just one per server. These options are called shared and dedicated proxies, respectively.

Advantages or purposes of Proxy Server
***Improve Performance :** Proxy servers *can dramatically improve performance for groups of users.* This is because it saves the results of all requests for a certain amount of time.
***Filter Requests** *:* Proxy servers can also be used to *filter requests*. For example, a company might use a proxy server to prevent its employees from accessing a specific set of Web sites.
***Advanced access control** (intermediate requester in firewalled DMZ, authentication & authorization)
*** Logging and auditing**
Disadvantages – Recognizing and avoiding stale (out of date) data

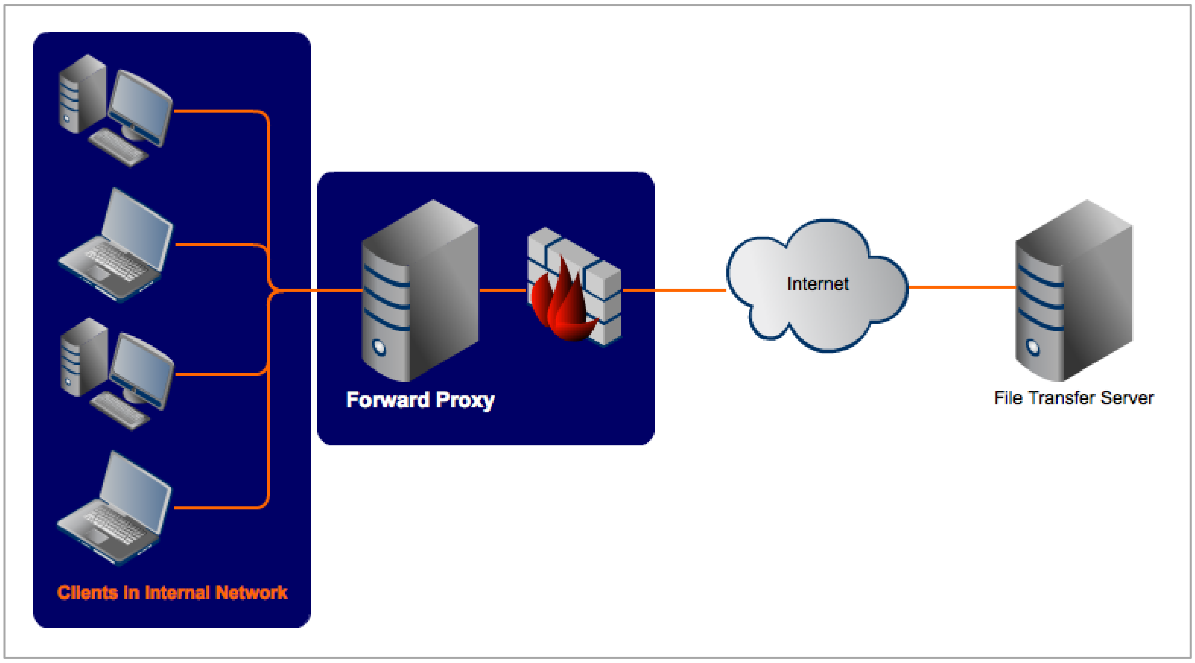
Proxy Server: Basic Operation

- Accept connection request from client – establishes new Socket client_sock
- Read HTTP request
- Parse HTTP request – reject invalid requests with appropriate response code – Request is REQUIRED to be in absoluteURI form
- Connect to (towards) requested server – establishes new socket serv_sock

- Send original HTTP request to server – or to next proxy on path to server
- Read response from Server – If time-out server connection, then issue – 504 Gateway Timeout
- Copy object in response to cache, if allowed
- Send response to client
- If **Connection: close** header received, close client connection (client_sock)

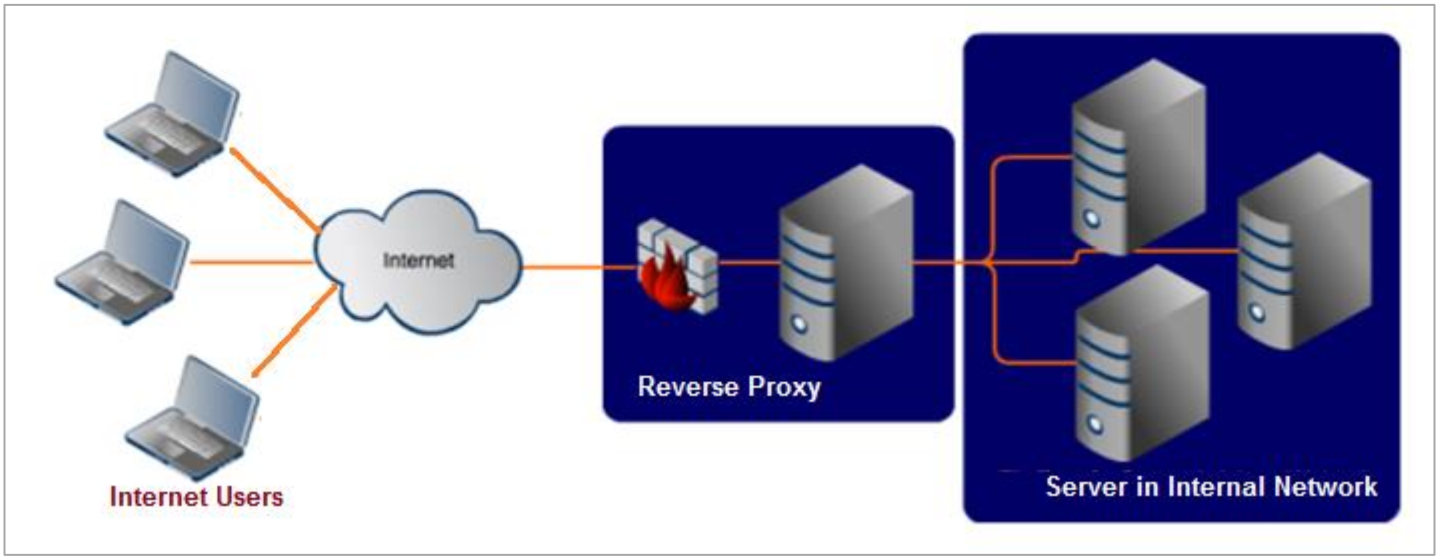
Types of Proxy Servers and their Uses:

1. Forward Proxies :A forward proxy is the same one described above where the proxy server forwards the client's request to the target server to establish a communication between the two. Here the client specifies the resources to be fetched and the target server to connect to, so that the forward proxy server acts accordingly.



2. Open Proxy : An **open proxy** is a type of *forwarding proxy* that is openly available to any Internet user. Most often, an open proxy is used by Internet users to hide their IP address so that they remain anonymous/ hides/ undefined during their web activity.

3. Reverse Proxy : A reverse proxy does the exact opposite of what a forward proxy does used for the benefit of the web server rather than its clients. . While a forward proxy proxies in behalf of clients (or requesting hosts), a reverse proxy proxies in behalf of servers. A reverse proxy accepts requests from external clients on behalf of servers stationed behind it. A forward proxy hides the identities of clients, a reverse proxy hides the identities of servers. Basically, a reverse proxy is on the web server end which will cache all the static answers from the web server and reply to the clients from its cache to reduce the load on the web server. This type of setup is also known as Web Server Acceleration.



Reverse proxies are often used to reduce load on the actual server by load balancing, to enhance security and to cache static content, so that they can be served faster to the client. Often big companies like Google which gets a large number of hits maintain a reverse proxy so as to enhance the performance of their servers. It is not a surprise that whenever you are connecting to google.com, you are only connecting to a reverse proxy that forwards your search queries to the actual servers to return the results back to you.

Reverse proxies are used:

- To disable direct access to a website as a security measure.
- To allow for load balancing between severs.
- To stream internal content to Internet users.
- To disable access to a site, for example when an ISP or government wishes to block a website.

NOTE : If you want to protect clients in your internal network, put them behind a forward proxy. On the other hand, if your intention is to protect servers, put them behind a reverse proxy.

* RADIUS (Remote Authentication Dial In User Service)

RADIUS is a system procedure that offers centralized entrance, approval, as well as accounting administration for individuals or computers to add and utilize a network service. Individuals often need "Authentication" when they try to fix to a network. People have to face far more problems while connecting their computers to a telecommunication network. *For example, the telco wants to know the computer operator. When the identification is given, it may ask what services the user prefers and at the moment the telco collects billing dates concerning the consumed time or capability.*

A RADIUS server utilizes a central database to authenticate remote users. RADIUS functions as a client-server protocol, authenticating each user with a unique encryption key when access is granted.

