UNIT 5 Application layer

The layer where all the applications are found is called Application layer.

Application-Layer Paradigms

Traditional Paradigm: Client-Server

The traditional paradigm is called the **client-server paradigm**.

In this paradigm, the service provider is an application program, called the server process; it runs continuously, waiting for another application program, called the client process, to make a connection through the Internet and ask for service.

Normally few server processes are available that can provide a specific type of service, but there are many clients that request service from any of these server processes.

The server process must be running all the time; the client process is started when the client needs to receive service.

For example, a telephone directory center in any area can be a server; a subscriber that calls and asks for a specific telephone number can be thought of as a client.

The directory center must be ready and available all the time; the subscriber can call the center for a short period when the service is needed.

Figure (below) shows an example of a **client-server communication**in which three clients communicate with one server to receive the services provided by this server.

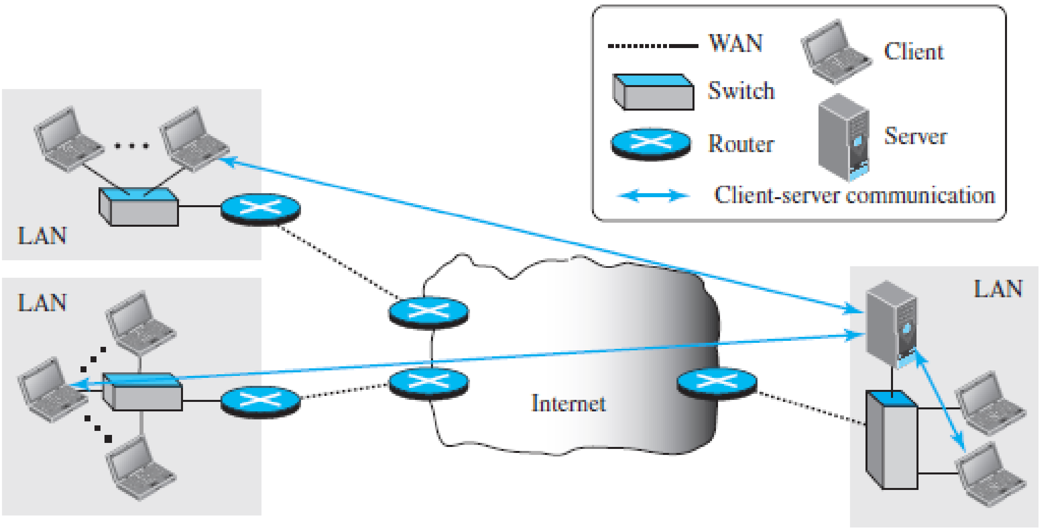


Fig:client server paradigm.. [Source : Data Communications and Networking by Behrouz A. Forouzan]

Peer-to-Peerparadigm

In this paradigm, thereis no need for a server process to be running all the time and waiting for the clientprocesses to connect.

The responsibility is shared between peers. A computer connected to the Internet can provide service at one time and receive service at another time. A computer can even provide and receive services at the same time.

Figure (below) shows an example of communication in this paradigm.

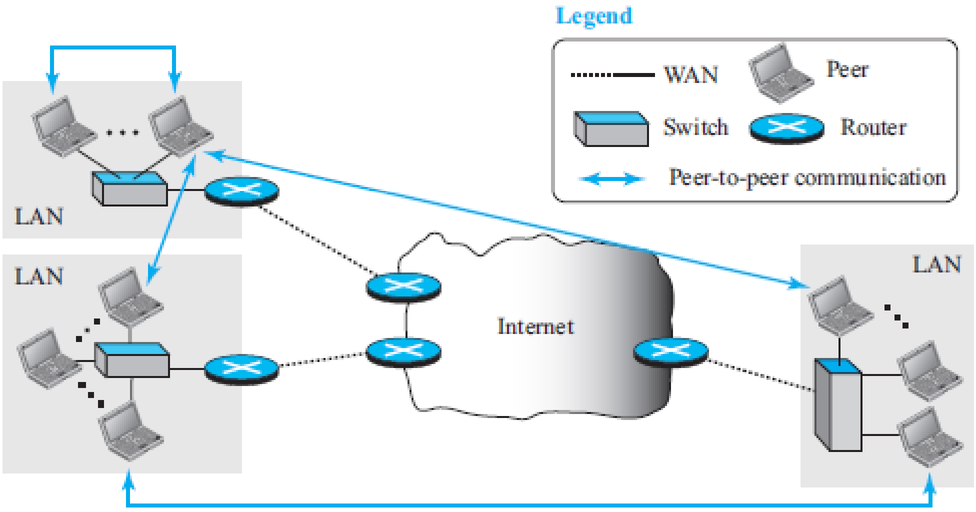


Fig: peer to peer paradigm.. [Source : Data Communications and Networking by Behrouz A. Forouzan]

Communicationby phone is a peer-to-peer activity; no party needs to wait for the other party to call.

The peer-to-peer paradigm can be used in a situation, when some computers connected to the Internet have something to sharewith each other.

For example, if an Internet user has a file available to share with otherInternet users, there is no need for the file holder to become a server and run a server processall the time waiting for other users to connect and to get the file.

CLIENT-SERVER PROGRAMMING

In a client-server paradigm, communication at the application layer is between two running application programs called **processes**: a client and a server.

A client is a running program that initializes the communication by sending a request; a server is another application program that waits for a request from a client.

The server handles the request received from a client, prepares a result, and sends the result back to the client.

The lifetime of a server is infinite: it should be started and run forever, waiting for the clients. The lifetime of a client is finite,it sends a finite number of requests to the corresponding server, receives the responses, and stops.

Application Programming Interface

If we need a process to be able to communicate with another process, we need a new set of instructions to tell the lowest four layers of the TCP/IP suite to open the connection, send and receive data from the other end, and close the connection.

A set of instructions of this typeis called as an **application programming interface (API)**.

An interface in programming is a set of instructions between two entities.

In this case, one of the entities is the process at the application layer and the other is theoperating system that encapsulates the first four layers of the TCP/IP protocol suite.

A computer manufacturer build the first four layers of the suite in the operating system and include an API.

Here, the processes running at the application layer are able to communicate with the operating system when sending and receiving messages through the Internet.

Several APIs have been designed for communication. Three among them are common: **socket interface, Transport Layer Interface** (**TLI**)**,** and
STREAM.

Socket Interface

Socket interface started in the early 1980s at UC Berkeley as part of a UNIX environment.

The socket interface is a set of instructions that provide communication between
the application layer and the operating system,

It is a set ofinstructions that can be used by a process to communicate with another process.

For example, in most computer languages, like C, C++, or Java, we have several instructions that can read and write data to other sources and sinks such as a keyboard (a source), a monitor (a sink), or a file (source and sink).

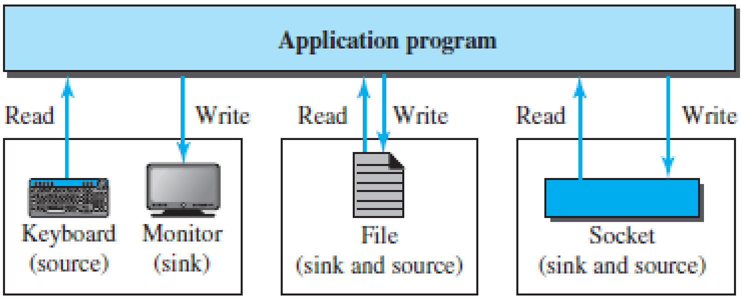


Fig: Socket format. [Source : Data Communications and Networking by Behrouz A. Forouzan]

In application layer, communication between a client process and a server process is the communication between two sockets.

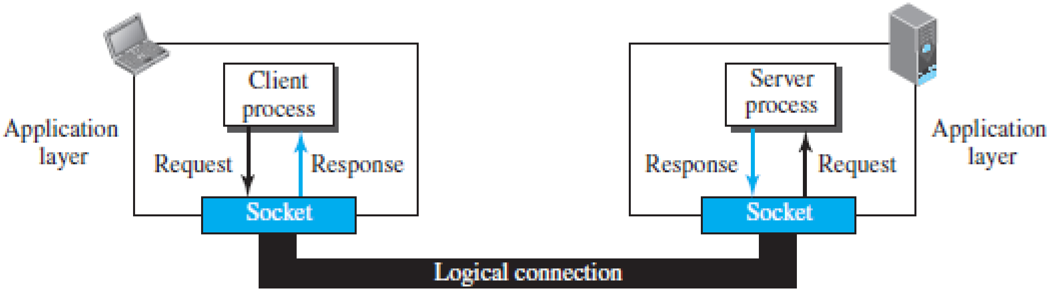


Fig: Socket in process to process communication. [Source : Data Communications and Networking by Behrouz A. Forouzan]

Socket Addresses

The interaction between a client and a server is two-way communication.

In a two-way communication, we need a pair of addresses: local (sender) and remote (receiver). The local address in one direction is the remote address in the other direction and vice versa.

Since communication in the client-server paradigm is between two sockets, we need a pair of **socket addresses** for communication: a local socket address and a remote socket address.

A socket address should first define the computer on which a client or a server is running. A computer in the Internet is defined by its IP address.

Finding Socket Addresses

How can a client or a server find a pair of socket addresses for communication?

The situation is different for each site.

Server Site
The server needs a local (server) and a remote (client) socket address for communication.

Local Socket Address The local (server) socket address is provided by the operating system. The operating system knows the IP address of the computer on which the server process is running. The port number of a server process, needs to be assigned.

If the server process is a standard one defined by the Internet authority, a port number is already assigned to it.
For example, the assigned port number for a Hypertext Transfer Protocol (HTTP) is the integer 80, which cannot be used by any other process.

Remote Socket Address

The remote socket address for a server is the socket address of the client that makes the connection. Since the server can serve many clients, it does
not know previously, the remote socket address for communication.

The server can find this socket address when a client tries to connect to the server. The client socket address, which is contained in the request packet sent to the server, becomes the remote socket address that is used for responding to the client.

World Wide Web

The Web was first proposed by Tim Berners-Lee in 1989 at *CERN*†, theEuropean Organization for Nuclear Research, to allow several researchers at differentlocations throughout Europe to access each others' researches.

The commercial Webstarted in 1990s.

The **webpages,**are distributed all over the world and related documents are linked together.

Today, the Web is used to provide electronic shopping and gaming.

Architecture of WWW

The WWW is a distributed client-server service, in which a client using a browser can access a service using a server.

The service provided is distributed over many locations called sites.

