Unit 6

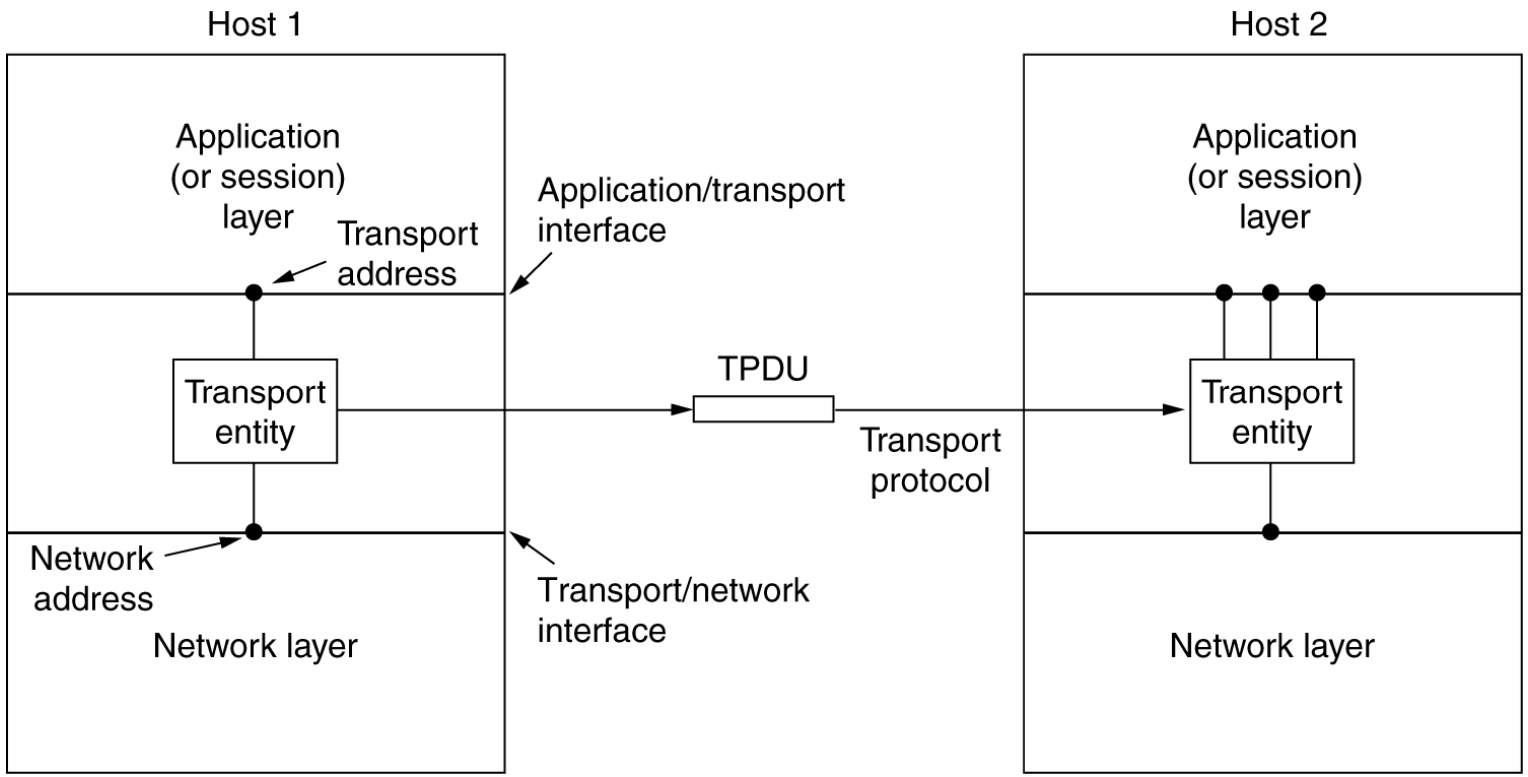
The Transport Layer

UNIT VI: Transport Layer: Transport Services, Connection establishment, Connection release and TCP and UDP protocols.
Application Layer: Domain name system, FTP, HTTP, SMTP, WWW.

The Transport Service

- Services Provided to the Upper Layers
- Transport Service Primitives
- Berkeley Sockets

Services Provided to the Upper Layers



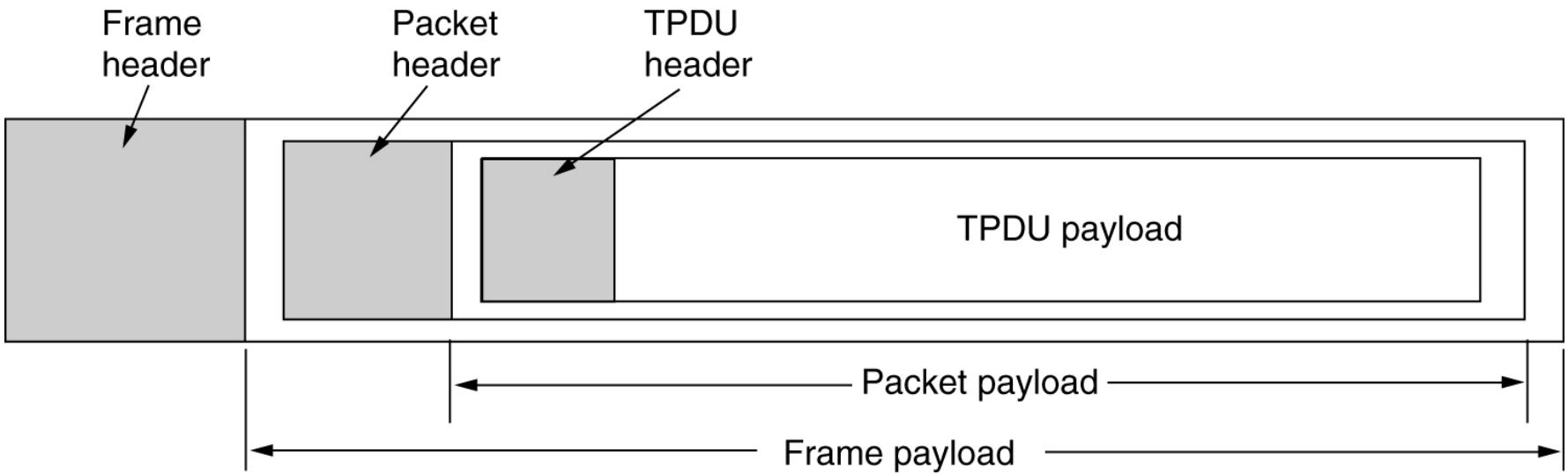
The network, transport, and application layers.

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

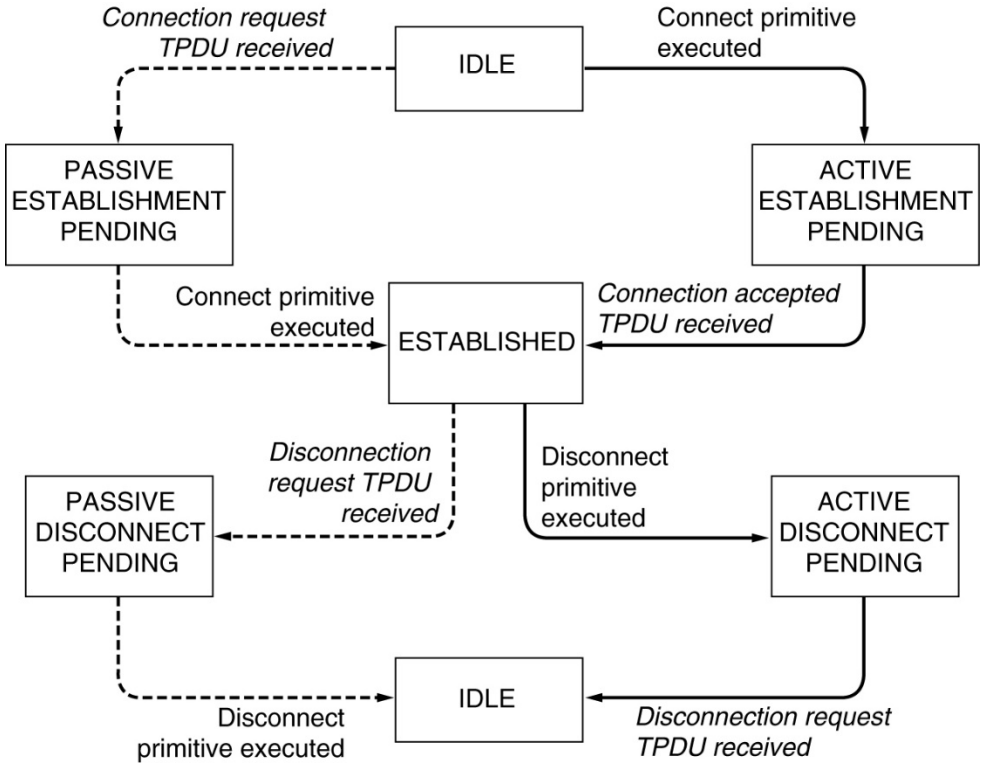
The primitives for a simple transport service.

Transport Service Primitives (2)



The nesting of TPDUs, packets, and frames.

Transport Service Primitives (3)



A state diagram for a simple connection management scheme. Transitions labeled in italics are caused by packet arrivals. The solid lines show the client's state sequence. The dashed lines show the server's state sequence.

Berkeley Sockets

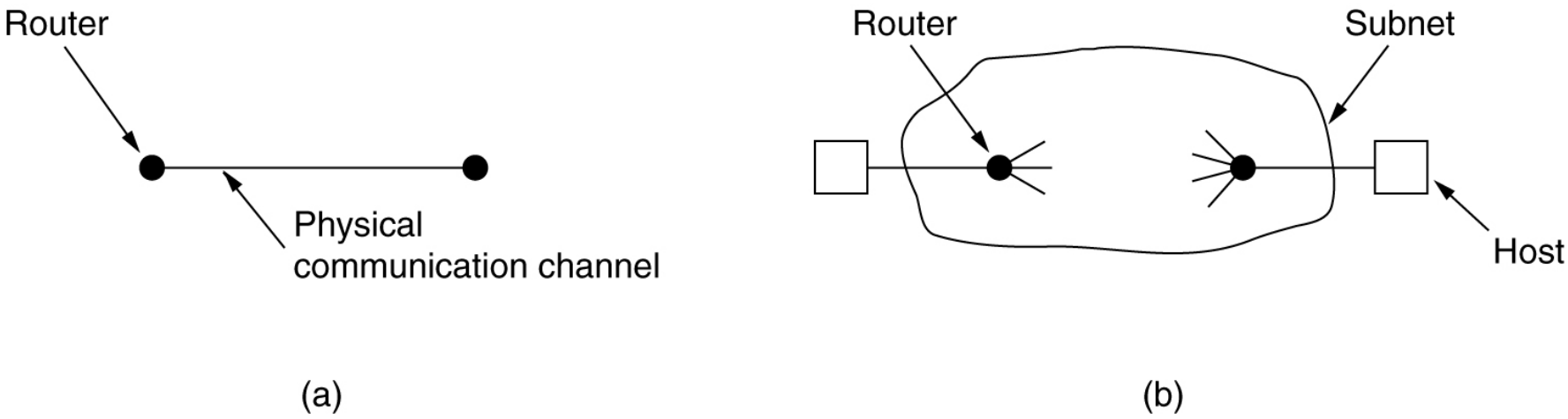
Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

The socket primitives for TCP.

