CHAPTER 3

3. Protocols and Client/Server Applications 6 Hrs.

3.1 Standard protocols: SMTP, E-mail, Message (RFC22), PGP, POP, IMAP, HTTP, FTP

Email

'Messages distributed by electronic means from one computer user to one or more recipients via a network.'

Short for **electronic mail**, **e-mail** or **email** is text messages that may contain files, images, or other attachments sent through a network to a specified individual or group of individuals. Ray Tomlinson sent the first e-mail in 1971. By 1996, more electronic mail was being sent than postal mail.

- **The email message** - Instead of using a pen to write a letter on paper, you're using your keyboard to type an email message in an email program on your computer.
- **Sending the email** - When the email is finished and has been addressed to the recipient's email address, you don't put a stamp on it and post it but press the *Send* button in the email program. This makes the email message go on its journey.
- **Email transport** - Like postal services transport letters and parcel, email servers transmit email messages from sender to recipient. Usually, emails are not delivered to the recipient directly, though, but waiting at the "nearest" mail server to be picked up by them.
- **Fetching new mail** - If you've got new mail in your mailbox, you go and fetch it. Similarly, your email program can check for new email messages at your mail server and download them for you to read.

Simple Mail Transfer Protocol (**SMTP**) is an Internet standard for electronic mail (e-mail) transmission. First defined by RFC 821 in 1982, it was last updated in 2008 with the Extended SMTP additions by RFC 5321 - which is the protocol in widespread use today.

SMTP by default uses TCP port 25. The protocol for mail submission is the same, but uses port 587. SMTP connections secured by SSL, known as SMTPS, default to port 465.

While electronic mail servers and other mail transfer agents use SMTP to send and receive mail messages, user-level client mail applications typically use SMTP only for sending messages to a mail server for relaying. For receiving messages, client applications usually use either POP3 or IMAP.

While proprietary systems (such as Microsoft Exchange and Lotus Notes/Domino) and webmail systems (such as Hotmail, Gmail and Yahoo! Mail) use their own non-standard protocols to access mail box accounts on their own mail servers, all use SMTP when sending or receiving email from outside their own systems.

IMAP and POP are the two most prevailing methods or protocols for retrieving email from a mail server. Almost all popular mail client programs like Outlook, Thunderbird and Apple Mail support both of these protocols.

When your mail client reads an email it can either download the email from the mail server to your local desktop and delete it from the mail server, or just allow you to see the email contents, without saving it locally, similar to your viewing a webpage. In the first case, where you download an email to your local machine, POP is used. In the second case, where you view the email, it actually stays on the mail server.

POP3

POP3 stands for Post Office Protocol. POP3 allows an email client to download an email from an email server. The POP3 protocol is simple and does not offer many features except for download. Its design assumes that the email client downloads all available email from the server, deletes them from the server and then disconnects. POP3 normally uses port 110.

Benefits	Disadvantages
Mail is stored locally you can access it when offline	Need to use workaround to read email on multiple machines
Easier to maintain mail quotas	All mail can be lost if your hard drive or computer dies and you don't have a backup
Easy to backup and archive locally or offsite	Harder to access mail anywhere
Faster – only need to read mail from server once	
Faster when you have to search email	

IMAP

IMAP stands for Internet Message Access Protocol. IMAP shares many similar features with POP3. It, too, is a protocol that an email client can use to download email from an email server. However, IMAP includes many more features than POP3. The IMAP protocol is designed to let users keep their email on the server. IMAP requires more disk space on the server and more CPU resources than POP3, as all emails are stored on the server. IMAP normally uses port 143.

Benefits	Disadvantages
Mail is stored remotely, you can access anywhere	Takes up a lot of server space – may easily exceed your quota **
Keeps a copy of everything you do on the mail server. Privacy concerns may apply.	All mail can be lost if your hard drive or computer dies and you don't have a backup
Easy to backup and archive locally or offsite	Slower in reading email – index of all messages downloaded
Faster – only need to read mail from server once	Harder to backup locally and remotely.
Faster when you have to search email	All mail can be lost if mail account becomes corrupted, server crashes and backup is bad
	Slow when searching email – all emails are downloaded and read by mail client.

Suppose you use ***hMailServer*** as your email server to send an email to bill@microsoft.com.

1. You click *Send* in your email client, say, Outlook Express.
2. Outlook Express delivers the email to ***hMailServer*** using the SMTP protocol.
3. ***hMailServer*** delivers the email to Microsoft's mail server, mail.microsoft.com, using SMTP.
4. Bill's Mozilla Mail client downloads the email from mail.microsoft.com to his laptop using the POP3 protocol (or IMAP).

Main differences

POP – Downloads email locally

IMAP – Mail is stored on the mail server

Host-host control message formats (RFC22) (https://tools.ietf.org/html/rfc22)

All Host-Host control messages consist of sequences of 8-bit bytes of the form:

<control byte> <parameter byte 1> ... <parameter byte n>

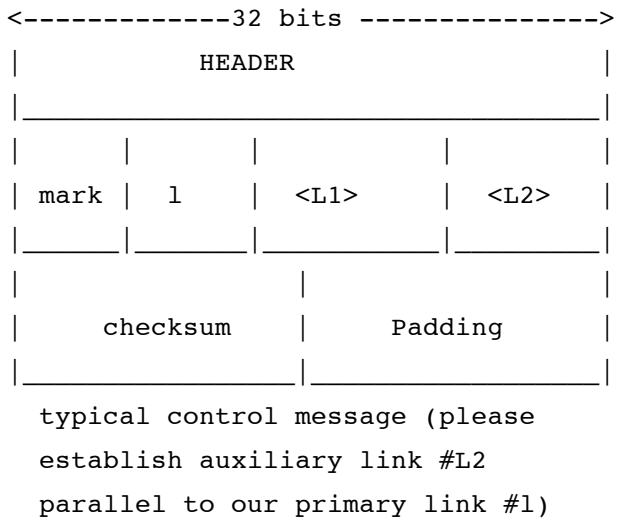
It is reasonable to transmit more than one control message in any given packet, although this is not mandatory.

Presently, 9 control messages have been defined by UCLA; these are given in the table below along with their parameters. The interpretation is given from the point of view of the transmitting host. ("L" or "Li" mean Link#, and are binary values.)

Control byte	Parameter	Interpretation
<0>	<L>	Please establish primary connection; our output link # is L
<1>	<L,> <L2>	Please establish auxiliary connection parallel to our primary output link L. The auxiliary output link is L2.
<2>	<L1> <L2>	DK primary. Your primary output link to us was L; our primary output link to you is L2.
<3>	<L1> <L2>	OK auxiliary. Your auxiliary output link is Li, our auxiliary output link is L2.
<4>	<L>	Not OK primary. We cannot establish a primary connection. Your primary output link number was L.
<5>	<Li> <L2>	Not OK auxiliary. We cannot establish an auxiliary connection. Your primary output link no was L2.
<6>	<L>	Please stop transmitting over link number L. This is called the CEASE directive.
<7>	<L>	We are CLOSING our output link number L. You may get this message before

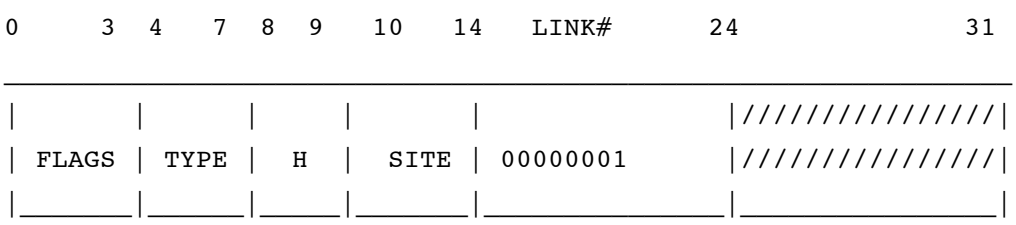
		the last message arrives over this link since control messages are higher priority than regular data messages.
<8>	<L>	UNCEASE: that is, you may resume transmitting over output link number L.