A web page can be simple or composite. A simple web page has no links to other web pages; a composite web page has one or more links to other web pages. Each web page is a file with a name and address.

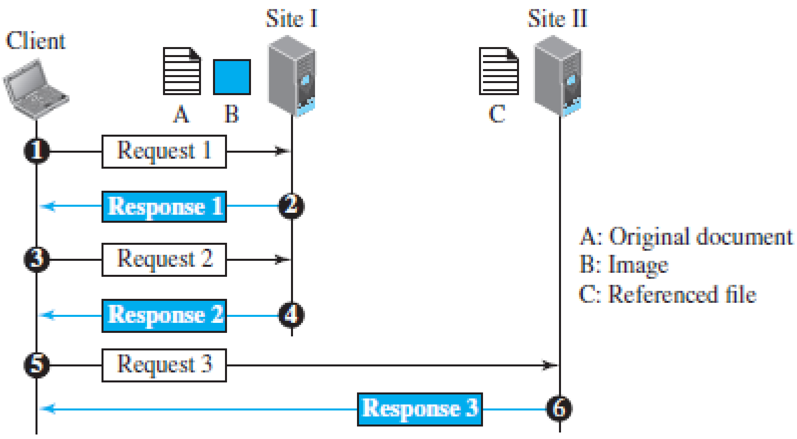


Fig: The web architecture.. [Source : Data Communications and Networking by Behrouz A. Forouzan]

The first transaction (request/response) retrieves a copy of the main document (file A), which has references (pointers) to the second and third files.

When a copy of the main document is retrieved and browsed, the user can click on the reference to the image to invoke the second transaction and retrieve a copy of the image (file B).

If the user needs to see the contents of the referenced text file, she can click on its reference (pointer) invoking the third transaction and retrieving a copy of file C.

Web Client (Browser)

A variety of vendors offer commercial **browsers** that interpret and display a web page, and all of them use the same architecture.

Each **browser has three parts**: a controller, client protocols, and interpreters.

The controller receives input from the keyboard or the mouse and uses the client programs to access the document.

After the document has been accessed, the controlleruses one of the interpreters to display the document on the screen.

The client protocolcan be one of the protocols described later, such as HTTP or FTP.

The interpreter canbe HTML, Java, or JavaScript, depending on the type of document.

Some commercialbrowsers include Internet Explorer, Netscape Navigator, and Firefox.

Web Server

The web page is stored at the server. Each time a request arrives, the correspondingdocument is sent to the client. To improve efficiency, servers normally store requestedfiles in a cache in memory; memory is faster to access than a disk.

Uniform Resource Locator (URL)

A web page, as a file, needs to have a unique identifier to distinguish it from other web pages.

URL is a standard for specifying any kind of information on the internet.

URL defines four things: protocol, host computer, port and path.

Protocol. It is the client-server program thatwe need to access the web page.

Example protocol is HTTP (HyperText Transfer Protocol) and FTP (File Transfer Protocol).

Host. The host identifier can be the IP address of the server or the unique name given to the server. IP addresses can be defined in dotted decimal notation.

Port. The port, a 16-bit integer, is normally predefined for the client-server application. For example, if the HTTP protocol is used for accessing the web page, the well-known port number is 80. However, if a different port is used, the number canbe explicitly given.

Path. The path identifies the location and the name of the file in the underlying operating system.

The format of this identifier normally depends on the operatingsystem.

In UNIX, a path is a set of directory names followed by the file
name, all separated by a slash.

For example, /top/next/last/myfile is a path that uniquely defines a file named my file, stored in the directory last, which itself ispart of the directory next, which itself is under the directory top.

To combine these four pieces together, the **uniform resource locator (URL)** is used. It uses three different separators between the four pieces.

Web Documents

The documents in the WWW can be grouped into three broad categories: static, dynamic,and active.

Static Documents

Static documents are fixed-content documents that are created and stored in a server.

The client can get a copy of the document only.

The contents inthe server can be changed, but the user cannot change them.

When a client accesses thedocument, a copy of the document is sent. The user can then use a browser to see thedocument.

Static documents are prepared using one of several languages: HyperTextMarkup Language (HTML), Extensible Markup Language (XML), Extensible StyleLanguage (XSL), and Extensible Hypertext Markup Language (XHTML).

Dynamic Documents

A **dynamic document** is created by a web server whenever a browser requests the document.
When a request arrives, the web server runs an application program or a script thatcreates the dynamic document.

The server returns the result of the program or script as aresponse to the browser that requested the document. Because a fresh document is createdfor each request, the contents of a dynamic document may vary from one request toanother.
A simple example of a dynamic document is the retrieval of the time anddate from a server.

Time and date are kinds of information that are dynamic in that theychange from moment to moment. The client can ask the server to run a program such asthe date program in UNIX and send the result of the program to the client.

Active Documents

For many applications, we need a program or a script to be run at the client site. These arecalled **active documents.**

For example, if we want to run a program that createsanimated graphics on the screen or a program that interacts with the user. The program definitely needs to be run at the client site where the animation or interaction takes place.

When a browser requests an active document, the server sends a copy of the document ora script.

The document is then run at the client (browser) site. One way to create an activedocument is to use Java applets, a program written in Java on the server. It is compiledand ready to be run. The document is in bytecode (binary) format.

HyperText Transfer Protocol (HTTP)

The standard web transfer protocol is HTTP.

The **HyperText Transfer Protocol (HTTP)** is used to define how the client-serverprograms can be written to get web pages from the Web.

An HTTP client sends arequest; an HTTP server returns a response. The server uses the port number 80; the clientuses a temporary port number. HTTP uses the services of TCP, which is a connection-oriented and reliable protocol.

Non persistent versus Persistent Connections

Non persistent Connections

In a **nonpersistent connection**, one TCP connection is made for each request/response.

The steps in this strategy:

1. The client opens a TCP connection and sends a request.
2. The server sends the response and closes the connection.
3. The client reads the data until it encounters an end-of-file marker; it then closes the connection.

In this strategy, if a file contains links to N different pictures in different files (all located on the same server), the connection must be opened and closed N + 1 times.

The nonpersistent strategy imposes high overhead on the server because the server needs N + 1 different buffers each time a connection is opened.

Figure shows an example of a nonpersistent connection.

The client needs to access a file that contains one link to an image. The text file and image are located on the same server. Here we need two connections. For each connection, TCP requires at least three handshake messages to establish the connection, but the request can be sent with the third one.

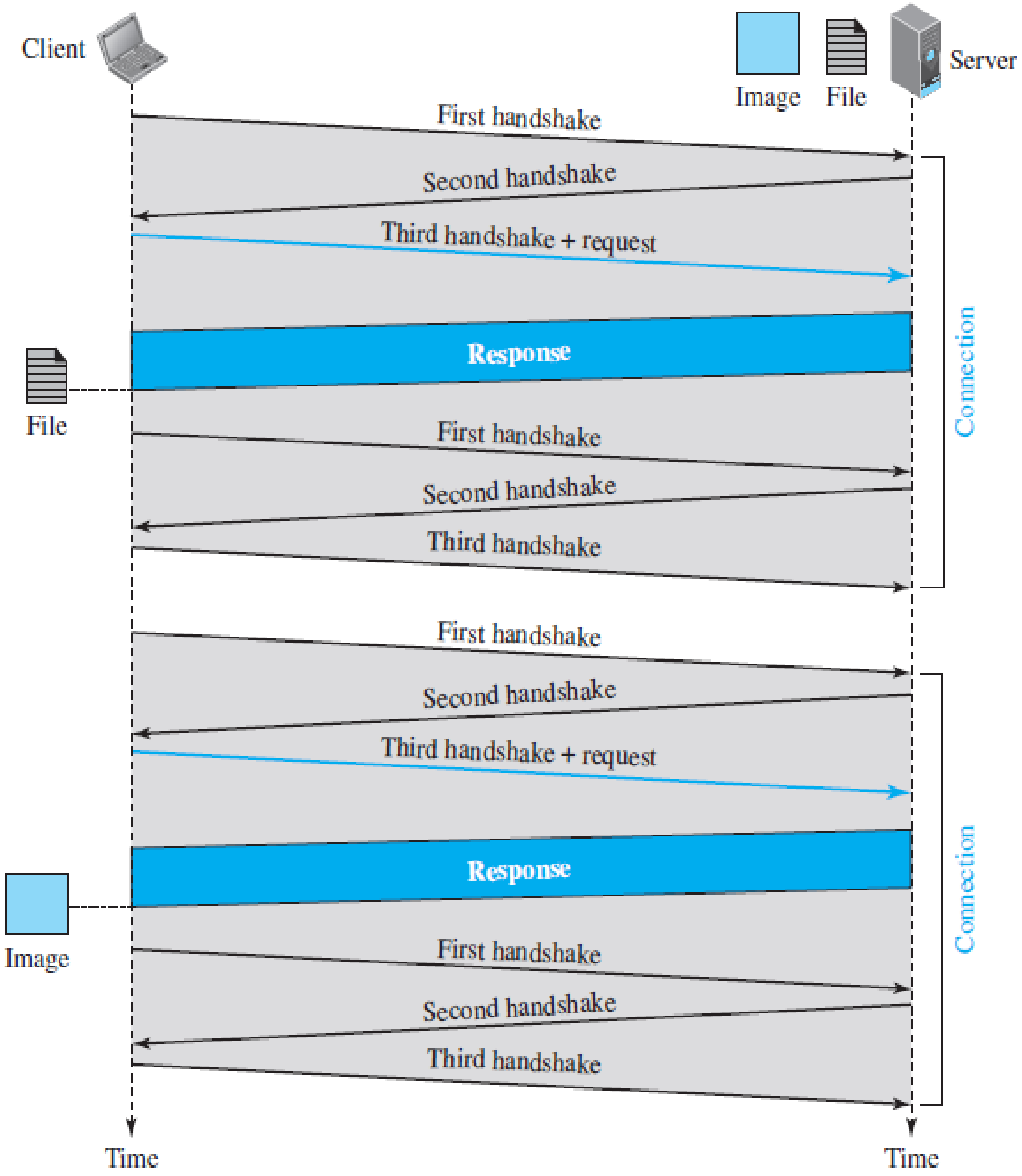


Fig: Nonpersistent connection .. [Source : Data Communications and Networking by Behrouz A. Forouzan]

After the connection is established, the object can be transferred. After receiving an object, another three handshake messages are needed to terminate the connection.

The client and server are involved in two connection establishments and two connection terminations.

Persistent Connections

HTTP 1.1 make persistent connections .

