UNIT V

COMPUTER NETWORKS: TRANSPORT LAYER

Goal of transport layer is to provide efficient, reliable, and cost-effective service to its users, normally processes in the application layer. To achieve this goal, the transport layer makes use of the services provided by the network layer.

Transport entity

The hardware and/or software within the transport layer that does the work are called the transport entity. The transport entity can be located in the operating system kernel or on the network interface card.

Types of Transport service

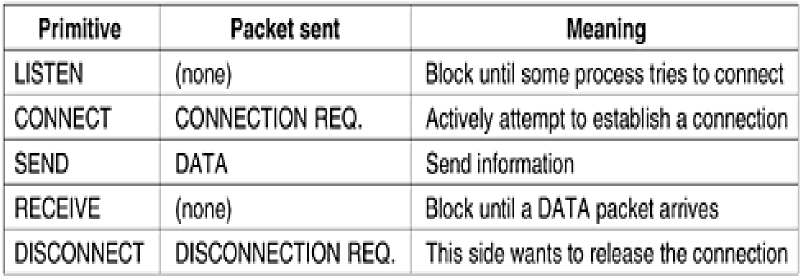
There are two types of transport service. In connection-oriented transport service is connections have three phases: establishment, data transfer, and release. The connectionless transport service is also very similar to the connectionless network service.

Need for transport layer

The users have no real control over the network layer, so they cannot solve the problem of poor service by using better routers or putting more error handling in the data link layer.

The only possibility is to put on top of the network layer another layer that improves the quality of the service. The bottom four layers can be seen as the transport service provider, whereas the upper layer(s) are the transport service user.

Transport Service Primitives



Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

ELEMENTS OF TRANSPORT PROTOCOLS

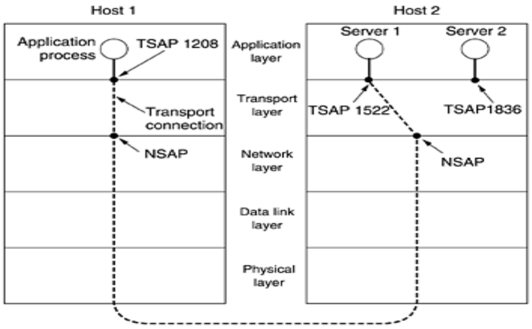
1. Addressing

When an application (e.g., a user) process wishes to set up a connection to a remote application process, it must specify which one to connect to. In the Internet, these end points are called **ports** known as **TSAP (Transport Service Access Point).**

End points in the network layer (i.e., network layer addresses) are then called NSAPs.

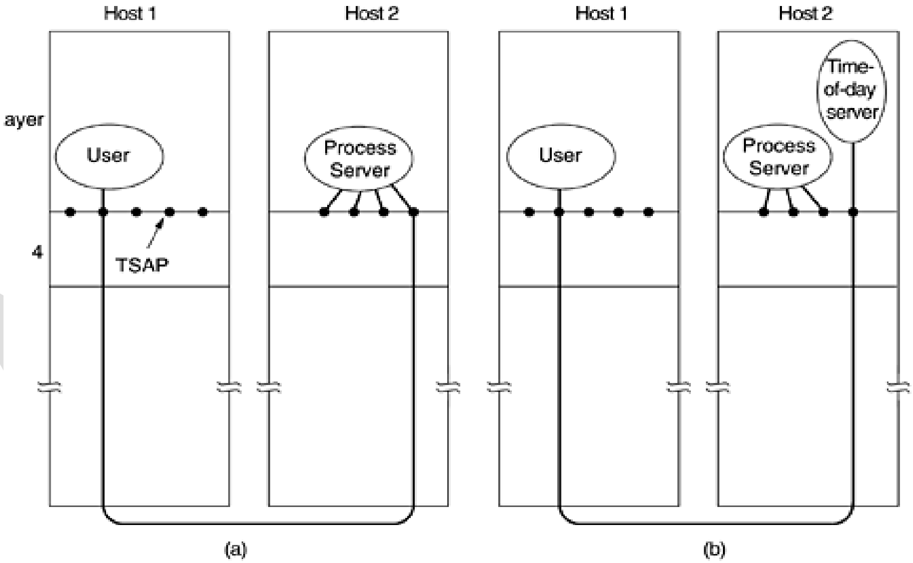
- IP addresses are examples of NSAPs.

Figure 6-8. TSAPs, NSAPs, and transport connections.



Process Server

- Instead of every server listening at a well-known TSAP, each machine to offer services to remote users has a special **process server** that acts as a proxy for less heavily used servers.
- Process server listens to a set of ports at the same time, waiting for a connection request.
- Users of a service send CONNECT request, specifying the TSAP address of the service. If no server is waiting for them, they get a connection to the process server.
- After getting request the process server initiate the requested server, allowing it to inherit the existing connection with the user.