Radius Features

- Client/Server Model: centralized authentication for remote connections
  - NAS works as a client for the Radius server i.e. enables remote access servers (NAS-network access server) to communicate with a central server.
  - Radius server is responsible for getting user connection requests, authenticating the user, and then returning all the configuration information necessary for the client to deliver service to the user.
  - A Radius server can act as a proxy client to other Radius servers.
- Network Security
  - Transactions between a client and a server are authenticated through the use of a shared key. This key is never sent over the network.
  - Password is encrypted before sending it over the network.
- Flexible Authentication Mechanisms

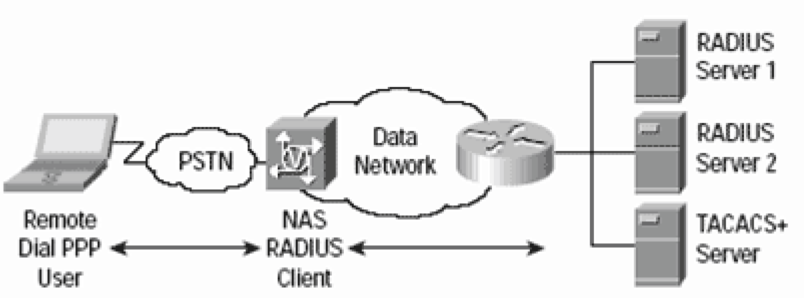
  Radius supports the following protocols for authentication purpose:
  - Point-to-Point Protocol (PPP) - used to establish a direct connection between two nodes.
  - Password Authentication Protocol (PAP)
- Extensible Protocol
  - Radius is extensible; most vendors of Radius hardware and software implement their own dialects.
  - Stateless protocol, using UDP, runs at port 1812.

How a RADIUS server works depends upon the exact nature of the RADIUS ecosystem.

- First, the user initiates authentication to the network access server (NAS).
- The network access server then requests either a username and password or a challenge (CHAP).
- The user replies.
- Upon receiving the user's reply, the RADIUS client sends the username and the uniquely encrypted password to the RADIUS server.
- The RADIUS server accepts or rejects the user.

Once Client is configured properly then:

- The Client starts with Access-Request.
- The Server sends either Access-Accept, Access-Reject, or Access-Challenge.
- Access-Accept keeps all the required attributes to provide service to the user.



The Network Access Server (NAS) is a service element that clients dial in order to get access to the network. An NAS is a device having interfaces both to the backbone and to the POTS or ISDN and receives calls from hosts that want to access the backbone by dialup services. NAS is located at an Internet provider's point of presence to provide Internet access to its customers.

RADIUS is a protocol for carrying information related to authentication, authorization, and configuration between a Network Access Server that desires to authenticate its links and a shared Authentication Server.

RADIUS server has following functions — **AAA**

Authentication, : Verify the user is who he/she claims to be :-

- Use Password, Special Token card, Caller-ID, etc.
- May issue additional 'challenge'

Authorization:

- Check that the user may access the services he/she wishes.
- Check database or file information about the user

Accounting.

- Record what the user has done.
- Time online. Bytes sent/received. Services accessed. Files downloaded. etc.

The main advantage of the centralized AAA capabilities of a RADIUS server are heightened security and better efficiency. RADIUS servers provide each business with the ability to preserve the privacy and security of both the system and each individual user.

Hence, RADIUS enables centralized running of certification data like usernames and passwords. The RADIUS server can accumulate these certified data locally, but it may also store authentication data in an outdoor SQL database or even an external Unix file. In fact, the RADIUS is an exceptional option to do accounting without any hassle. It can also improve safety by enabling password executive centralization. Overall, if people take over the RADIUS server, they would have everything.

Applications

1. **Telecom**: the telco wants to know the computer operator. When the identification is given, it may ask what services the user prefers and at that moment, the telco collects billing dates concerning the consumed time or capability.

To solve all these problems and allow people to easily connect their computers to the telecommunication network, most the widespread open source and decorum systems use RADIUS. Telcos and other companies frequently put systems associated with RADIUS into services to identify their customers or employees with ease. RADIUS is good to use because it can easily determine the users' authorized rights to execute and create a testimony of the entrance in the server's "Accounting" feature.

2. **ISPs** : use to verify authentication, authorization and to track accounting of users.

Overall, RADIUS is good for Internet service providers and companies to identify their customers or workers with ease. It can help users connect their computers to telecommunication without hassle.

***Mail Server**

Email service is one of the most often *used services globally*. Today almost *everyone has at least one* email account. Although clicking on the email send button and delivery of an email message appear seamless, *a lot of events take place behind the scenes to make sure that the email reaches its final destination.*

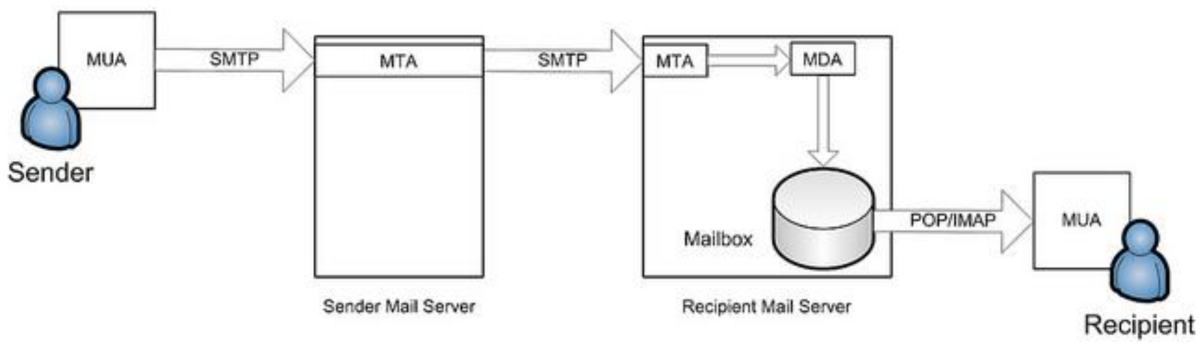
The *functionality of a mail server can be divided broadly into two processes*: sending and receiving emails. The following two protocols oversee these processes.

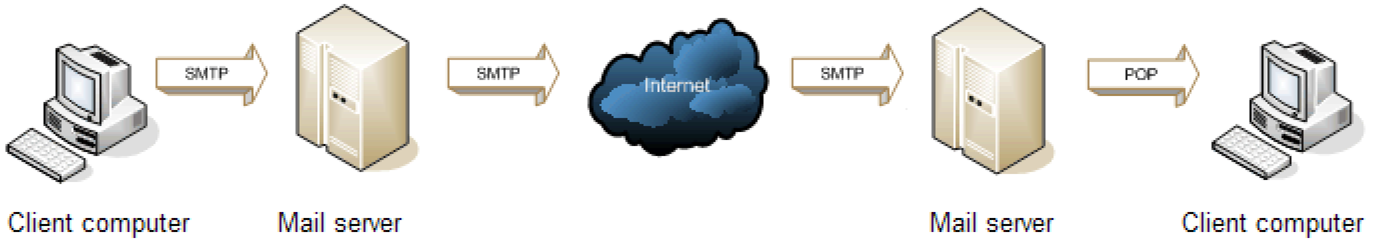
- **Sending email:** Simple Mail Transfer Protocol (SMTP)
- **Receiving email:** Post Office Protocol (POP) / Internet Message Access Protocol (IMAP)

Terminology :

- **Mail User Agent (MUA):** The MUA is a component which *interacts with end users directly*. Examples of MUA *are Thunderbird, MS Outlook, Zimbra Desktop. Web mail interfaces like Gmail and Yahoo! are also MUA.*
- **Mail Transfer Agent (MTA):** The MTA is responsible for transferring an email from a sending mail server all the way to a recipient mail server. *Examples of MTA are send mail and postfix.*
- **Mail Delivery Agent (MDA):** Within a destination mail server, local MTA accepts an incoming email from remote MTA. The email is then *delivered to user's mailbox by MDA.*
- **POP/IMAP:** POP and IMAP protocols are used to *fetch emails from a recipient server's mailbox* to recipient MUA.
- **Mail Exchanger Record (MX):** *The MX record is the DNS entry for mail servers.* This record points to the IP address towards which emails should be shot. The lowest MX record always wins, i.e., gets the highest priority. For example, MX 10 is better than MX 20.

Block Diagram - Mail Server Operation





When a sender clicks on the send button, SMTP (MTA) ensures end to end delivery of an email from a sender-side server to a destination server. Upon reaching the destination server, the MTA local to the destination server accepts the email, and hands it over to the local MDA. The MDA then writes the email to a receiver's mailbox. When the recipient checks for emails, they are fetched by MUA by using protocols like POP or IMAP.