Each control message is embedded in the appropriate message structure e.g.:



typical control message (please establish auxiliary link #L2 parallel to our primary link #1)

The header for all HOST-HOST control messages is given below:

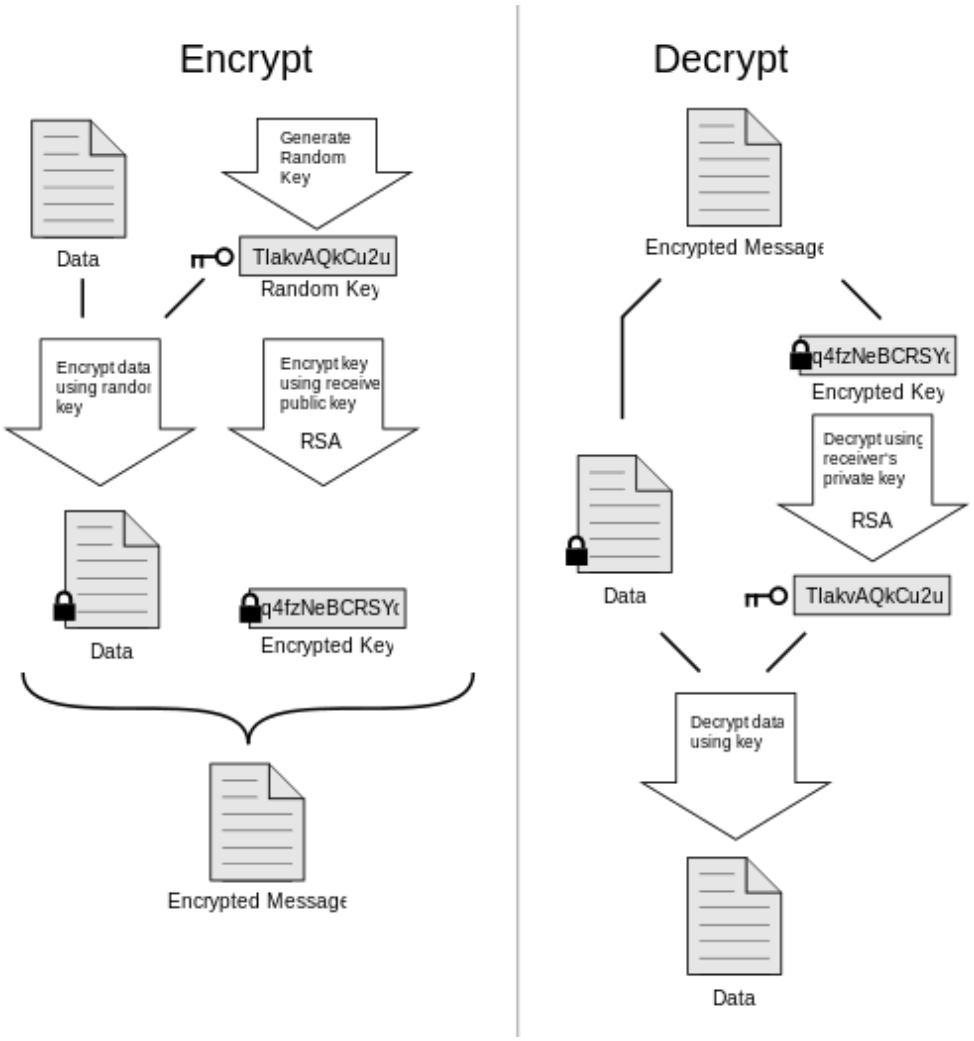


where FLAGS - 0000
TYPE - 0000 (regular message)
H - host #(0-3) at SITE (usually 0 for single HOST sites)
SITE - Site #
LINK# - 00000001 (HOST-HOST control link)

[This RFC was put into machine readable form for entry]
[into the online RFC archives by Alison De La Cruz 12/00]

PGP (Pretty Good Privacy) is the most widely recognized public key encryption program in the world. It can be used to protect the privacy of email, data files, drives and instant messaging.

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 and to increase the security of e-mail communications. It was created by Phil Zimmermann in 1991 while working at PKWARE, Inc.



Traffic on the Internet is vulnerable to interfering by third parties. Data packets can be captured and stored for years. Even mail servers will often indefinitely store messages, which can be read now or at a future point, sometimes long after the author has changed his or her point of view. Email, unlike a phone call or letter, is not legally protected as private communication, and can therefore be read by third parties, legal or otherwise, without permission or knowledge of the author. Many privacy watchdog groups advocate, *if you aren't using encryption, don't include anything in an email you wouldn't want to see published.* Ideally this includes personal information as well, such as name, address, phone number, passwords, and so on.

PGP encryption provides privacy missing from online communication. It changes plain, readable text into a complex code of characters that is completely unreadable. The email or instant message travels to the destination or recipient in this cyphered form. The recipient uses PGP to decrypt the message back into readable form. Whether you are concerned about protecting privacy rights, a corporate whistleblower, or a citizen that simply wants to chat with friends without allowing people to "listen in," PGP is the answer.

The simple but ingenious method behind public key encryption is based around the creation of a customized *key pair*. The key pair consists of a **public key** and a **private key**. The public key encrypts messages, while the private key decrypts them.

Using PGP, Mr. Pandey would generate a key pair by entering a real name or nickname to be associated with the keys and a password. The two keys are interlocking algorithms that appear as small bits of text code. Mr. Pandey can freely share the public key with anyone who wishes to send an encrypted message to him. For example, let's say Mr. Pandey gives his public key to Mr. Sharma. He can copy and paste it into an email and send it to him.

Mr. Sharma receives the public key and copies it to his *public key ring* in PGP. After he writes an email to Mr. Pandey, the email is encrypted using the associated public key, obtained from the key ring. The encrypted email is now sent. If someone captures the email en route, or even if it is stored on a server, it will be unreadable.

When Mr. Pandey receives the email, his *private key* decrypts the message. Thus the communication is kept private, even though it travels over public channels. The encryption and decryption can be done automatically, as PGP seamlessly interfaces with most major email clients.

To send an encrypted email to someone using PGP, you only need his or her public key. Each public key is unique and works with the associated private key as a key pair. If you encrypt a message with the public key of someone other than the recipient, the recipient will not be able to decrypt the message.

When creating a key pair in PGP, the option exists for your public key to be sent to a **public key server**. This makes it possible for strangers to send you encrypted mail by simply looking up your public key. To avoid spam, you may choose instead to email your public key discretely to handpicked

friends. Others attach their public key as part of their signature on public posts to newsgroups and Web chat boards.