Name Server or a Directory Server

- To find the TSAP address corresponding to a given service name, a user sets up a connection to the name server (which listens to a well-known TSAP).
- The user then sends a message specifying the service name, and the name server sends back the TSAP address.
- Then the user releases the connection with the name server and establishes a new one with the desired service.

2. Connection Establishment

To establish connection, one transport entity sends a CONNECTION REQUEST TPDU to the destination and waits for a CONNECTION ACCEPTED reply. Though it is a simple process it may introduce problems as,

1. Existence of delayed duplicates.
2. Router crash

It can be attacked in various ways, none of them very satisfactory.

1. Throw-away transport addresses

Each time a transport address is needed, a new one is generated.

When a connection is released, the address is discarded and never used again.

2. Give each connection a connection identifier

- Connection identifier is chosen by the initiating party and put in each TPDU, including the one requesting the connection.
- After each connection is released, each transport entity could update a table listing outdated connections as (peer transport entity, connection identifier) pairs.
- Whenever a connection request comes in, it could be checked against the table, to see if it belonged to a previously-released connection.

Problem

- It requires each transport entity to maintain a certain amount of history information indefinitely.
- If a machine crashes and loses its memory, it will no longer know which connection identifiers have already been used.

3. A mechanism to kill off aged packets

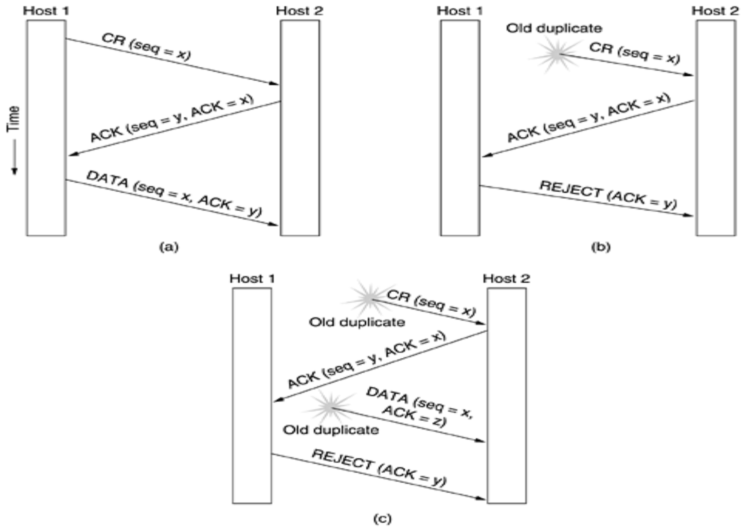
- Packet lifetime can be restricted to a known maximum using one (or more) of the following techniques:
  - Restricted subnet design.
  - Putting a hop counter in each packet.
  - Time stamping each packet.
- With packet lifetime is bounded, it is possible to devise a foolproof way to establish connections safely.

Host Crash

- A problem occurs when a host crashes. When it comes up again, its transport entity does not know where it was in the sequence space.
- One solution is to require transport entities to be idle for T sec after a recovery to let all old TPDUs die off.
- However, in a complex internetwork, T may be large, so this strategy is unattractive.
- To avoid requiring T sec of dead time after a crash, it is necessary to introduce a new restriction on the use of sequence numbers.

Three-way handshaking

Figure 6-11. Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST. (a) Normal operation. (b) Old duplicate CONNECTION REQUEST appearing out of nowhere. (c) Duplicate CONNECTION REQUEST and duplicate ACK.



Connection Release

1. Asymmetric -when one party hangs up, the connection is broken.

2. Symmetric - treats the connection as two separate unidirectional connections and requires each one to be released separately.

Figure 6-12. Abrupt disconnection with loss of data.

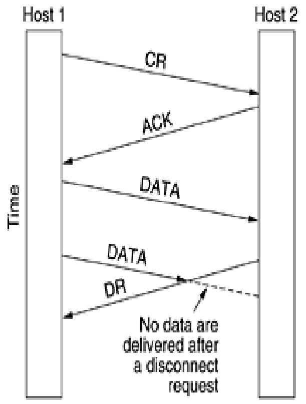
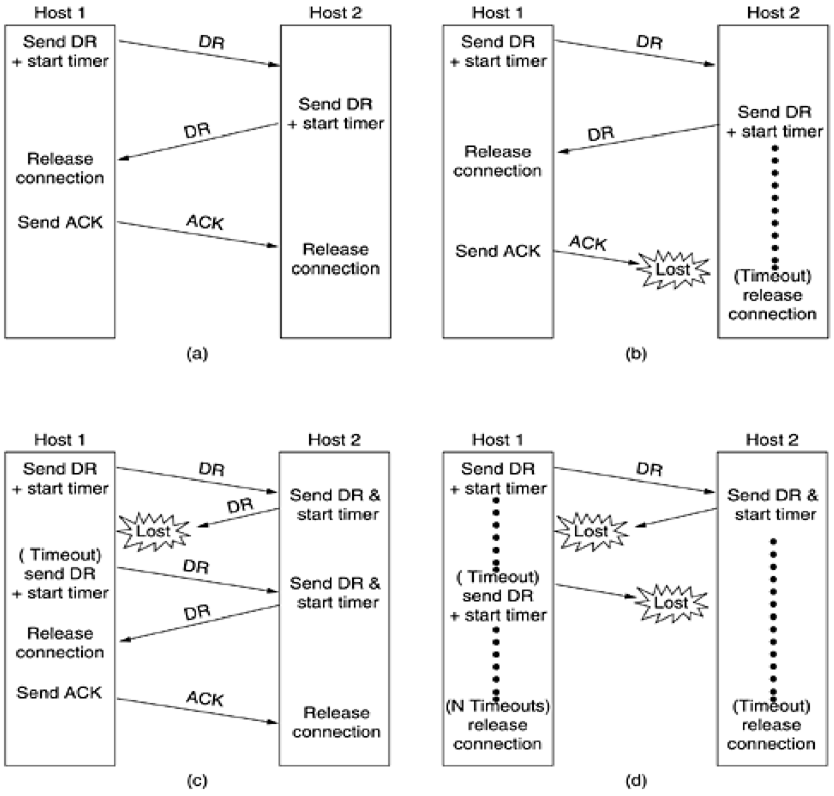


