CSE-3101

Computer Networking

Mala Rani Barman
Assistant Professor, Department of Computer Science and Engineering
Sheikh Hasina University

Application layer

- ✔The application layer is responsible for providing services to the user. We are familiar with several network applications : Web, e-mail, DNS, remote file access and transfer etc.

- ✔The application layer is where network applications and their application-layer protocols reside.

- ✔To understand the difference between an application layer protocol and an application, think of all the different World Wide Web browsers that are available (Firefox, Safari, Internet Explorer etc.). The reason that you can use any one of these application programs to access a particular site on the Web is because they all conform to the same application layer protocol: HTTP (Hyper Text Transport Protocol).

The application layer determines how a specific user application (for example e-mail) should use a network. When a new application is developed, its software must be able to run on multiple machines, so it does not need to be rewritten for networking devices, such as routers, that function at the network layer.

This lecture will deal with:

- ✔**Domain Name System (DNS)**
- ✔**SMTP-The Simple Mail Transfer Protocol**
- ✔**The World Wide Web**
- ✔**URLs-Uniform Resource Locators**
- ✔**Web Caching (Proxy Server)**
- ✔**Telnet and Secure Shell**
- ✔**File Transfer Protocol (FTP)**
- ✔**Network Management**

Domain Name System (DNS)

- Although programs theoretically could refer to hosts, mailboxes, and other resources by their network (e.g., IP) addresses, these addresses are hard for people to remember. Nevertheless, the network itself understands only numerical addresses, so some mechanism is required to convert the ASCII strings to network addresses.
- DNS is a distributed hierarchical and global directory that translates machine or domain names to numerical IP address. DNS can be thought as a distributed database system used to map host names to IP address and vice-versa.

A **distributed database** is a database in which storage devices are not all attached to a common processing unit such as the CPU, and which is controlled by a distributed database management system (together sometimes called a distributed database system). It may be stored in multiple computers, located in the same physical location; or may be dispersed over a network of interconnected computers.

✔DNS is an application layer protocol, and every Internet service provider has a DNS server. In the normal mode of operation, a hosts send UDP (connection less) queries to a DNS server. The DNS server either replies or directs the queries to another server.

The DNS Name Space

- In theory at least, a single name server could contain the entire DNS database and respond to all queries about it. In practice, this server would be so overloaded as to be useless.
- To avoid the problems associated with having only a single source of information, the DNS name space is divided into non-overlapping **zones**.
- Conceptually, the Internet is divided into over 200 top-level domains, where each domain covers many hosts. Each domain is partitioned into sub-domains, and these are further partitioned, and so on.
- All these domains can be represented by a tree, as shown in fig.1 of next slide. The leaves of the tree represent domains that have no sub-domains (but do contain machines, of course). A leaf domain may contain a single host, or it may represent a company and contain thousands of hosts.

The top-level domains come in two flavors: generic and countries. The original generic domains were *com* (commercial), *edu* (educational institutions), *gov* (the U.S. Federal Government), *int* (certain international organizations), *mil* (the U.S. armed forces), *net* (network providers), and *org* (nonprofit organizations). The country domains include one entry for every country

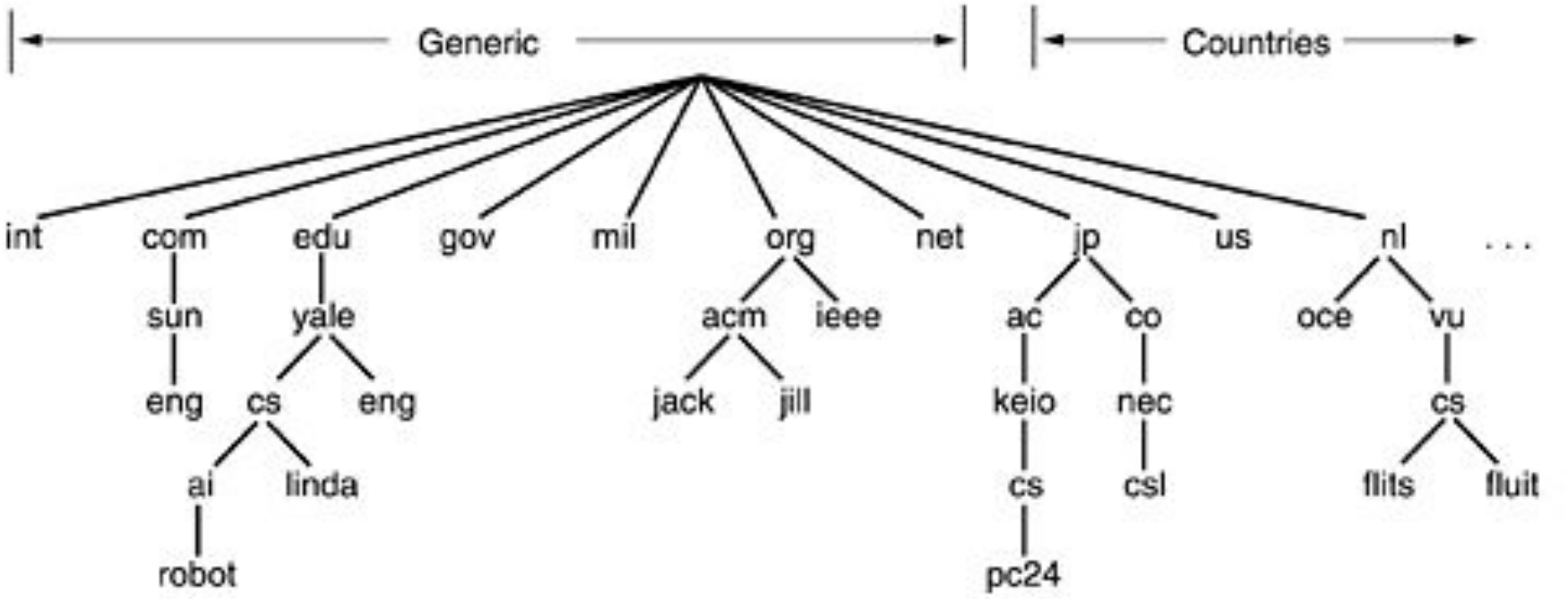


Fig.1 A portion of the Internet domain name space

Top-Level Internet Domains

Domain	Contents
com	Commercial organizations
edu	Educational institutions
gov	U.S. federal government agencies
mil	U.S. military
net	Network support centers, Internet service providers, and other network-related organizations
org	Nonprofit organizations
us	U.S. state and local government agencies, schools, libraries, and museums
country code	ISO standard 2-letter identifier for country-specific domains (e.g., au, ca, uk)
biz	Dedicated exclusively for private businesses
info	Unrestricted use
name	Individuals, for email addresses and personalized domain names.
museum	restricted to museums, museum organizations, and individual members of the museum profession
coop	Member-owned cooperative organizations, such as credit unions
aero	Aviation community
pro	Medical, legal, and accounting professions
arpa	Temporary ARPA domain (still used)
int	International organizations

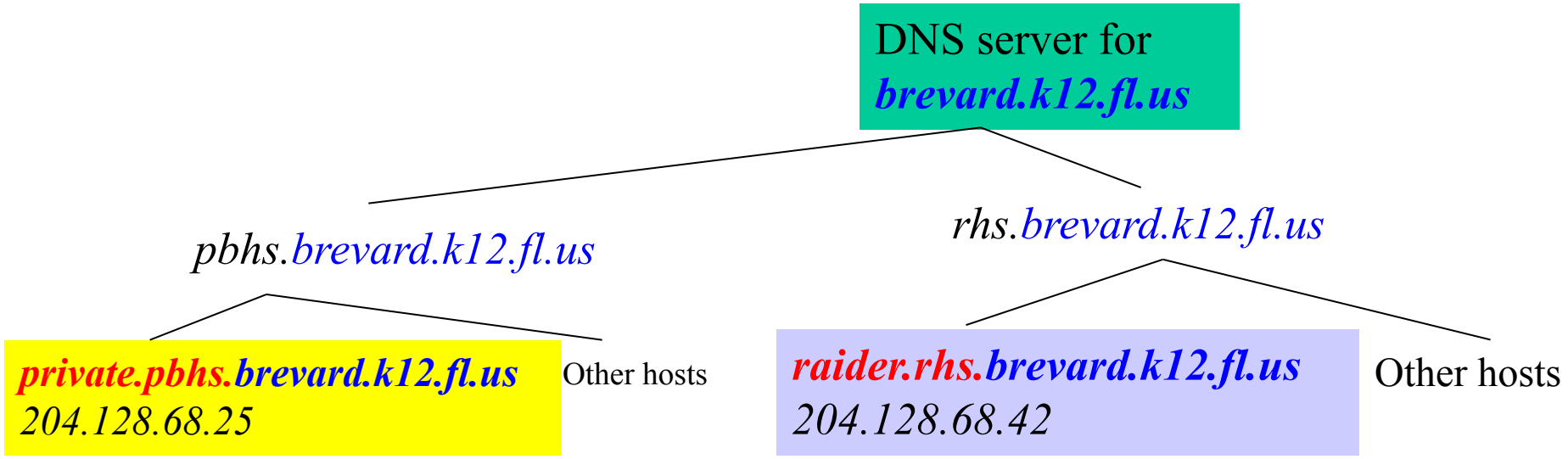
Example-1

The domain name *cs.purdue.edu* contains three labels: *cs, purdue* and *edu*. It gives the domain name for Computer Science Department at Purdue University. Again *purdue.edu* gives the domain name of Purdue University.