Elements of Transport Protocols

- Addressing
- Connection Establishment
- Connection Release

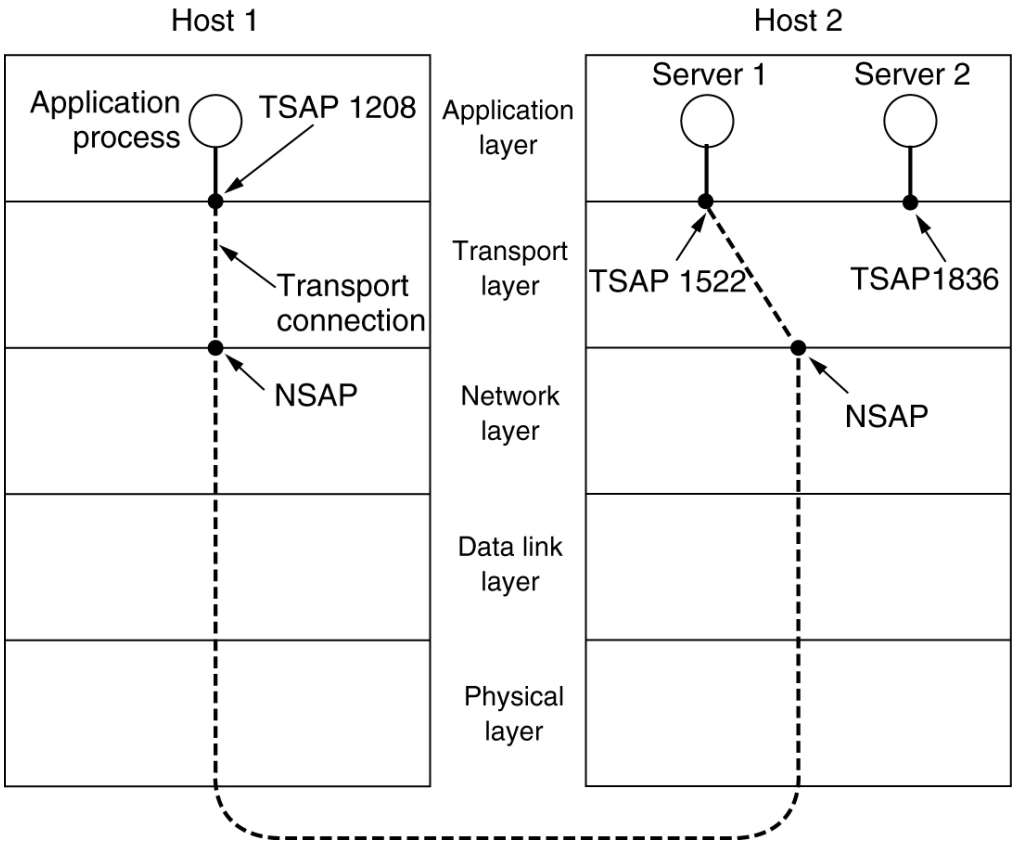
Transport Protocol



(a) Environment of the data link layer.

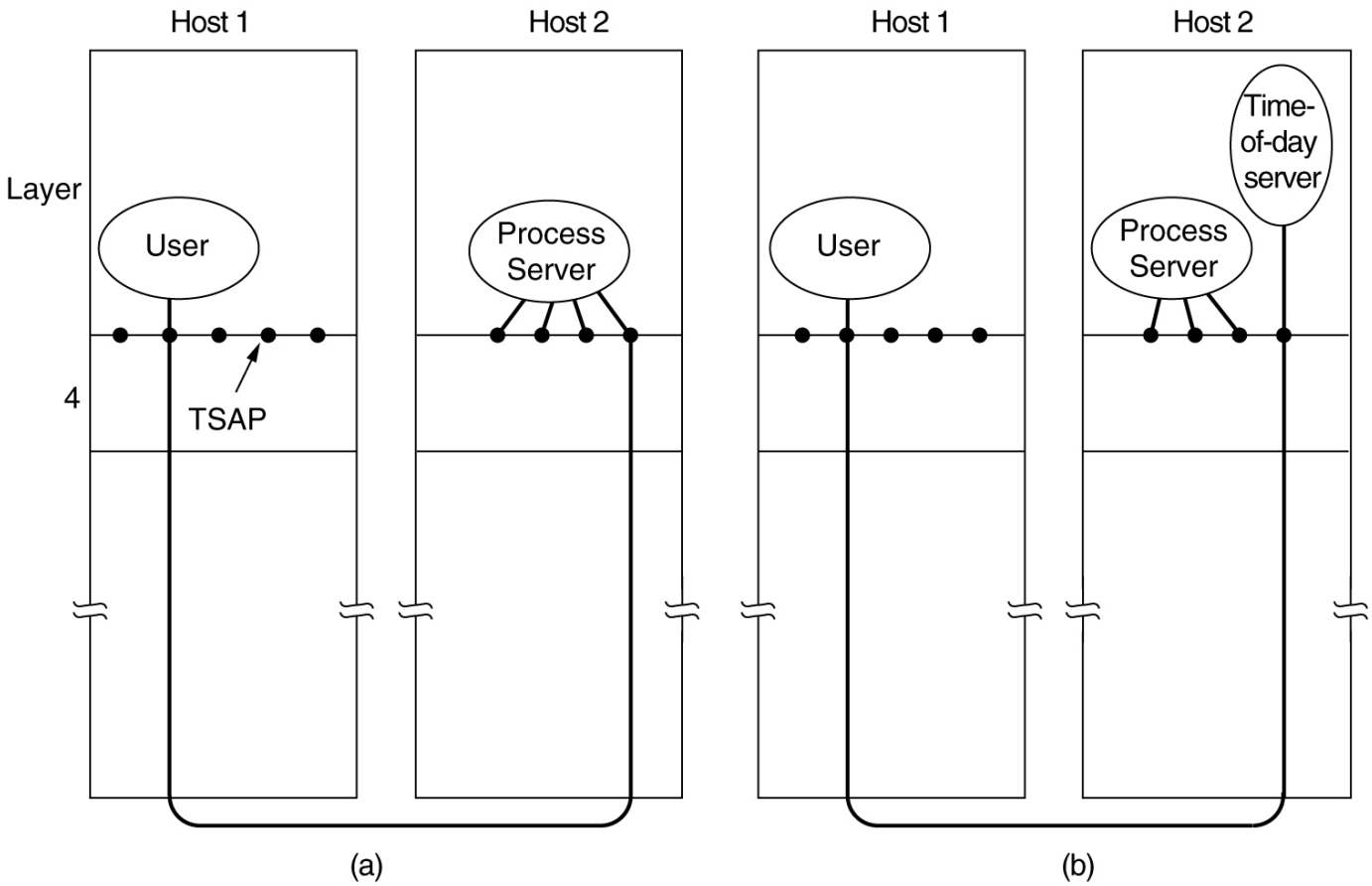
(b) Environment of the transport layer.

Addressing



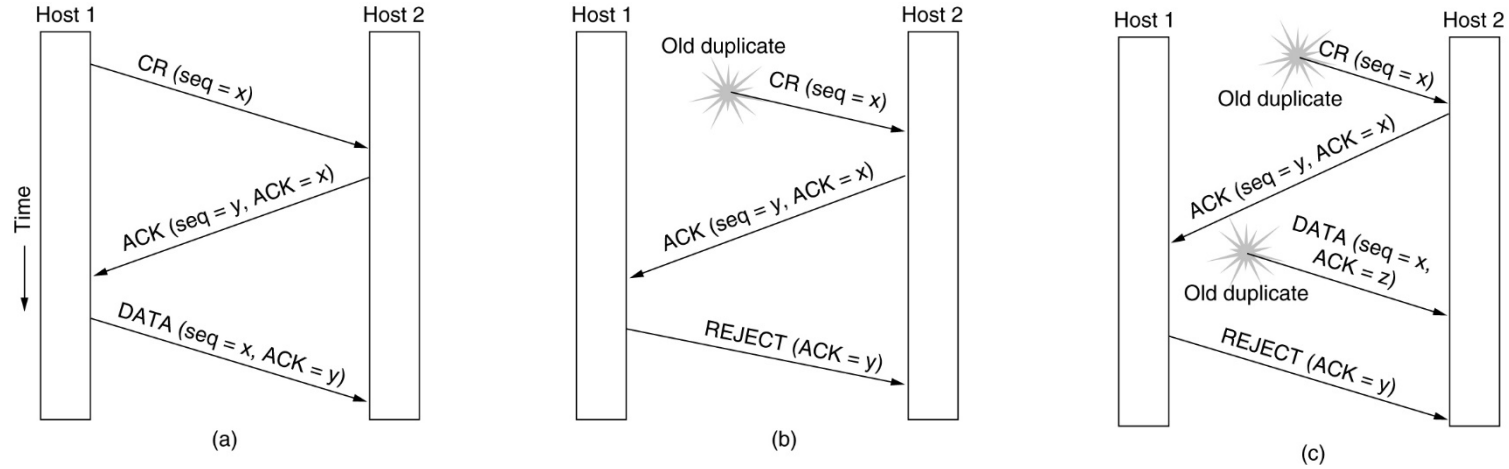
TSAPs, NSAPs and transport connections.

Connection Establishment



How a user process in host 1 establishes a connection with a time-of-day server in host 2.