Figure 6-14. Four protocol scenarios for releasing a connection. (a) Normal case of three-way handshake. (b) Final ACK lost. (c) Response lost. (d) Response lost and subsequent DRs lost.



E-mail

The first e-mail systems simply consisted of file transfer protocols, with the convention that the first line of each message (i.e., file) contained the recipient's address.

Complaints

1. Sending a message to a **group of people** was inconvenient.
2. Messages had no internal structure, making computer processing difficult. For example, if a forwarded message was included in the body of another message, extracting the forwarded part from the received message was difficult.
3. The originator (sender) never knew if a message arrived or not.
4. If someone was planning to be away on business for several weeks and wanted all incoming e-mail to be handled by his secretary, this was not easy to arrange.
5. The user interface was poorly integrated with the transmission system requiring users first to edit a file, then leave the editor and invoke the file transfer program.
6. It was not possible to create and send messages containing a mixture of text, drawings, facsimile, and voice.

Architecture and services

E-mail system consists of two subsystems: User Agent and Message Transfer Agent

The user agent

- Allow people to read and send e-mail
- UA are local programs that provide a command-based, menu-based, or graphical method for interacting with the e-mail system.

The message transfer agent

Move the messages from the source to the destination.

MTA are typically system daemons, i.e., processes that run in the background. Their job is to move e-mail through the system.

Functions of E-mail system

- E-mail systems support five basic functions.
    - Composition

  - Transfer
  - Reporting
  - Displaying
  - Disposition

Composition

- Creating messages and answers.
- Any text editor can be used for the body of the message, the system itself can provide assistance with addressing and the numerous header fields attached to each message.
- For example, when answering a message, the e-mail system can extract the originator's address from the incoming e-mail and automatically insert it into the proper place in the reply.

Transfer

- Moving messages from the originator to the recipient
- This requires establishing a connection to the destination or some intermediate machine, outputting the message, and releasing the connection.
- The e-mail system should do this automatically, without bothering the user.

Reporting

- Telling the originator what happened to the message. Was it delivered? Was it rejected? Was it lost? Numerous applications exist in which confirmation of delivery is important and may even have legal significance.

Displaying

- Incoming messages is displayed so people can read their e-mail.
- Sometimes conversion is required or a special viewer must be invoked.
- Simple conversions and formatting are sometimes attempted as well.

Disposition

- Is concerned what the recipient does with the message after receiving it.
- Possibilities include throwing it away before reading, throwing it away after reading, saving it, and so on. It should also be possible to retrieve and reread saved messages, forward them, or process them in other ways.

Additional features

- When people move or when they are away for some period of time, they may want their e-mail forwarded, so the system should be able to do this automatically.
- Most systems allow users to create mailboxes to store incoming e-mail.
- Mailing list, which is a list of e-mail addresses. When a message is sent to the mailing list, identical copies are delivered to everyone on the list.
- Carbon copies, blind carbon copies, high-priority e-mail, secret (i.e., encrypted) e-mail, alternative recipients if the primary one is not currently available, and the ability for secretaries to read and answer their bosses' e-mail.