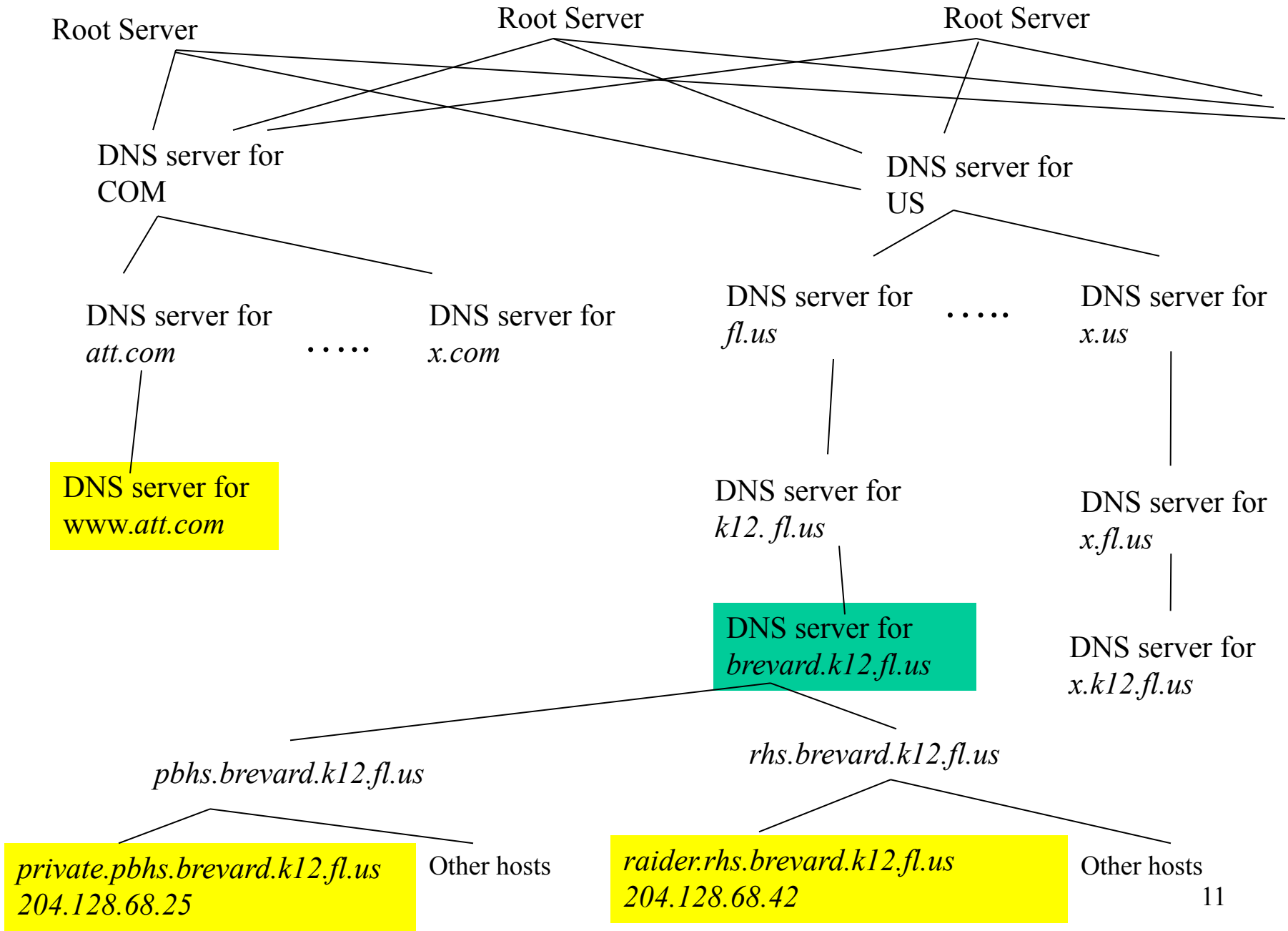
The domain names are written with the local label first and the top domain last.

Example-2

Let the host *private* (whose machine/domain name is *private.pbhs.brevard.k12.fl.us* and IP address is *204.128.68.25*) wants to connect the Web server *raider*, which is in the *rhs.brevard.k12.fl.us* domain. The web browser on *private* places a DNS query to the DNS server for *brevard.k12.fl.us*, which is authorized for the *brevard.k12.fl.us* domain. This DNS server looks up the information in its database and returns the IP address.



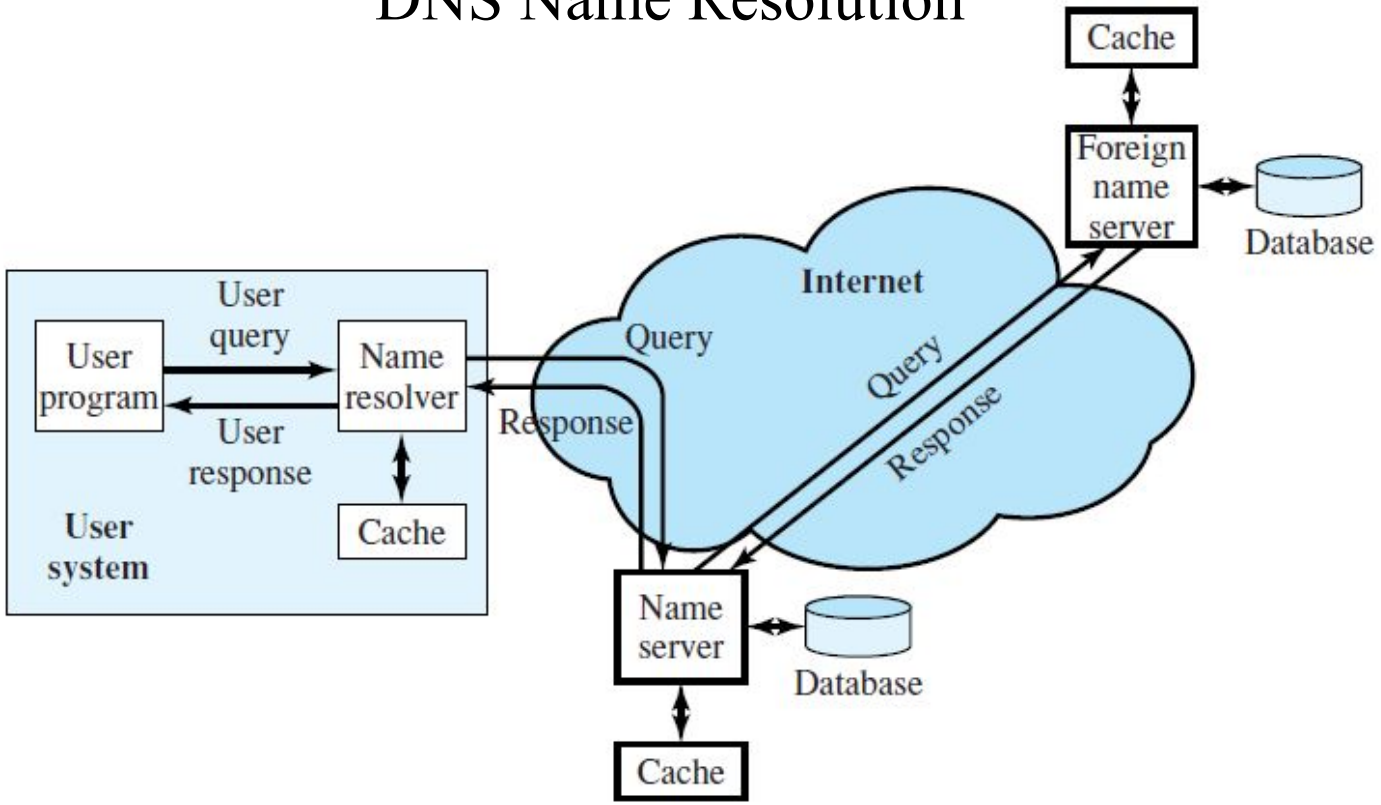
A **Web server** is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files that form **Web** pages to users, in response to their requests, which are forwarded by their computers' HTTP clients. Dedicated computers and appliances may be referred to as **Web servers** as well.


Root Server
Root Server
Root Server
DNS server for COM
DNS server for US
DNS server for att.com
DNS server for x.com
DNS server for fl.us
DNS server for x.us
DNS server for www.att.com
DNS server for k12. fl.us
DNS server for x.fl.us
DNS server for brevard.k12.fl.us
DNS server for x.k12.fl.us
pbhs.brevard.k12.fl.us
rhs.brevard.k12.fl.us
private.pbhs.brevard.k12.fl.us 204.128.68.25
Other hosts
raider.rhs.brevard.k12.fl.us 204.128.68.42
Other hosts

✔Now suppose that user wants to connect the web server *www.att.com*. A similar query is made to the *brevard.k12.fl.us* DNS server. This time it does not have any information about *www.att.com* domain and hence cannot resolve the name. Each DNS server has the address of other name server including at least one root server.