A PGP user can also use his or her private key to *digitally sign* outgoing mail so that the recipient knows that the mail originated from the named sender. A third party would not have access to the private key, so the digital signature authenticates the sender.

Sensitive data files stored on your hard drive or on removable media can also be protected using PGP. You can use your public key to encrypt the files and your private key to decrypt them. Some PGP versions also allow the user to encrypt an entire disk. This is especially useful for laptop users in the event the laptop is lost or stolen.

HTTP

[Self Study, Refer: Web Technology]

FTP

[Self Study, Refer: Web Technology]

3.2 N-Tiered Client/Server Architecture

Layer Partitioning

Presentation layer:

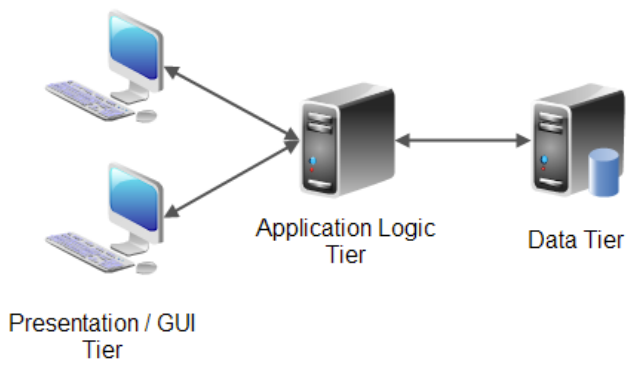
It contains the components dealing with user interface and user interactions.

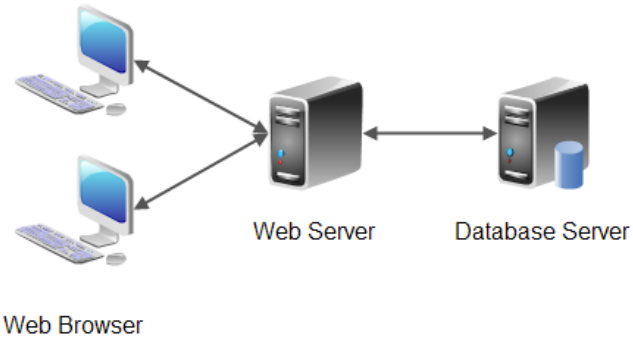
Business logic layer:

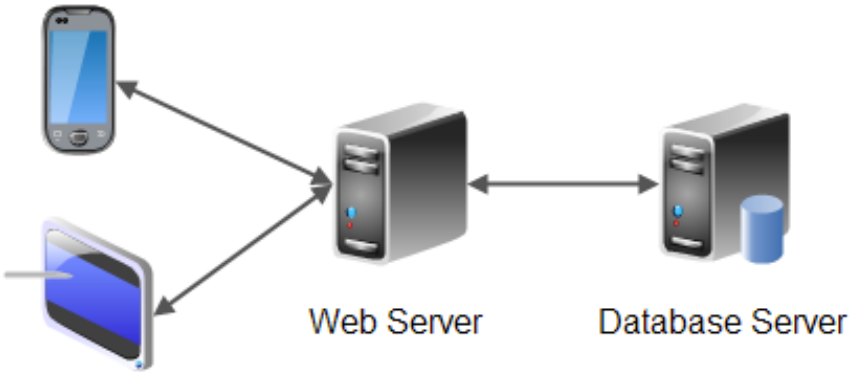
This layer contains components that includes high performance engine such as database driver, catalog engines and pricing engines.

Data layer:

The entire database required for the application resides in this layer.







Two-Tier Client/Server Architecture:

Here the client communicates directly with the database server. This system was initially used in LANs in late eighties. Then its application extended to all other systems using file sharing techniques and systems accessing databases.

Fat client model: The client side becomes more and more fat for increase in complexity of applications there by reducing the effective bandwidth of the network. This system combines the presentation layer and business logic layer where in the data access layer is a separate tier.

Disadvantages of two-tier architecture:

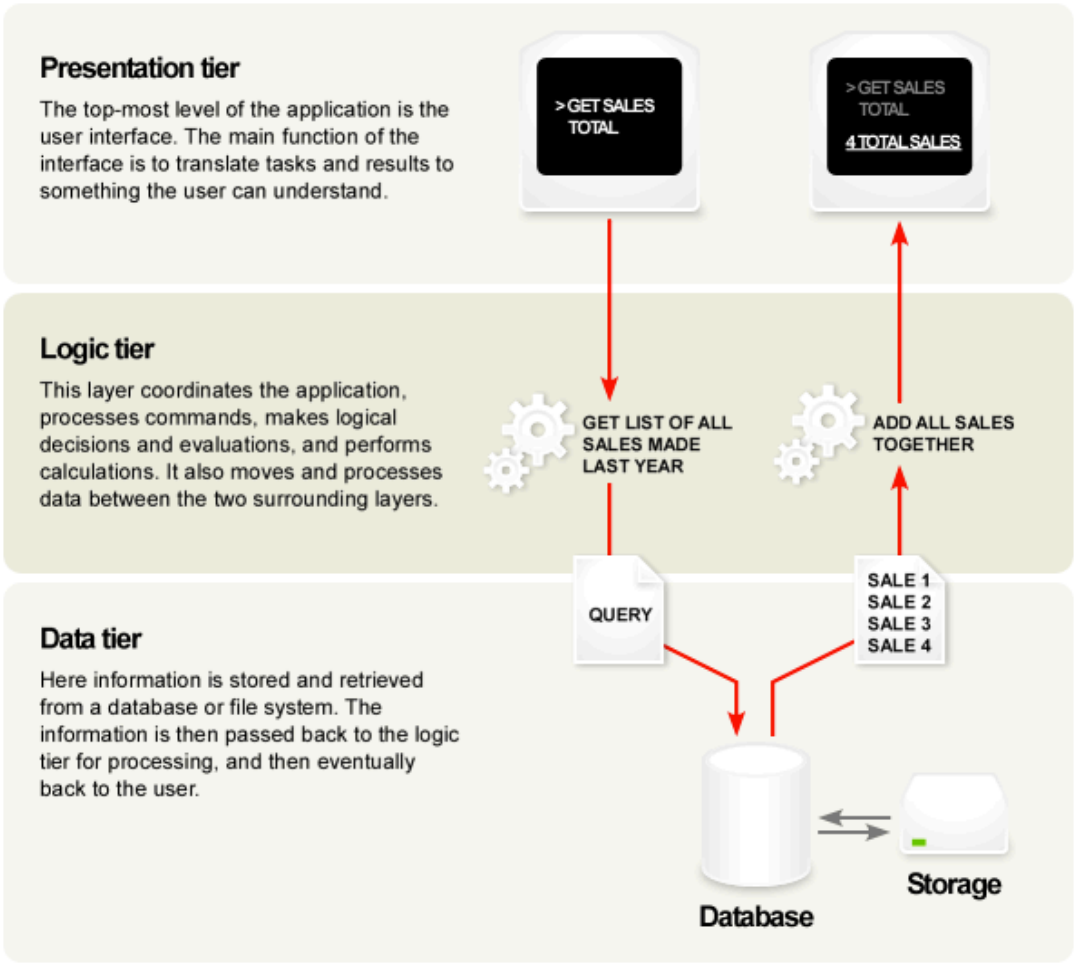
- Network performance suffers reducing the amount of bandwidth of other users.
- Changing business logic layer involves recompiling and redeploying the client tier.
- Fat clients are bound to database API such as relational databases. This involves not only redeploying each client but the client code should also change to suit the new database type.
- Every client needs to establish its own database connection. Therefore, database connection costs are high.
- Database drivers must be installed and configured on each of the client tiers, which include high deployment cost.