In a persistent connection,the server keeps the connection open for more requests after sending a response.
The server can close the connection at the request of a client or if a time-out has been reached.

The sender usually sends the length of the data with each response.

There are some occasions ,when the sender does not know the length of the data. This happens when a document is created dynamically or actively.

In these cases, the serverinforms the client that the length is not known and closes the connection after sendingthe data so the client knows that the end of the data has been reached.

Time andresources are saved using persistent connections. Only one set of buffers and variablesneeds to be set for the connection at each site.
The **round trip time** for connection establishmentand connection termination is saved.

What is RTT?

RTT is the time it takes for a packet to travel from client to server and then back to the client.

Figure (below) shows the example for persistent connections.

Here Only one connection establishment and connection termination is used, but the request for the image is sent separately.

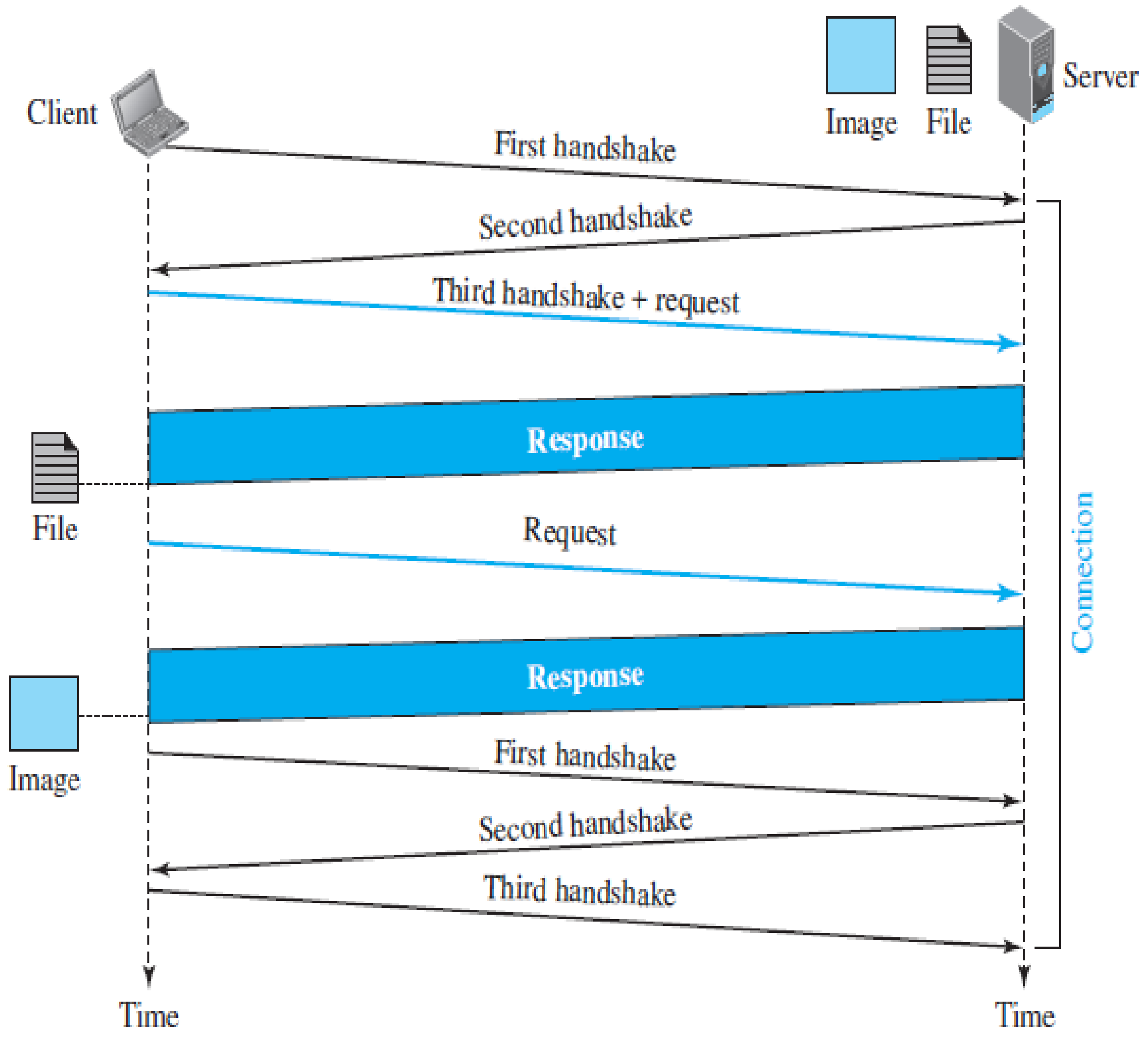


Fig: Persistent connection .. [Source : Data Communications and Networking by Behrouz A. Forouzan]

HTTP Messages

Two types: request , response

Message Formats
The HTTP protocol defines the format of the request and response messages, as shown in Figure .

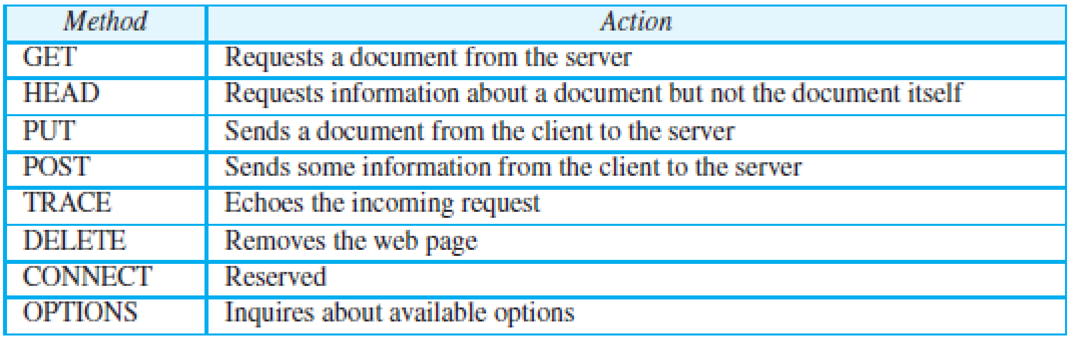
Each message is made of four sections. The first section in the request message is called therequest line; the first section in the response message is called the status line.

Request Message

The first line in a request message is called a request line.

There are three fields in this line separated by one space and terminated by two characters (carriage return and line feed).

The fields are called method, URL, and version.



Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
PUT	Sends a document from the client to the server
POST	Sends some information from the client to the server
TRACE	Echoes the incoming request
DELETE	Removes the web page
CONNECT	Reserved
OPTIONS	Inquires about available options

The method field defines the request types.

The client uses the GET method to send a request. In this case, the body of the message is empty. The HEAD method is used when the client needs only some information about the web page from the server, such as the last time it was modified. It is used to test the validity of a URL.

The response message in this case has only the header section; the body section is empty. The PUT method is the inverse of the GET method; it allows the client to post a new web page on the server (if permitted).

The POST method is like PUT method, but it is used to send some information to the server to be added to the web page or to modify the web page.

The TRACE method is used for debugging; the client asks the server to echo back the request to check whether the server is getting the requests.

The DELETE method allows the client to delete a web page on the server if the client has permission to do so.

The CONNECT method was made as a reserve method; it may be used by proxy servers. The OPTIONS method allows the client to ask about the properties of a web page from the server.

The second field, URL defines the address and name of the corresponding web page. The third field, gives the version of the protocol; the current version of HTTP is 1.1.

After the request line, we have request header lines.
Each header line sends additional information from the client to the server.

For example, the client can request that the document be sent in a special format. Each header line has a header name, a colon, a space, and a header value

The value field defines the values associated with each header name.

The body can be present in a request message. It contains the comment
to be sent or the file to be published on the website when the method is PUT or
POST.

Response Message

A response message consists of a status line, header lines, a blank line, and sometimes a body.

The first line in a response message is called the status line. There are three fields in this line separated by spaces and terminated by a carriage return and line feed.

The first field defines the version of HTTP protocol, currently 1.1.

The status code field defines the status of the request. It consists of three digits.

The codes in the 100 range are only informational, the codes in the 200 range indicate a successful request. The codes in the 300 range redirect the client to another URL, and the codes in the 400 range indicate an error at the client site.

The codes in the 500 range indicate an error at the server site. The status phrase explains the status code in text form.

After the status line, we can have zero or more response header lines.

Each headerline sends additional information from the server to the client. For example, the sendercan send extra information about the document. Each header line has a header name, acolon, a space, and a header value. The body contains the document to be sent from the server to the client.

Domain Name System (DNS)

The DNS is a distributed database that resides on multiple machines on the internet. It provide e-mail routing information.

The DNS protocol runs over UDP and uses port 53.

Figure (below) shows how TCP/IP uses a DNS client and a DNS server to map a name to an address.

A user wants to use a file transfer client to access the corresponding file transfer server running on a remote host. The user knows only the file transfer server name, such as afilesource.com.

The TCP/IP suite needs the IP address of the file transfer server to make the connection.

The following six steps map the host name to an IP address:

1. The user passes the host name to the file transfer client.
2. The file transfer client passes the host name to the DNS client.
3. Each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.
4. The DNS server responds with the IP address of the desired file transfer server.
5. The DNS server passes the IP address to the file transfer client.
6. The file transfer client now uses the received IP address to access the file transfer server.

Domain Name Space

To have a hierarchical name space, a **domain name space** was designed.

In this design the names are defined in an inverted-tree structure with the root at the top. The tree has 128 levels: level 0 (root) to level 127 (see Figure).

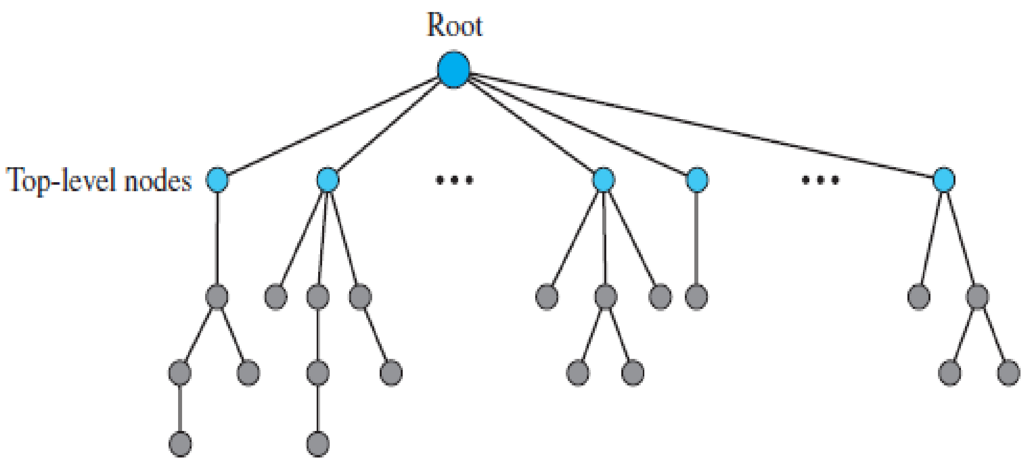


Fig: DNS tree. [Source : Data Communications and Networking by Behrouz A. Forouzan]

Label

Each node in the tree has a **label**, which is a string with a maximum of 63 characters. The root label is a null string (empty string).