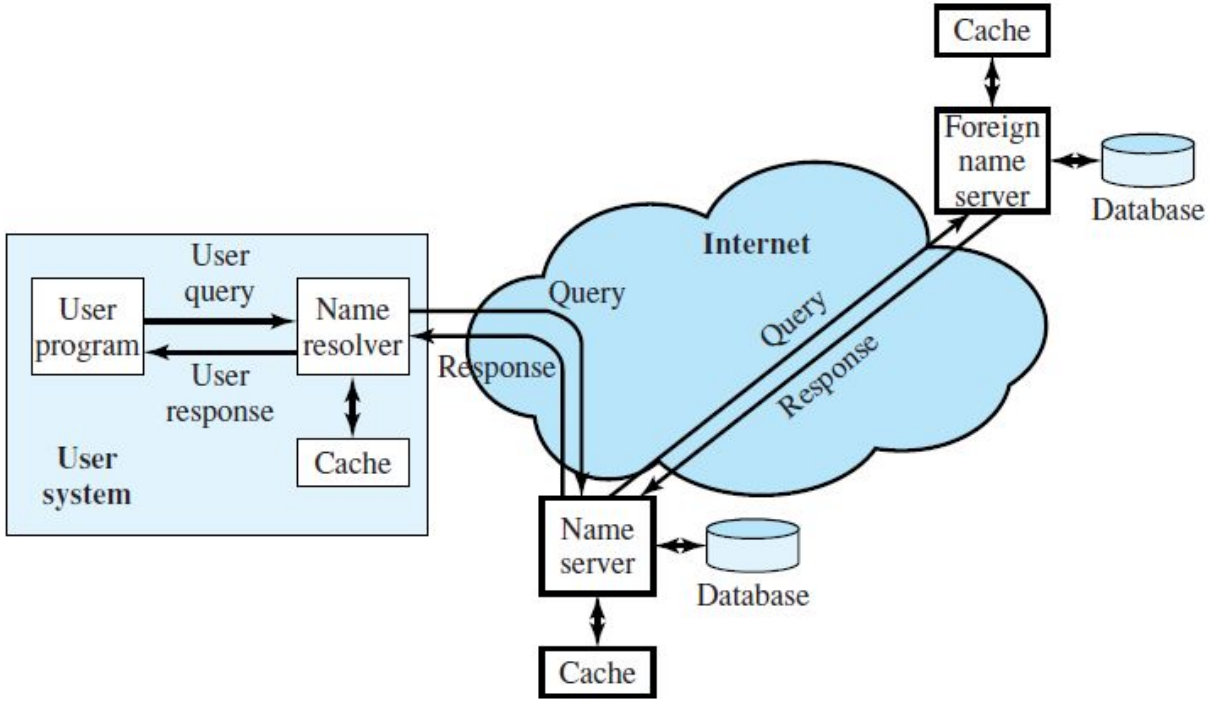
✔If DNS server can not resolve the name it replies by specifying the name server that should be connected next. Eventually a root server gets involved. Root server maintain information about all the authorative name server for each top level domain. A root server will eventually provide it with the *att.com* DNS server's address. The *brevard* name server then contacts the *att.com* name server, which will return the address 192.20.3.54

DNS Name Resolution



DNS operation typically includes the following steps (Figure above):

1. A user program requests an IP address for a domain name.
2. A resolver module in the local host or local ISP formulates a query for a local name server in the same domain as the resolver.
3. The local name server checks to see if the name is in its local database or cache, and, if so, returns the IP address to the requestor. Otherwise, the name server queries other available name servers, starting down from the root of the DNS tree or as high up the tree as possible.



4. When a response is received at the local name server, it stores the name/address mapping in its local cache and may maintain this entry for the amount of time specified in the time to live field of the retrieved RR (*resource records*).
5. The user program is given the IP address or an error message.

Resource Records

Every domain, whether it is a single host or a top-level domain, can have a set of **resource records** associated with it**.** For a single host, the most common resource record is just its IP address, but many other kinds of resource records also exist. The primary function of DNS is to map domain names onto resource records.

A resource record is a five-tuple:
Domain_name Time_to_live Class Type Value

The *Domain_name* tells the domain to which this record applies.

The *Time_to_live* field gives an indication of how stable the record is.

The third field of every resource record is the *Class*. For Internet information, it is always *IN*. For non-Internet information, other codes can be used, but in practice, these are rarely seen.

The *Type* field tells what kind of record this is. The most important types are listed in Table. below.

The principal DNS resource record types for IPv4.

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

Example-3 What do you mean by record:

ait.ac.th 86400 IN MX kuddus.ait.ac.th

MX record specifies the name of the host prepared to accept e-mail for the specified domain. It is used because not every machine is prepared to accept e-mail. If someone wants to send e-mail to, for example, *bill@ait.ac.th*, the sending host needs to find a mail server at *ait.ac.th* that is willing to accept e-mail. The *MX* record can provide this information.

ait.ac.th 86400 IN MX kuddus.ait.ac.th

*T*he mail *bill@ait.ac.th* will be accepted by the machine with DNS of *kuddus.ait.ac.th*

Example-4

CNAME records allow aliases to be created. For example, a person familiar with Internet naming in general and wanting to send a message to someone whose login name is *paul* in the computer science department at M.I.T. might guess that *paul@cs.mit.edu* will work. Actually, this address will not work, because the domain for M.I.T.'s computer science department is *lcs.mit.edu*. However, as a service to people who do not know this, M.I.T. could create a *CNAME* entry to point people and programs in the right direction. An entry like this one might do the job:

```
cs.mit.edu    86400    IN    CNAME    lcs.mit.edu
```

```
; Authoritative data for cs.vu.nl
cs.vu.nl.        86400   IN  SOA     star boss (9527,7200,7200,241920,86400)
cs.vu.nl.        86400   IN  TXT     "Divisie Wiskunde en Informatica."
cs.vu.nl.        86400   IN  TXT     "Vrije Universiteit Amsterdam."
cs.vu.nl.        86400   IN  MX      1 zephyr.cs.vu.nl.
cs.vu.nl.        86400   IN  MX      2 top.cs.vu.nl.

flits.cs.vu.nl.  86400   IN  HINFO   Sun Unix
flits.cs.vu.nl.  86400   IN  A       130.37.16.112
flits.cs.vu.nl.  86400   IN  A       192.31.231.165
flits.cs.vu.nl.  86400   IN  MX      1 flits.cs.vu.nl.
flits.cs.vu.nl.  86400   IN  MX      2 zephyr.cs.vu.nl.
flits.cs.vu.nl.  86400   IN  MX      3 top.cs.vu.nl.
www.cs.vu.nl.    86400   IN  CNAME   star.cs.vu.nl
ftp.cs.vu.nl.    86400   IN  CNAME   zephyr.cs.vu.nl

rowboat                  IN  A       130.37.56.201
                         IN  MX      1 rowboat
                         IN  MX      2 zephyr
                         IN  HINFO   Sun Unix

little-sister            IN  A       130.37.62.23
                         IN  HINFO   Mac MacOS

laserjet                 IN  A       192.31.231.216
                         IN  HINFO   "HP Laserjet IIISi" Proprietary
```

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

Domain_name **Time_to_live** **Class** **Type** **Value**

A portion of a possible DNS database for cs.vu.nl

```
; Authoritative data for cs.vu.nl
cs.vu.nl.       86400   IN  SOA    star boss (9527,7200,7200,241920,86400)
cs.vu.nl.       86400   IN  TXT    "Divisie Wiskunde en Informatica."
cs.vu.nl.       86400   IN  TXT    "Vrije Universiteit Amsterdam."
cs.vu.nl.       86400   IN  MX     1 zephyr.cs.vu.nl.
cs.vu.nl.       86400   IN  MX     2 top.cs.vu.nl.
```

Two entries giving the first and second places to try to deliver e-mail sent to *person@cs.vu.nl*. The zephyr (a specific machine) should be tried first. If that fails, the top should be tried as the next choice.

The first noncomment line gives some basic information about the domain, which will not concern us further. The next two lines give textual information about where the domain is located.

flits.cs.vu.nl.	86400	IN	HINFO	Sun Unix
flits.cs.vu.nl.	86400	IN	A	130.37.16.112
flits.cs.vu.nl.	86400	IN	A	192.31.231.165
flits.cs.vu.nl.	86400	IN	MX	1 flits.cs.vu.nl.
flits.cs.vu.nl.	86400	IN	MX	2 zephyr.cs.vu.nl.
flits.cs.vu.nl.	86400	IN	MX	3 top.cs.vu.nl.
www.cs.vu.nl.	86400	IN	CNAME	star.cs.vu.nl
ftp.cs.vu.nl.	86400	IN	CNAME	zephyr.cs.vu.nl

The *flits* is a Sun workstation running UNIX and giving both of its IP addresses. Then three choices are given for handling e-mail sent to flits.cs.vu.nl. First choice is naturally the flits itself, but if it is down, the zephyr and top are the second and third choices..

Next comes an alias, www.cs.vu.nl, so that this address can be used without designating a specific machine. Creating this alias allows cs.vu.nl to change its World Wide Web server without invalidating the address people use to get to it. A similar argument holds for ftp.cs.vu.nl

```
rowboat          IN  A       130.37.56.201
                 IN  MX      1 rowboat
                 IN  MX      2 zephyr
                 IN  HINFO   Sun Unix

little-sister    IN  A       130.37.62.23
                 IN  HINFO   Mac MacOS

laserjet         IN  A       192.31.231.216
                 IN  HINFO   "HP Laserjet IIISi" Proprietary
```

The next four lines contain a typical entry for a workstation, in this case, *rowboat.cs.vu.nl.* The information provided contains the IP address, the primary and secondary mail drops, and information about the machine. Then comes an entry for a non-UNIX system that is not capable of receiving mail itself, followed by an entry for a laser printer that is connected to the Internet.