Connection Establishment (3)



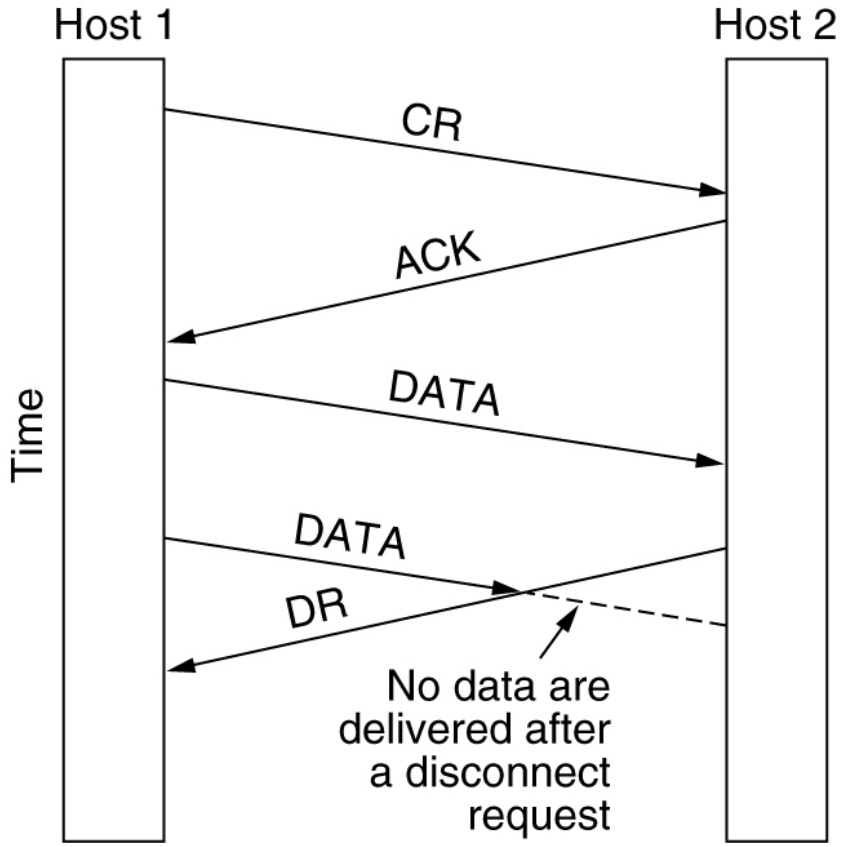
Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST.

(a) Normal operation,

(b) Old CONNECTION REQUEST appearing out of nowhere.

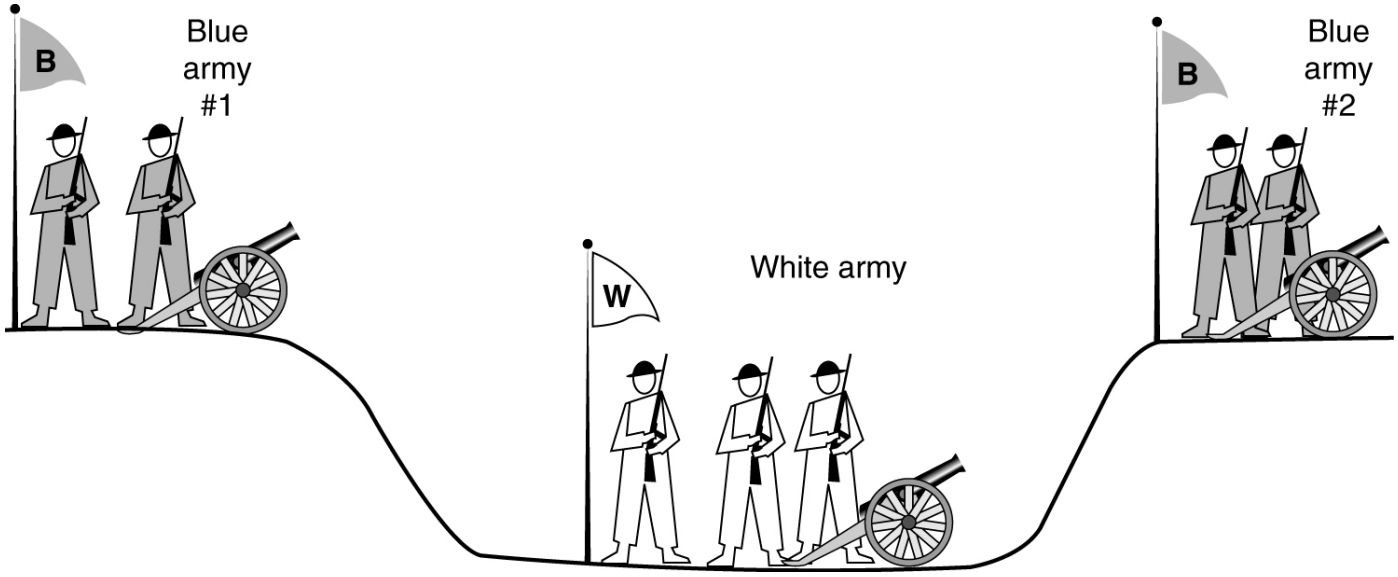
(c) Duplicate CONNECTION REQUEST and duplicate ACK.

Connection Release



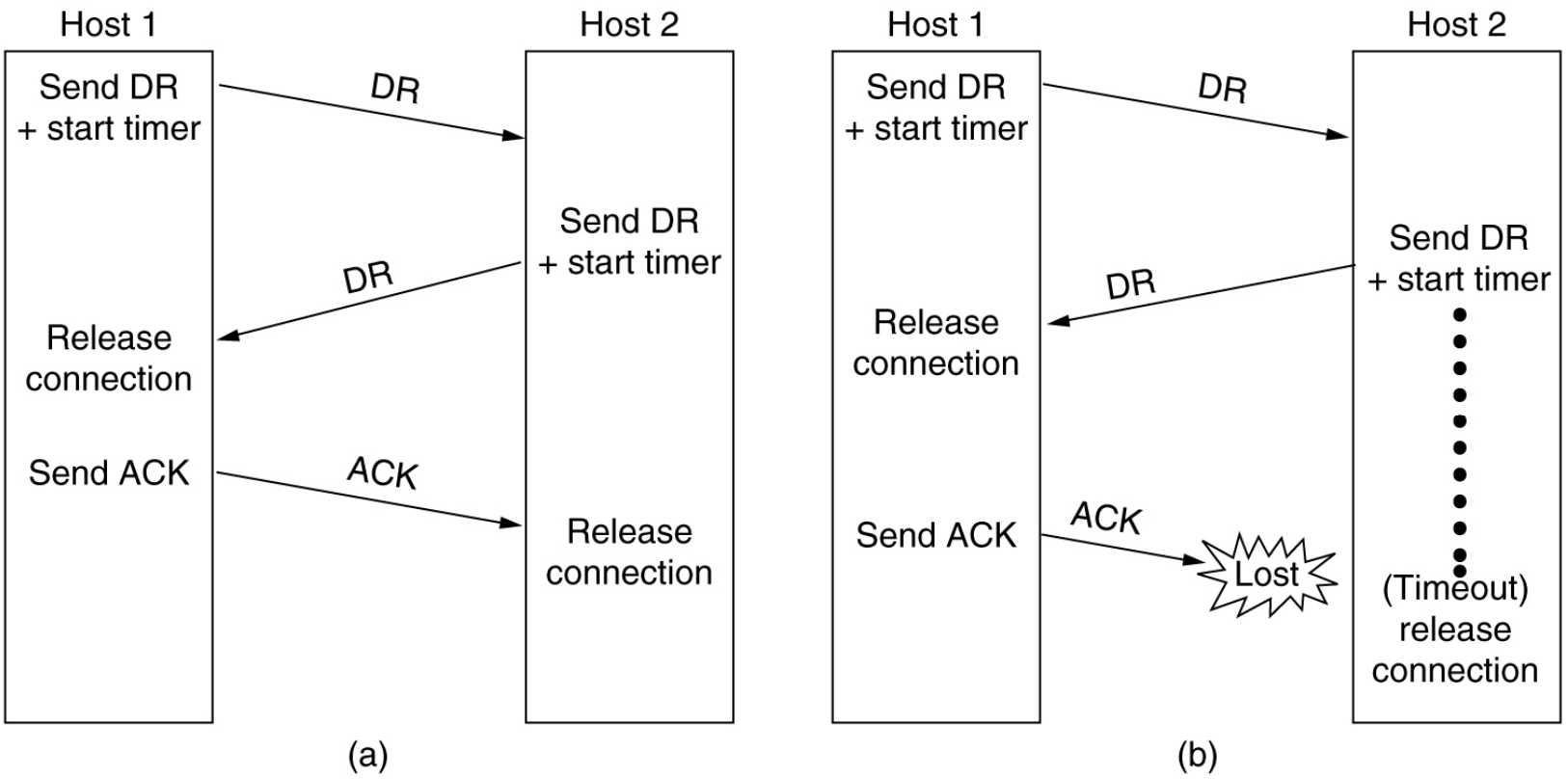
Abrupt disconnection with loss of data.

Connection Release (2)



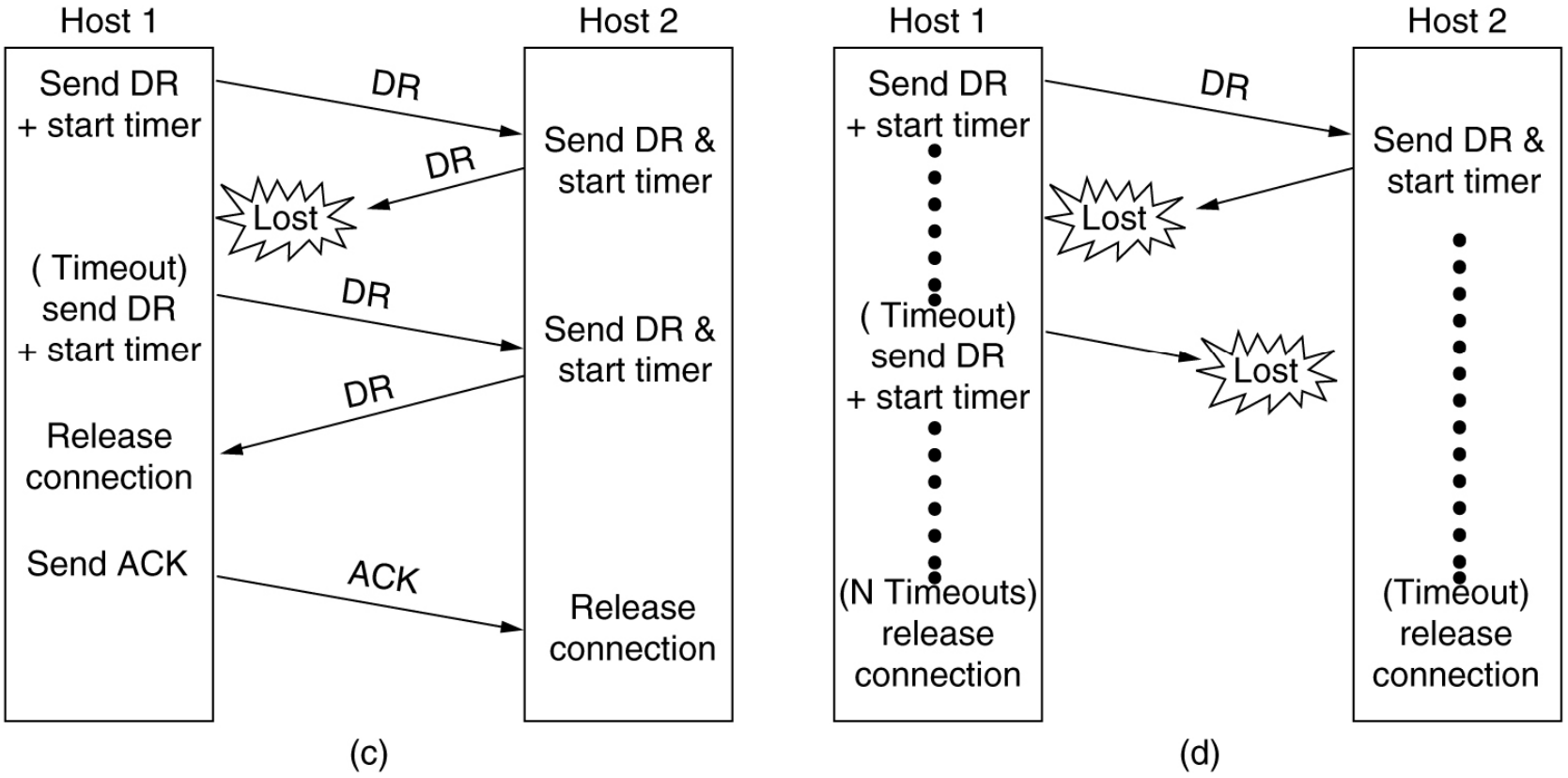
The two-army problem.