Fat Server Model: In this system, the stored procedure is placed with in the database. So whenever the business logic changes, the procedure must be modified. Thus the server side is fatter in this system.

Development of stored procedures, which enhances portability, is an improvement in this system though does not solve most of the problems of Fat Client Model.

Three-Tier Client/Server Architecture:

In this system, the client implements presentation logic (thin client). The application server implements the business logic and the data resides on database server.



Multi-Tier Architecture:

The front-end component is responsible for providing portable presentation logic. The back-end component acts as database server. The middle tier component allows the users to share and control business logic by isolating it from actual application. This system is fat in the middle. The client system interacts with the middle tier through a standard protocol such as HTTP or RPC. The middle tier interacts with the backend server through standard database protocols such as SQL, ODBC and JDBC.

N-Tier Architecture:

N-tier application architecture provides a model for developers to create a flexible and reusable application. By breaking up an application into tiers, developers only have to modify or add a specific layer, rather than have to rewrite the entire application over, if they decide to change technologies or scale up.

With four or more tiers, each layer can be further decomposed to allow various parts of the system to scale independently. More sophisticated multi-tier solutions appear in this model. This architecture leads to reduction of network traffic, faster network communications, greater reliability and greater overall performance.

Benefits of N-Tier Architecture:

- Database drivers are installed and configured on the server-side, rather than on client machines. Hence deployment costs are low.
- Data base switching costs are low: There is a middle tier for data access. This enables the users to migrate database schemas, or change the different database drivers without redeploying the clients.
- Changing the business logic layer may not necessitate the redeploying the client tier.
- By placing a firewall between the presentation and business logic tiers, high security can be provided to the data easily.
- Rather than, the business components acquiring and releasing connections to the resources such as databases, the resources can be pooled and reused for different client requests. Resource pooling can also be applied to other resources such as threads and socket connections. Multiple clients can pool business components themselves.
- Since there are many tiers and each tier is independent, the database images can be added while minimizing the changes and recompiling other tiers.
- If one tier is overloaded, other tier can still function properly there by improving the performance.
- If critical error occurs, it is localized to a single tier.

Disadvantages of N-Tier Architecture:

- Since the tiers are physically separate, they must communicate across the process boundaries, machine boundaries or enterprise domain boundaries. This results in high communications overhead.
- Software installation costs, software upgrade costs and other administration costs are high.

Application of N-Tier Architecture:

It plays a major role in Internet and intranet services, transaction processing monitors, distributed computing and most other growing software technologies. N-Tier provides a wide range of benefits to the companies longing for flexible and reliable solution to complex, and constantly changing problems. It also provides information and tools to solve some of the challenges faced by IT professionals.

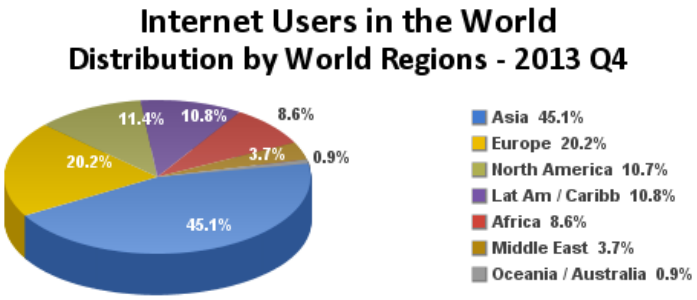
3.3 Universal Internet Browsing

Internet Browsing

'In digital communications media, the vast majority of participants are active creators of information as well as recipients. This type of symmetry has previously only been found in media like the telephone. But while the telephone is almost entirely a medium for private one-to-one communication, computer network applications such as electronic mailing lists, conferences, and bulletin boards, serve as a medium of group or 'many-to-many' communication.

The new forums atop computer networks are the great levelers and reducers of organizational hierarchy. Each user has, at least in theory, access to every other user, and an equal chance to be heard.' - Mitchell Kapor, Electronic Frontier Foundation Information, 1993.

The Internet provides *universal access*, giving the same powerful capabilities to everyone who has access to the network no matter where they are.



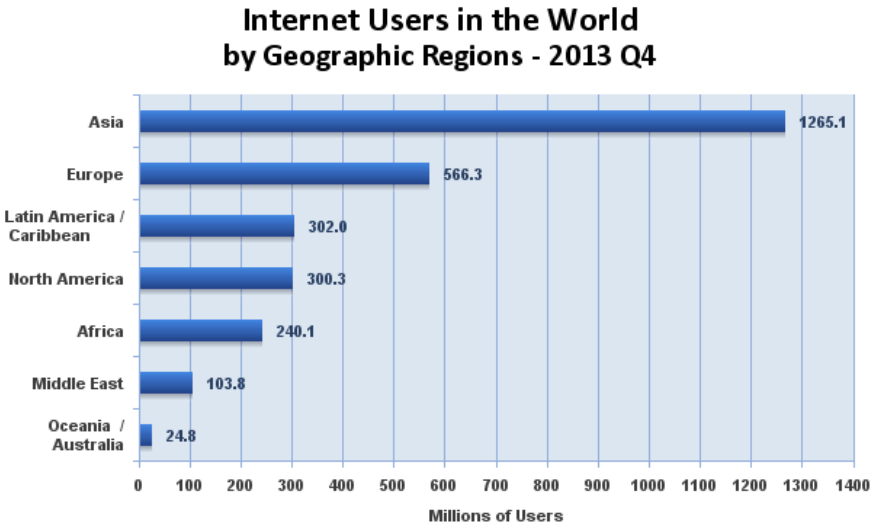
The Internet is based on a common standard, the *TCP/IP* network protocol, which provides all computers with access to the network with the same technical interface and capabilities. This common foundation makes all of the Internet technologies equally available to anyone connected to the Internet.

"Exploration of the World Wide Web by following one interesting link to another, usually with a definite objective but without a planned search strategy. In comparison 'surfing' is exploration without a definite objective or search strategy, and 'searching' is exploration definite in both objective and strategy."