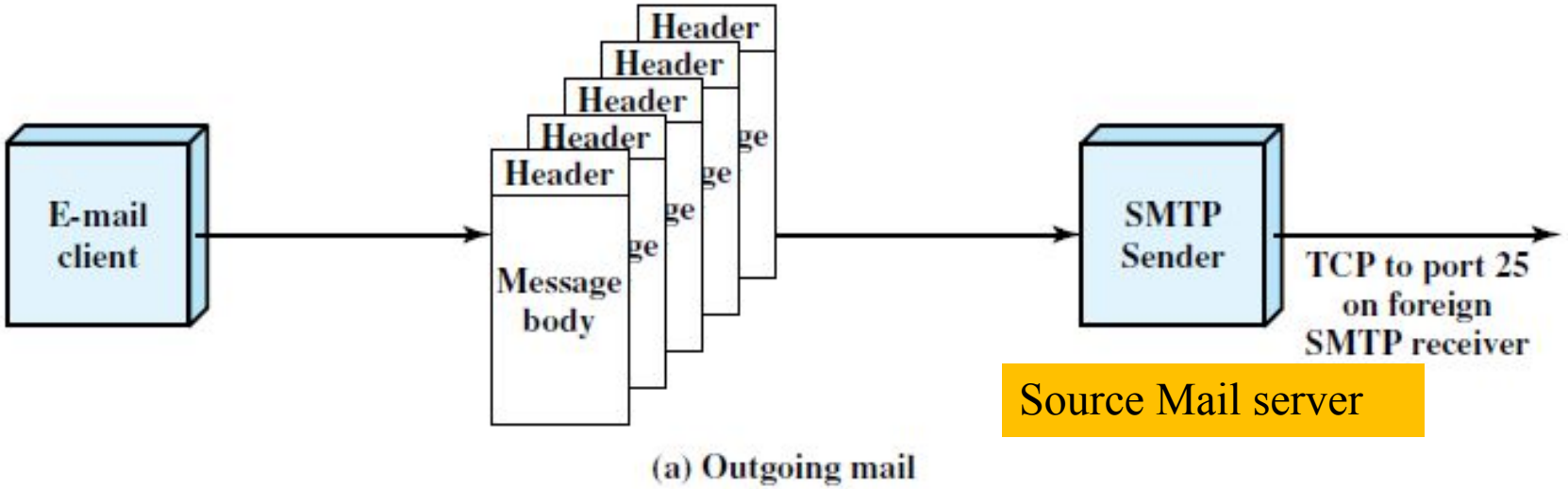
SMTP-The Simple Mail Transfer Protocol

✔Electronic mail, or e-mail, as it is known to its many fans, has been around for over two decades. Before 1990, it was mostly used in academia.

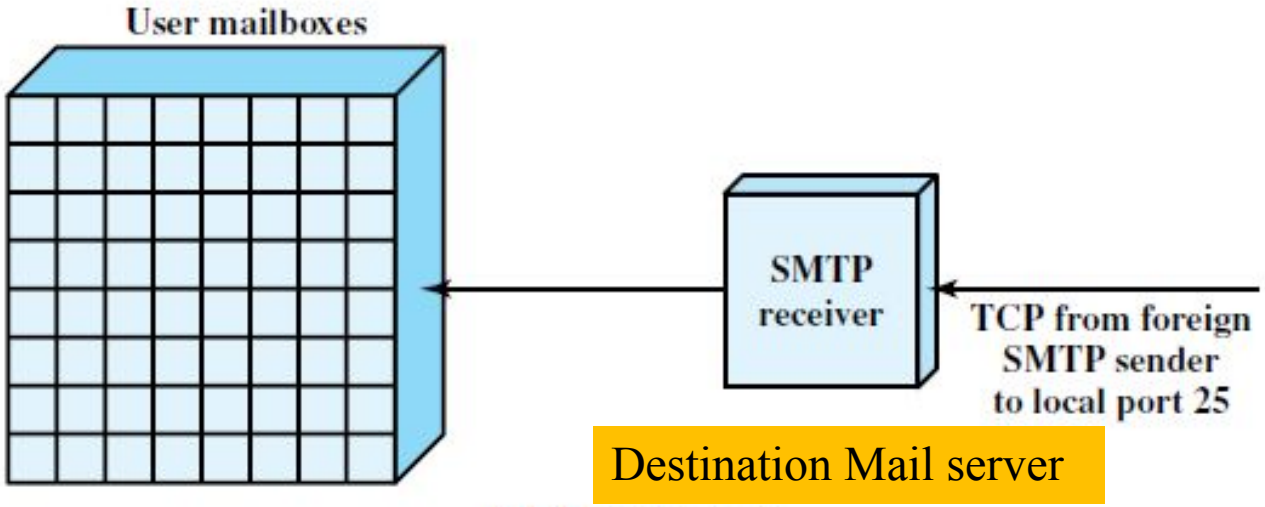
✔The **Simple Mail Transfer Protocol (SMTP)** provides a basic electronic mail transport facility. Each created message consists of a header that includes the recipient's e-mail address and other information, and a body containing the message to be sent.

✔This protocol transfers e-mail from the mail server of a source to the mail servers of destinations through TCP link. SMTP is older than HTTP, the web communication protocol, and impose certain restrictions, such as limits on the size of the e-mail contents.

Figure below illustrates the overall flow of mail in a typical system.



(a) Outgoing mail



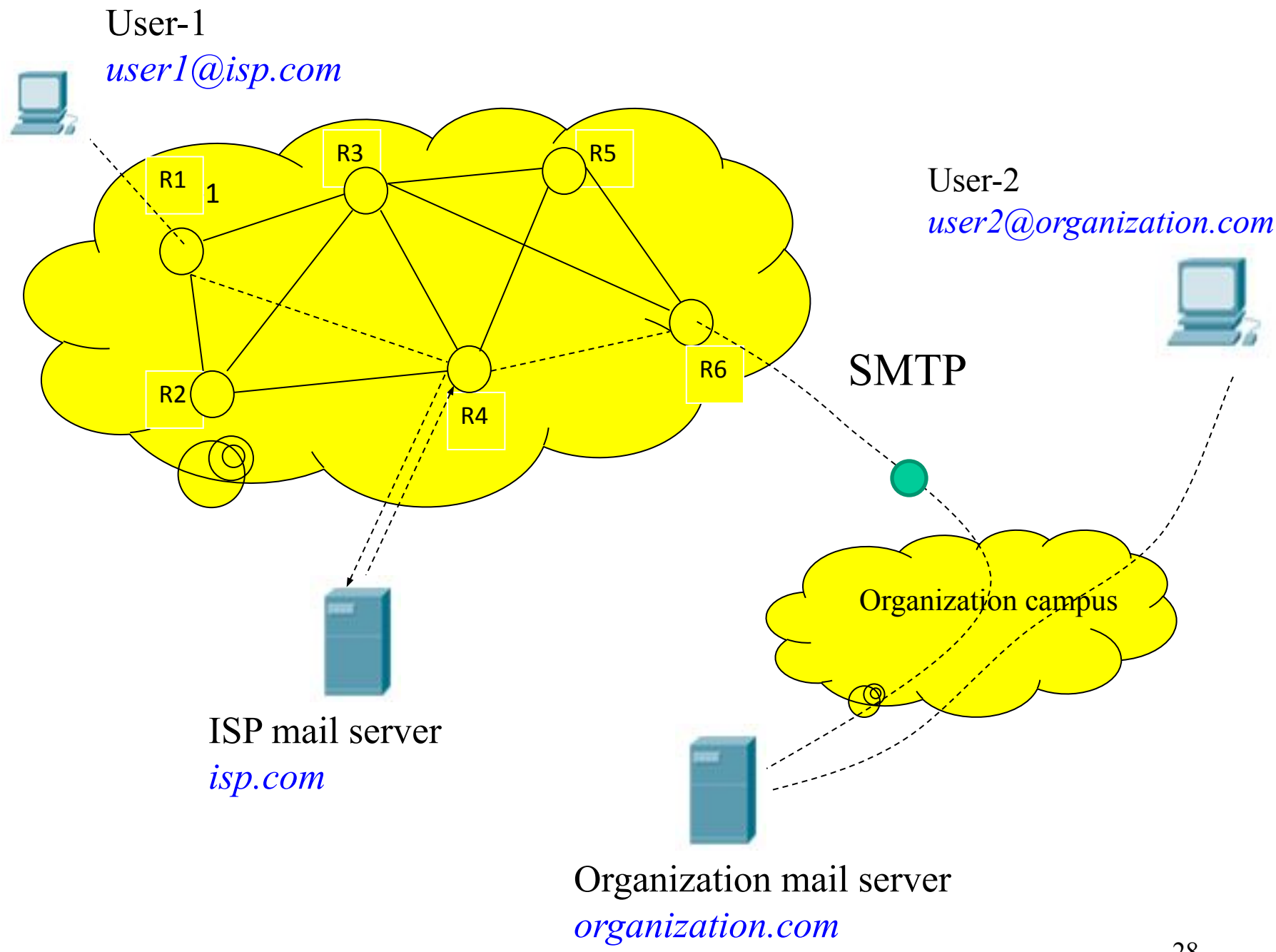
(b) Incoming mail

✔Before an individual can use e-mail, he or she must have at least one electronics mailbox. A mailbox is a space in the mail server allocated to the user to keep its e-mail.

✔In order for the mail to be sent properly from mailbox to mailbox, each mailbox must have a unique e-mail address. The e-mail address is multipart hierarchical address. The first portion of the address indicates the user and the later portion indicates the location of the mailbox.

Example-1

In fig. of next slide, user1 is in a residential area, has an Internet Service Provider (ISP), and is sending an e-mail to user2 working in an organization. Suppose that the mail servers are *isp.com* and *organization.com* respectively. The e-mail addresses of user1 and user2 are *user1@isp.com* and *user2@organization.com* respectively.


User-1
user1@isp.com
R1
1
R3
R5
R2
R4
R6
User-2
user2@organization.com
SMTP
Organization campus
ISP mail server
isp.com
Organization mail server
organization.com

The procedure for e-mail exchange between user 1 and user 2 are as follows.

1. User1 provides user2's e-mail address (*user2@organization.com*) and composes its message.
2. User1 sends the message to its mail server (*isp.com*)
3. Server *isp.com* places the message in its queue.
4. SMTP on user1's mail server notices the message in the queue and opens a TCP connection with the organization mail server (*organization.com*)
5. Initial handshaking takes place between the two servers.
6. The message is sent to the organization mail server using the established TCP connection.
7. Users2's mail server receives the message and then puts it in user 2's mailbox, ready to be retrieved by user2.

A user mail box is a space in the mail server allocated to the user to keep its e-mail.