Connection Release (3)



Four protocol scenarios for releasing a connection. (a) Normal case of a three-way handshake. (b) final ACK lost.

Connection Release (4)

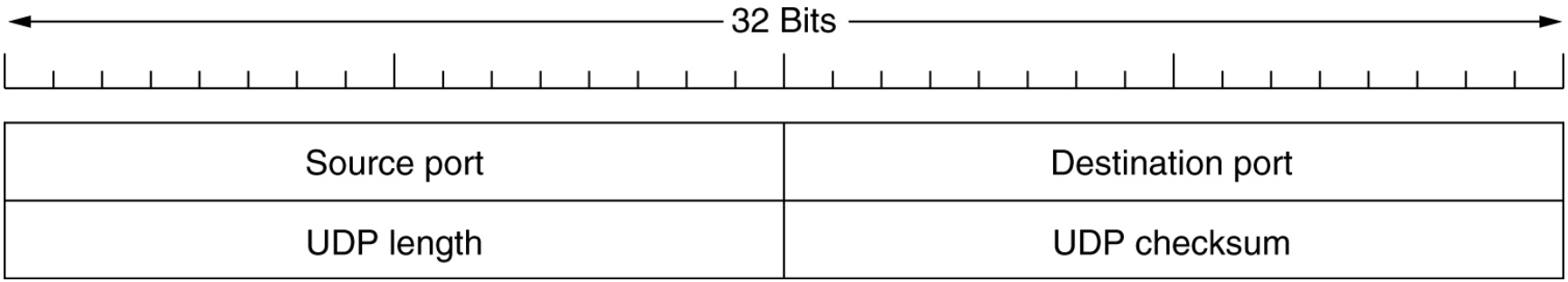


(c) Response lost. (d) Response lost and subsequent DRs lost.

The Internet Transport Protocols: UDP

- Introduction to UDP

Introduction to UDP



The UDP header.

The Internet Transport Protocols: TCP

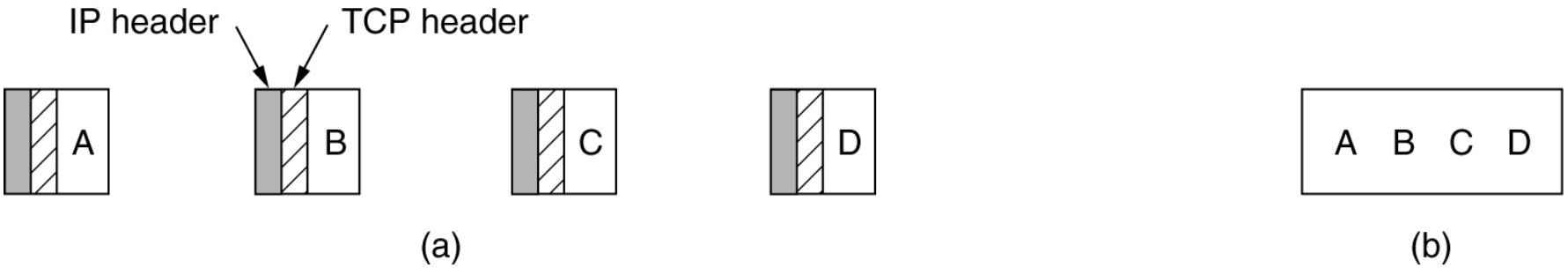
- Introduction to TCP
- The TCP Service Model
- The TCP Protocol
- The TCP Segment Header
- TCP Connection Establishment
- TCP Connection Release

The TCP Service Model

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

Some assigned ports.

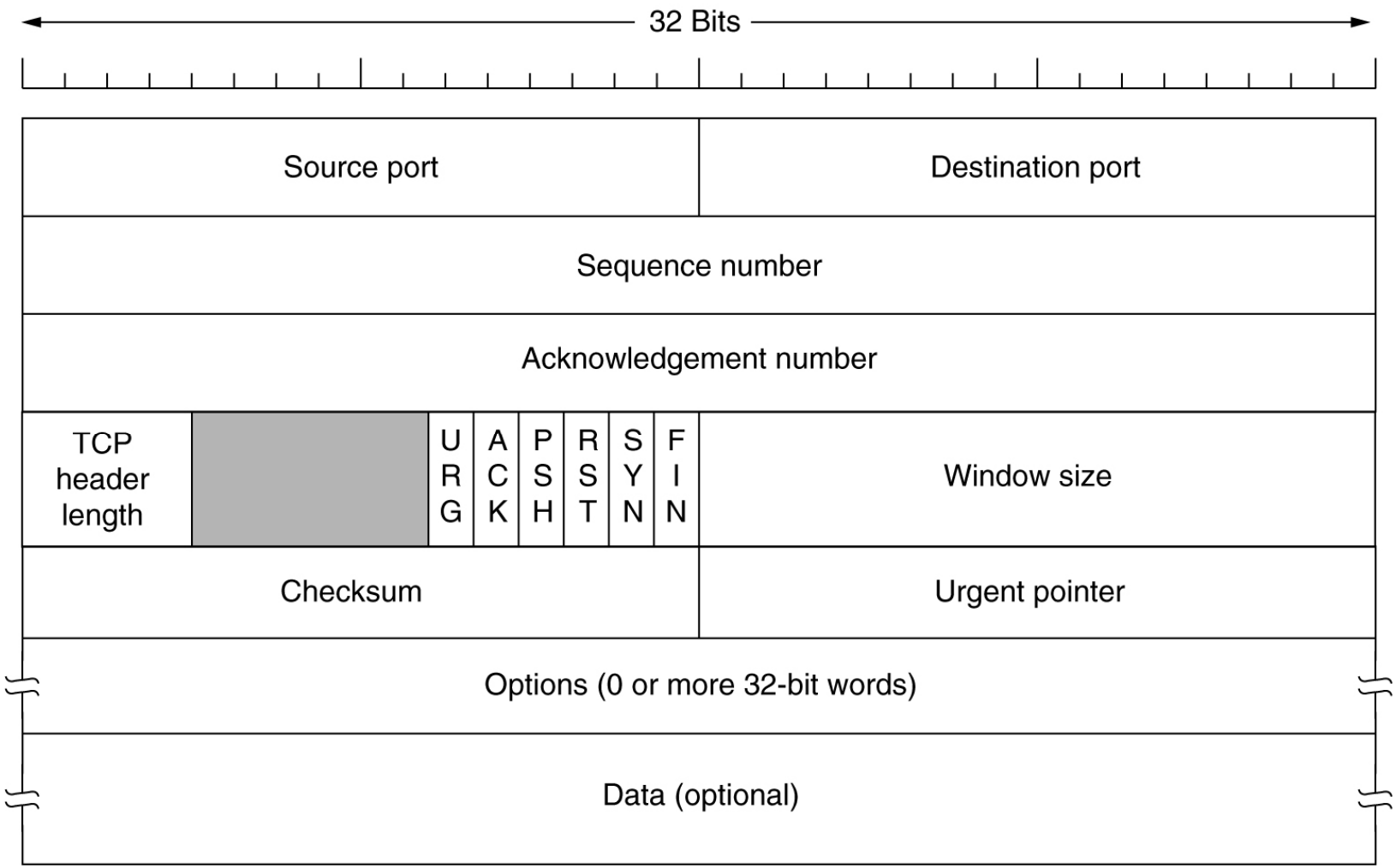
The TCP Service Model (2)



(a) Four 512-byte segments sent as separate IP datagrams.

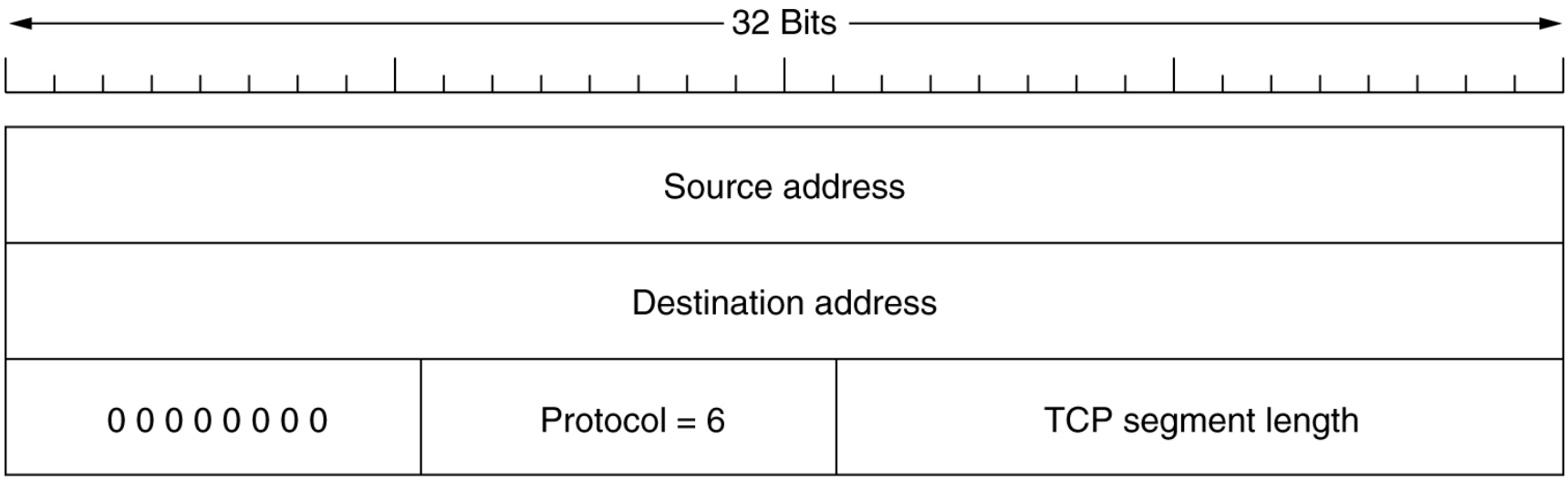
(b) The 2048 bytes of data delivered to the application in a single READ CALL.