Domain Name

Each node in the tree has a domain name. A full **domain name** is a sequence of labels separated by dots (**.**). The domain names are always read from the node up to the root.

The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

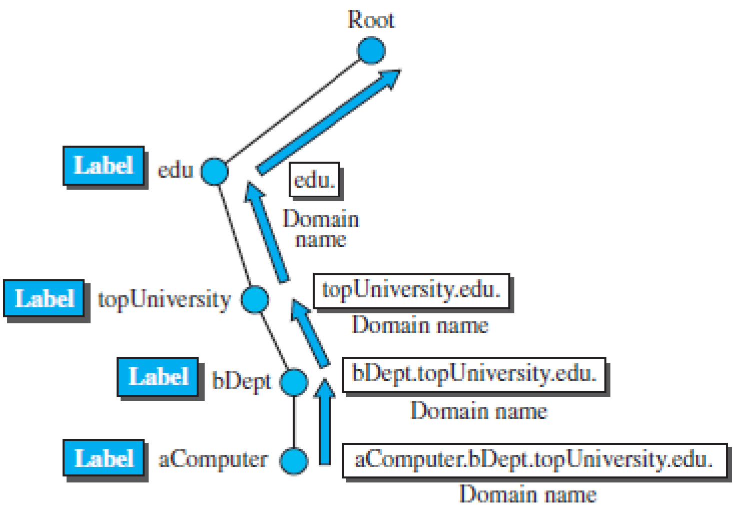


Fig: Domain names. [Source : Data Communications and Networking by Behrouz A. Forouzan]

If a label is terminated by a null string, it is called a **fully qualified domain name (FQDN)**. The name must end with a null label, but because null means nothing, the label ends with a dot.

If a label is not terminated by a null string, it is called a **partially qualified domain name (PQDN).** A PQDN starts from a node, but it does not reach the root. It is used when the name to be resolved belongs to the same site as the client.
Here the resolver can supply the missing part, called the suffix, to create an FQDN.

Domain

A **domain** is a subtree of the domain name space. The name of the domain is the name of the node at the top of the subtree. Figure (below) shows some domains. A domain is divided into domains.

Distribution of Name Space

The information contained in the domain name space must be stored.
It is inefficient and not reliable to have just one computer store such a huge amount of information. It is inefficient because responding to requests from all over the world places a heavy load on the system.

Hierarchy of Name Servers

To distribute the information among many computers called **DNS servers.** One way to do this is to divide the whole space into many domains based on the first level.

DNS allows domains to be divided further into smaller domains (subdomains). Each server can be responsible (authoritative) for either a large or small domain.

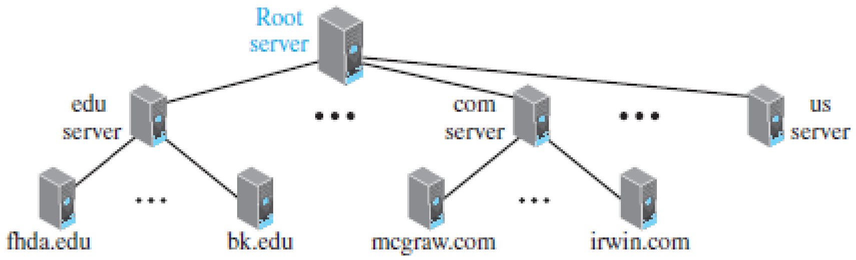


Fig: Domain name hierarchy. [Source : Data Communications and Networking by Behrouz A. Forouzan]

Zone

If a server accepts responsibility for a domain and does not divide the domain into smallerdomains, the “domain” and the “zone” refer to the same thing. The server makes a databasecalled a zone file and keeps all the information for every node under that domain.

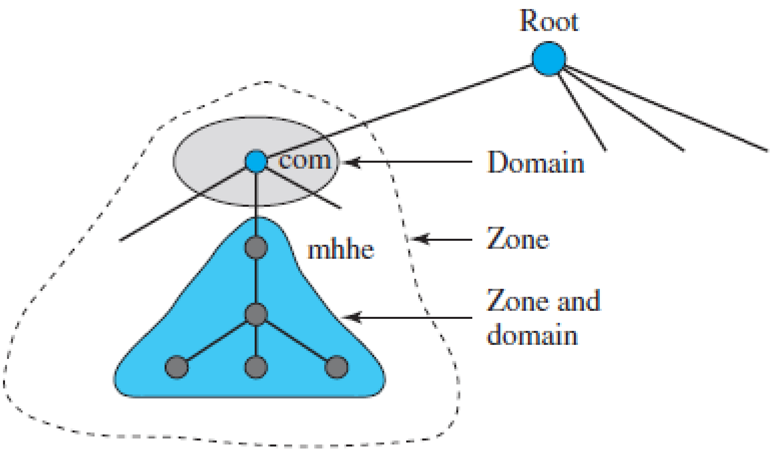


Fig: Zone [Source : Data Communications and Networking by Behrouz A. Forouzan]

The information about the nodes in the subdomains is stored in the servers at the lower levels, withthe original server keeping some sort of reference to these lower-level servers.

Root Server

If zone consists of the full tree then that zone server is called root server.

A root server does not store any information about domains .

A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file.

A secondary server is a server that loads all information from the primary server. Secondary server cannot perform any operation on zone file.

DNS in the Internet

In the Internet, the domain name space (tree) was divided into three sections: generic domains, country domains, and the inverse domains.

Generic Domains

The **generic domains** define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database.

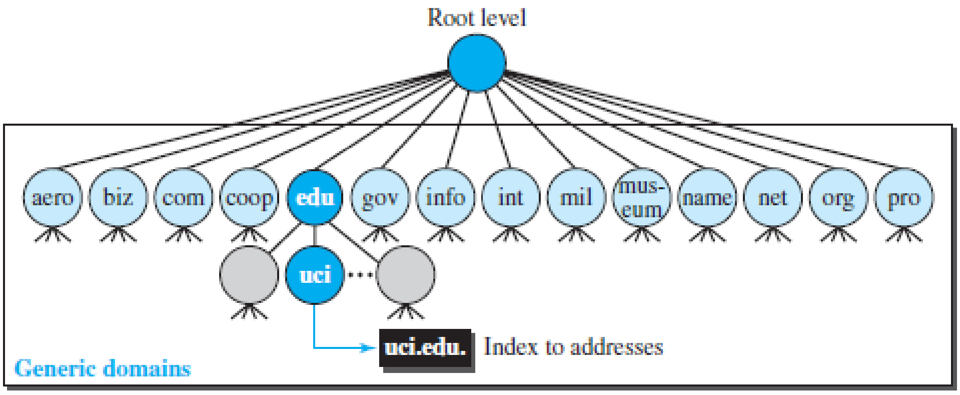


Fig: Generic domain [Source : Data Communications and Networking by Behrouz A. Forouzan]

Looking at the tree, the first level in the **generic domains** section allows 14 possible labels.

Country Domains

The **country domains** section uses two-character country abbreviations (e.g., us for United States). Second labels can be organizational, or they can be more specific national designations.

Examplefor the country domains section.
The address **uci.ca.us.** can be translated to University of California, Irvine, in the state of California in the United States.

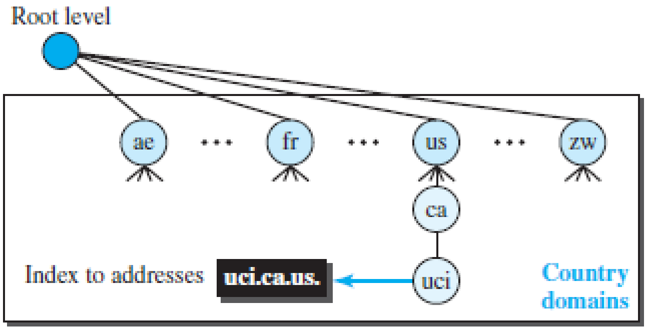


Fig: Country domain [Source : Data Communications and Networking by Behrouz A. Forouzan]

Inverse domain

Inverse domain is used to find the name of a host when given the IP address.

Resolution

Mapping a name to an address is called name-address resolution.

DNS is designed as a client-server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a **resolver.**

The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.

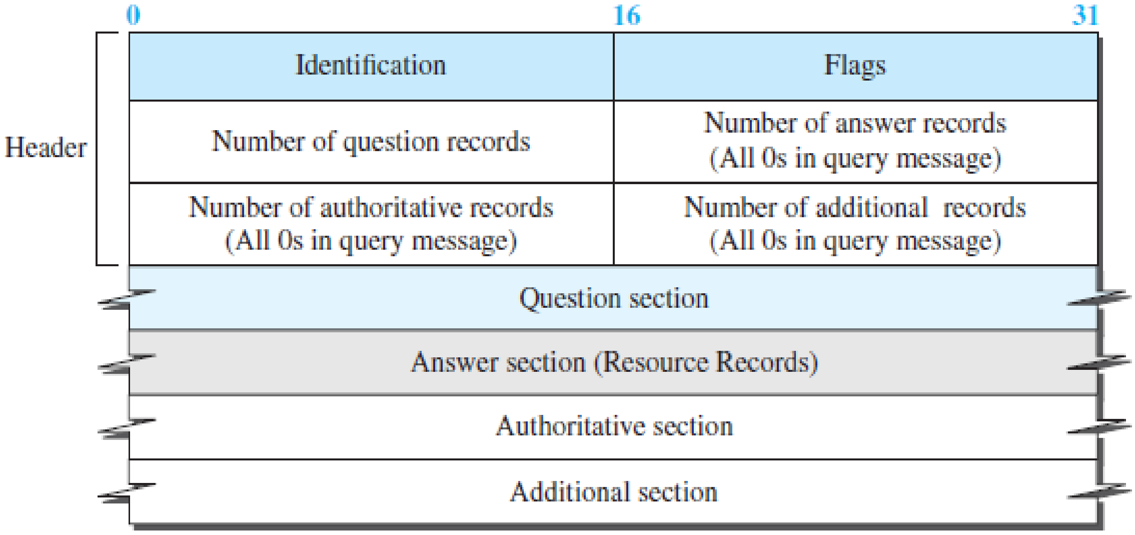
Recursive Resolution

A client request complete translation.
If the server is authority for the domain name, it checks its database and responds.
If the server is not authority, it sends the request to another server and waits for the response.