5.4. Cookies: *HTTP State Management*

We said earlier that HTTP is a stateless protocol. We also said that stateful protocols can provide improved performance. This feature is usually established by the idea of a "session" between client and server. So, cookies enable HTTP sessions.

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

Authentication cookies *are the most common method that is used by web servers to know whether the user is logged in or not, and which account they are logged in under.*

The cookies consist of several values;

- The name and value that are encoded into the cookie and represent the state. The interpretation is that the name has an associated value.
- The expires field indicates when the cookie is valid. Expired cookies are not to be given out. The cookie will be deleted at the end of the session if this field is not present.
- The domain states the domain for that cookie is intended. It consists of the last n fields of the domain name of a server. *For example, domain=.adv.com indicates that the cookie is to be sent to any requesting server in the adv.com domain. A domain field should have at least one embedded "." in it. It is not required that a cookie is sent from a host in the domain.*
- The path further limits the dissemination of the cookie. When a Web server requests a cookie, it procures a domain. Cookies that match the domain may be sent to the server. If the server indicates a path, the path should be the leading substring of the path specified in the cookie.
- If the secure field is set, the cookie will be sent over only secured connections.

Characteristics of Cookie

- ✓ Cookies are domain specific i.e. a domain e.g. facebook.com cannot read or write to a cookie created by another domain e.g. yahoo.com. This is done by the browser for security purpose.
- ✓ Cookies are browser specific. Each browser stores the cookies in a different location. The cookies are browser specific and so a cookie created in one browser (e.g. in Google Chrome) will not be accessed by another browser (Internet Explorer/Firefox).
- ✓ Most of the browsers store cookies in text files in clear text. So, it's not secure at all and no sensitive information should be stored in cookies.
- ✓ Most of the browsers have restrictions on the length of the text stored in cookies. It is 4096(4kb) in general but could vary from browser to browser.
- ✓ Some browsers limit the number of cookies stored by each domain (20 cookies). If the limit is exceeded, the new cookies will replace the old cookies.

✓ Cookies can be disabled by the user using the browser properties. So, unless you have control over the cookie settings of the users (for e.g. intranet application), cookies should not be used.

✓ Cookie names are case-sensitive. E.g. UserName is different than username.

Advantages of using cookies

✓ Cookies are simple to use and implement.

✓ Occupies less memory, do not require any server resources and are stored on the user's computer so no extra burden on server.

✓ We can configure cookies to expire when the browser session ends (session cookies) or they can exist for a specified length of time on the client's computer (persistent cookies).

✓ Cookies persist a much longer period of time than Session state.

Disadvantages of using cookies

✓ As mentioned previously, cookies are not secure as they are stored in clear text they may pose a possible security risk as anyone can open and tamper with cookies. You can manually encrypt and decrypt cookies, but it requires extra coding and can affect application performance because of the time that is required for encryption and decryption

✓ Several limitations exist on the size of the cookie text (4kb in general), number of cookies (20 per site in general), etc.

✓ User has the option of disabling cookies on his computer from browser's setting.

✓ Cookies will not work if the security level is set to high in the browser.

✓ Users can delete a cookie.

✓ Users browser can refuse cookies, so your code has to anticipate that possibility.

✓ Complex type of data not allowed (e.g. dataset etc.). It allows only plain text (i.e. cookie allows only string content)

Types of cookie:

***Session cookie:** A user's session cookie for a website *remains only while the user is reading and navigating the website.* A session cookie is created when an expiry date is not set at cookie creation time. Web browsers generally 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 the initial value set in which cookie would be sent back to the server every time the user visited the server within a year. This would be used to record an important piece of information such as *how the user initially came to this website*. Because of this reason persistent cookies are also called *tracking cookies.*

***Secure cookie:** A secure cookie has the *secure attribute that is enabled and is only used via HTTPS, assuring that the cookie is always encrypted when transmitting from client to server.*

***Http Only cookie**: The Http Only cookie is supported by most of the modern browsers.

***First-party cookie:** First-party cookies are cookies set with the same domain in our browser's address bar.

***Third-party cookie:** Third-party cookies are those cookies being *set with various domains from the one shown on the address bar i.e.* the web pages on that domain can have content from a third-party domain - *e.g.* an advertisement run by www.advexample.com showing advert banners from other domain e.g. <img src="http://www. advexample.com/banner.jpg">

***Supercookie :** A supercookie is a type of tracking cookie inserted into an HTTP header by an internet service provider (ISP) to collect data about a user's internet browsing history and habits. Supercookies can be used to collect a wide array of data on users' personal internet browsing habits including the websites users visit and the time they visit them. It does not matter which browser is being used or if users switch browsers.

***Zombie cookie:** A zombie cookie is *any cookie which is automatically recreated after deleting it*. This is accomplished by a script that stores the content of the cookie in some other locations, such as the local storage that is 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.

How Cookies Works ?

- Consider user browse a new webpage
- At first, webpage request to server, the web server issues a cookie.
- The server sends back with requested page and cookies to the web browser.
- The browser stores the cookie in memory and sends back to the server with each subsequent request.
- The server inspects each request, the cookie is present, the server maintain state regarding the user (identity, old or new user, activity)

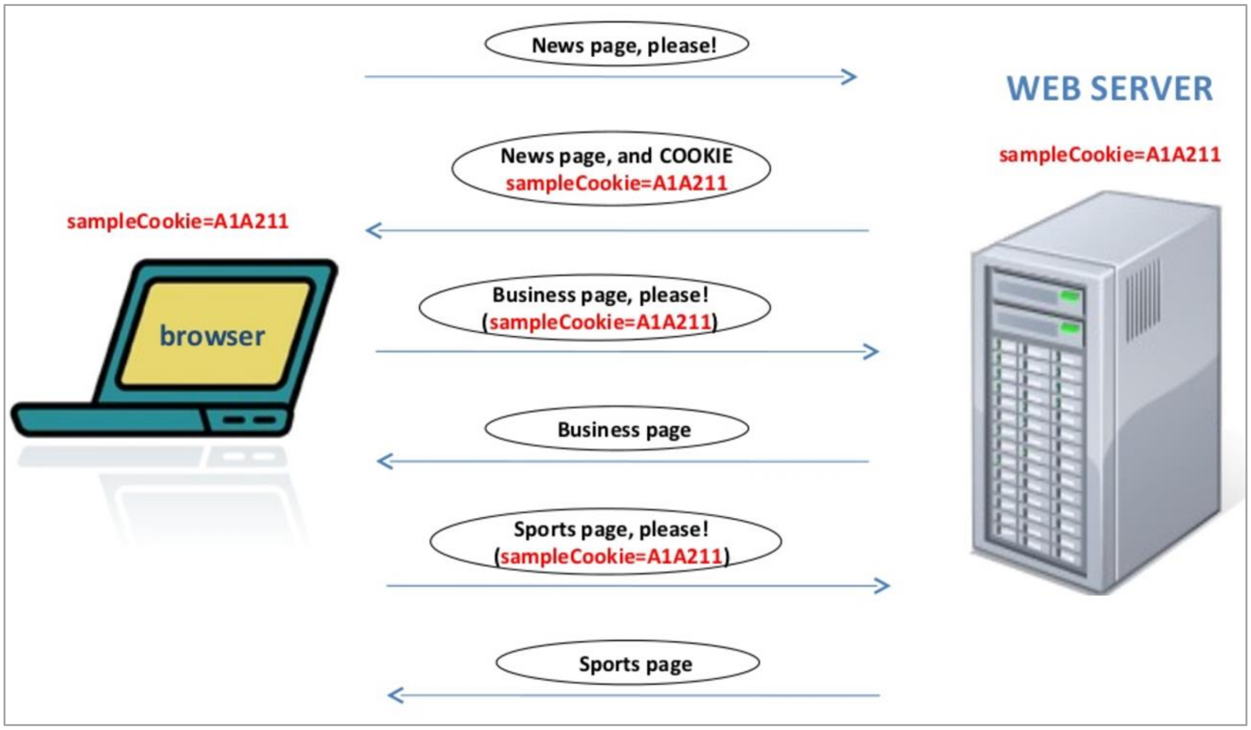

News page, please!
WEB SERVER
sampleCookie=A1A211
News page, and COOKIE sampleCookie=A1A211
sampleCookie=A1A211
browser
Business page, please! (sampleCookie=A1A211)
Business page
Sports page, please! (sampleCookie=A1A211)
Sports page

Fig. How Cookie Works ?