The TCP Segment Header



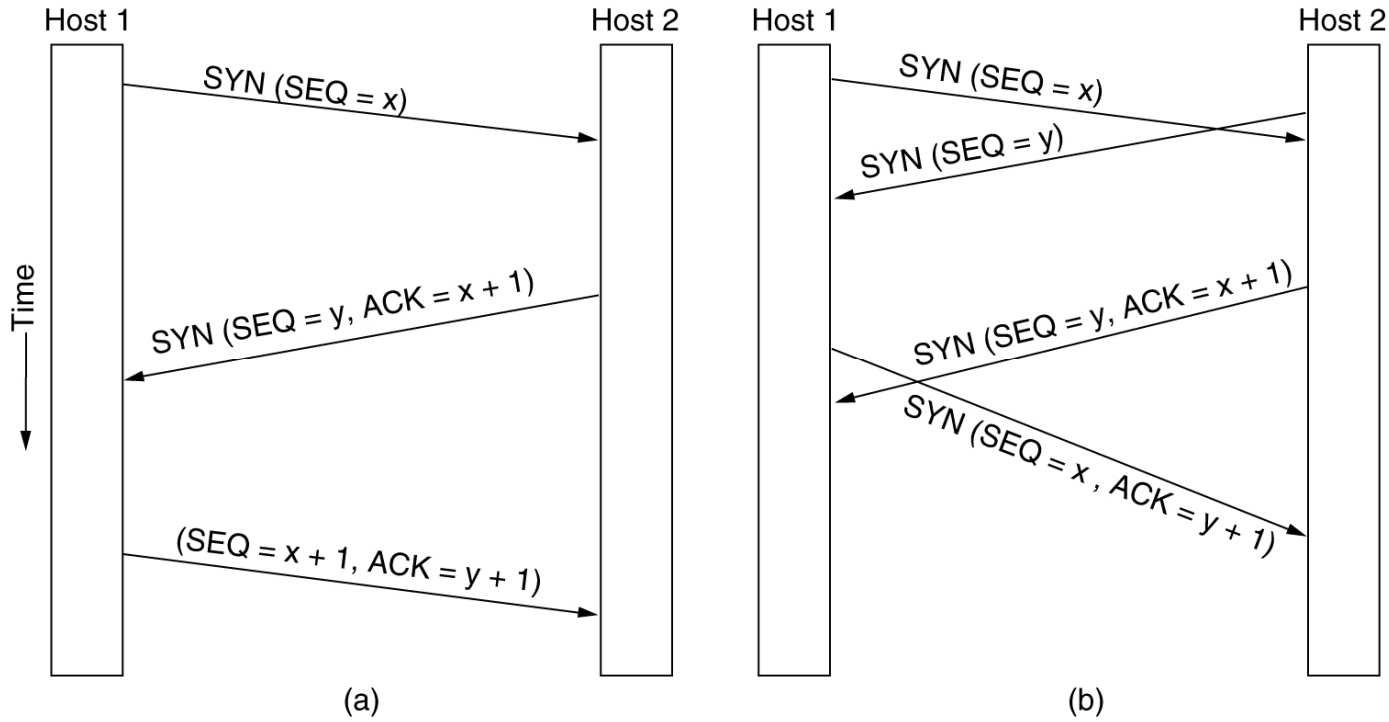
TCP Header.

The TCP Segment Header (2)



The pseudoheader included in the TCP checksum.

TCP Connection Establishment



(a) TCP connection establishment in the normal case.

(b) Call collision.

The Application Layer

The Domain Name System

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.
Also, sending e-mail to *tana@128.111.24.41* means that if Tana's ISP or organization moves the mail server to a different machine with a different IP address, her e-mail address has to change.
Consequently, ASCII names were introduced to decouple machine names from machine addresses. In this way, Tana's address might be something like *tana@art.ucsb.edu*.
Nevertheless, the network itself understands only numerical addresses, so some mechanism is
required to convert the ASCII strings to network addresses.

DNS – The Domain Name System

- The DNS Name Space
- Resource Records
- Name Servers

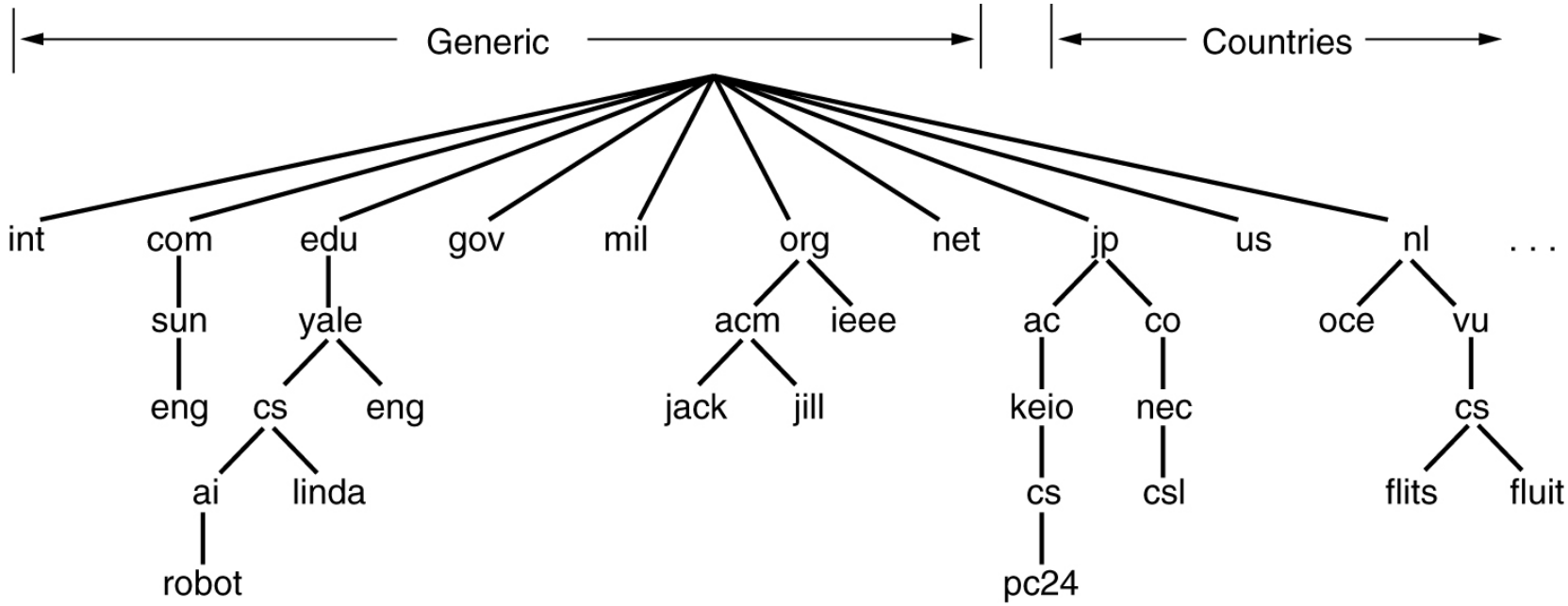
DNS

- DNS is the invention of a hierarchical, domain-based naming scheme and a distributed database system for implementing the naming scheme.

- DNS is used to map a name onto an IP address, an application program calls a library procedure called the Resolver, passing it the name as a parameter.

- The Resolver sends a UDP packet to a local DNS server, which then looks up the name and returns IP address, the program can then establish a TCP connection with the destination or send it UDP packets.

The DNS Name Space



A portion of the Internet domain name space.

DNS Name Space

- Internet is divided into 200 top-level domains, where each domain covers many hosts.
- Each domain is partitioned into subdomains, and these are further partitioned, and so on.
- Top-level domains : Generic

  Countries.
- Generic Domains were com(commercial), edu(educational institutions), gov, int(international organizations), mil, net(network providers), and org(non profit organization).
- The Country domains include one entry for every country.
- Each domain is named by the path upward from it to the (unnamed) root.

DNS(contd..)

- Domain names can be either absolute or relative.
- An absolute domain name always ends with a period (e.g.,*eng.sun.com*)
- Relative does not.
- Domain names are case insensitive.
- Domains can be inserted into the tree in two different ways.
- Each domain controls how it allocates the domains under it.
- To create a new domain, permission is required of the domain in which it will be included. So that name conflicts are avoided.
- Once domain has been created & registered, it can create subdomains, without getting permission from anybody higher up the tree.

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 single host, the most common resource record is just its IP address, but many other kinds of resource records also exist.
- When a resolver gives a domain name to DNS, it gets back resource records associated with that name. Thus, the primary function of DNS is to map domain names onto resource records.
- A resource record is a five-tuple. The format is as follows:

  Domain_name Time_to_live Class Type Value
- **Domain_name** :tells the domain to which this record applies
- **Time_to_live** field gives an indication of how stable the record is.
- **Class:** For Internet information, it is always IN

  For non-Internet information, other codes can be used.

Resource Records

Type: This field tells what kind of record this is. The important types are listed below

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

The principal DNS resource records types.