When the query is resolved ,the response travel back until it finally reaches the requesting client. This is called recursive resolution.

DNS Messages

To get information about hosts, DNS uses two types of messages: *query* and *response*. Both messages have the same format as shown in Figure .


0
16
31
Header
Identification
Flags
Number of question records
Number of answer records
(All 0s in query message)
Number of authoritative records
(All 0s in query message)
Number of additional records
(All 0s in query message)
Question section
Answer section (Resource Records)
Authoritative section
Additional section

Note:
The query message contains only the question section.
The response message includes the question section,
the answer section, and possibly two other sections.

Fig: DNS messages [Source : Data Communications and Networking by Behrouz A. Forouzan]

The identification field is used by the client to match the response with the query.
The flag field defines whether the message is a query or response. It also includes status of error.

The next four fields in the header define the number of each record type in the message.
The question section consists of one or more question records. It is present in both query and response messages.

The answer section consists of one or more resource records. It is present only in response messages.
The authoritative section gives information (domain name) about one or more authoritative servers for the query.

The additional information section provides additional information that may help the resolver.

Encapsulation

DNS can use either UDP or TCP. The port used by the server is port 53.

UDP is used when the size of the response message is less than 512 bytes because most UDP packages have a 512-byte packet size limit.
If the size of the response message is more than 512 bytes, a TCP connection is used.

Security of DNS

Applications such as Web access or e-mail are dependent on the proper operation of DNS. DNS can be attacked in several ways :

1. The attacker may read the response of a DNS server to find the nature or names of sites the user mostly accesses. This type of information can be used to find the user's profile. To prevent this attack, DNS messages need to be confidential .

ELECTRONIC MAIL

Electronic mail (or e-mail) allows users to exchange messages. Electronic mail is used for sending a single message like text, voice,video or grapics to one or more recipients.

Email is fast,easy to distribute and inexpensive.

The sender and receiver use three different *agents*: a **user agent (UA),** a **message transfer agent(MTA),** and a **message access agent (MAA).**

The mail server uses a queue (spool) to store messages waiting to be sent. The message, needs to be sent through the Internet from client to server site using an MTA.

Here two message transfer agents are needed: one client and one server.

User Agent

The first component of an electronic mail system is the **user agent (UA).**

It provides service to the user to make the process of sending and receiving a message easier.
A user agent is a software package (program) that composes, reads, replies to, and forwards messages. It also handles local mailboxes on the user computers.

There are two types of user agents: command-driven and GUI-based.

A command-driven user agent normally accepts a one character command from the keyboard to perform its task.

Sending Mail

To send mail, the user, through the UA, creates mail that looks very similar to postalmail. It has an envelope and a message

The message contains the header and the body. The header of the message defines the sender, the receiver, the subject of the message, and some other information. The body of the message contains the actual information to be read by the recipient.

Receiving Mail

The user agent is triggered by the user (or a timer). If a user has mail, the UA informs the user with a notice. If the user is ready to read the mail, a list is displayed in which each line contains a summary of the information about a particular message in the mailbox..

Addresses

To deliver mail, a mail handling system must use an addressing system with unique addresses.

In the Internet, the address consists of two parts: a local part and a domain name, separated by an @ sign.

SMTP

Simple Mail Transfer Protocol (SMTP) act as message transfer agent.

Commands and Responses

SMTP uses commands and responses to transfer messages between an MTA client and an MTA server. The command is from an MTA client to an MTA server; the response is from an MTA server to the MTA client.

Commands

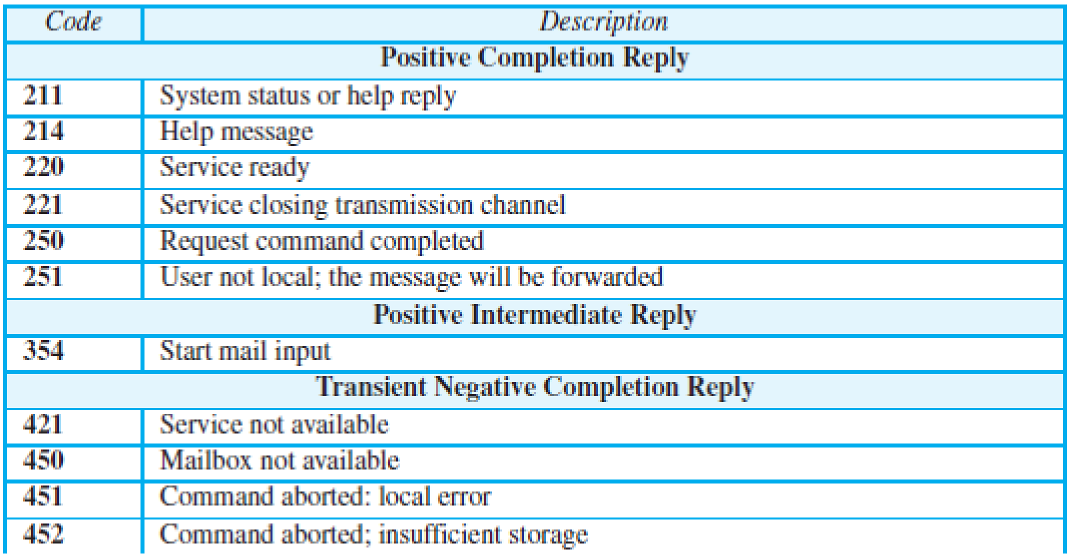
Commands are sent from the client to the server.

Keyword: argument(s)

SMTP defines 14 commands.

Responses

Responses are sent from the server to the client. A response is a three digit code that may be followed by additional textual information.



Code	*Description*
	Positive Completion Reply
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
	Positive Intermediate Reply
354	Start mail input
	Transient Negative Completion Reply
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage

Mail Transfer Phases

The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

Connection Establishment After a client has made a TCP connection to the well known port 25, the SMTP server starts the connection phase. This phase involves the following three steps:

1. The server sends code 220 (service ready) to tell the client that it is ready to receive mail. If the server is not ready, it sends code 421 (service not available).

2. The client sends the HELO message to identify itself, using its domain name address. This step is necessary to inform the server of the domain name of the client.

3. The server responds with code 250 (request command completed) or some other code depending on the situation.

Message Transfer After connection has been established between the SMTP client and server, a single message between a sender and one or more recipients can be exchanged. This phase involves eight steps. Steps 3 and 4 are repeated if there is more than one recipient.

1. The client sends the MAIL FROM message to introduce the sender of the message. It includes the mail address of the sender (mailbox and the domain name). This step is needed to give the server the return mail address for returning errors and reporting messages.
2. The server responds with code 250 or some other appropriate code.
3. The client sends the RCPT TO (recipient) message, which includes the mail addressof the recipient.
4. The server responds with code 250 or some other appropriate code.
5. The client sends the DATA message to initialize the message transfer.
6. The server responds with code 354 (start mail input) or some other appropriatemessage.
7. The client sends the contents of the message in consecutive lines. Each line is terminatedby a two-character end-of-line token (carriage return and line feed). Themessage is terminated by a line containing just one period.
8. The server responds with code 250 (OK) .

Connection Termination After the message is transferred successfully, the client terminates the connection.

Two steps.

.
1. The client sends the QUIT command.
2. The server responds with code 221 or some other appropriate code.

POP3 (Post Office Protocol, version 3)

Post Office Protocol, version 3 (POP3) is simple in its function.

The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.

Mail access starts with the client when the user needs to download its e-mail from the mailbox on the mail server. The client opens a connection to the server on TCP port 110.

It then sends its user name and password to access the mailbox. The user can then list and retrieve the mail messages, one by one.

Figure (below) shows an example of downloading using POP3.

POP3 has two modes: the *delete* mode and the *keep* mode.

In the delete mode, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval.

The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying.

The keep mode is normally used when the user accesses her mail away from her primary computer (for example, from a laptop).

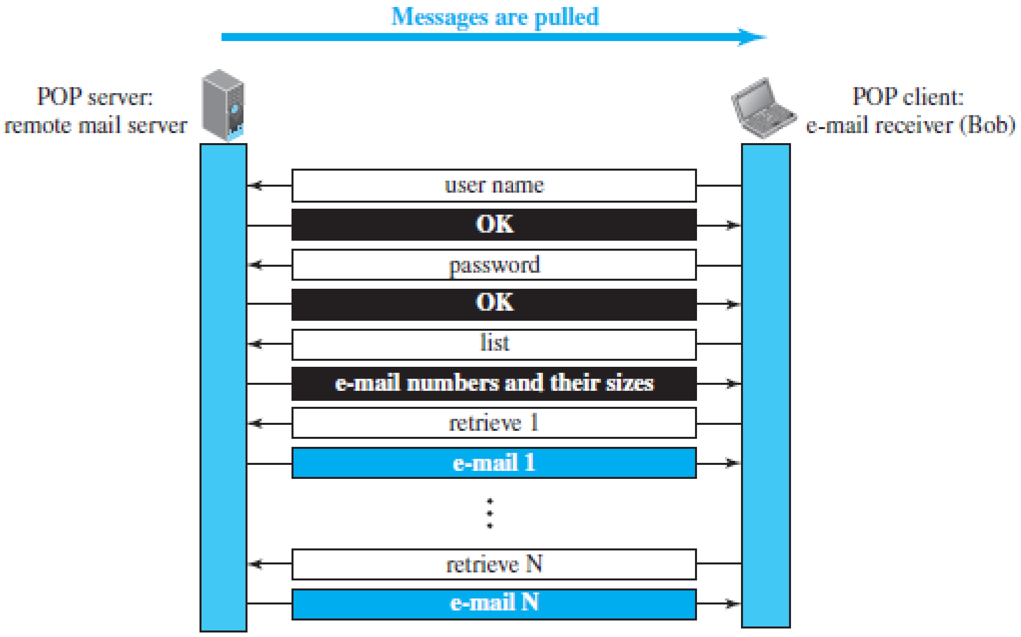


Fig: POP3 format. [Source : Data Communications and Networking by Behrouz A. Forouzan]

POP3 has some drawbacks.