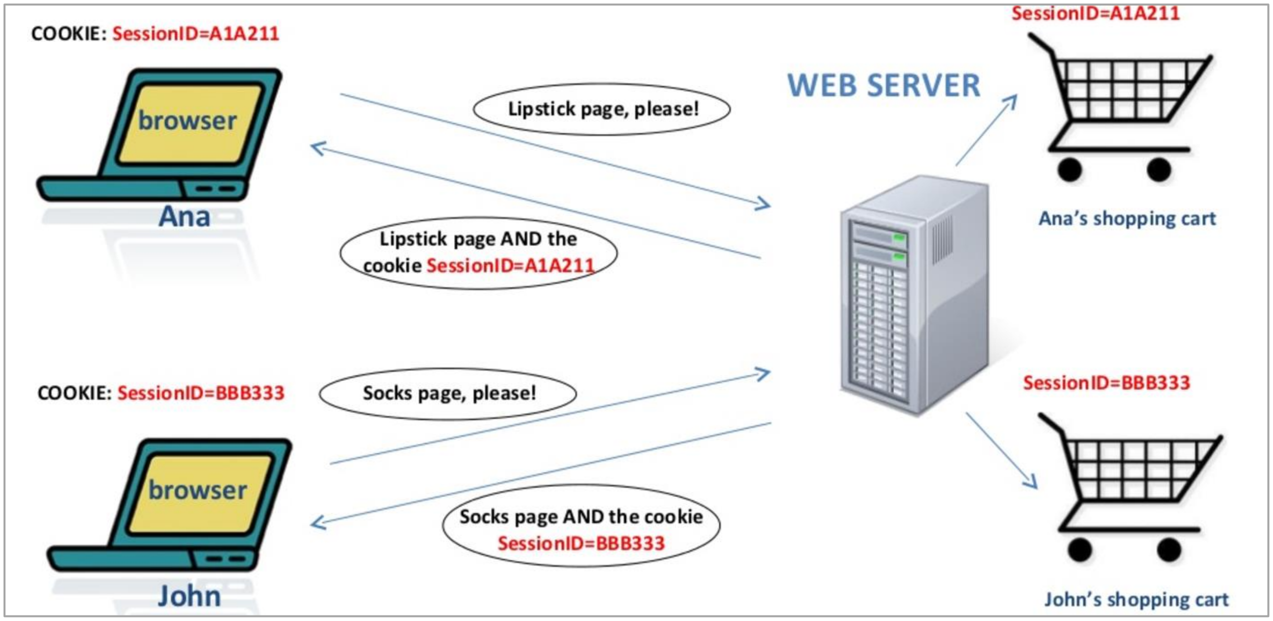


Fig. Cookie used in e-commerce shopping cart

Uses of Cookies:

*Session Management: Cookies may be used in maintaining data that is related to the user during navigation, possibly across multiple visits. *Cookies were introduced to procure a way to implement a "shopping cart", a virtual device into which users can store items what they want to purchase as they navigate throughout the site.*

Allowing users to log in to a website is an often use of cookies. Generally, the web server will first send a cookie that contains a unique session identifier. Then only users submit their credentials and the web application authenticates the session and it allows the user access to services.

*Personalization: Cookies are also used to remember the information about the user regarding their visit a website in order to show relevant content in the future. For example, a web server can send a cookie that contains the username last used to login to a website so that it can be filled in for future visits.

*Tracking: Tracking cookies can be *used to track internet user's web browsing* which can also be done in part by using the IP address of the computer that requests the page or the referrer field of the HTTP request header, but the cookies allow for greater precision.

5.5. Load Balancing: Proxy Arrays

Load Balancing improves the distribution of workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units, or disk drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. Load balancing usually involves dedicated software or hardware, such as a multilayer switch or a Domain Name System server process. Load balancing divides traffic between network interfaces on a network socket (OSI model layer 4) basis

Problem: Single physical Origin or Proxy Server may not be able to handle its load
Solution: install multiple servers and distribute the requests.
How do we distribute requests among the servers?

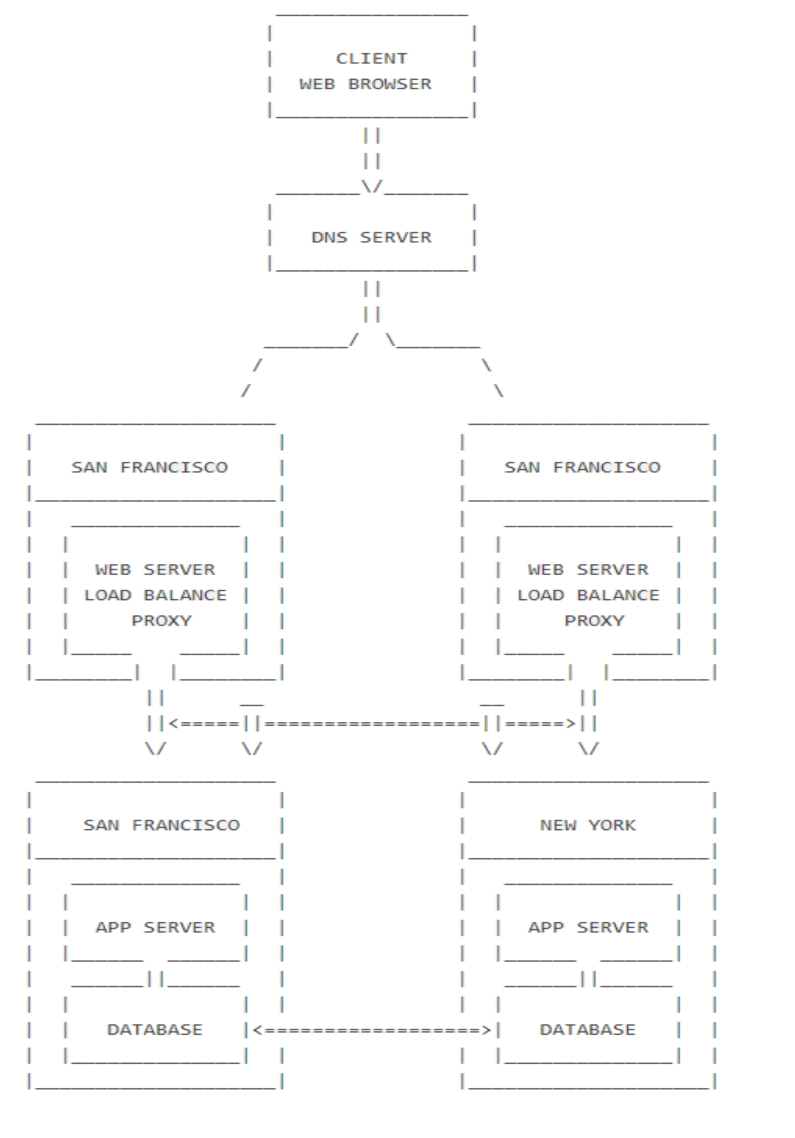


Fig. DNS Round Robin

*DNS Round Robin

- DNS RR is a simple technique of load balancing various Internet services such as Web server, e-mail server by creating multiple DNS A records with the same name.
  DNS is configured so multiple IP Addresses correspond to a single host name
- *Modify the DNS server to round-robin through the IP addresses for each new request*
- This way, different clients are pointed to different servers

How Does It Works?

You configure DNS server to send a list of IP addresses of several servers with same hostname. For example, foo.dnsknowledge.com may be configured to return two IP address as follows:

- foo.dnsknowledge.com – 202.54.1.2
- foo.dnsknowledge.com – 202.54.1.3

Half of the time when a user make foo.dnsknowledge.com request will go to 202.54.1.2 and rest will go to 202.54.1.3. In other words, all clients would receive service from two different server, thus distributing the overall load among servers.

Round Robin DNS Usage

1. Load distribution.
2. Load balancing.
3. Fault-tolerance service.

***ICP Internet Cache Protocol**

- ICP is a UDP-based protocol used for coordinating web caches by *querying proxy servers for cached documents*
  - o Its purpose is to find out the most appropriate location to retrieve a requested object from in the situation where multiple caches are in use at a single site. The goal is to use the caches as efficiently as possible, and to minimize the number of remote requests to the originating server.
  - o Typically used by proxy servers to check other proxy server's cache

Using the Internet Cache Protocol (ICP)

The Internet Cache Protocol (ICP) is an object location protocol that enables caches to communicate with one another. Caches can use ICP to send queries and replies about the existence of cached URLs and about the best locations from which to retrieve those URLs. In a typical ICP exchange, one cache will send an ICP query about a particular URL to all neighboring caches. Those caches will then send back ICP replies that indicate whether they contain that URL. If the caches do not contain the URL, they send back **miss**. If they do contain the URL, they send back **hit**.

Routing Through ICP Neighborhoods

ICP can be used for communication among proxies located in different administrative domains. It enables a proxy cache in one administrative domain to communicate with a proxy cache in another administrative domain. It is effective for situations in which several proxy servers want to communicate, but cannot all be configured from one master proxy as they are in a proxy array. Figure shows an ICP exchange between proxies in different administrative domains.