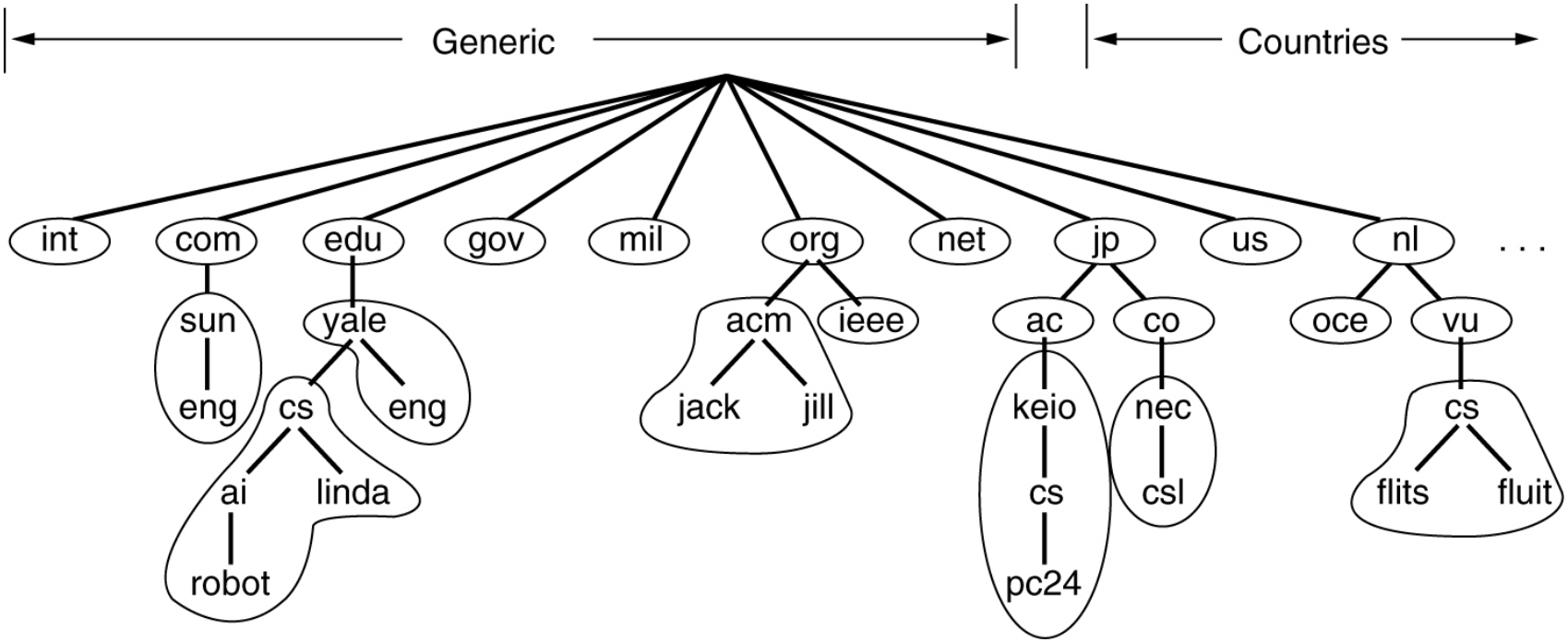
Value: This can be a number, a domain name, or an ASCII string.

Resource Records (2)

```
; Authoritative data for cs.vu.nl
cs.vu.nl.          86400   IN   SOA      star boss (952771,7200,7200,2419200,86400)
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
```

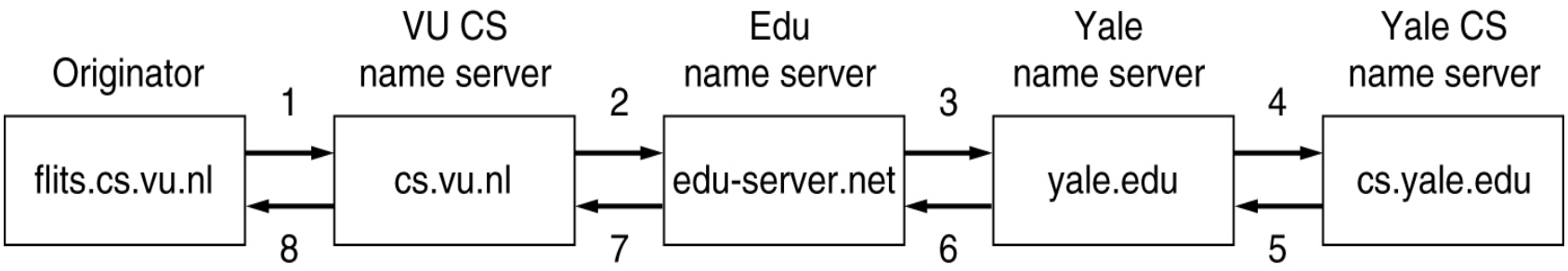
A portion of a possible DNS database for *cs.vu.nl*.

Name Servers



Part of the DNS name space showing the division into zones.

Name Servers (2)



How a resolver looks up a remote name in eight steps.

The World Wide Web

- Architectural Overview
- Static Web Documents
- Dynamic Web Documents
- HTTP – The HyperText Transfer Protocol
- Performance Ehnancements
- The Wireless Web

Architectural Overview

WELCOME TO THE UNIVERSITY OF EAST PODUNK'S WWW HOME PAGE

- Campus Information
  - Admissions information
  - Campus map
  - Directions to campus
  - The UEP student body
- Academic Departments
  - Department of Animal Psychology
  - Department of Alternative Studies
  - Department of Microbiotic Cooking
  - Department of Nontraditional Studies
  - Department of Traditional Studies

Webmaster@eastpodunk.edu

(a)

THE DEPARTMENT OF ANIMAL PSYCHOLOGY

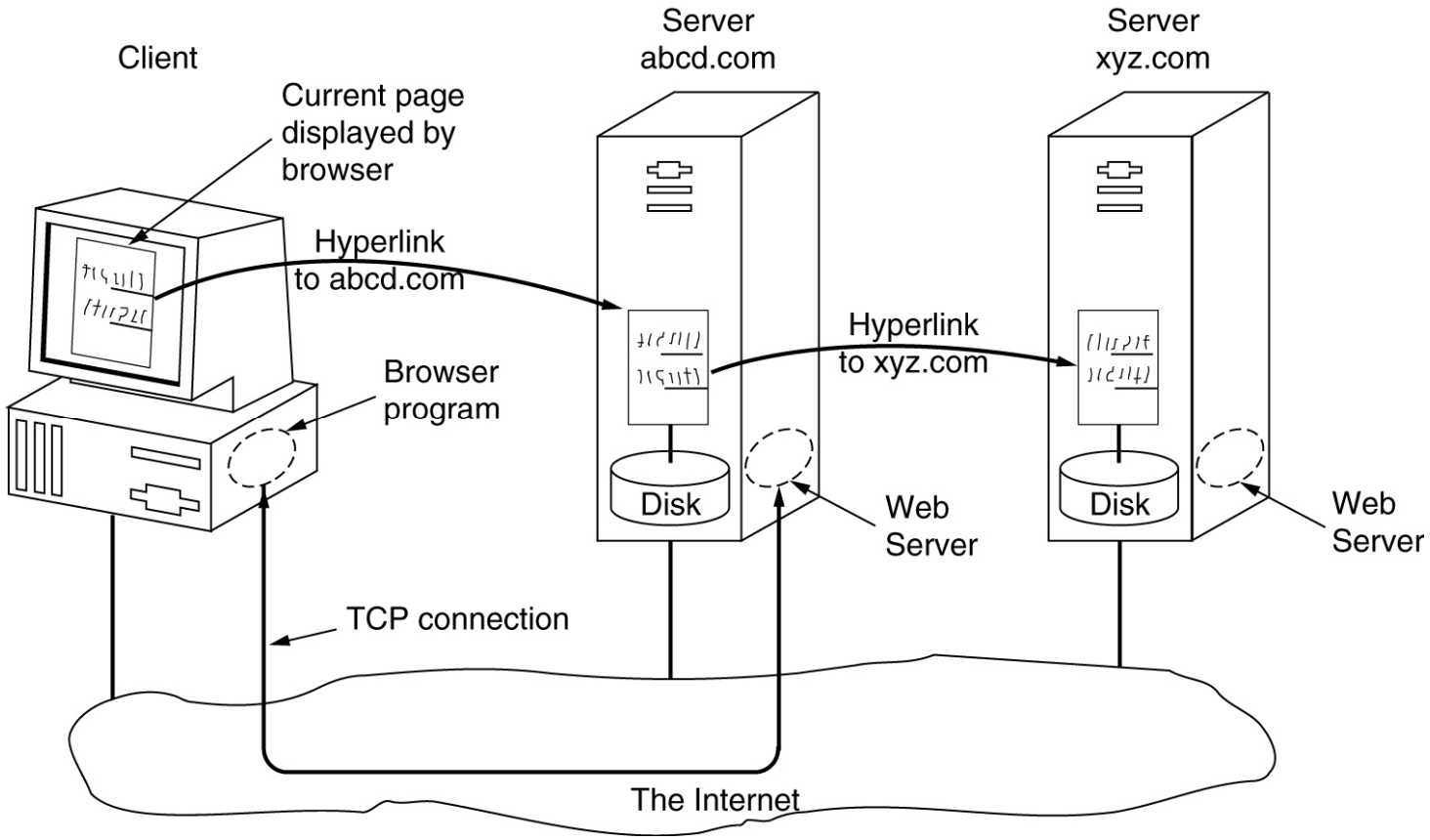
- Information for prospective majors
- Personnel
  - Faculty members
  - Graduate students
  - Nonacademic staff
- Research Projects
- Positions available
- Our most popular courses
  - Dealing with herbivores
  - Horse management
  - Negotiating with your pet
  - User-friendly doghouse construction
- Full list of courses

Webmaster@animalpsyc.eastpodunk.edu

(b)

(a) A Web page (b) The page reached by clicking on Department of Animal Psychology.