USER AGENT

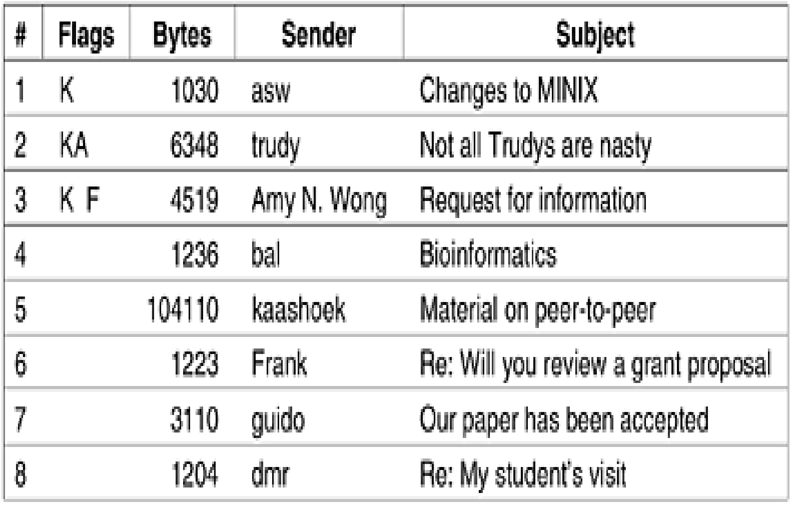
- A user agent is normally a program (sometimes called a mail reader) that accepts a variety of commands for composing, receiving, and replying to messages, as well as for manipulating mailboxes.
- Some user agents have a fancy menu- or icon-driven interface that requires a mouse, whereas others expect 1-character commands from the keyboard.

Sending E-mail

- A user must provide the message, the destination address, and possibly some other parameters.
- The destination address must be in a format that the user agent can deal with. Many user agents expect addresses of the form user@dns-address.
- Most e-mail systems support mailing lists, so that a user can send the same message to a list of people with a single command.

Reading E-mail

- When a user agent is started up, it looks at the user's mailbox for incoming e-mail before displaying anything on the screen. Each line of the display contains several fields extracted from the envelope or header of the corresponding message.
- In a simple e-mail system, the choice of fields displayed is built into the program.



#	Flags	Bytes	Sender	Subject
1	K	1030	asw	Changes to MINIX
2	KA	6348	trudy	Not all Trudys are nasty
3	K F	4519	Amy N. Wong	Request for information
4		1236	bal	Bioinformatics
5		104110	kaashoek	Material on peer-to-peer
6		1223	Frank	Re: Will you review a grant proposal
7		3110	guido	Our paper has been accepted
8		1204	dmr	Re: My student's visit

Field1: message number.

Field 2: Flags

K - The message is not new but was read previously and kept in the mailbox

A- The message has already been answered;

F - The message has been forwarded to someone else. (Other flags are also possible).

Field 3 - how long the message is,

Field 4: tells who sent the message.

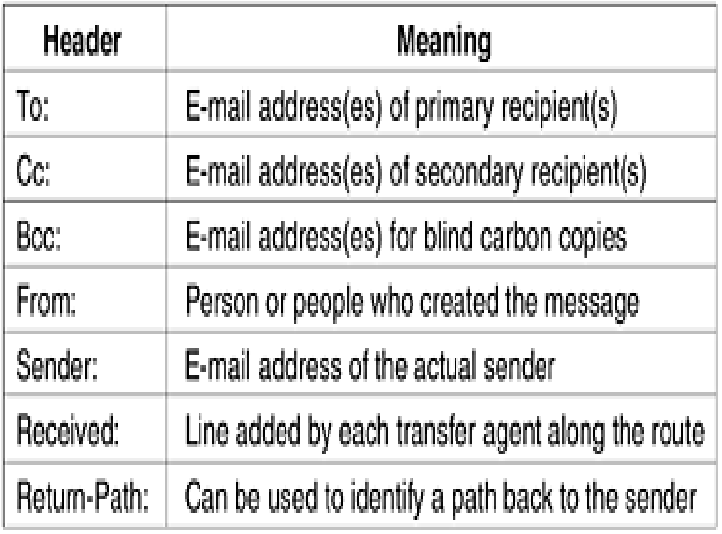
Field 5: Subject field gives a brief summary of what the message is about.

People who fail to include a Subject field often discover that responses to their e-mail tend not to get the highest priority.

After the headers have been displayed, the user can perform any of several actions, such as displaying a message, deleting a message, and so on. Usually, the user selects a message with the mouse and then clicks on an icon to type, answer, delete, or forward it.

Message Formats

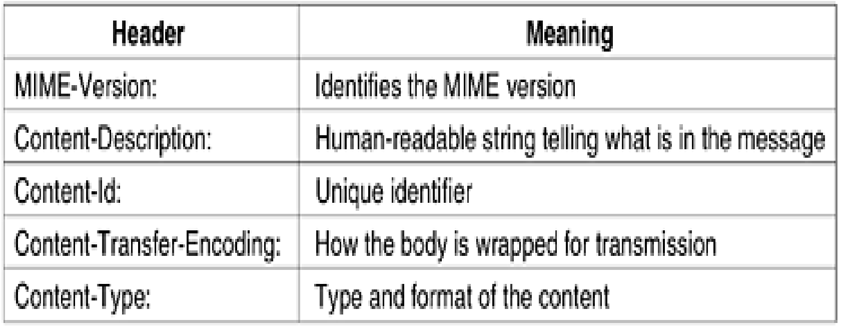
RFC 822 header fields related to message transport.