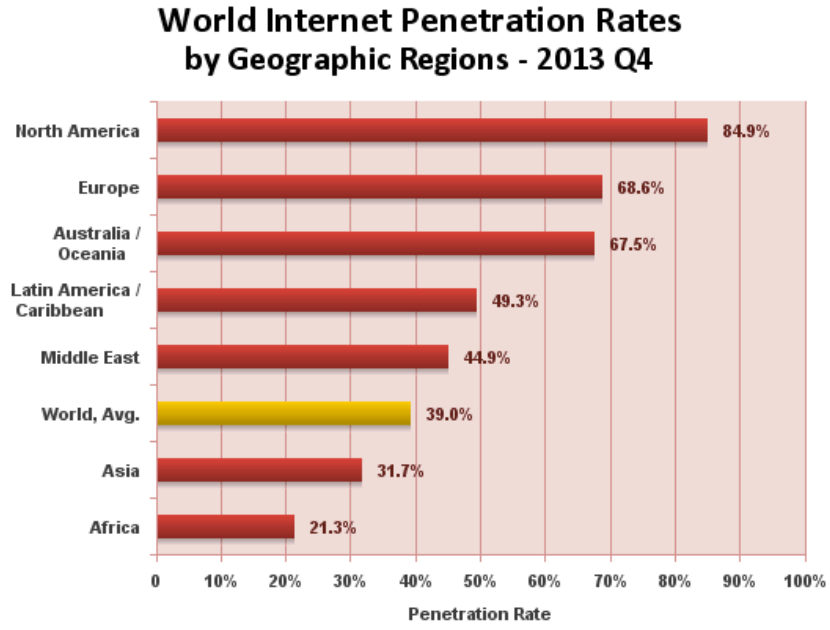
This architecture gives everyone the ability to make information like text, audio, and video accessible to a world wide audience at an extremely low cost, since website storage space and lots of bandwidth can be rented from *web hosting* providers for low fees. Because the Internet has a "many-to-many" architecture, with everyone having the same capabilities as anyone else, it allows anyone to become a global publisher.

"Two decades after its inception, the Internet is celebrated as the most powerful force of human development in recorded history. And for many of us, this ascendence of the web in our consciousness has made the technology indispensable."

The Internet is the current communication frontier. You should feel free to approach the Internet with a spirit of exploration, and don't need to have a task or a question to answer -- you can *surf* from link to link or try random searches just to see what turns up, like exploring a new city.



Efforts have been made by different companies and organizations to provide tailored solutions to individuals who have different access needs. Dedicated browsers, toolbars and add-ons have been created so that the content is accessible. Sites should be able to be navigated b y a pointing device and more critically for some users by keyboard equivalents. Others require text only for screen reading with other users needing just text to speech options, magnification systems, screen tinting or removal of video, graphics or ads. Sensible use of search engines can also satisfy to different user's needs. Google is the most commonly used search engine, with Yahoo, Bing, Ask, MSN and may others competing for user loyalty.



'Since a web page can be interpreted differently by different browsers with different capabilities, and since the language of a web page- HTML, is constantly evolving, accessibility must be considered to make a page usable by as many people as possible. The keys to making your page accessible are graceful degradation, standards compliance, fast loading, and intelligent organization.'

If you feel moved to set up a website about your favorite hobby, go ahead. The Internet is universally empowering - everyone can participate.

3.4 Multiprotocol Support

Networking technology evolved along different paths, resulting in several different systems to format data for transmission. Suites of programs were developed to comply with one system or another. Programs with multiple protocol support can interact with many different underlying protocols.

Features

- An example of multiple protocol application is a messenger user interface that can represent many different underlying instant messenger services. The program is written to be independent of the transport method or messenger format. Examples of these systems are *Kopete, Miranda IM and Adium*.

Function

- Programs with multiple protocol support are written in two ways. One is to keep the program generic so that it does not perform any processing, but merely displays the results of other systems. The other category of programs, or protocols, with multiple protocol support is one written to be extended by additional programs. Web browsers fall into this category.

Protocols

- There are protocols with multiple protocol support. The Point-to-Point Protocol (PPP) has to interact with many different networking systems. It does this through a library of network control protocols, each one fitting a different network standard.

CHAPTER 5

5. Designing Internet Systems and Servers 8 Hrs.

5.1 Designing of Internet System Network Architecture

Network architecture refers to the layout of the network, consisting of the hardware, software, connectivity, communication protocols and mode of transmission, such as wired or wireless. Know about the **types of network** classified according to the areas covered such as LAN, MAN and WAN. Learn about the **network topologies** categorized according to the layout of equipment and computers such as star, loop, bus, or mesh topologies. There are many communication protocols used in the networking technology. It is important to know about the **network architecture** as networks play a very important role in today's world.

Network architecture is the design of a communications network. It is a framework for the specification of a network's physical components and their functional organization and configuration, its operational principles and procedures, as well as data formats used in its operation.

Fortunately, nobody owns the Internet, there is no centralized control, and nobody can turn it off. Its evolution depends on rough consensus about technical proposals, and on running code. Engineering feedback from real implementations is more important than any architectural principles.

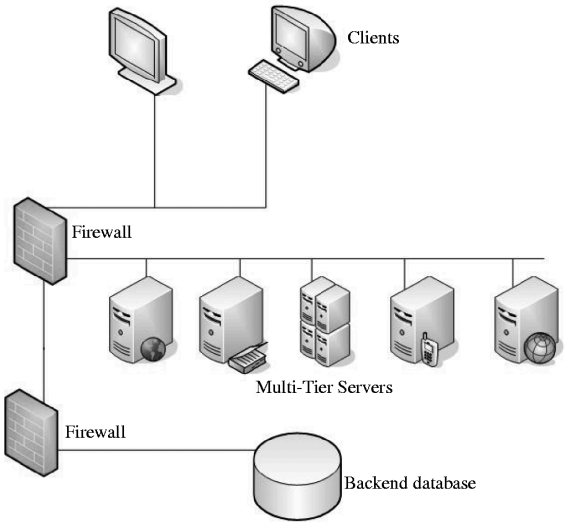
In telecommunication, the specification of a network architecture may also include a detailed description of products and services delivered via a communications network, as well as detailed rate and billing structures under which services are compensated.

The network architecture of the Internet is predominantly expressed by its use of the Internet Protocol Suite, rather than a specific model for interconnecting networks or nodes in the network, or the usage of specific types of hardware links.

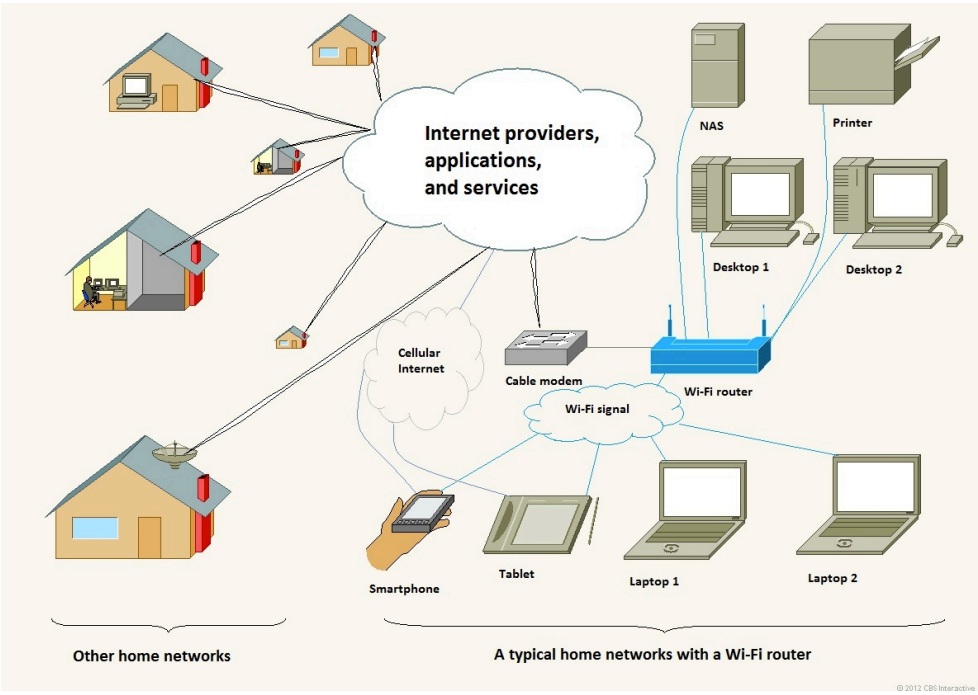
Internet's architecture

The Internet's architecture is described in its name, a short from of the compound word "inter-networking". This architecture is based in the very specification of the standard *TCP/IP* protocol, designed to connect any two networks, which may be very different in internal hardware, software, and technical design. Once two networks are interconnected, communication with TCP/IP is enabled end-to-end, so that any node on the Internet has the near magical ability to communicate with

any other no matter where they are. This openness of design has enabled the Internet architecture to grow to a global scale.