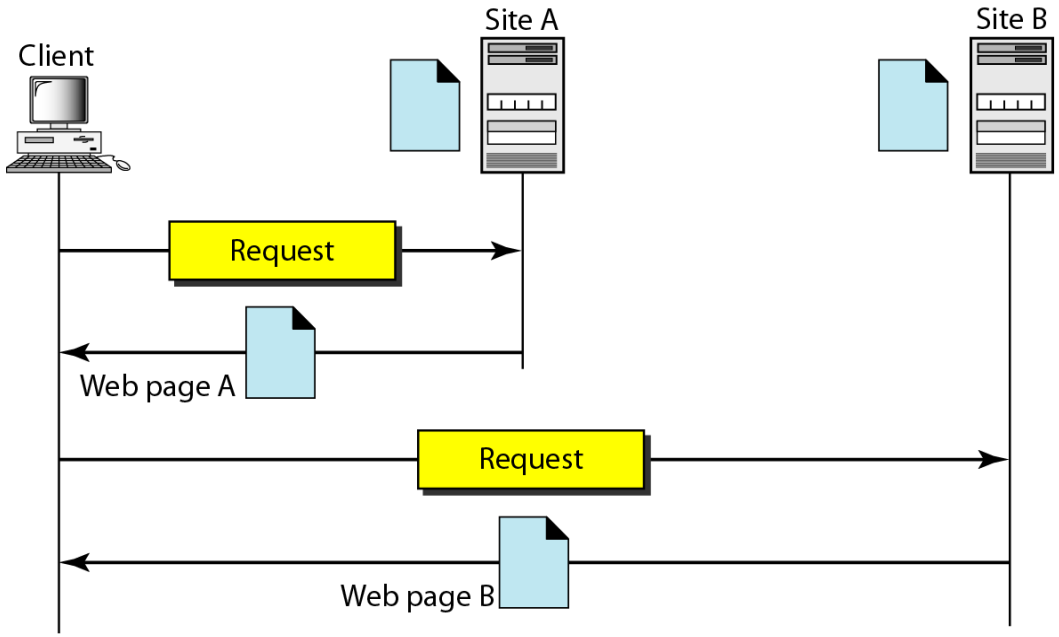
Architectural Overview (2)



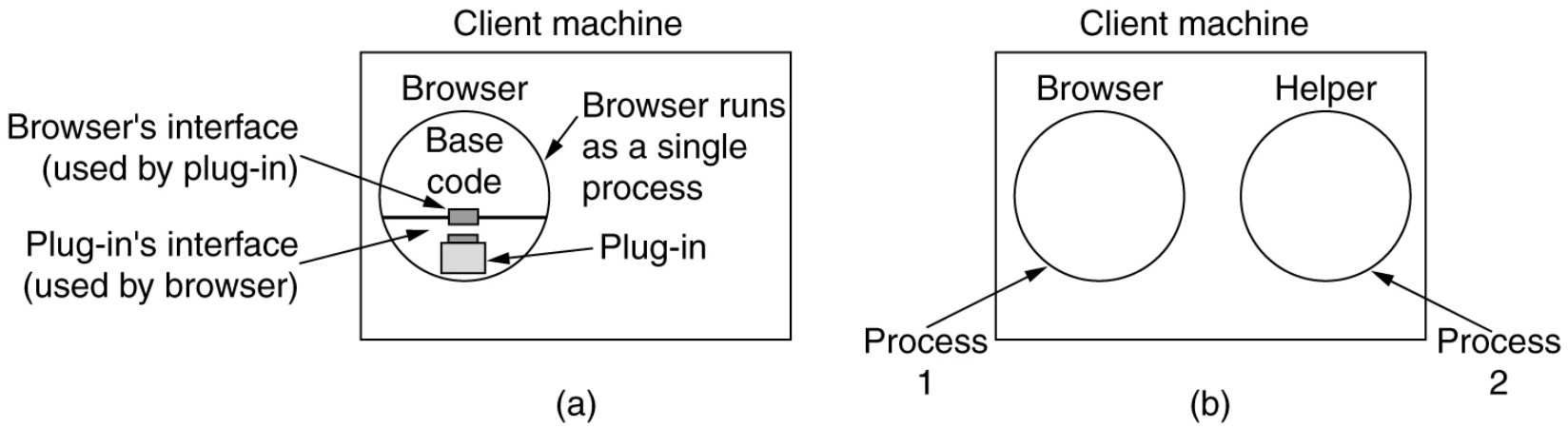
The parts of the Web model.

Architecture of WWW

The WWW today is a distributed client/server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called sites.

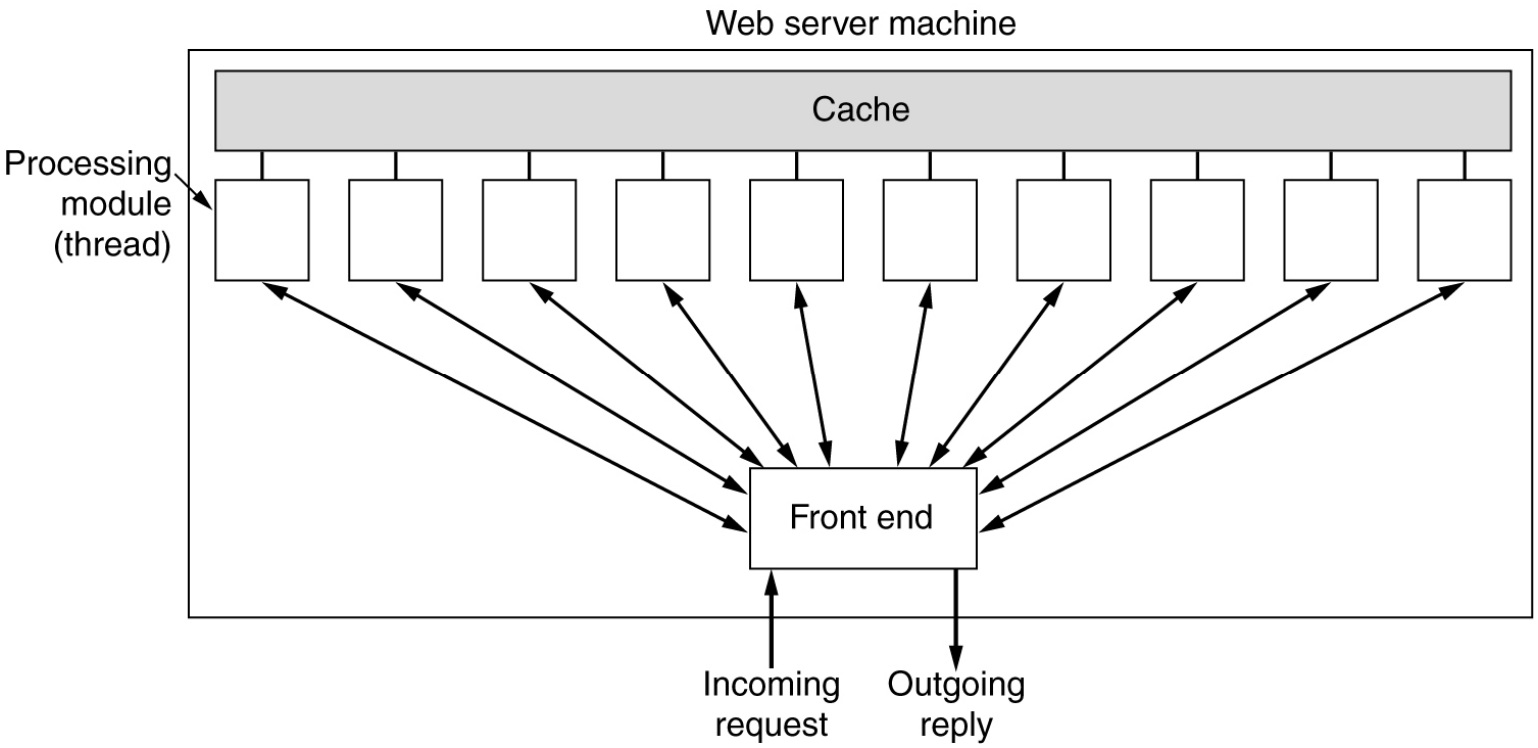


The Client Side



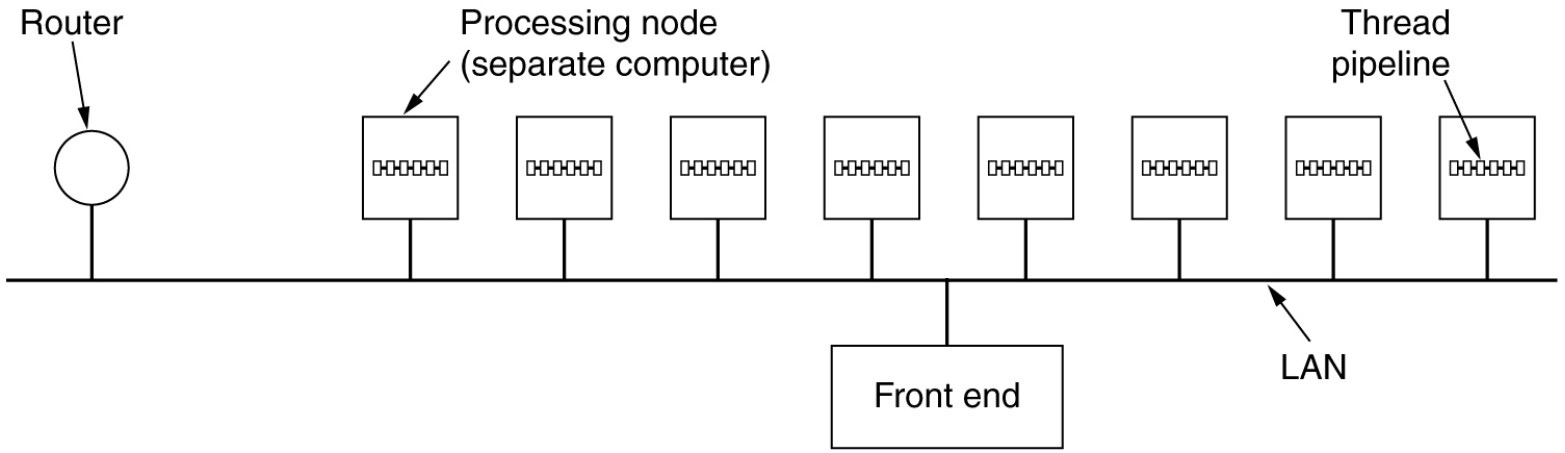
(a) A browser plug-in. (b) A helper application.

The Server Side



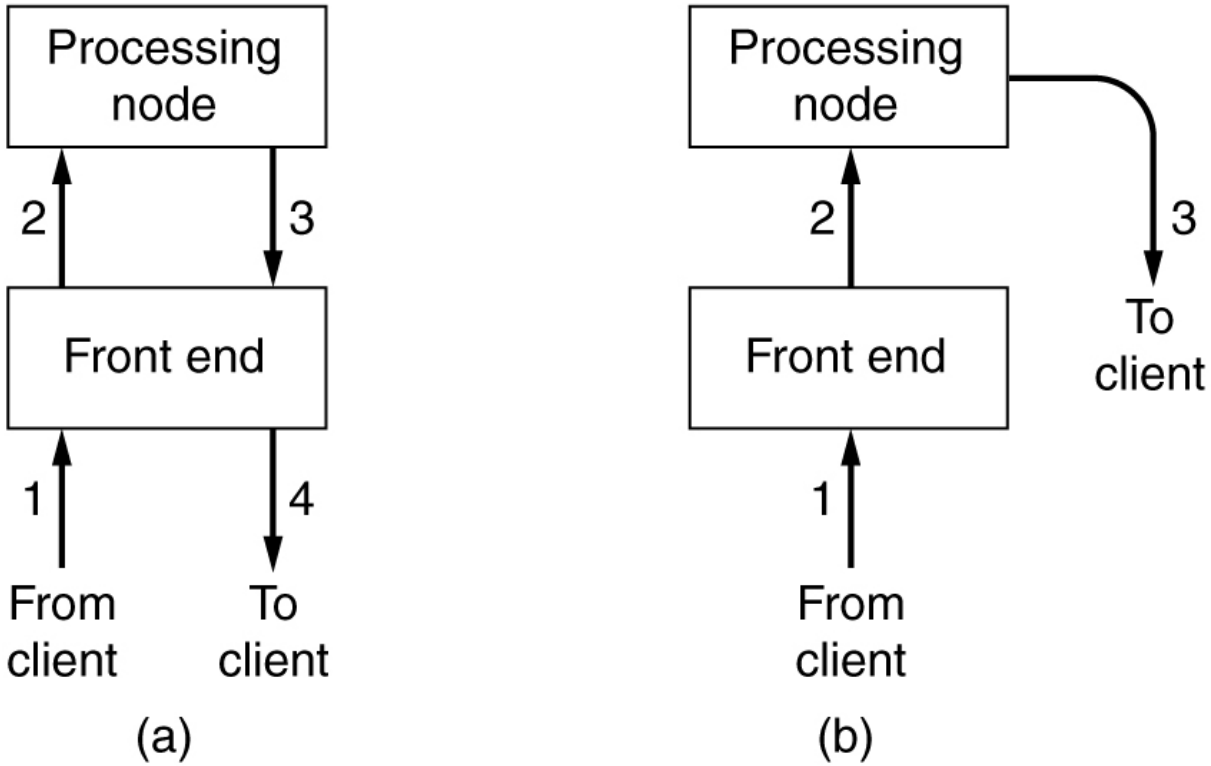
A multithreaded Web server with a front end and processing modules.