Header	Meaning
To:	E-mail address(es) of primary recipient(s)
Cc:	E-mail address(es) of secondary recipient(s)
Bcc:	E-mail address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender

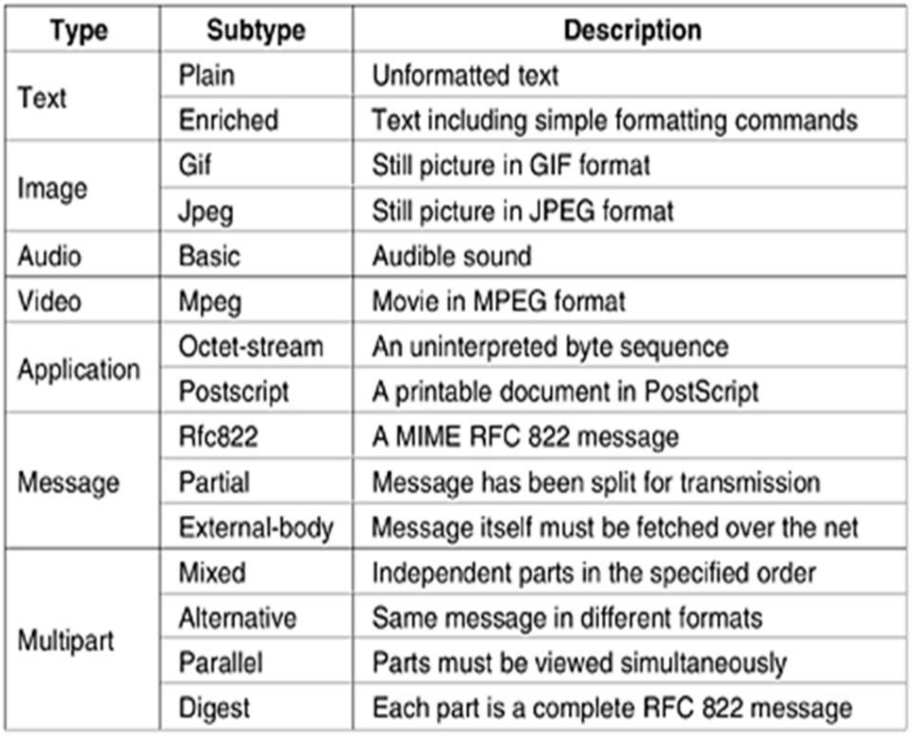
MIME (Multipurpose internet mail extension)

- In the early days of the ARPANET, e-mail consisted exclusively of text messages written in English and expressed in ASCII. For this environment, RFC 822 did the job completely: it specified the headers but left the content entirely up to the users.
- Nowadays, on the worldwide Internet, this approach is no longer adequate. The problems include sending and receiving messages not containing text at all (e.g., audio or images). The solution, called MIME (Multipurpose Internet Mail Extensions) is now widely used.
- The basic idea of MIME is to continue to use the RFC 822 format, but to add structure to the message body and define encoding rules for non-ASCII messages.
- By not deviating from RFC 822, MIME messages can be sent using the existing mail programs and protocols.
- All that has to be changed are the sending and receiving programs, which users can do for themselves.



Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-Id:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content

RFC 822 headers added by MIME.



Type	Subtype	Description
Text	Plain	Unformatted text
	Enriched	Text including simple formatting commands
Image	Gif	Still picture in GIF format
	Jpeg	Still picture in JPEG format
Audio	Basic	Audible sound
Video	Mpeg	Movie in MPEG format
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in PostScript
Message	Rfc822	A MIME RFC 822 message
	Partial	Message has been split for transmission
	External-body	Message itself must be fetched over the net
Multipart	Mixed	Independent parts in the specified order
	Alternative	Same message in different formats
	Parallel	Parts must be viewed simultaneously
	Digest	Each part is a complete RFC 822 message

The MIME types and subtypes defined in RFC 2045.

MESSAGE TRANSFER SYSTEM

The message transfer system is concerned with relaying messages from the originator to the recipient.

The simplest way to do this is to establish a transport connection from the source machine to the destination machine and then just transfer the message.

SMTP—The Simple Mail Transfer Protocol

- SMTP is a simple ASCII protocol.
- Within the Internet, e-mail is delivered by having the source machine establish a TCP connection to port 25 of the destination machine.
- Listening to this port is an e-mail daemon that speaks SMTP (Simple Mail Transfer Protocol).
- This daemon accepts incoming connections and copies messages from them into the appropriate mailboxes.

Note:

- Daemon: A program or process that sits idly in the background until it is invoked to perform its task.
- If a message cannot be delivered, an error report containing the first part of the undeliverable message is returned to the sender.
- After establishing the TCP connection to port 25, the sending machine, operating as the client, waits for the receiving machine, operating as the server, to talk first.
- The server starts by sending a line of text giving its identity and telling whether it is prepared to receive mail.
- If it is not, the client releases the connection and tries again later.
- If the server is willing to accept e-mail, the client announces whom the e-mail is coming from and whom it is going to.
- If such a recipient exists at the destination, the server gives the client the go-ahead to send the message.
- client sends the message and the server acknowledges it. No checksums are needed because TCP provides a reliable byte stream.
- When all the e-mail has been exchanged in both directions, the connection is released.