The proxies that communicate with each other through ICP are called **neighbors**. You cannot have more than 64 neighbors in an ICP neighborhood. The two types of neighbors in an ICP neighborhood are **parents** and **siblings**. Only parents can access the remote server if no other neighbors have the requested URL. Your ICP neighborhood can have no parents or it can have more than one parent. Any neighbor in an ICP neighborhood that is **not** a parent is considered a sibling. Siblings cannot retrieve documents from remote servers unless the sibling is marked as the default route for ICP, and ICP uses the default.

Each neighbor in an ICP neighborhood must have at least one ICP server running. If a neighbor does not have an ICP server running, it cannot answer the ICP requests from their neighbors. Enabling ICP on your proxy server starts the ICP server if it is not already running.

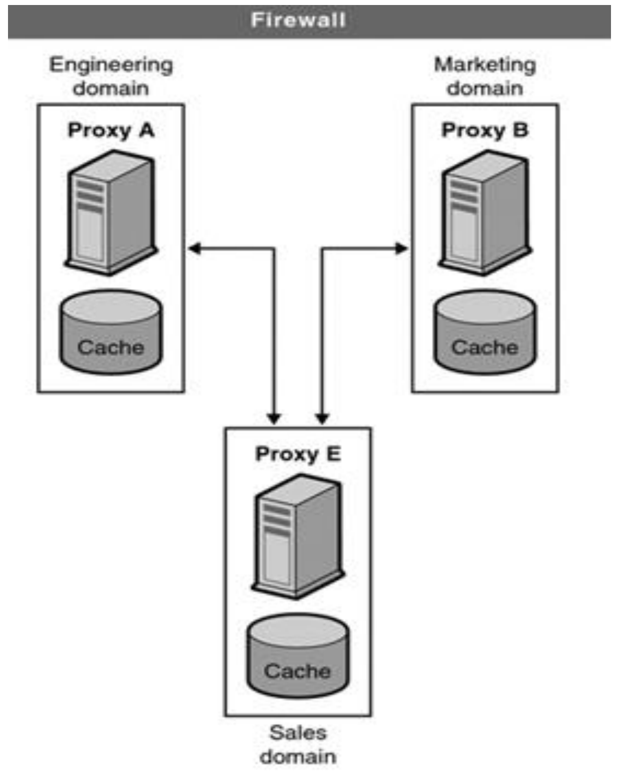


Fig. ICP Exchange

*Non-redundant Proxy Load Balancing

- Proxy selection based on a hash function
- Hash value is calculated from the URL
- Use resulting hash value to choose proxy
- Use Host name in hash function to ensure request routed to same proxy server (why?)

Cache Array Routing Protocol (CARP)

The **Cache Array Routing Protocol** (CARP) is used in load-balancing HTTP requests across multiple proxy cache servers. It works by generating a hash for each URL requested. A different hash is generated for each URL and by splitting the hash namespace into equal parts (or unequal parts if uneven load is intended) the overall number of requests can be distributed to multiple servers.

Caching Array Routing Protocol (CARP) is implemented as a series of algorithms that are applied on top of Hypertext Transfer Protocol (HTTP). CARP allows a Web browser or downstream proxy server to determine exactly where in the proxy array the information for a requested Uniform Resource Locator (URL) is stored.

CARP enables proxy servers to be tracked through an array membership list that is automatically updated using a Time to Live (TTL) countdown function. This function regularly checks for active proxy servers in the array. CARP uses hash functions and combines the hash value of each requested URL with each proxy server. The URL/proxy server hash with the highest value becomes the owner of the information cached. This results in a deterministic location for all cached information in the array, which enables a Web browser or downstream proxy server to know exactly where a requested URL is locally stored, or where it will be located once it has been cached. The hash functions result in cached information being statistically distributed (load balanced) across the array. Using hashing means that massive location tables for cached information need not be maintained—the Web browser simply runs the same hashing function on the object to locate where it is cached.

CARP provides two main benefits:

- It saves network bandwidth by avoiding the query messaging between proxy servers .
- It eliminates the duplication of content that occurs when proxy servers are grouped in arrays, resulting in faster response times and more efficient use of server resources.

Hash-based proxy selection mechanism

- No queries - hashing used to select server
- Highly scalable
  - *performance improves as size of array increases*
  - *automatically adjusts to additions/deletions of servers*
- Eliminates cache redundancy
- No new protocols!

How CARP Works

- Given an array of Proxy servers
- Assume array membership is tracked using a membership list
- A hash value ***Hs*** is computed for the name of each proxy server in list (only when list changes)
- A hash value ***Hu*** is computed for the name of each requested URL

- *For each request, a combined hash value **Hc** = F(**Hs** , **Hu**) is computed for all servers and Use highest **Hc** to select server*

CARP: Hierarchical Routing

- One server acts as director using Hash routing.
- Cache hit rate is maximized (why?)
- Single point of failure (use DNS RR?)

CARP: Distributed Routing

- Requests can be sent directly to ANY member of the Array.
- Route request to best score if not me.
- Don't cache response if redirected

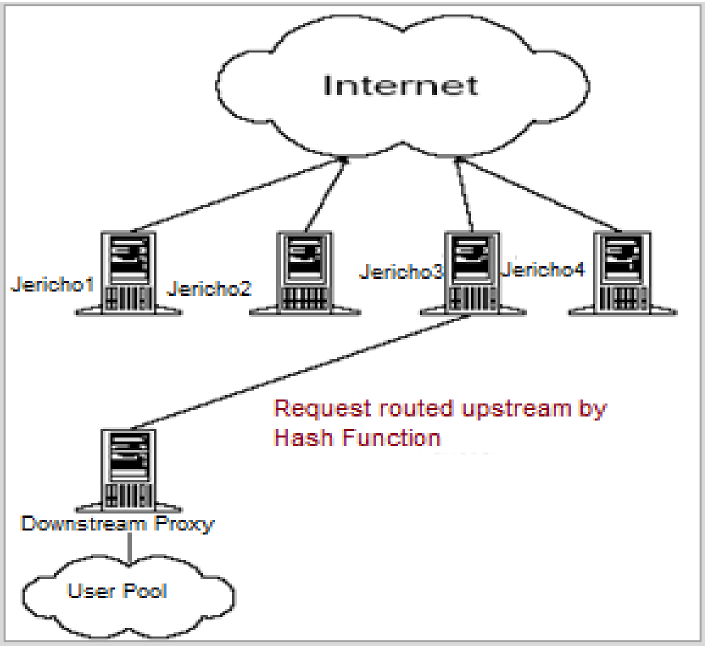


Fig. CARP: Hierarchical Routing

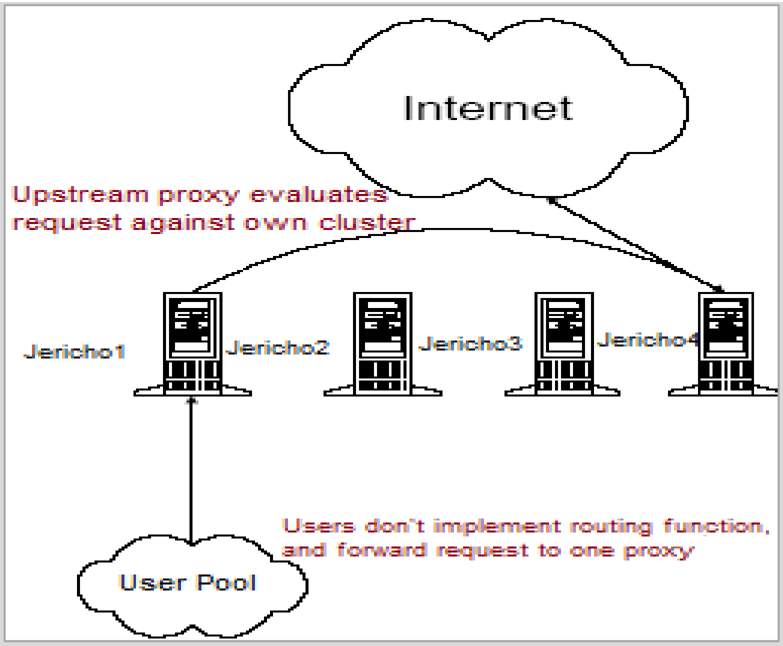

Internet
Upstream proxy evaluates
request against own cluster
Jericho1
Jericho2
Jericho3
Jericho4
Users don't implement routing function,
and forward request to one proxy
User Pool

Fig. CARP: Distributed Routing

CARP Features

- Assume the membership stays the same
- Then a given URL always maps to the same Proxy (because the hash functions are deterministic)