In practice, the Internet technical architecture looks a bit like a multi-dimensional river system, with small branches feeding medium-sized streams feeding large rivers. For example, an individual's access to the Internet is often from home over a modem to a local Internet service provider who connects to a regional network connected to a national network. At the office, a desktop computer might be connected to a local area network with a company connection to a corporate Intranet connected to several national Internet service providers. In general, small local Internet service providers connect to medium-sized regional networks, which connect to large national networks, which then connect to very large bandwidth networks on the Internet *backbone*. Most Internet service providers have several redundant network cross-connections to other providers in order to ensure continuous availability.



The companies running the Internet backbone operate very high bandwidth networks relied on by governments, corporations, large organizations, and other Internet service providers. Their technical infrastructure often includes global connections through underwater cables and satellite links to enable communication between countries and continents. As always, a larger scale introduces new phenomena: the number of packets flowing through the switches on the backbone is so large that it exhibits the kind of complex non-linear patterns usually found in natural, analog systems like the flow of water or development of the rings of Saturn.

Each communication *packet* goes up the hierarchy of Internet networks as far as necessary to get to its destination network where local *routing* takes over to deliver it to the addressee. In the same way, each level in the hierarchy pays the next level for the bandwidth they use, and then the large backbone companies settle up with each other. Bandwidth is priced by large Internet service providers by several methods, such as at a fixed rate for constant availability of a certain number of megabits per second, or by a variety of use methods that amount to a cost per gigabyte.

5.2 Choice of platforms

Popular server operating systems include Windows Server, Mac OS X Server, and variants of Linux such as Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server.

5.3 Server Concepts: WEB, Proxy, RADIUS, and MAIL

In a general network environment the following types of servers may be found

- Application server, a server dedicated to running certain software applications
- Catalog server, a central search point for information across a distributed network
- Communications server, carrier-grade computing platform for communications networks
- Compute server, a server intended for intensive (esp. scientific) computations
- Database server, provides database services to other computer programs or computers
- Fax server, provides fax services for clients
- File server, provides remote access to files
- Game server, a server that video game clients connect to in order to play online together
- Home server, a server for the home
- Mail server, handles transport of and access to email
- Mobile Server, or Server on the Go is an Intel Xeon processor based server class laptop form factor computer.
- Name server or DNS

- Print server, provides printer services
- Proxy server, acts as an intermediary for requests from clients seeking resources from other servers
- Sound server, provides multimedia broadcasting, streaming.
- Stand-alone server, a server on a Windows network that neither belongs to nor governs a Windows domain
- Web server, a server that HTTP clients connect to in order to send commands and receive responses along with data contents

Almost the entire structure of the Internet is based upon a client–server model. High-level root name servers, DNS, and routers direct the traffic on the Internet. There are millions of servers connected to the Internet, running continuously throughout the world.

- World Wide Web
- Domain Name System
- E-mail
- FTP file transfer
- Chat and instant messaging
- Voice communication
- Streaming audio and video
- Online gaming
- Database servers

Proxy Server

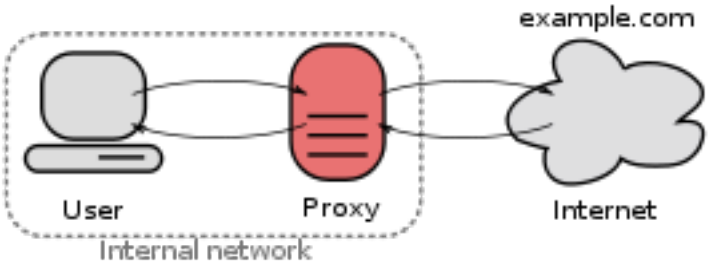
In computer networks, a **proxy server** is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource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

Types of proxy

A proxy server may reside on the user's local computer, or at various points between the user's computer and destination servers on the Internet.

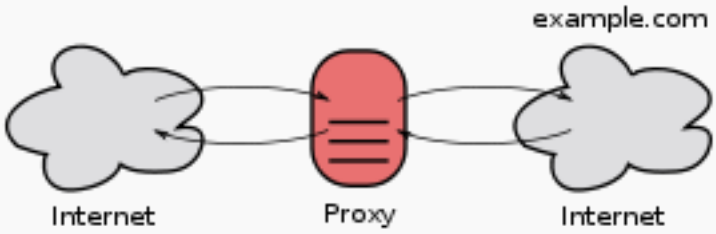
- A proxy server that passes requests and responses unmodified is usually called a gateway or sometimes a *tunneling proxy*.
- A forward proxy is an Internet-facing proxy used to retrieve from a wide range of sources (in most cases anywhere on the Internet).
- A reverse proxy is usually an Internet-facing proxy used as a front-end to control and protect access to a server on a private network. A reverse proxy commonly also performs tasks such as load balancing, authentication, decryption or caching.

Forwarding proxies



A forward proxy taking requests from an internal network and forwarding them to the Internet. Forward proxies are proxies in which 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 servers.

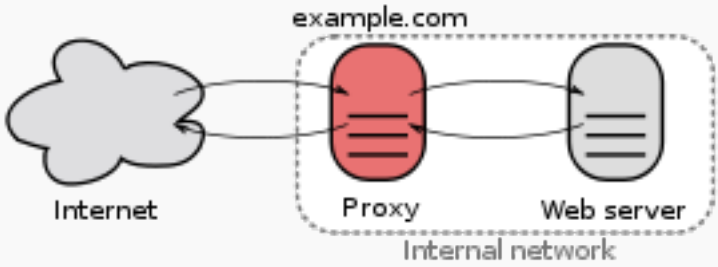
Open proxies



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

An open proxy is a forwarding proxy server that is accessible by any Internet user. Gordon Lyon estimates there are "hundreds of thousands" of open proxies on the Internet. 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 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** is a proxy server that appears to clients to be an ordinary server. Requests are forwarded to one or more proxy servers, which handle the request. The response from the proxy server is returned as if it came directly from the origin server, leaving the client no knowledge of the origin servers. 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. See Secure Sockets Layer. 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).