It does not allow the user to organize her mail on the server; the user cannot have different folders on the server. In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.

IMAP4

IMAP4 is**Internet Mail Access Protocol, version 4 (IMAP4).**
IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.

IMAP4 provides the following extra functions:

A user can check the e-mail header prior to downloading.
A user can search the contents of the e-mail for a specific string of characters prior to downloading.
A user can partially download e-mail.

This is very useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.

A user can create, delete, or rename mailboxes on the mail server.
A user can create a hierarchy of mailboxes in a folder for e-mail storage.

MIME

Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail.

MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through the Internet.

The message at the receiving site is transformed back to the original data.

MIME means a set of software functions that transforms non-ASCII
data to ASCII data .

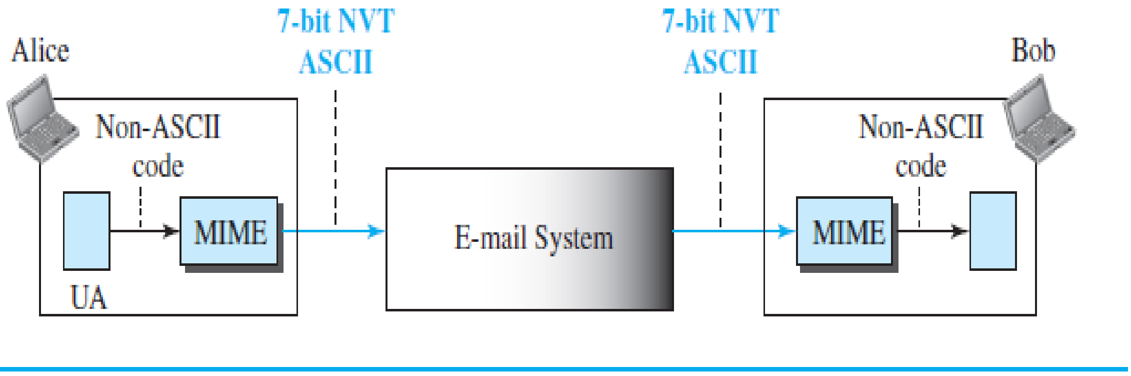


Fig: MIME. [Source : Data Communications and Networking by Behrouz A. Forouzan]

MIME Headers
MIME defines five headers, as shown in Figure , which can be added to the original e-mail header section to define the transformation parameters:

Figure 26.19 *MIME header*

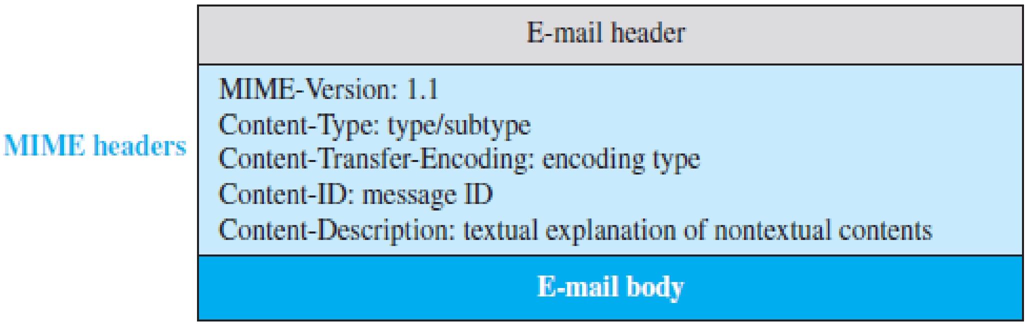
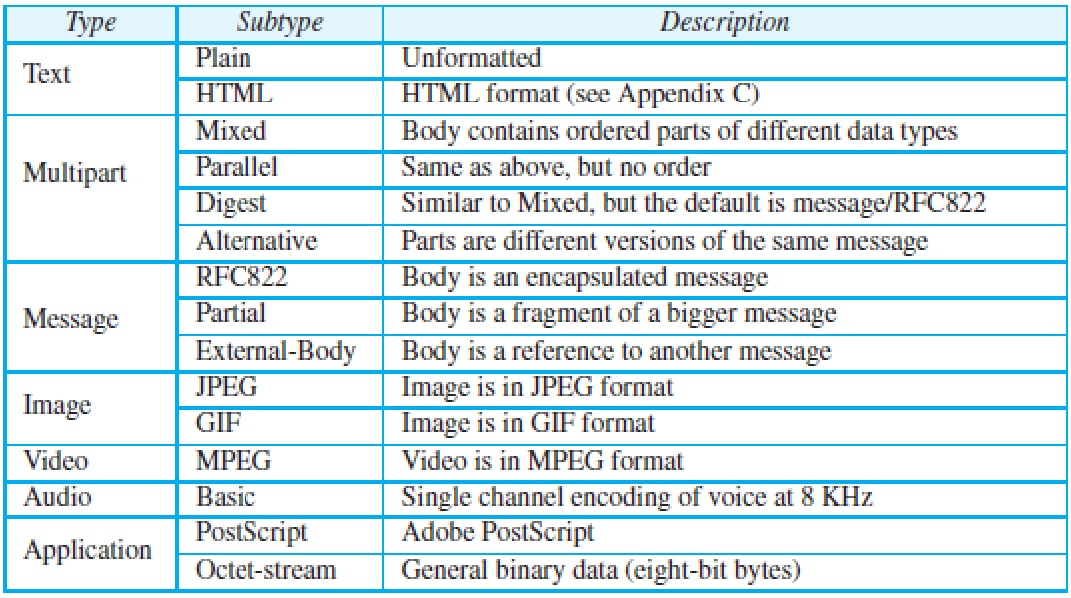

E-mail header
MIME headers
MIME-Version: 1.1
Content-Type: type/subtype
Content-Transfer-Encoding: encoding type
Content-ID: message ID
Content-Description: textual explanation of nontextual contents
E-mail body

Fig: MIME header. [Source : Data Communications and Networking by Behrouz A. Forouzan]

MIME-Version The current version is 1.1.
Content-Type This header defines the type of data used in the body of the message. The content type and the content subtype are separated by a slash. Depending on the subtype, the header may contain other parameters.

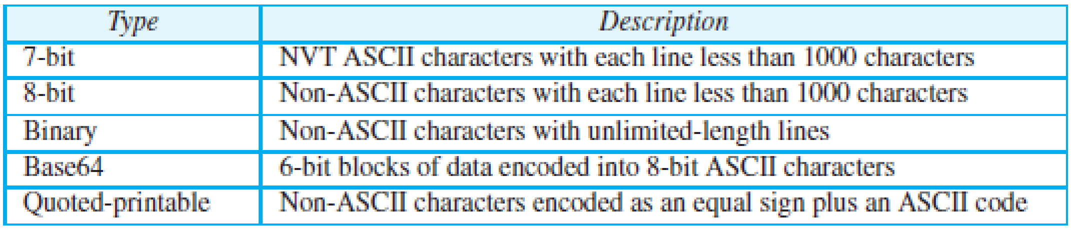
MIME allows seven different types of data,



Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Appendix C)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

Content-Transfer-Encoding This header defines the method used to encode the messages into 0s and 1s for transport.

The five types of encoding methods are shown in Table.



Type	*Description*
7-bit	NVT ASCII characters with each line less than 1000 characters
8-bit	Non-ASCII characters with each line less than 1000 characters
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equal sign plus an ASCII code

In the Base 64 encoding, data, as a string of bits, is first divided into 6-bit chunks.

Each 6-bit section is then converted into an ASCII character.

Cryptography and Network Security

Information security is important in communication.

Security Goals

Three Security goals: confidentiality, integrity, and availability.

Confidentiality

Confidentiality is the most common aspect of information security.
Confidentiality means preventing access to unauthorized person.

We needto protect our confidential information. An organization needs to guard against thosemalicious actions that endanger the confidentiality of its information.

When we send a piece of information to be stored in a remote computer orwhen we retrieve a piece of information from a remote computer, we need to conceal itduring transmission.

Integrity
Information needs to be changed constantly. In a bank, when a customer deposits orwithdraws money, the balance of her account needs to be changed.

Integrity means thatchanges need to be done only by authorized entities and through authorized mechanisms.

Availability

The third component of information security is availability. The information created and stored by an organization needs to be available to authorized entities.

Security Attacks

Attacks Threatening Confidentiality

In general, two types of attacks threaten the confidentiality of information: snooping and traffic analysis.

Snooping
Snooping refers to unauthorized access to or interception of data

To prevent snooping, the data can be made nonintelligible to the intercepter by using encipherment techniques.

Symmetric-Key Ciphers

A **symmetric-key cipher** uses the same key for both encryption and decryption, and the key can be used for bidirectional communication, which is why it is called symmetric.

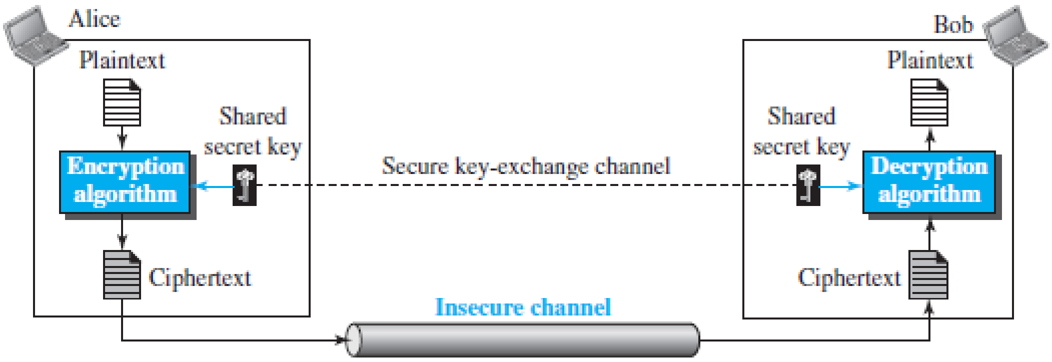


Fig: symmetric-key cipher . [Source : Data Communications and Networking by Behrouz A. Forouzan]

In Figure, an entity called Alice and Bob are used.

Consider the scenario, Alice can send a message to another entity, Bob, over aninsecure channel with the assumption that it is safe.

The original message from Alice to Bob is called **plaintext;** the message that is sent through the channel is called **ciphertext.**

To create the cipher text from the plain text,Alice uses an **encryption algorithm** and a shared secret key.