The **SMTP protocol** is used to transfer a message from the SMTP sender to the SMTP receiver over a TCP connection. SMTP attempts to provide reliable operation but does not guarantee to recover from lost messages. SMTP does not return an end-to-end acknowledgment to a message's originator to indicate that a message is successfully delivered to the message's recipient. Also, SNMP (simple network management protocol) does not guarantee to return error indications. However, the SMTP-based mail system is generally considered reliable.

Post Office Protocol

The most popular protocol used to transfer e-mail messages from a permanent mailbox to local computer is known as version 3 of the Post Office Protocol (POP3); a secure version of the protocol is known as POP3S. The user invokes a POP3 client, which creates a TCP connection to a POP3 server on the mailbox computer. The user first sends a login and password to authenticate the session. Once authentication has been accepted, the client sends commands to retrieve a copy of one or more messages and to delete the message from the permanent mailbox.

Multipurpose Internet Mail Extensions (MIME)

The multipurpose Internet mail Extensions (MIME) were defined to allow transmission of non-ASCII data through e-mail. MIME does not change or replace protocols such as SMTP, POP3. Instead MIME allows arbitrary data to be encoded in ASCII and then transmitted in a standard e-mail message.

Fig. below illustrates a MIME message that contains a JPEG where the image is converted to 7-bit ASCII (base64 encoding) representation.

From: karim@juniv.edu
To: john@yahoo.com
MIME-Version:1.0
Content-Type: Image/jpeg
Content-Transfer-Encoding: base64

…data for the image…

Each MIME message includes information that tells the receipent the type of data and the encoding used. A receiver's mail system must first convert from base64 encoding back to binary, and then run an application that displays a JPEG image on the user's screen.

The World Wide Web

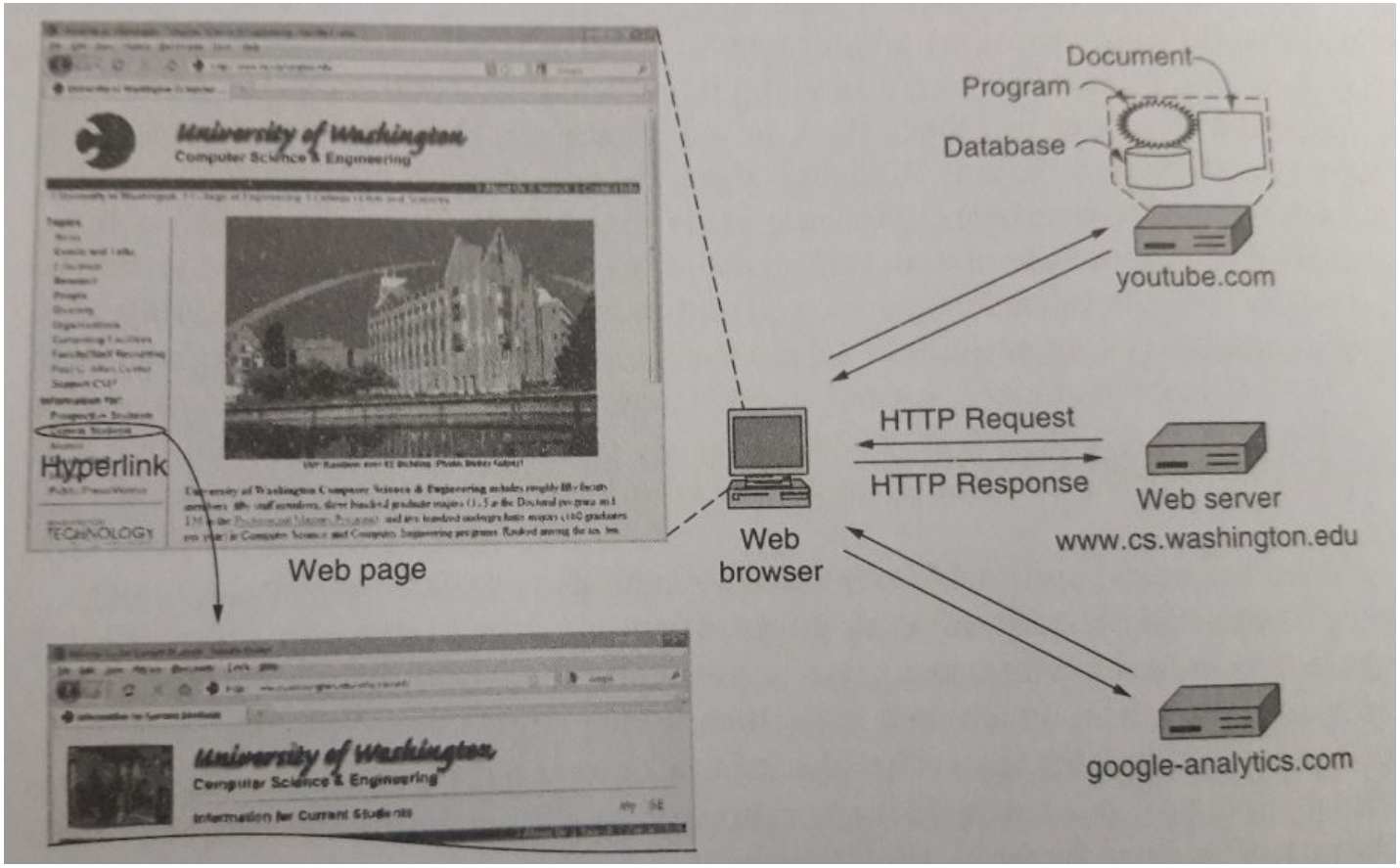
- Conceptually, the web consists of a vast, worldwide collection of contents in the form of **Web pages**, often called pages for short. The **Web pages** are accessible over the Internet. Each page may contain links to other pages anywhere in the world.

- The idea of having one page point to another called **hypertext**. A piece of text, icon, image and so on associated with another page is called a **hyperlin**k.

- The **World Wide Web** is a system of interlinked hypertext documents accessed via the Internet.

✔With a web browser (application programs like Internet Explorer, Netscape Navigator, Mozilla Firefox, Chrome etc) , one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks. A browser becomes a *client* that contacts the appropriate *web server* to obtain a copy of the specified page.

✔The browser fetches the page requested, interprets the text and formatting commands on it, and displays the page, properly formatted, on the screen.



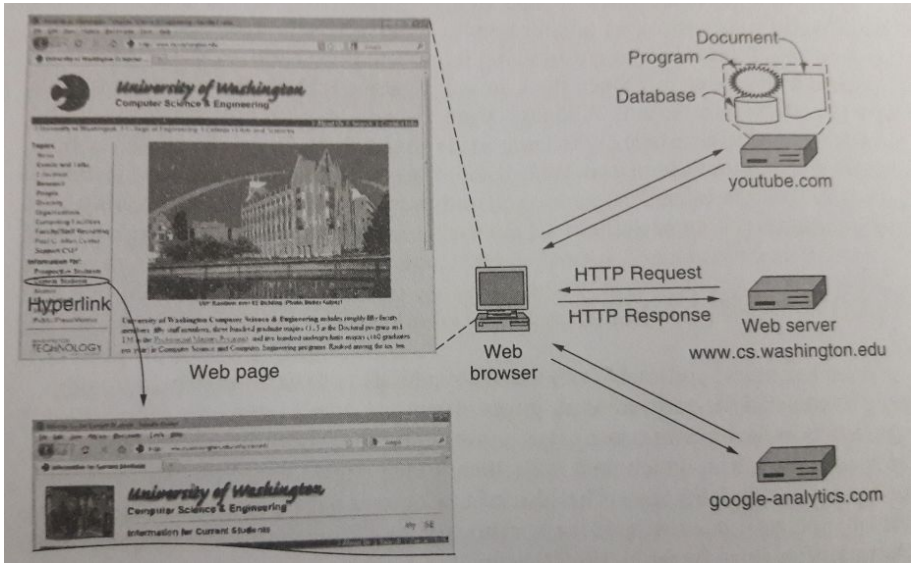
Architecture of the web

The Client Side

Let us now examine the client side of fig. below in more detail. In essence, a browser is a program that can display a Web page. When a client host *requests* an object, a Web server *responds* by sending the requested object through browsing tools.

The hyperlink needs a way to name any other page on the Web. Pages are named using URLs (Uniform Resource Locators). A typical URL is

http://www.abcd.com/products.html



- For the moment, it is sufficient to know that a **URL** has three parts: the name of the protocol (*http*), the DNS name of the machine where the page is located (*www.abcd.com*), and (usually) the name of the file containing the page (*products.html*).

- When a user clicks on a hyperlink, the browser carries out a series of steps in order to fetch the page pointed to. Suppose that a user is browsing the Web and finds a link on *Internet telephony* that points to ITU's home page, which is http://www.itu.org/home/index.html.

Let us trace the steps that occur when this link is selected.

1. The browser determines the URL (by seeing what was selected).
2. The browser asks DNS for the IP address of *www.itu.org*.
3. DNS replies with 156.106.192.32.
4. The browser makes a TCP connection to 156.106.192.32 on port 80 i.e. HTTP protocol.
5. It then sends over an HTTP request asking for file /home/index.html.
6. The www.itu.org server sends the file */home/index.html* as an HTTP response.
7. The TCP connection is released.
8. The browser displays all the text in */home/index.html*.
9. The browser fetches and displays all images in this file.

- Although a browser is basically an **HTML interpreter**, most browsers have numerous buttons and features to make it easier to navigate the Web. Most have a button for going back to the previous page, a button for going forward to the next page (only operative after the user has gone back from it), and a button for going straight to the user's own start page.

The Server Side

When the user types in a URL or clicks on a line of hypertext, the browser parses the URL and interprets the part between *http://* and the next slash as a DNS name to look up. Armed with the IP address of the server, the browser establishes a TCP connection to port 80 on that server. Then it sends over a command containing the rest of the URL, which is the name of a file on that server. The server then returns the file for the browser to display.