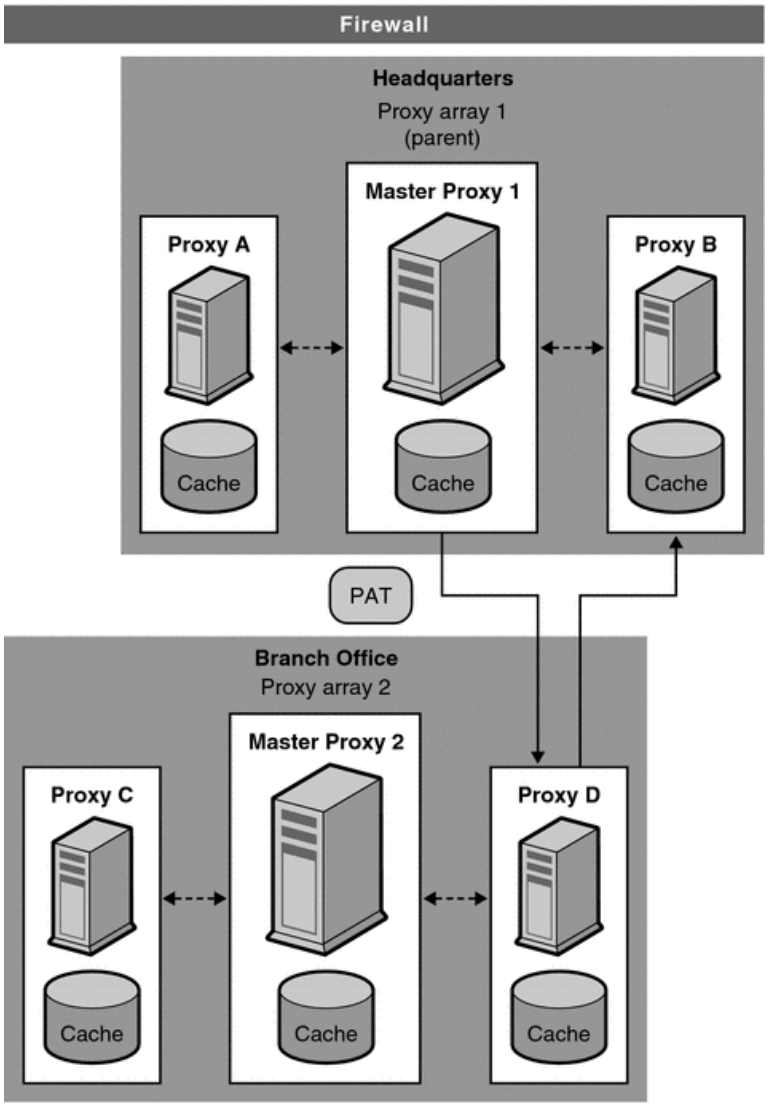
- **Serve/cache static content**: A reverse proxy can offload the web servers by caching static content like pictures and other static graphical content.

- **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 Web Server specific attacks. However, it does not provide any protection from 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 firewall 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.

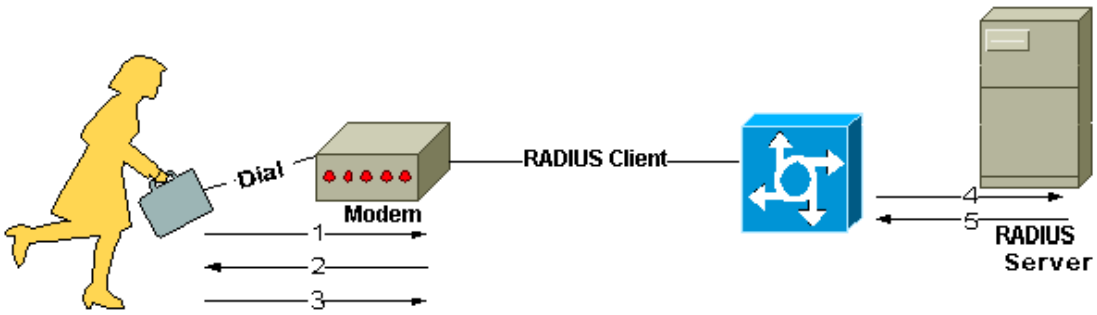


Examples of web proxy servers include Apache (with mod_proxy or Traffic Server), IIS configured as proxy (e.g., with Application Request Routing), Nginx, Privoxy, Squid, Varnish (*reverse proxy only*), WinGate, Ziproxy, Tinyproxy etc.

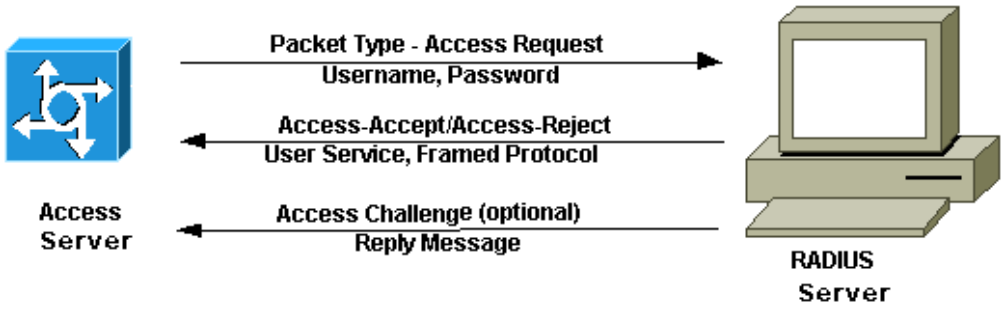
Uses of proxy servers

- Monitoring and filtering
- *Content-control software*
- *Bypassing filters and censorship*
- *Logging and eavesdropping*
- Improving performance
- Translation
- Accessing services anonymously

***R**emote **A**uthentication **D**ial-**I**n **U**ser **S**ervice **(RADIUS)**,* an authentication and accounting system used by many Internet Service Providers (ISPs). When you dial in to the ISP you must enter your username and password. This information is passed to a RADIUS server, which checks that the information is correct, and then authorizes access to the ISP system.



1. User initiates PPP authentication to the NAS.
2. NAS prompts for username and password (if Password Authentication Protocol [PAP]) or challenge (if Challenge Handshake Authentication Protocol [CHAP]).
3. User replies.
4. RADIUS client sends username and encrypted password to the RADIUS server.
5. RADIUS server responds with Accept, Reject, or Challenge.
6. The RADIUS client acts upon services and services parameters bundled with Accept or Reject.



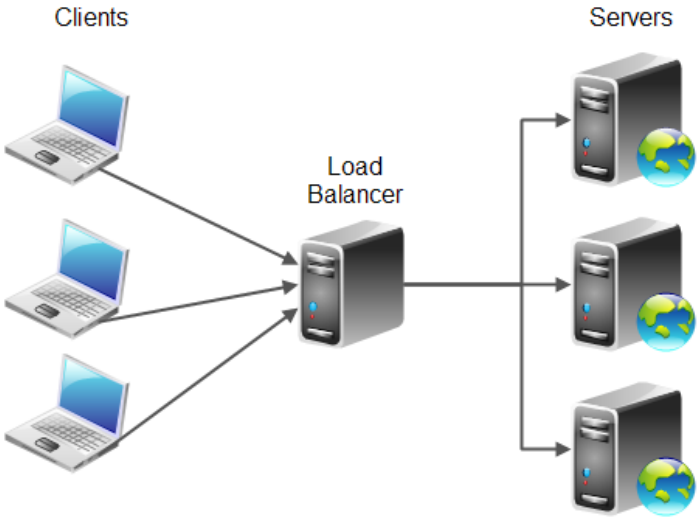
Radius is a server for remote user authentication and accounting. Its primary use is for Internet Service Providers, though it may as well be used on any network that needs a centralized authentication and/or accounting service for its workstations.