To create the plaintext from ciphertext, Bob uses a **decryption algorithm** and thesame secret key.

The encryption and decryption algorithms are called as **ciphers.**

A **key**isa set of values (numbers) that the cipher, as an algorithm, operates on.

The symmetric-key encipherment uses a single key for both encryption and decryption. In addition, the encryption and decryption algorithms are inverses of each other.

If P is the plaintext, C is the ciphertext, and K is the key, the encryption algorithm Ek(x) creates the ciphertext from the plaintext; the decryption algorithm Dk(x) creates the plaintext from the ciphertext.

Here, Dk(Ek(x)) = Ek(Dk(x)) = x.

Encryption is like, locking the message in a box; decryption is like unlocking the box. In symmetric-key encipherment, the same key locks and unlocks.

The symmetric-key ciphers can be divided into traditional ciphers and modern ciphers.

Traditional ciphers are simple, ie, character-oriented ciphers that are not secure.

Modern ciphers are bit oriented ciphers that are more secure.

Traditional Symmetric-Key Ciphers

Traditional ciphers are divided into substitution ciphers and transposition ciphers.

Substitution Ciphers

A **substitution cipher** replaces one symbol with another.

Substitution ciphers can be categorized as mono alphabetic ciphers or polyalphabetic ciphers.

Monoalphabetic Ciphers

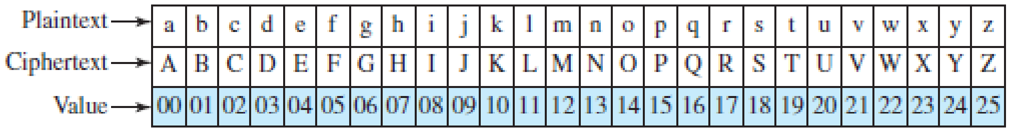
In a **monoalphabetic cipher,** a character (or a symbol) in the plaintext is always changed to the same character (or symbol) in the ciphertext regardless of its position in the text.

For example, if the algorithm says that letter A in the plaintext is changed to letter D, every letter A is changed to letter D.
Here the relationship between letters in the plaintext and the ciphertext is one-to-one.

The simplest mono alphabetic cipher is the **additive cipher** (or **shift cipher**).

Assume that the plaintext consists of lowercase letters (a to z), and that the ciphertext consists of uppercase letters (A to Z).

To apply mathematical operations on the plaintext and ciphertext, we assign numerical values to each letter (lowercase or uppercase), as shown in Figure .



Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Fig: Representation of cipher . [Source : Data Communications and Networking by Behrouz A. Forouzan]

In Figure (above), each character (lowercase or uppercase) is assigned an integer inmodulo 26. The secret key between Alice and Bob is also an integer in modulo 26.

The encryption algorithm adds the key to the plaintext character; the decryption algorithm subtracts the key from the ciphertext character.

Polyalphabetic Ciphers

In a **polyalphabetic cipher,** each occurrence of a character may have a different substitute.

The relationship of a character in the plaintext to a character in the ciphertext is one-to-many.

For example, “a” could be enciphered as “D” at the beginning of the text, but as “N” in the middle.

To create a polyalphabetic cipher, we need to make each ciphertext character dependent on both the corresponding plaintext character and the position of the plaintext character in the message.

This implies that our key should be a stream of subkeys, in which each subkey depends somehow on the position of the plaintext character that uses that subkey for encipherment.

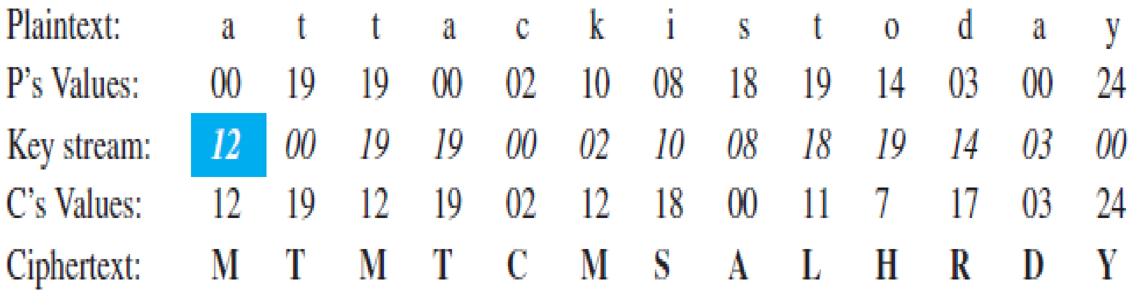
We need to have a key stream k = (k1, k2, k3, …) in which keys used to encipher the ith character in the plaintext to create the ith character in the ciphertext.

Example -Polyalphabetic cipher

Assume that Alice and Bob agreed to use an autokey cipher with initial key value *k*1 = 12. Now Alice wants to send Bob the message “Attack is today”.

Enciphering is done character by character.
Each character in the plaintext is first replaced by its integer value. The first subkey is added to create the first ciphertext character.
The rest of the key is created as the plaintext characters are read.

Note that the cipher is polyalphabetic because the three occurrences of “a” in the plaintext are encrypted differently. The three occurrences of “t” are also encrypted differently.



Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P’s Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	*12*	*00*	*19*	*19*	*00*	*02*	*10*	*08*	*18*	*19*	*14*	*03*	*00*
C’s Values:	12	19	12	19	02	12	18	00	11	7	17	03	24
Ciphertext:	M	T	M	T	C	M	S	A	L	H	R	D	Y

Fig: Example of cipher with key . [Source : Data Communications and Networking by Behrouz A. Forouzan]

Transposition Ciphers

A **transposition cipher** does not substitute one symbol for another; instead it changes the location of the symbols.

A symbol in the first position of the plaintext may appear in the tenth position of the ciphertext.
A symbol in the eighth position in the plaintext may appear in the first position of the ciphertext.

A transposition cipher reorders (transposes) the symbols.

Suppose Alice wants to secretly send the message “Enemy attacks tonight” to Bob.
The encryption and decryption is shown in Figure (below).

Note that we added an extra character (z) to the end of the message to make the number of characters a multiple of 5.

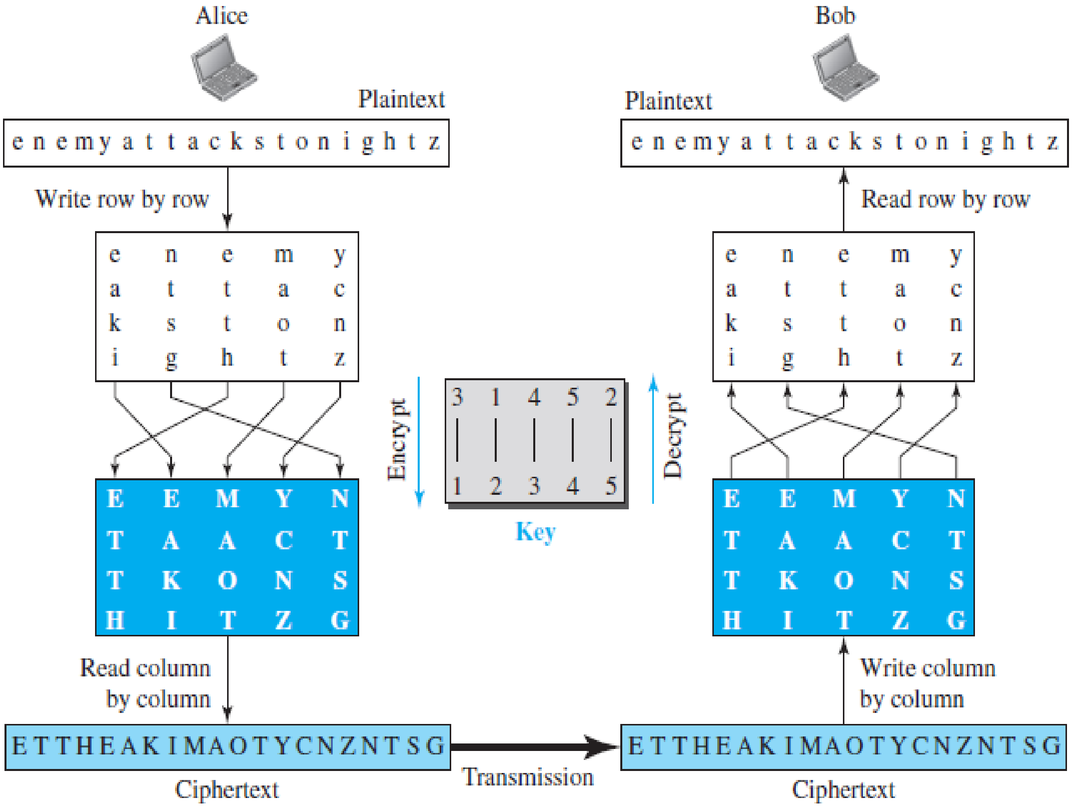


Fig: Transposition ciphers. [Source : Data Communications and Networking by Behrouz A. Forouzan]

The first table is created by Alice writing the plaintext row by row. The columns are permuted using a key.

The ciphertext is created by reading the second table column by column.

In the receiving side, Bob does the same three steps in the reverse order.

He writes the ciphertext column by column into the first table, permutes the columns, and then reads the second table row by row.

Here the same key is used for encryption and decryption, but the algorithm uses the key in reverse order.

Asymmetric-Key Ciphers

In asymmetric-key cryptography, the secret is personal (unshared); each person creates and keeps his or her own secret.

In asymmetric-key cryptography, the plaintext and ciphertext are numbers; encryption and decryption are mathematical functions that are applied to numbers to create other numbers.

Asymmetric key cryptography uses two separate keys: one private and one public.
If encryption and decryption are applied for locking and unlocking padlocks with keys, then the padlock that is locked with a public key can be unlocked only with the corresponding private key.

Figure shows that if Alice locks the padlock with Bob's public key, then only Bob's private key can unlock it.