The steps that the server performs in its main loop are:

1. Accept a TCP connection from a client (a browser).
2. Get the name of the file requested.
3. Get the file (from disk).
4. Return the file to the client.
5. Release the TCP connection.

- A problem with this design is that every request requires making a disk access to get the file. The result is that the Web server cannot serve more requests per second than it can make disk accesses. A high-end SCSI (**Small Computer System Interface,** communicate with peripheral hardware such as **disk** drives, tape drives, CD-ROM etc.) disk has an average access time of around 5 msec, which limits the server to at most 200 requests/sec, less if large files have to be read often. For a major Web site, this figure is too low.

- One obvious improvement (used by all Web servers) is to maintain a cache in memory of the *n* most recently used files. Before going to disk to get a file, the server checks the cache. If the file is there, it can be served directly from memory, thus eliminating the disk access.

- The next step for building a faster server is to make the server multithreaded. In one design, the server consists of a front-end module that accepts all incoming requests and *k* processing modules, as shown in Fig. below. When a request comes in, the front end accepts it and builds a short record describing it. Then it hand the record to one of the processing modules.

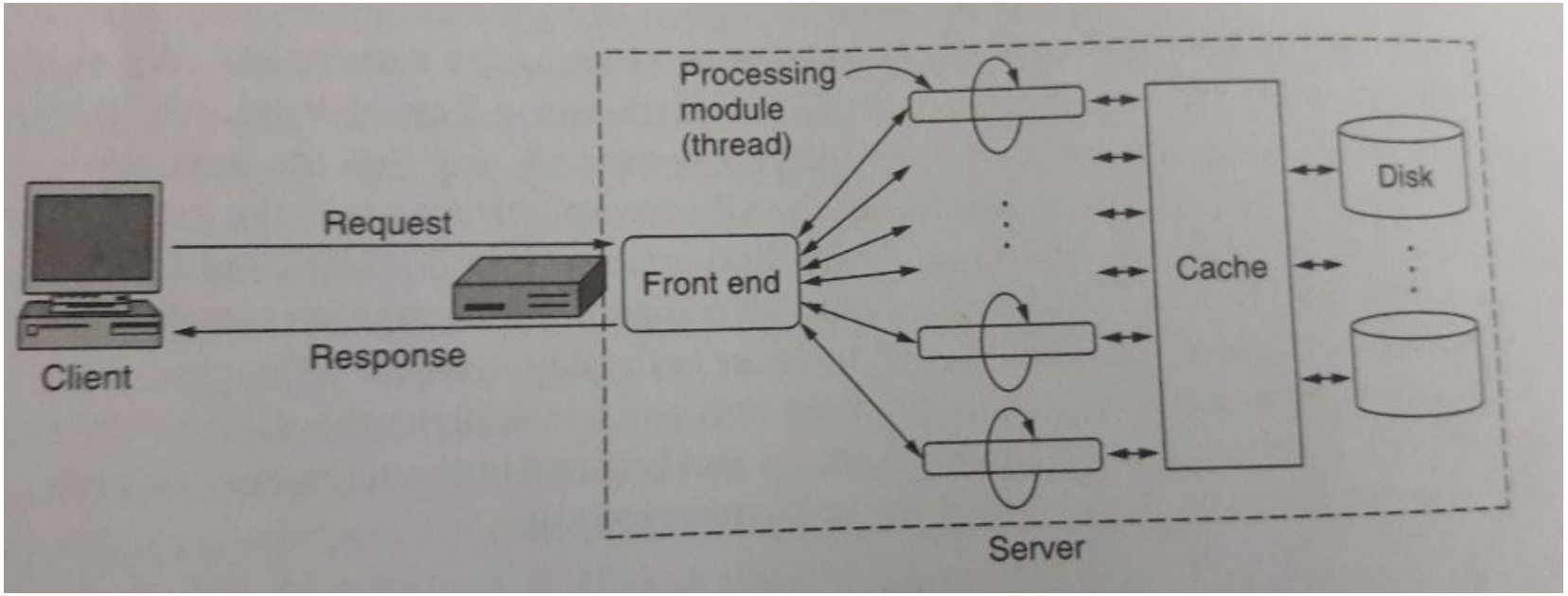


Figure . A multithreaded Web server with a front end and processing modules

- ✔The processing module first checks the cache to see if the file needed is there. If so, it updates the record to include a pointer to the file in the record. If it is not there, the processing module starts a disk operation to read it into the cache (possibly discarding some other cached files to make room for it). When the file comes in from the disk, it is put in the cache and also sent back to the client.

- ✔The advantage of this scheme is that while one or more processing modules are blocked waiting for a disk operation to complete (and thus consuming no CPU time), other modules can be actively working on other requests.

URLs—Uniform Resource Locators

- If every page were somehow assigned a unique name, there would not be any ambiguity in identifying pages.

- URLs have three parts: the protocol (also known as the scheme), the DNS name of the machine on which the page is located, and a local name uniquely indicating the specific page (usually just a file name on the machine where it resides). As an example, the Web site for the author's department contains several videos about the university and the city of Amsterdam. The URL for the video page is

  http://www.cs.vu.nl/video/index-en.html

- This URL consists of three parts: the protocol (*http),* the DNS name of the host (*www.cs.vu.nl*), and the file name (v*ideo/index-en.html*), with certain punctuation separating the pieces. The file name is a path relative to the default Web directory at cs.vu.nl.

❖URL scheme is open-ended in the sense that it is straightforward to have browsers use multiple protocols to get at different kinds of resources. In fact, URLs for various other common protocols have been defined. Slightly simplified forms of the more common ones are listed in Table-1

Table-1 Some common URLs

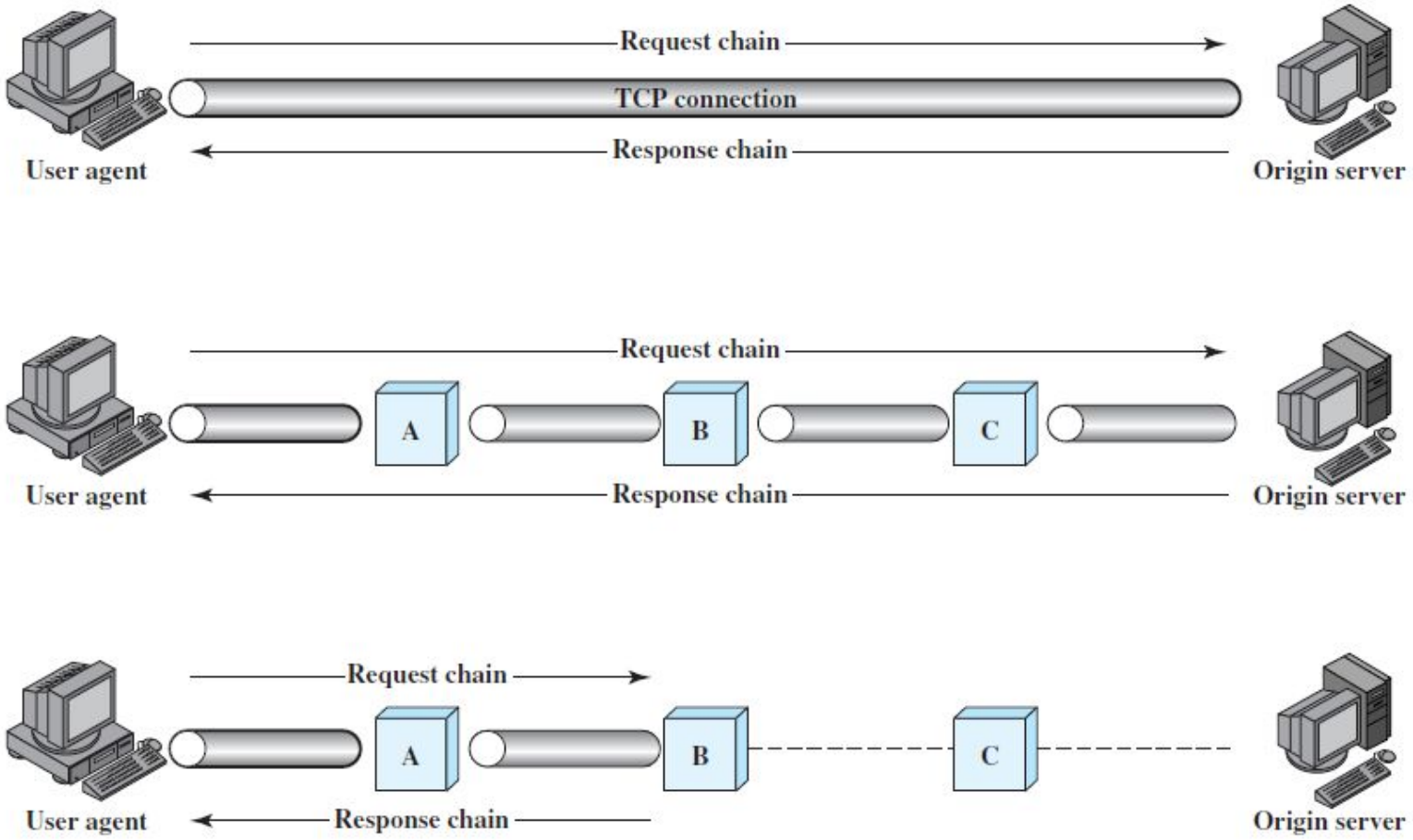
Name	Used for	Example
http	Hypertext (HTML)	http://www.ee.uwa.edu/~rob/
https	Hypertext with security	https://www.bank.com/accounts/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
mailto	Sending email	mailto:JohnUser@acm.org
rtsp	Streaming media	rtsp://youtube.com/montypython.mpg
sip	Multimedia calls	sip:eve@adversary.com
about	Browser information	about:plugins

HTTP Overview

The request-response protocol for fetching pages is a simple text-based protocol that runs over TCP, called HTTP (Hyper Text Transfer Protocol). The most typical use of HTTP is between a Web browser and a Web server or between intermediate machines and a Web browser.