Problems of SMTP

- Message length: Some older implementations cannot handle messages exceeding 64 KB.
- Timeouts: If the client and server have different timeouts, one of them may give up while the other is still busy, unexpectedly terminating the connection.
- Infinite mail-storms

  if host 1 holds mailing list A and host 2 holds mailing list B and each list contains an entry for the other one, then a message sent to either list could generate a never-ending amount of e-mail traffic unless somebody checks for it.
- To get around some of these problems, extended SMTP (ESMTP) has been defined in RFC 2821.
- Clients wanting to use it should send an *EHLO message instead of HELO initially.*
- *If this is rejected, then the server is a regular SMTP server, and the client should proceed in the usual way.*
- *If the EHLO is accepted, then new commands and parameters are allowed.*

FINAL DELIVERY

E-mail is delivered by having the sender establish a TCP connection to the receiver and then ship the e-mail over it.

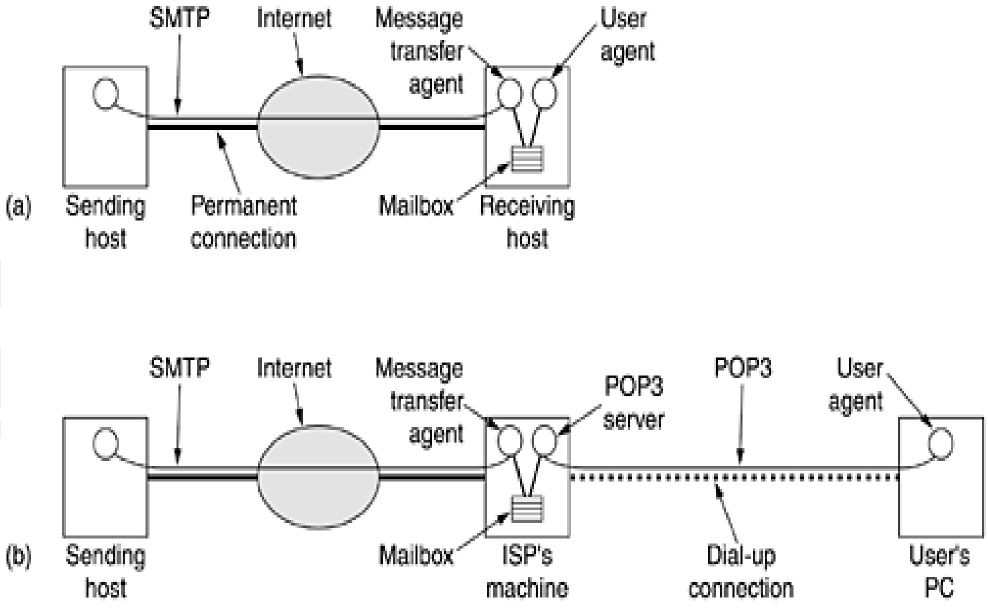
This model worked fine for decades when all Internet hosts were on-line all the time to accept TCP connections.

- When people who access the Internet by calling their ISP over a modem, it breaks down.
- One solution is to have a message transfer agent on an ISP machine accept e-mail for its customers and store it in their mailboxes on an ISP machine.
- Since this agent can be on-line all the time, e-mail can be sent to it 24 hours a day.

POP3

This protocol allows user transfer agents (on client PCs) to contact the message transfer agent (on the ISP's machine) and allow e-mail to be copied from the ISP to the user.

One such protocol is POP3 (Post Office Protocol Version 3), which is described in RFC 1939.



- POP3 begins when the user starts the mail reader. The mail reader calls up the ISP and establishes a TCP connection with the message transfer agent at port 110. Once the connection has been established, the POP3 protocol goes through three states in sequence:

1. Authorization.
2. Transactions.
3. Update.

Authorization - deals with having the user log in.

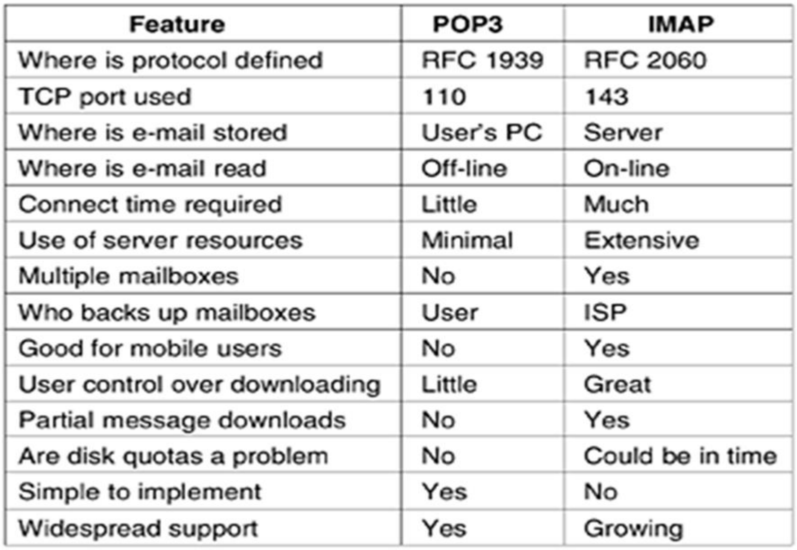
Transaction - deals with the user collecting the e-mails and marking them for deletion from the mailbox.