– Thus, a given page always resides in the same proxy – So caching works And pages are not stored redundantly

- When a membership of size n changes by one, only 1/n th of the URLs are remapped as shown in Fig2

The CARP Hash Functions

- Host (server) Hash

```
– Computations use 32 bit UNSIGNED integers
HS = 0; // initially
for each character Ci in host name
        HS += R(HS, 19) + Ci // where R(x,n) ::= logical left rotate x by n
End for
HS += HS*0x62531965
HS = R(HS , 21)
```

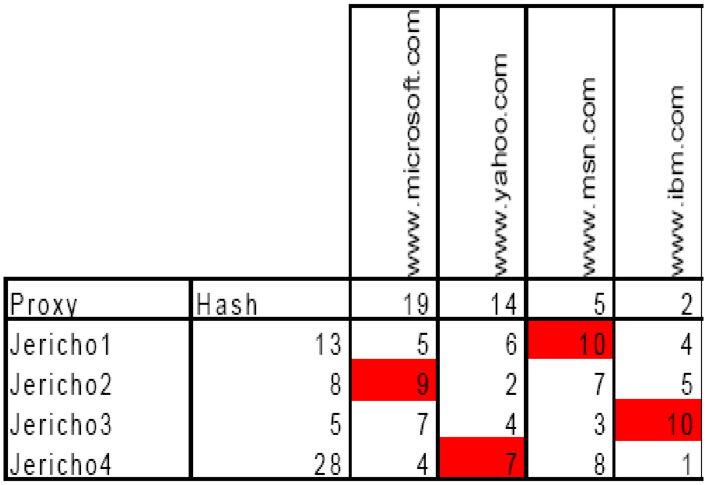
- URL Hash

```
– Computations use 32 bit UNSIGNED integers
HU = 0; // initial HU = 0;
for each character Ci in URL
        HU += R(HU, 19) + Ci
End for
```

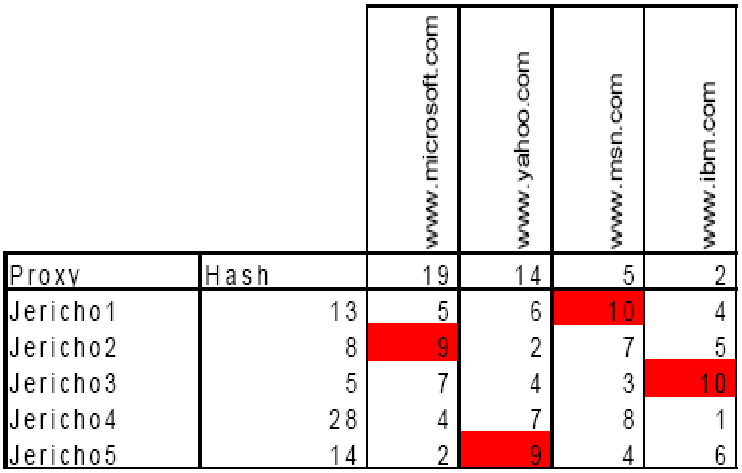
- Combining Hash Function

```
– Again, all computations are performed using 32-bit unsigned integers
HC = HU ^ HS // [exclusive OR]
HC += HC*0x62531965
HC = R(HC , 21)
```

CARP Example



		www.microsoft.com	www.yahoo.com	www.msn.com	www.ibm.com
Proxy	Hash	19	14	5	2
Jericho1	13	5	6	**10**	4
Jericho2	8	**9**	2	7	5
Jericho3	5	7	4	3	**10**
Jericho4	28	4	**7**	8	1



		www.microsoft.com	www.yahoo.com	www.msn.com	www.ibm.com
Proxy	Hash	19	14	5	2
Jericho1	13	5	6	**10**	4
Jericho2	8	**9**	2	7	5
Jericho3	5	7	4	3	**10**
Jericho4	28	4	7	8	1
Jericho5	14	2	**9**	4	6

Note the distribution of URL across servers

Fig 1.CARP: adding a new server

Fig2.A 5th server is added and effects only 1/5 of the existing mappings

5.6. Server Setup and Configuration Guidelines

•**Hardware/The Basics: Environment**

•Redundant Power -Two power supplies
UPS source – protects against grid failure ▫"Dirty" source – protects against UPS failure

•Redundant Cooling ▫What happens if one of the fans fail? Facility has air-conditioning backup ▫...or some other cooling system?

•Redundant processors ▫Consideration also, but less important ▫Partner router device is better •Redundant interfaces ▫Redundant link to partner device is better

Redundant cabling ▫Cable break inside facility can be quickly patched by using "spare" cables ▫Facility should have two diversely routed external cable paths

•RAID •RAID 0 •RAID 1 •RAID 5 •RAID 10 •RAID 15

Operating System / Firewall

Platform ▫Windows ▫Linux

•Should have corporate level firewall ▫Packet filtering ▫Application level ▫IDS

Number of sessions and load balancing

•Threading should be increased •Beside Threading there should be load balancing Hardware that should be responsible for load balancing in the server either packet wise or session wise in the replicated server.

**** Activation :*** When joining the new *domain the activation of Windows should occur automatically via KMS activation and should be confirmed.* Note that this could take a couple of restarts, following updates for example.

**** Updates & Antivirus :*** The server *should have the latest OS and application updates applied*. If necessary, Windows Updates should be installed first thing.

- A well---*defined update and antivirus strategy should be established*.
- *Routine maintenance,* including patching and updating, should be regularly scheduled and documented.
- Antivirus software should be *installed*, updated and activated.
- Consider *disabling the automatic clean---up of malware on mission critical application servers*. It is best to configure the antivirus software to isolate/quarantine the infection for manual evaluation as false positives can cause downtime or data loss.
- Evaluate the need for on---*access scanning in antivirus software*. On---Access scanning will be initiated with each read and/or write to the disk --- this can create additional overhead and unnecessary processing/resource drain on busy servers.
- Configure a regularly *scheduled full system scan.*
- In some scenarios it will be best practice to *configure email alerts or notifications* in the antivirus software. In the case of an unattended server, for example, no one will be logged in to see the default desktop alerts.

* ***Windows Firewall and Services :*** *The firewall should be enabled and configured with the most restrictive settings possible*.

The following practices are a small set of a Windows firewall and services strategy.

- Configure the firewall *with the most restrictive settings* possible and *to allow only the IP range(s)* expected.
- *Unused services* should not be allowed to *start as 'automatic' and ports* should be evaluated.

- *Remove all unnecessary services, features or applications* from the server. These may differ based on the role the server will fill.
  - o Some examples of services to *disable are Telnet and FTP.*
  - o *Disable web browsing* on servers unless running a terminal server.

* ***User Accounts and Passwords :*** *Default accounts and default or weak passwords should be disabled, renamed, or modified.*

The following should be part of an account and password strategy.

- *Disable or rename* the *Administrator account, all generic Guest accounts, default passwords*, or at a minimum change to very strong passwords.
- *Do not* allow *auto---login* and *Restrict* the use of *blank passwords*.
- *Enable screen saver, screen---locking.*
- Disable the setting that reads as "Interactive logon: Do not require CTRL+ALT+DEL."

* ***Remote Desktop and Access Control :*** The file system should have *a well---documented access control strategy allowing for only authenticated user access.* The following should be part of an access control and remote desktop strategy:

- *Remote access should be disabled or restricted* to specific IP addresses by default.
- *Directories, files, and shares should be evaluated for permissions*, including close analysis that the Everyone group not be given access to shares with sensitive/secure data.
- *Administrative shares*, should be disabled or audited for access.
- No open or non---authenticated file sharing should be allowed.

* ***Auditing, Backup & Recovery :*** The following should be part of an auditing and event log management strategy:

- A strategy should be *established for regularly reviewing audit logs*, either manually or programmatically. This should include system logs and service logs.
- Audit *account logon events, account management, directory service access, policy change, system events.*
- Auditing of *privileged accounts* should be enabled, specifically on failure and *administrative share access*, if used, should be enabled.
- *Disaster recovery* planning for each server should be documented and include details about the back---up methods, recovery and restoration of the system and applications as well as data.
- *At least one backup* should be stored in a different location as the server itself.

* ***File Transfers :*** Follow the best practices and guidelines for securing the server listed above first, and then consider the role your server will play in file transfer – *will it be sending only, receiving only, or both*? Using WinSCP for the sending of files to another server is a secure method of transferring files out. If establishing a receiving server for *file transfers be sure to use SFTP and strong user/password combinations, firewall settings, and all other OS best practices.*

**** Disposal :*** Media destruction takes several forms, including physical destruction or electronic destruction.

- *Physical destruction* of media can take several forms, such as drilling holes in the physical drive.
- *Electronic destruction* needs to follow certain guidelines, and ITS suggests the standard DoD and NIST guidelines for 3---pass wiping of a drive.