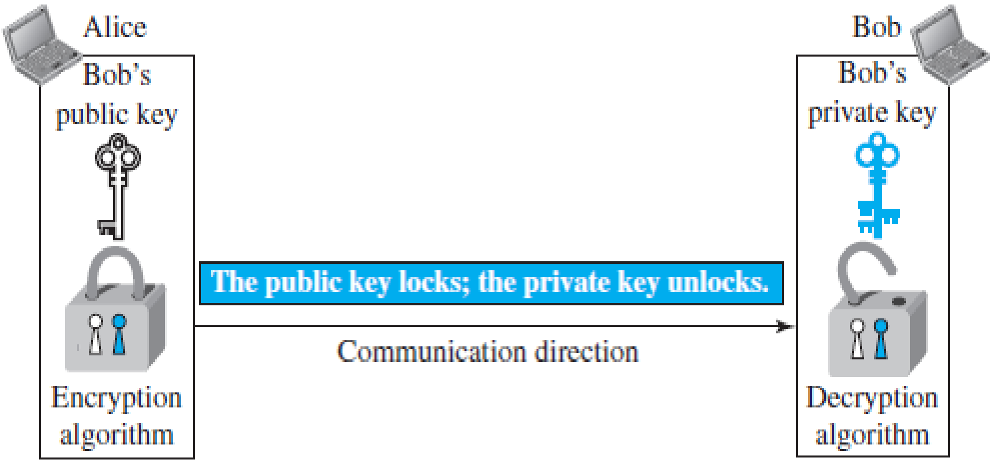


Fig: Asymmetric key ciphers. [Source : Data Communications and Networking by Behrouz A. Forouzan]

General Idea of asymmetric key method

The general idea of asymmetric-key cryptography as used for encipherment.

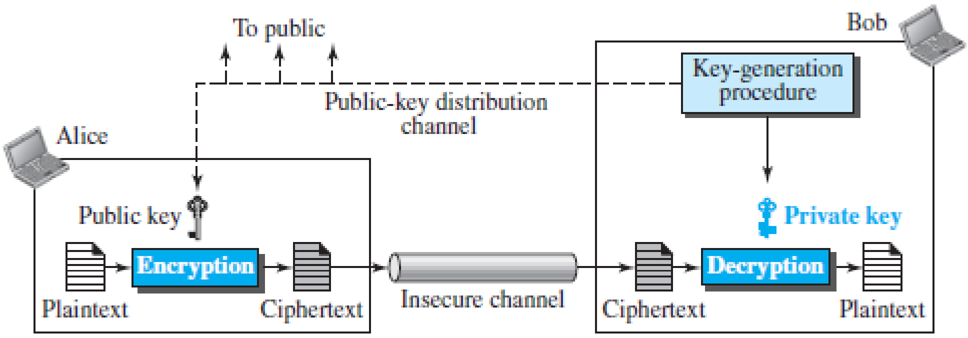


Fig: Asymmetric key cryptosystem. [Source : Data Communications and Networking by Behrouz A. Forouzan]

The burden of providing security is mostly on the shoulders of the receiver (Bob, in this case).

Bob needs to create two keys: one private and one public.

Bob is responsible for distributing the public key to the community. This can be done through a public-key distribution channel. Eve should not be able to advertise her public key to the community pretending that it is Bob's public key.

Second, asymmetric-key cryptography means that Bob and Alice cannot use the same set of keys for two-way communication. Each entity in the community should create its own private and public keys.

Figure (above) shows how Alice can use Bob's public key to send encrypted messages to Bob. If Bob wants to respond, Alice needs to establish her own private and public keys.

Third, asymmetric-key cryptography means that Bob needs only one private key to receive all correspondence from anyone in the community, but Alice needs *n* public keys to communicate with *n* entities in the community, that is one public key for each entity.

FIREWALLS

A **firewall** is a device (usually a router or a computer) installed between the internal network of an organization and the rest of the Internet. It is designed to forward some packets and filter (not forward) others.

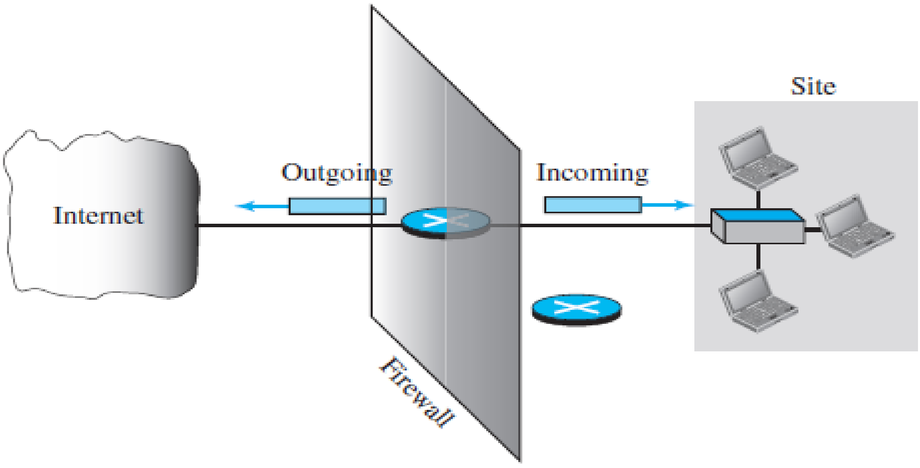


Fig: Firewall [Source : Data Communications and Networking by Behrouz A. Forouzan]

For example, a firewall may filter all incoming packets destined for a specific host or a specific server such as HTTP.

A firewall can be used to deny access to a specific host or a specific service in the organization.

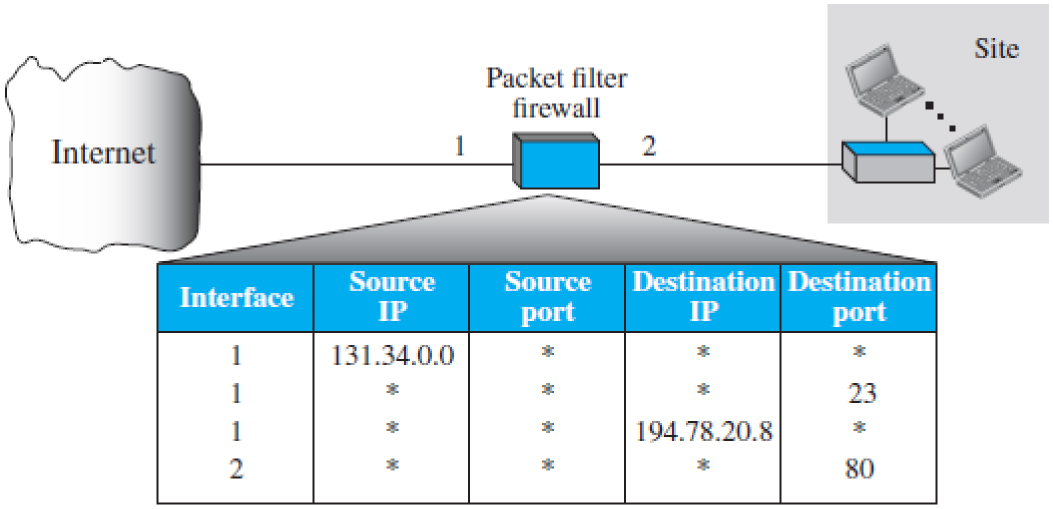
A firewall has two classifications.

Packet-filter firewall or a proxy-based firewall.

Packet-Filter Firewall

A firewall can be used as a packet filter. It can forward or block packets based on the information in the network-layer and transport-layer headers: source and destination IP addresses, source and destination port addresses, and type of protocol (TCP or UDP). A **packet-filter firewall** is a router that uses a filtering table to decide which packets must be discarded (not forwarded).

Figure shows an example of a filtering table for this kind of a firewall.



Interface	Source IP	Source port	Destination IP	Destination port
1	131.34.0.0	*	*	*
1	*	*	*	23
1	*	*	194.78.20.8	*
2	*	*	*	80

Fig: Packet filter firewall [Source : Data Communications and Networking by Behrouz A. Forouzan]

According to the figure, the following packets are filtered:

1. Incoming packets from network 131.34.0.0 are blocked (security precaution). Note that the * (asterisk) means “any.”
2. Incoming packets destined for any internal TELNET server (port 23) are blocked.
3. Incoming packets destined for internal host 194.78.20.8 are blocked. The organization wants this host for internal use only.
4. Outgoing packets destined for an HTTP server (port 80) are blocked. The organization does not want employees to browse the Internet.

Proxy Firewall

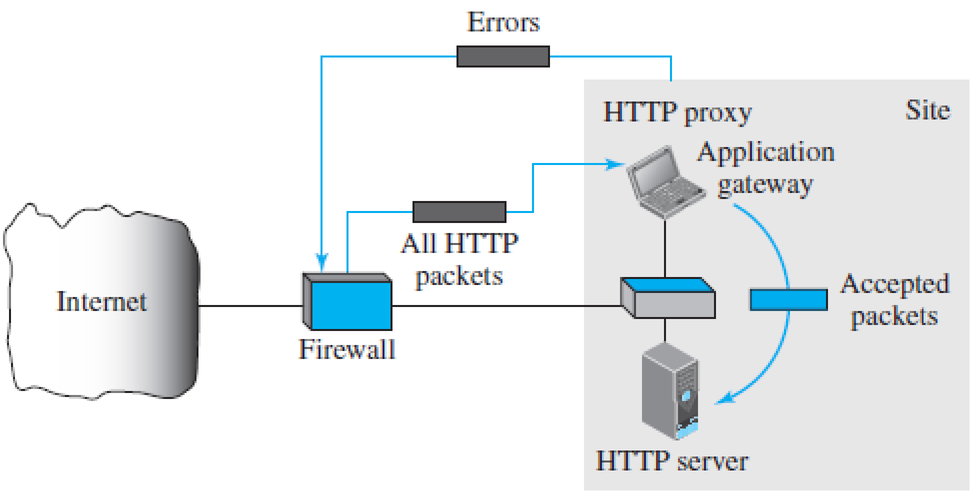


Fig: Proxy firewall [Source : Data Communications and Networking by Behrouz A. Forouzan]

The packet-filter firewall is based on the information available in the network layer andtransport layer headers (IP and TCP/UDP). However, sometimes we need to filter amessage based on the information available in the message itself (at the applicationlayer).

As an example, assume that an organization wants to implement the following

policies regarding its web pages: only those Internet users who have previously establishedbusiness relations with the company can have access; access to other users mustbe blocked.

In this case, a packet-filter firewall is not feasible because it cannot distinguishbetween different packets arriving at TCP port 80 (HTTP). Testing must be doneat the application level (using URLs).

One solution is to install a **proxy firewall** (computer) (sometimes called an **applicationgateway**), which stands between the customer computer and the corporationcomputer.

When the user client process sends a message, the application gateway runsa server process to receive the request. The server opens the packet at the applicationlevel and finds out if the request is legitimate.

If it is, the server acts as a client processand sends the message to the real server in the corporation. If it is not, the message isdropped and an error message is sent to the external user.

In this way, the requests ofthe external users are filtered based on the contents at the application layer.