The update - causes the e-mails to be deleted.

IMAP (Internet Mail Access Protocol)

- The Internet Message Access Protocol is an Application Layer Internet protocol that allows an e-mail client to access e-mail on a remote mail server.
- IMAP supports both on-line and off-line modes of operation.
- E-mail clients using IMAP generally leave messages on the server until the user explicitly deletes them. This and other characteristics of IMAP operation allow multiple clients to manage the same mailbox.
- Most e-mail *clients* support IMAP in addition to Post Office Protocol (POP) to retrieve messages; however, fewer e-mail *services* support IMAP. Clients may store local copies of the messages, but these are considered to be a temporary cache.

- IMAP assumes that all the e-mail will remain on the server indefinitely in multiple mailboxes.
- IMAP provides extensive mechanisms for reading messages or even parts of messages, a feature useful when using a slow modem to read the text part of a multipart message with large audio and video attachments.
- Since the working assumption is that messages will not be transferred to the user's computer for permanent storage, IMAP provides mechanisms for creating, destroying, and manipulating multiple mailboxes on the server.



Feature	POP3	IMAP
Where is protocol defined	RFC 1939	RFC 2060
TCP port used	110	143
Where is e-mail stored	User's PC	Server
Where is e-mail read	Off-line	On-line
Connect time required	Little	Much
Use of server resources	Minimal	Extensive
Multiple mailboxes	No	Yes
Who backs up mailboxes	User	ISP
Good for mobile users	No	Yes
User control over downloading	Little	Great
Partial message downloads	No	Yes
Are disk quotas a problem	No	Could be in time
Simple to implement	Yes	No
Widespread support	Yes	Growing

Difference between pop3 and IMAP

DOMAIN NAME SYSTEM

- DNS is a globally distributed, scalable and reliable database. DNS does the address conversion function i.e., *domain name system* is usually used to translate a host name into an IP address.

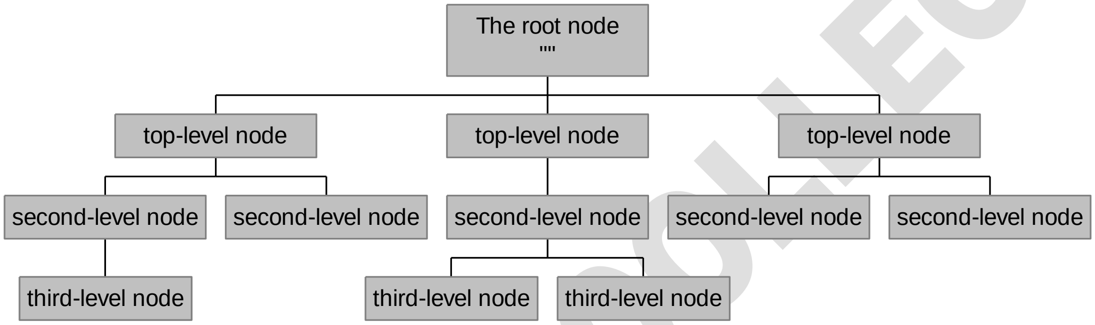
- Host names
  - Mnemonic name appreciated by humans
  - Variable length, full alphabet of characters
  - Provide little (if any) information about location
  - Examples: www.cnn.com and bbc.co.uk
- IP addresses
  - Numerical address appreciated by routers
  - Fixed length, binary number
  - Hierarchical, related to host location
  - Examples: 64.236.16.20 and 212.58.224.131

- DNS has three components
  - A "name space"
  - Servers making that name space available