5.7. Security and System Administration Issues, Firewalls and Content Filtering

The system administrator basically responsible for following things:

- User administration (setup and maintaining account)
- Maintaining system
- Verify that peripherals are working properly
- Quickly arrange repair for hardware in occasion of hardware failure
- Monitor system performance
- Create file systems
- Install software
- Create a backup and recovery policy
- Monitor network communication
- Update system as soon as new version of OS and application software comes out
- Implement the policies for the use of the computer system and network
- Setup security policies for users. A sysadmin must have a strong grasp of computer security (e.g. firewalls and intrusion detection systems)
- Documentation in form of internal wiki
- Password and identity management

Network Security Issues and Solutions

1. Non-complex or Weak Network Access Passwords

Most network system administrators are open to an "old school" exploit known as brute forcing. In order to correct this network security password vulnerability, they have *implemented "CAPTCHA Technology."* A common type of CAPTCHA requires the user to type letters or digits from a distorted image that appears on screen, which is commonly used to prevent unwanted internet bots from accessing websites and networks. *This technology has given network security administrators a false sense of security, in regard to countering brute forcing.*

The solution? A complex password. In order to create a complex password, you need seven or more characters combined with at least three numbers and one special character (capital letters, @ or # signs, etc.). Network security administrators should require the creation of complex passwords as well as implement a *password expiration system* to help remind users to change their passwords often. A restriction on how soon a password can be reused is also another handy precaution, that way someone isn't cycling between two different passwords every month or so.

2. Outdated Server Application or Software

Companies constantly *release patches in order to ensure that your system* is not vulnerable to new public threats. Hackers consistently release new threats and exploits which could allow harm to befall your network if these patches are not in place. A simple solution is to ensure your system administrator is *regularly informed of new threats and is updating your applications on a monthly basis.*

3. Web Cookies

Although cookies do not carry viruses and cannot install malware on the host computer, the tracking of cookies and third-party tracking cookies are commonly used ways to compile records of individuals' browsing histories. *Unencrypted cookies are a major network security issue* because they can open your system to a XSS (Cross Site Scripting) vulnerability and that is a major privacy concern. With 'Open Cookies' anyone could have access to any login data cookies (saved password sessions) on the network, which creates a major vulnerability on your network security system.

The solution is to ensure all of your network *cookies are encrypted and have an encoded expiration time.* Your network administrator should also force users to re-login any time they are accessing sensitive directories in your network.

4. Plain Hashes

Anyone who knows their stuff can decrypt a Hash that is not Salted.

Hashing is used to index and retrieve items in a database and Plain Hashes are also used in many encryption algorithms. A **Salt** (which is another type of encryption) is added to Hashes in order to make a lookup table assisted Directory Attack (or Brute-Force) impractical or extremely difficult, provided the Salt is large enough. Basically, an attacker wouldn't be able to use a pre-computed look up table to assist in exploiting your network, which adds a whole new level of complexity to your network security system. *So even if an attacker gains access and compromises your database (table), it will still be very difficult for the attacker to retrieve the information.*

The best way to ensure safety in regard to Hashes is for your network administrator to *hide the Salt (or encryption key),* because if the hacker is able to gain access to your Salt encryption they can access your network system. Salt all of your Hashes. No Salt means no security.

5. Share Hosting (not Cloud Server Base)

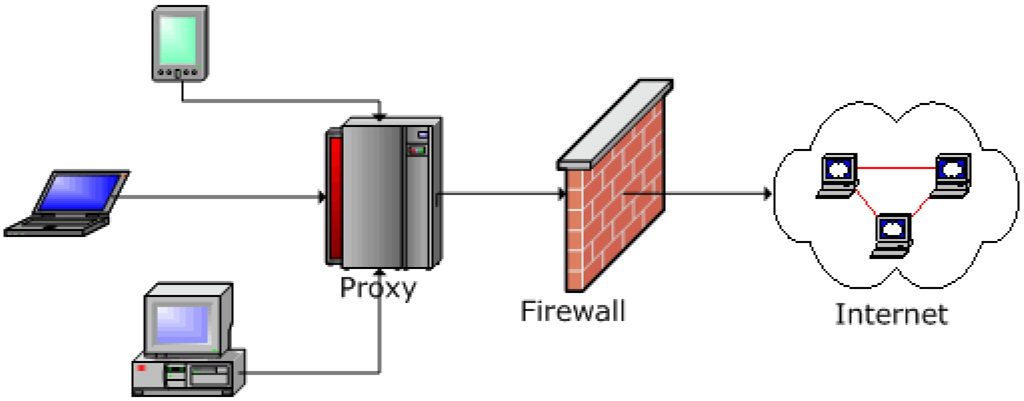
If you are running a sincere business and have a website with access to your internal network, Shared Hosting is not the way to go! A shared web hosting service is where many websites reside on one web server connected to the Internet. Each site sits on its own partition, or section or space on the server, to keep it separate from other sites. This is generally the most economical option for hosting, because people share the overall cost of server maintenance. Think of it this way: shared hosting is like sharing a house with other people, and if someone breaks into your roommate's bedroom or any other area of the home for that matter, they'll also be able to access your own room! This same concept is applied to Shared Hosting. When an attacker is inside one area of the shared server, it's almost as if they have a skeleton key that fits all of the locks. *The best solution is to have dedicated Server Hosting and/or Secure Cloud Hosting.*

*FIREWALLS

A firewall is a network security system designed *to prevent unauthorized access to or from a private network.* Firewalls can be implemented in both *hardware and software, or a combination of both.* Network firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially *intranets*. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria

Hardware and Software Firewalls *:* Firewalls can be either hardware or software but the ideal configuration will consist of both. In addition to *limiting access to your computer and network, a firewall is also useful for allowing remote access to a private network* through secure authentication certificates and logins.

- **Hardware firewalls** can be purchased as a stand-alone product but are also typically found in broadband routers, and should be considered an important part of your system and network set-up. Most hardware firewalls will have a minimum of four network ports to connect other computers, but for larger networks, business networking firewall solutions are available.
- **Software firewalls** are installed on your computer (like any software) and you can customize it; allowing you some control over its function and protection features. A software firewall will protect your computer from outside attempts to control or gain access your computer.



Types of Firewall

- Personal Firewall : generally used to protect on personal computer in small network
- Departmental Firewall : used to protect small business, for limited no of computers
- Enterprises Firewall : used to protect large no of user

Common Firewall Filtering Techniques

Firewalls are used to protect both home and corporate networks. A typical firewall program or hardware device filters all information coming through the Internet to your network or computer system. There are several types of firewall techniques that will prevent potentially harmful information from getting through:

- **Packet Filter:** Looks at each packet entering or leaving the network and accepts or rejects it based on user-defined rules. Packet filtering is fairly effective and transparent to users, but it is difficult to configure. In addition, it is vulnerable to IP spoofing(hacker can modify source information). the packet is either permitted or denied passage through the interface

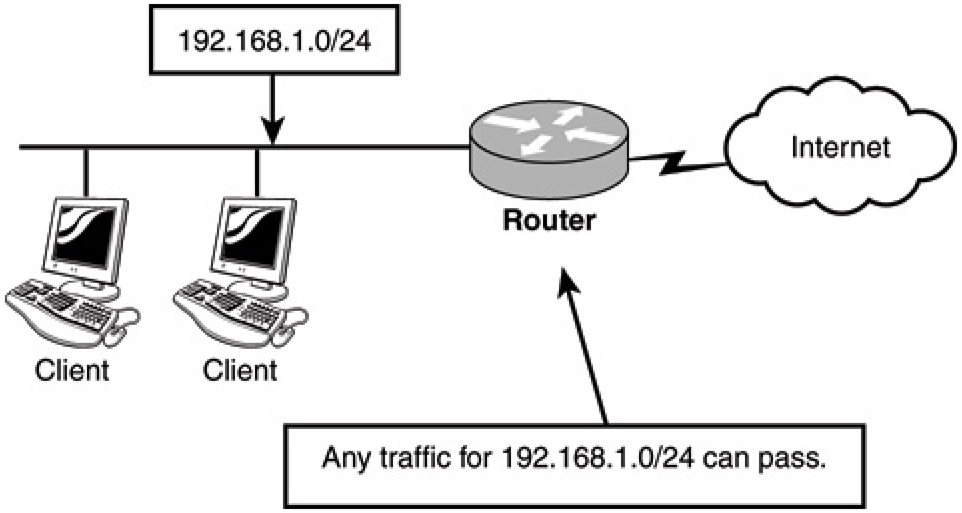
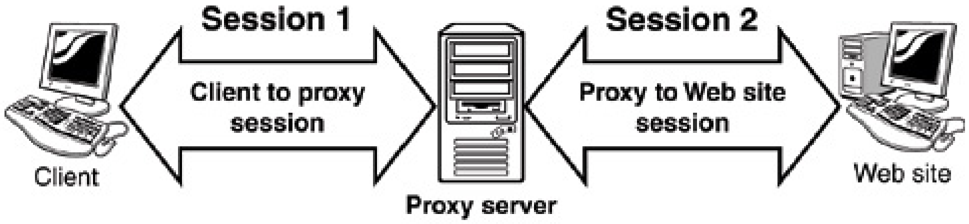


Fig. Basic packet filter.