5.4 Cookies

[Self Study; Refer: Web Technology]

5.5 Loading Balancing, Proxy Arrays

Load balancing is a method for distributing tasks onto multiple computers. For instance, distributing incoming HTTP requests (tasks) for a web application onto multiple web servers. There are a few different ways to implement load balancing. I will explain some common load balancing schemes in this text. Here is a diagram illustrating the basic principle of load balancing:



The primary purpose of load balancing is to distribute the workload of an application onto multiple computers, so the application can process a higher workload. Load balancing is a way to scale an application.

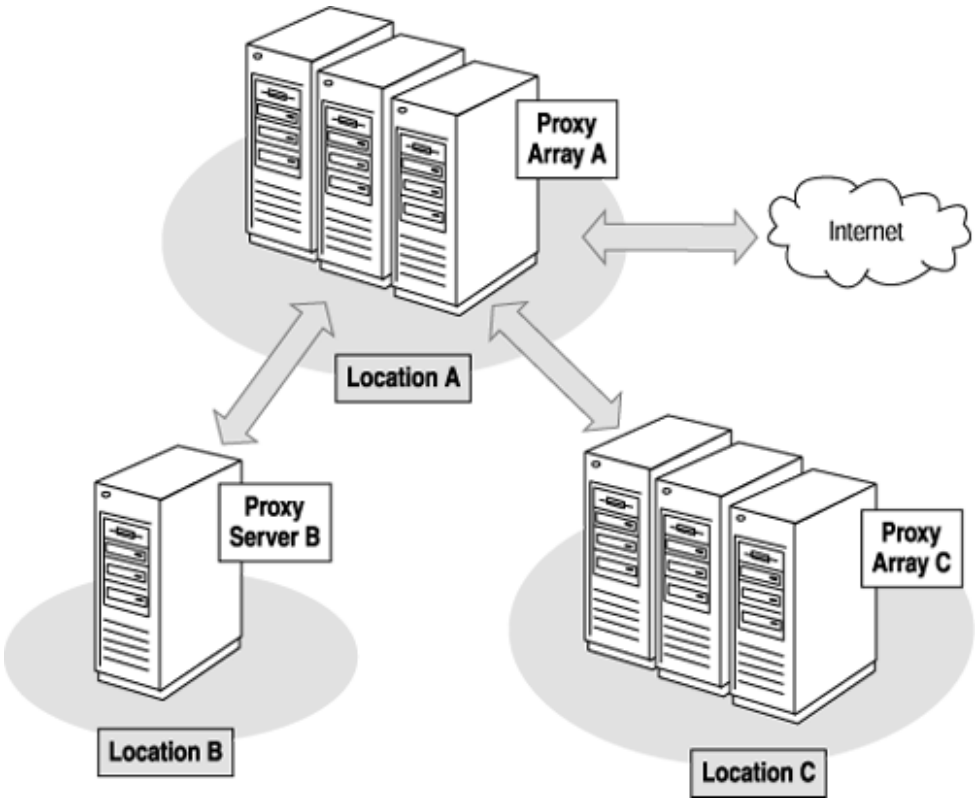
A secondary goal of load balancing is often (but not always) to provide redundancy in your application. That is, if one server in a cluster of servers fails, the load balancer can temporarily remove that server from the cluster, and divide the load onto the functioning servers. Having multiple servers

help each other in this way is typically called "redundancy". When an error happens and the tasks is moved from the failing server to a functioning server, this is typically called "failover".

A set of servers running the same application in cooperation is typically referred to as a "cluster" of servers. The purpose of a cluster is typically both of the above two mentioned goals: To distribute load onto different servers, and to provide redundancy / failover for each other.

Proxy arrays for distributed caching enable multiple proxies to serve as a single cache. Each proxy in the array will contain different cached URLs that can be retrieved by a browser or downstream proxy server. Proxy arrays prevent the duplication of caches that often occurs with multiple proxy servers. Through hash-based routing, proxy arrays route requests to the correct cache in the proxy array.

Proxy arrays also enable incremental scalability. If you decide to add another proxy to your proxy array, each member's cache is not invalidated. Only $1/n$ of the URLs in each member's cache, where n is the number of proxies in your array, will be reassigned to other members.



5.6 Server Setup and Configuration Guidelines

<< … … … … … … … … … >>

5.7 Security and System Administration Issues, Firewalls and Content Filtering

A **system administrator**, or **sysadmin**, is a person who is responsible for the upkeep, configuration, and reliable operation of computer systems; especially multi-user computers, such as servers. The system administrator seeks to ensure that the uptime, performance, resources, and security of the computers he or she manages meet the needs of the users, without exceeding the budget. To meet these needs, a system administrator may acquire, install, or upgrade computer components and software; provide routine automation; maintain security policies; troubleshoot; train and/or supervise staff; or technical support in projects.

Many organizations staff other jobs related to system administration. In a larger company, these may all be separate positions within a computer support or Information Services (IS) department. In a smaller group they may be shared by a few sysadmins, or even a single person.

- **A database administrator** (DBA) maintains a database system, and is responsible for the integrity of the data and the efficiency and performance of the system.
- **A network administrator** maintains network infrastructure such as switches and routers, and diagnoses problems with these or with the behavior of network-attached computers.
- **A security administrator** is a specialist in computer and network security, including the administration of security devices such as firewalls, as well as consulting on general security measures.
- **A web administrator** maintains web server services (such as Apache or IIS) that allow for internal or external access to web sites. Tasks include managing multiple sites, administering security, and configuring necessary components and software. Responsibilities may also include software change management.
- **A computer operator** performs routine maintenance and upkeep, such as changing backup tapes or replacing failed drives in a RAID. Such tasks usually require physical presence in the room with the computer; and while less skilled than sysadmin tasks require a similar level of trust, since the operator has access to possibly sensitive data.
- **A postmaster** administers a mail server.
- **A Storage (SAN) Administrator**. Create, Provision, Add or Remove Storage to/from Computer systems. Storage can be attached local to the system or from a Storage Area Network (SAN) or Network Attached Storage (NAS). Create File Systems from newly added storage.

Security & Firewall

The first line of security defense is to control access to your system. You can control and monitor system access by doing the following:

- Maintaining physical site security
- Maintaining login control
- Restricting access to data in files
- Maintaining network control
- Monitoring system usage
- Securing files
- Installing a firewall
- Reporting security problems

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.

Basically, a firewall is a barrier to keep destructive forces away from your property. In fact, that's why it's called a firewall. Its job is similar to a physical firewall that keeps a fire from spreading from one area to the next. As you read through this article, you will learn more about firewalls, how they work and what kinds of threats they can protect you from.

'A firewall is a 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. 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 firewall 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.

Common Firewall 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 susceptible to IP spoofing.

Application Gateway

Applies security mechanisms to specific applications, such as FTP and Telnet servers. This is very effective, but can impose 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 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. A firewall is considered a first line of defense in protecting private information. For greater security, data can be encrypted.