Figure on next slide illustrates three examples of HTTP operation. The simplest case is one in which a user agent establishes a direct connection with an origin server. The *origin server* is the server on which a resource of interest resides.

Each intermediate system acts as a relay, so that a request initiated by the client is relayed through the intermediate systems to the server, and the response from the server is relayed back to the client.


Request chain
TCP connection
Response chain
User agent
Origin server
Request chain
A
B
C
User agent
Response chain
Origin server
Request chain
A
B
C
User agent
Response chain
Origin server

- HTTP makes use of TCP connection between browser and Web server but the connection does not provide reliability or retransmission itself.

- The client then issues an HTTP request. Each HTTP request is stateless i.e. the server does not keep history of previous requests or sessions.

- In most cases, a browser requests a Web page, and the server transfers a copy to the browser. HTTP also allows transfers from browser to server (form upload).

- To improve response time, a browser caches a copy of each web page it retrieves. If user requests a page again, HTTP allows the browser to interrogate the server to determine whether the content of the page has changes since the copy was cached.

HTML-The HyperText Markup Language

- Web pages are written in a language called HTML (HyperText Markup Language). HTML allows users to produce Web pages that include text, graphics, and pointers to other Web pages.

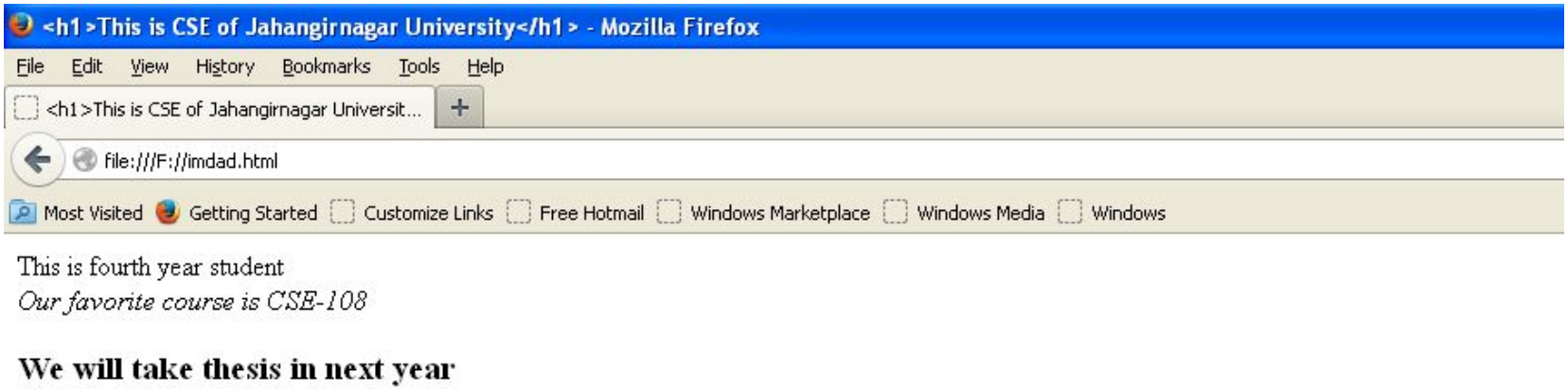
- In the simplest form, web pages are **static**. That is, they are just files sitting on some server that present themselves in the same way each time they are fetched and viewed.

- HTML is a markup language, a language for describing how documents are to be formatted. The term "markup" comes from the old days when copyeditors actually marked up documents to tell the printer-in those days, a human being-which fonts to use, and so on.

✔Markup languages thus contain explicit commands for formatting. For example, in HTML, means start boldface mode, and means leave boldface mode. The advantage of a markup language over one with no explicit markup is that writing a browser for it is straightforward: the browser simply has to understand the markup commands.

Example

```
<html>
<head>
<title>
<h1>This is CSE of Jahangirnagar University</h1>
</title>
</head>
<body style="backgroung-color:yellow;">
This is fourth year student <br/>
<i> Our favorite course is CSE-108 </i> <br/>
<h3> We will take thesis in next year </h3> <br/>
<p style="font-size:36px; text-align:center;">
Our M.Sc class will star soon </p>
<h3 style="font-family:arial;color:blue;"> We will take thesis in next year </h3> <br/>
</body>
</html>
```



Our M.Sc class will star soon

We will take thesis in next year

Static Documents

Static documents are fixed-content documents that are created and stored in a server. The client can get only a copy of the document. In other words, the contents of the file are determined when the file is created, not when it is used. Of course, the contents in the server can be changed, but the user cannot change them.

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 that creates the dynamic document. The server returns the output of the program or script as a response to the browser that requested the document.

- ✔A fresh document is created for each request, the contents of a dynamic document can vary from one request to another. A very simple example of a dynamic document is the retrieval of the time and date from a server. Time and date are kinds of information that are dynamic in that they change from moment to moment. The client can ask the server to run a program such as the *date* program in UNIX and send the result of the program to the client.

CSS (cascading Style Sheets) is used to represent appearance of pages.

PHP(Hypertext Preprocessor) for dynamic Web page for example e-commerce sites.

AJAX (Asynchronous JavaScript and XML) for full-featured and responsive Web applications: Google's Gmail, Maps etc.

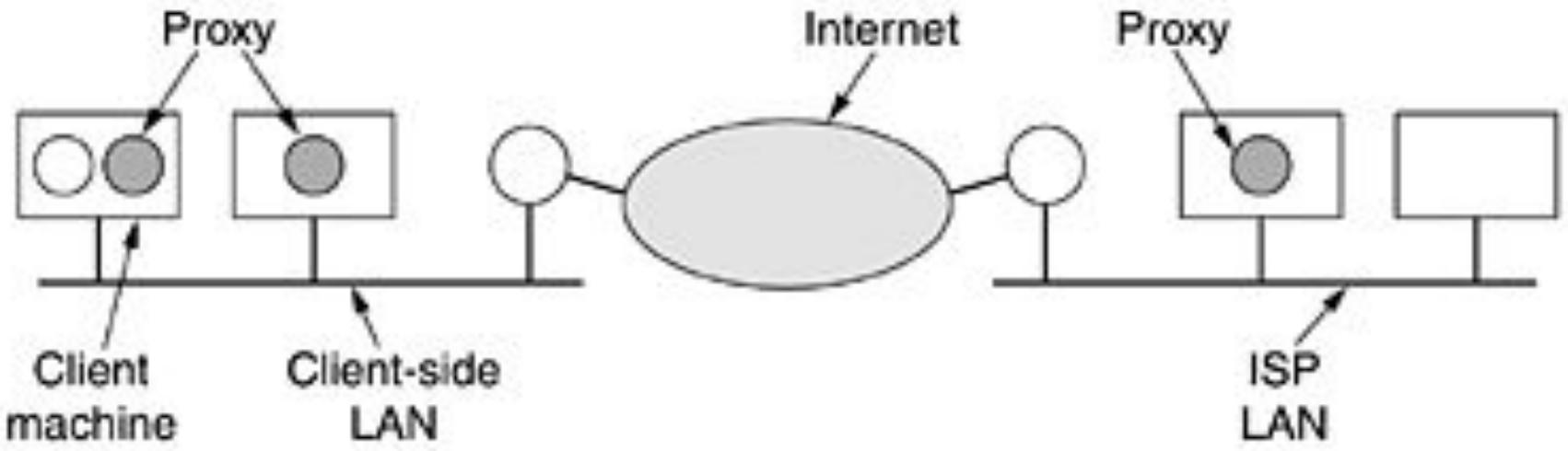
Web Caching (Proxy Server)

An HTTP request from a user is first directed to the network proxy server or web cache. Normally each organization or ISP should have its own cache providing a high speed link to its user.

A fairly simple way to improve performance is to save pages that have been requested in case they are used again. This technique is especially effective with pages that are visited a great deal, such as www.yahoo.com and www.cnn.com.

Individual PCs often run proxies so they can quickly look up pages previously visited. On a company LAN, the proxy is often a machine shared by all the machines on the LAN, so if one user looks at a certain page and then another one on the same LAN wants the same page, it can be fetched from the proxy's cache.

Requests first go to the local proxy. If that fails, the local proxy queries the LAN proxy. If that fails, the LAN proxy tries the ISP proxy. The latter must succeed, either from its cache, a higher-level cache, or from the server itself. A scheme involving multiple caches tried in sequence is called hierarchical caching. A possible implementation is illustrated in fig. below.



Hierarchical caching with three proxies

Web Caching Algorithm

- The user browser makes a TCP connection with the Web cache .
- The user browser transmits its HTTP request to the Web cache.
- If the Web cache has a copy of the requested object , the Web cache forwards the object to the user browser.
- Otherwise the Web cache establishes a TCP connection to the requested server and ask for the object. Once it receives the requested object, the Web cache stores a copy of it and forwards another copy to the requesting user browser over the existing TCP connection.

Figure below shows three Internet service providers (ISPs). A user in ISP domain 3 is browsing to find and watch an object namedhttp://www.filmmaker.comin ISP domain 1. The request for this object is directed to the Web cache, shown by dashed lines. In this example, the Web cache has no record of the requested object and therefore is establishing another TCP connection to update its record.