- **Application Gateway:** Applies security mechanisms to specific applications, *such as FTP and Telnet servers*. This is very effective, but can impose a performance degradation.
- **Circuit-level Gateway:** Applies security mechanisms when a TCP or UDP connection is established. Once the connection has been made, packets can flow between the hosts without further checking.
- **Proxy Filter(Server):** Intercepts all messages entering and leaving the network. The proxy server effectively hides the true network addresses. In practice, many firewalls use two or more of these techniques in concert. A firewall is considered a first line of defense in protecting private information. For greater security, data can be encrypted.



Proxy server sessions.

Next Generation Firewall (NGFW)

A newer class of firewalls, next generation firewall - *NGFW, filters network and Internet traffic based upon the applications or traffic types using specific ports. Next Generation Firewalls (NGFWs) blend the features of a standard firewall with quality of service (QoS) functionalities in order to provide smarter and deeper inspection*

Content Filtering

A Content Filter helps decide which content is acceptable for viewing and access through a given system. Software that controls content, which is also known as web-filtering programs or censor ware, is a term used for applications created and developed for managing what information or media is allowed to be seen by the end user (specifically content from the Internet).

Content filtering blocks or permits certain types of traffic based on the MIME type, file extension, and protocol command. The content filter controls file transfers across the gateway by checking traffic against configured filter lists.

The content filter module evaluates traffic before all other UTM modules, except Web Filtering. Therefore, if traffic meets criteria configured in the content-filter, the content-filter acts first upon this traffic.

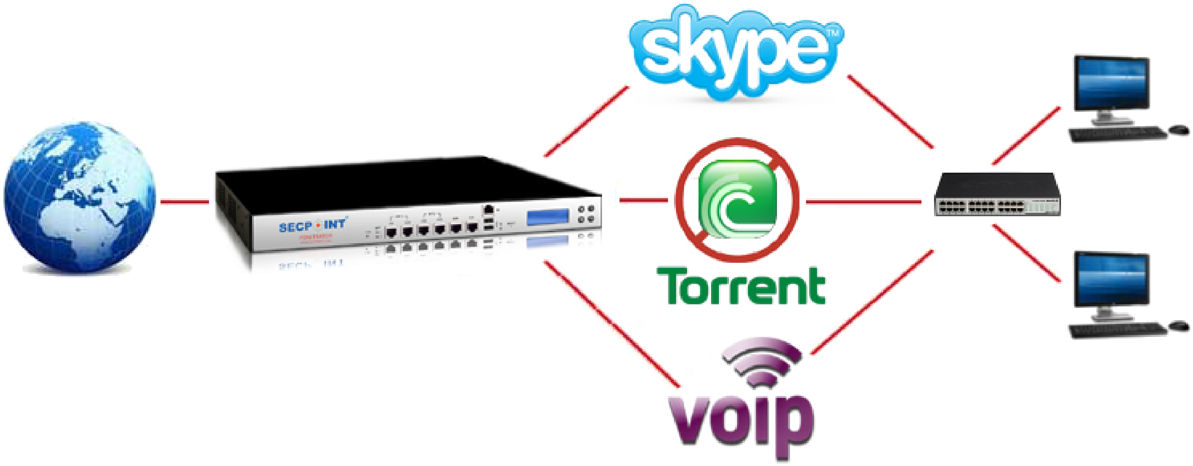
You can configure the following types of content filters:

- ✓ MIME Pattern Filter — MIME patterns (*Multipurpose Internet Mail Extensions (*MIME*) is an Internet standard that extends the format of email to support: Text in character sets other than ASCII. Non-text attachments: audio, video, images, application programs etc*) are used to identify the type of traffic in HTTP and MAIL protocols. There are two lists of MIME patterns that are used by the content filter to determine the action to be taken. The block MIME list contains a list of MIME type traffic that is to be blocked by the content filter. The MIME exception list contains MIME patterns that are not to be blocked by the content filter and are generally subsets of items on the block list. Note that the exception list has a higher priority than the block list. If you have MIME entries that appear on both lists, those MIME types are not blocked by the content filter because the exception list takes priority. Therefore, when adding items to the exception list, it is to your advantage to be specific.
- ✓ Block Extension List — Because the name of a file is available during file transfers, using file extensions is a highly practical way to block or allow file transfers. The content filter list contains a list of file extensions to be blocked. All protocols support the use of the block extension list.
- ✓ Protocol Command Block and Permit Lists — Different protocols use different commands to communicate between servers and clients. By blocking or allowing certain commands, traffic can be controlled on the protocol command level.

The block and permit command lists are intended to be used in combination, with the permit list acting as an exception list to the block list.

Why is Content Filtering needed?

It is important to control the content on your network and know how your resources are being used. Often, employees will tend to do private or illegal things in the work hours, due to boredom or other reasons. This will waste valuable work hours and can possibly put you at risk if your network is being abused for downloading copyrighted materials.



What does the Content Filtering consist off?

- Anti Free Mail : This blocks access to official free email providers such as Hotmail, Yahoo Mail, Google's Gmail, and so on. The use of free email providers can often indicate employees checking their private email during working hours.
- Anti Game : It is often a tempting to play network games such as Counterstrike or other addictive games during work hours.
- Instant Message Recording : This provides monitoring of the usage of MSN Instant Messaging to see if your employees are communicating with your business customers or with their friends.
- Anti Instant Message : If your security policy disallows all sorts of instant messengers, then this module can be used to block programs such as MSN Instant Messenger, Yahoo Messenger, Google Chat, Skype Chat, and so forth.
- Anti VoIP : This allows the blocking of services like Skype, Yahoo Talk, Google Talk, VoIP usage, and lots more. Employees can be talking to non-work-related contacts during work hours or even leak sensitive information without your knowledge.

- Anti P2P : If your security policy requires you to block all P2P file sharing services like BitTorrent, eDonkey2000, Emule, Kazaa, and Napster, then you should enable this module. Those programs are often used to share copyrighted materials such as music or movies. If this is done in your corporate perimeter, you will become responsible for this dilemma once a raid is started. In some countries, ISPs will outright close down the Internet connection of a guilty business, so such a case can become a very costly affair.
- File Filter : This option blocks downloading of specific file formats such as *.exe, *.zip, or *.rar files depending on the supervisor's choice. This applies to emails, web browsing, and other protocols.
- Protocol Filter : This allows blocking of specific protocols in your network. In some locations, POP3 traffic is forbidden since this is often used by employees to check their private email in working hours. You can customize which protocols to block as well.
- Block Websites : This allows blocking of websites of your choice. Often, employees will spend hours daily to read news sites, gossips, and websites of personal interest during working hours

Extra Notes

Designing of Internet System Network Architecture:

The term network architecture is generally used to define a set of abstract principles for the technical design of protocols and mechanisms for computer communication. It represents a group of designed choices out of many design alternatives in which the choices are informed by an understanding of the requirements. The architecture gives a guide for the many technical decisions that is needed to standardize network protocols and algorithms. The purpose of the architecture is to render coherence and consistency to these decisions and to ensure that the requirements are met.

Network architecture is a set of high-level design principles which guides the technical design of the network, generally the engineering of its protocols and algorithms. A network architecture must specify the following points:

- 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 perform.
- How communication functions are modularized, e.g., into “layers” to form a “protocol stack”.
- How network resources are categorized between flows and how end-systems react to this division, i.e., fairness and congestion control.
- Where security boundaries are shaped and how they are enforced.
- How management boundaries are shaped and selectively pierced.
- How differing QoS is requested and achieved?

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

- Internetworking: the existing networks should be interconnected.
- Robustness: Internet communication must continue even though there is loss of networks or routers.
- Heterogeneity: The Internet architecture must accommodate with different network
- Distributed management: The Internet architecture must favor distributed management of its resources
- Cost: The Internet architecture must be effective by cost.
- Ease of Attachment: The Internet architecture must favor host attachment with a low level of effort.
- Accountability: The resources that are used in the internet architecture must be accountable.