- Resolvers (clients) which query the servers about the name space

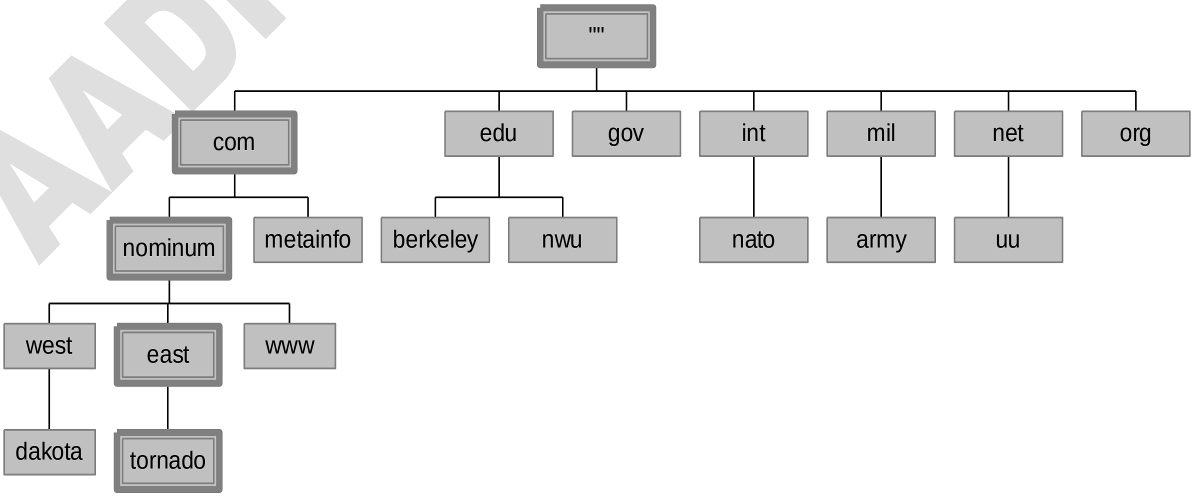
Namespace

The *name space* is the structure of the DNS database. An inverted tree with the root node at the top and each node has a label



Domain name

- A *domain name* is the sequence of labels from a node to the root, separated by dots ("."s), read left to right
  - The name space has a maximum depth of 127 levels
  - Domain names are limited to 255 characters in length
  - A node's domain name identifies its position in the name space



Name servers

Name servers store information about the name space in units called “zones”. The name servers that load a complete zone are said to “have authority for” or “be authoritative for” the zone. Usually, more than one name servers are authoritative for the same zone. Also, a single name server may be authoritative for many zones

Resolvers

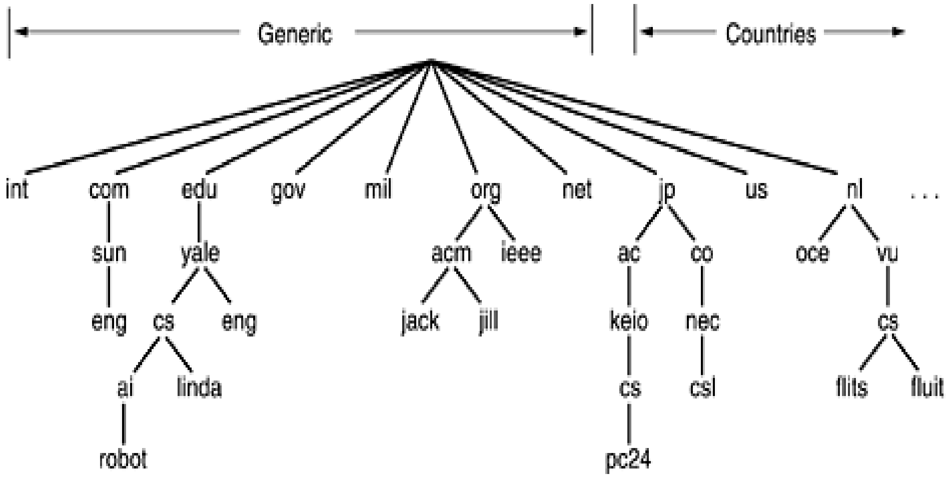
- *Name resolution* is the process by which resolvers and name servers cooperate to find data in the name space

Properties of DNS

  - Hierarchical name space divided into zones
  - Zones distributed over collection of DNS servers

- Hierarchy of DNS servers
  - Root (hardwired into other servers)
  - Top-level domain (TLD) servers
  - Authoritative DNS servers
- Performing the translations
  - Local DNS servers
  - Resolver software

Figure 7-1. A portion of the Internet domain name space.



While DNS is extremely important to the correct functioning of the Internet, all it really does is map symbolic names for machines onto their IP addresses. It does not help locate people, resources, services, or objects in general. For locating these things, another directory service has been defined, called LDAP (Lightweight Directory Access Protocol).

World Wide Web

World Wide Web is the very popular service of internet. It can be simply referred as www or just web or w3. World Wide Web is a system of interlinked, hypertext documents accessed via the Internet. With a web browser, a user views web pages or web documents which may contain text, images, videos and graphics. Web pages are linked together using hyperlinks.

The idea of the web was conceived by Tim Berners-Lee at CERN (European particle physics laboratory) in 1989. He developed three standards that made the web possible. They are,

- **HTTP (Hypertext transfer protocol)**

  It specifies that how the browser and server communicate with each other.

- **HTML (Hypertext Markup Language)**

  These define the structure and interpretation of hypertext documents.

- **URL(Uniform Resource Locator)**

  It described a way to give each document a unique address so that the can be found out easily.

Note:

A **hyperlink**, is a reference or navigation element in a document to another section of the same document or to another document that may be on a (different) website.

A **website** (alternatively, **web site** or **Web site**) is a collection of Web pages

Hypertext most often refers to text on a computer that will lead the user to other, related information on demand.