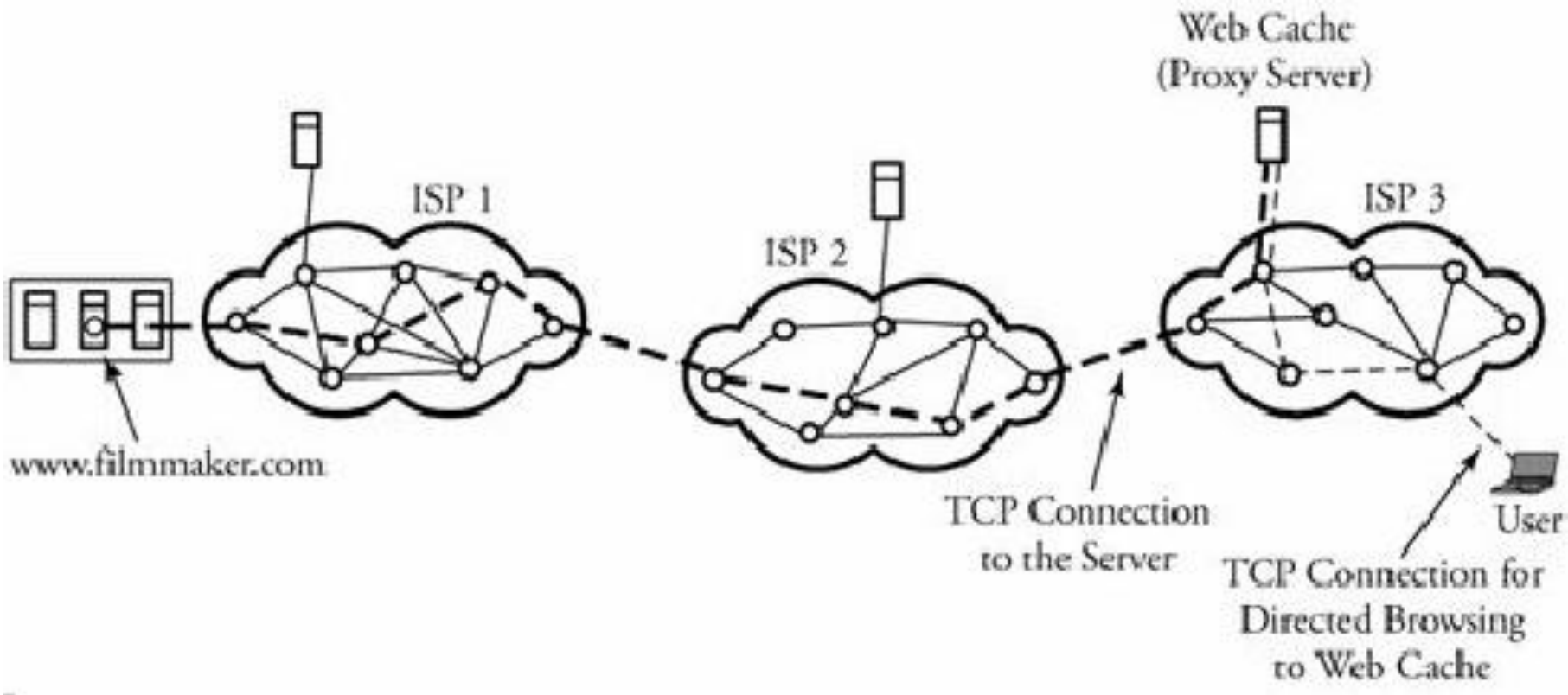


Figure. A user's browser requesting an object through the Web cache

Telnet and Secure Shell

Telnet (teletype network) is a remote login protocol uses TCP/IP standard for establishing connection to a remote system. Recently IT specialists realize that telnet traffic is plain text hence suffers from security. To combat the situation, a new protocol SSH (or Secure Shell) is used to log into a remote machine which ensure the transmission of data thorough encrypted channel. We can use the SSH server to enable an SSH client for encrypted communications with Cisco port. Because of security the telnet protocol is replaced by Secure Shell (SSH) protocol.

File Transfer Protocol (FTP)

✔File transfer is an important computer networking application. It is always essential that files and information geographically distributed over different location be shared among the numbers of working group.

✔File Transfer Protocol (FTP) is part of the TCP/IP suite and is very similar to TELNET. Both FTP and TELNET are built on the client server paradigm, and both allow a user to establish a remote connection. However, TELNET provides a broader access to a user, whereas FTP allows access only to certain files. The sequence of operation of FTP is like:

1. A user requests a connection to a remote server
2. The user waits for an acknowledgement.
3. Once connected, the user must enter a user ID, followed by password.
4. The connection is established over TCP session.
5. The desired files are transferred.
6. The user closes the FTP connection.

FTP can also run through a Web browser.

Well-Known TCP Port Numbers

In TCP/IP and UDP networks, a port is an endpoint to a logical connection and the way a client program specifies a specific server program on a computer in a network. The port number identifies what type of port it is. For example, port 80 is used for HTTP traffic.

20 FTP -- Data
21 FTP -- Control
22 SSH Remote Login Protocol
23 Telnet
25 SMTP
53 DNS
80 HTTP
110 POP3
161 SNMP

Network Management

✔The main purpose of network management is to monitor, manage, and control a network. A network can be structured with many links, routers, servers, and other physical-layer devices, which can be equipped with many network protocols that coordinate them.

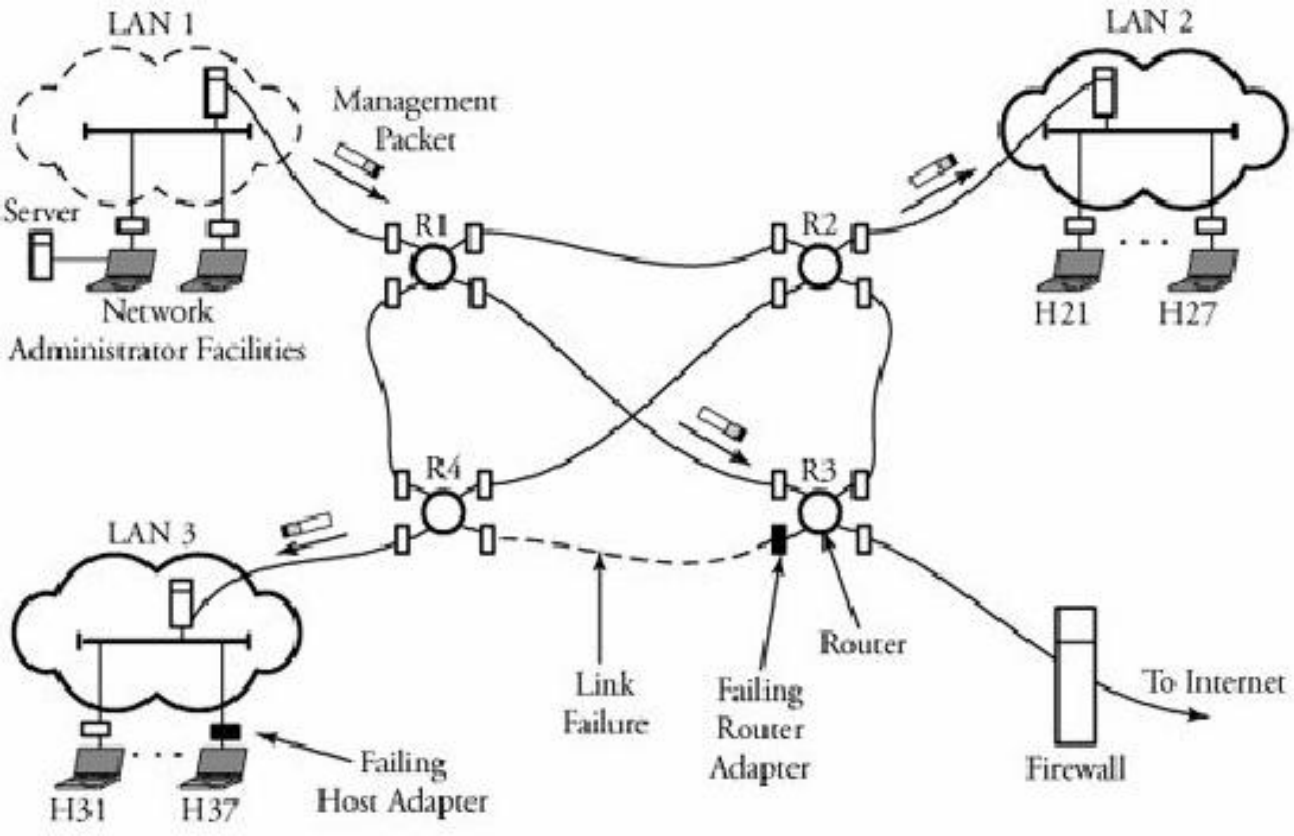
✔Imagine when thousands of such devices or protocols are tied together by an ISP and how drastic their management can become to avoid any interruptions in routine services.

✔In this context the purpose of network management is to monitor, test, and analyze the hardware, software, and human elements of a network and then to configure and control those elements to meet the operational performance requirements of the network

Network management tasks can be characterized as follows:

QoS and performance management. A network administrator periodically monitors and analyzes routers, hosts, and utilization of links and then redirect traffic flow to avoid any overloaded spots. Certain tools are available to detect rapid changes in traffic flow.

Network failure management. Any fault in a network, such as link, host, or router hardware or software outages, must be detected, located, and responded to by the network. Typically, increased checksum errors in frames is an indication of possible error. Figure shows adapter failures at router R3 and host H37; these failures can be detected through network management.



Simple network management in a scenario ofLANs connecting to the Internet

Configuration management. This task involves tracking all the devices under management and ensuring that all devices are connected and operate properly. If there is an unexpected change in routing tables, a network administrator wants to discover the misconfigured spot and reconfigure the network before the error affects the network substantially.

Security management. A network administrator is responsible for the security of its network. This task is handled mainly through firewalls. A firewall can monitor and control access points. In such cases, the network administrator wants to know about any intrusion from a suspicious source to the network. For example, a host in a network can be attacked by receiving a large number of SYN packets.

Billing and accounting management. The network administrator specifies user access or restrictions to network resources and issues all billing and charges, if any, to users.