The Server Side (2)



A server farm.

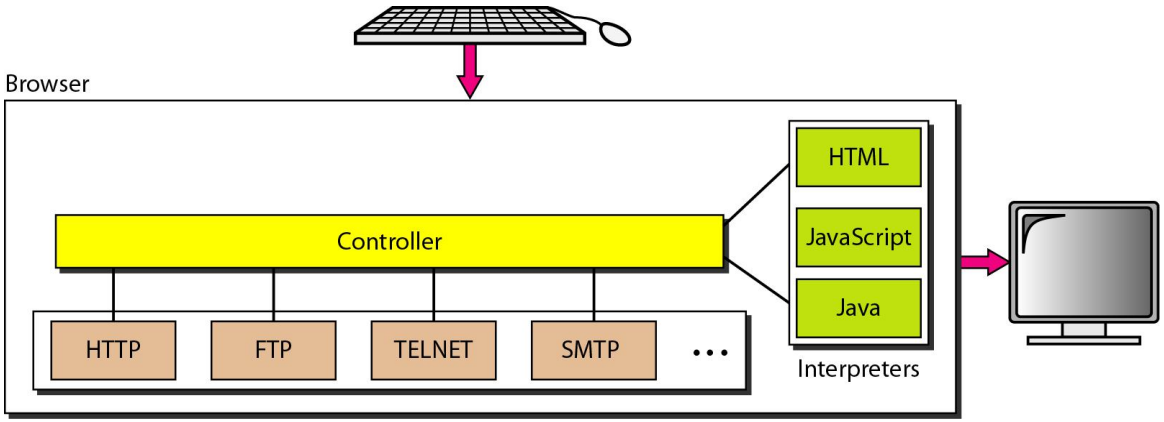
The Server Side (3)



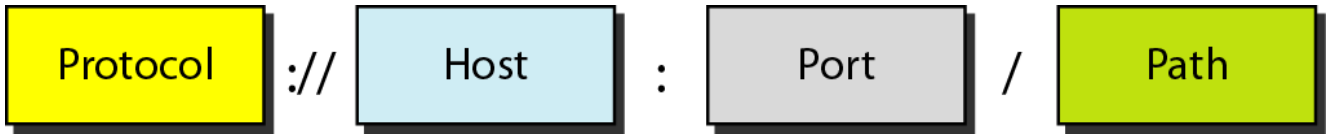
(a) Normal request-reply message sequence.

(b) Sequence when TCP handoff is used.

Browser and URL



Uniform Resource Locator



•

URLs – Uniform Resource Locaters

Name	Used for	Example
http	Hypertext (HTML)	http://www.cs.vu.nl/~ast/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
news	Newsgroup	news:comp.os.minix
news	News article	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Sending e-mail	mailto:JohnUser@acm.org
telnet	Remote login	telnet://www.w3.org:80

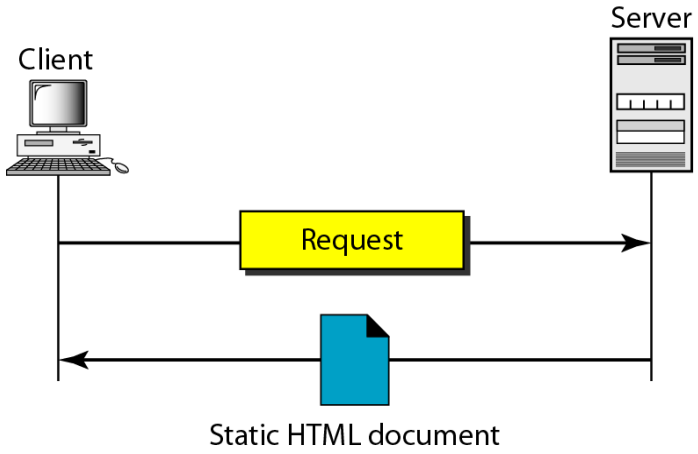
Some common URLs.

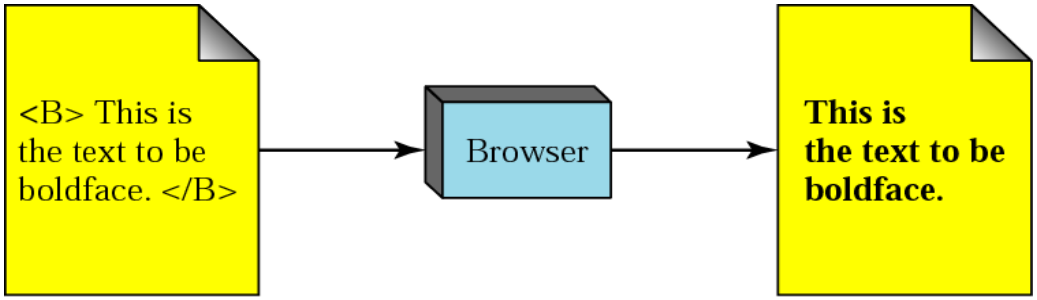
Statelessness and Cookies

Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jets	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No

Some examples of cookies.

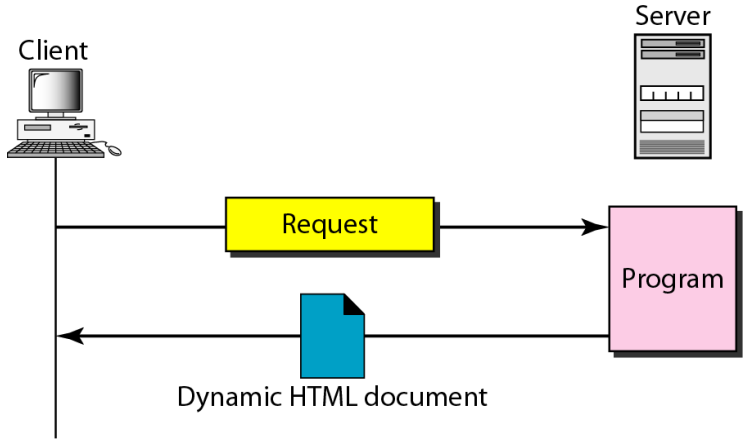
Static Documents: HTML



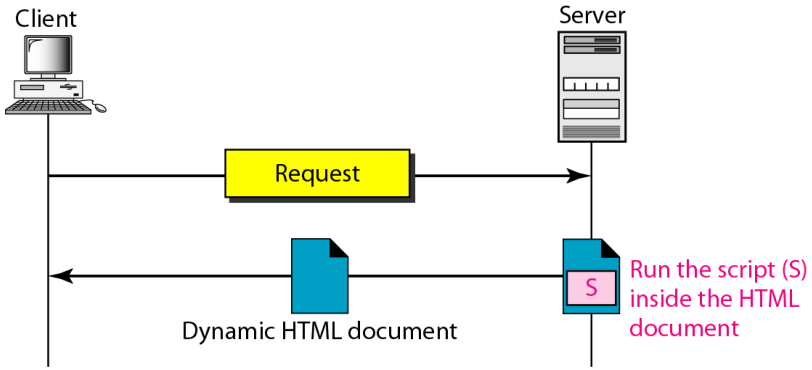


Dynamic Documents: CGI and Scripting

- Common Gateway Interface

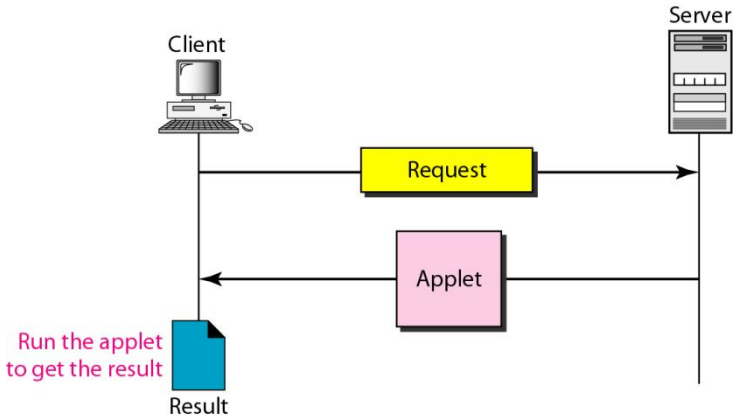


- Server-site Script

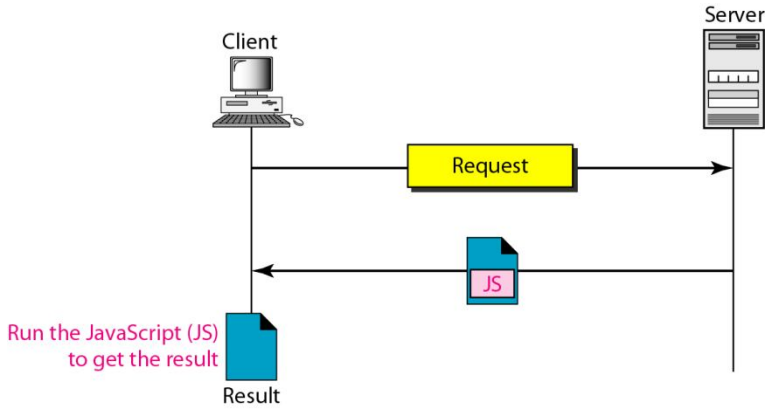


Active Documents: Java Applets and JavaScript

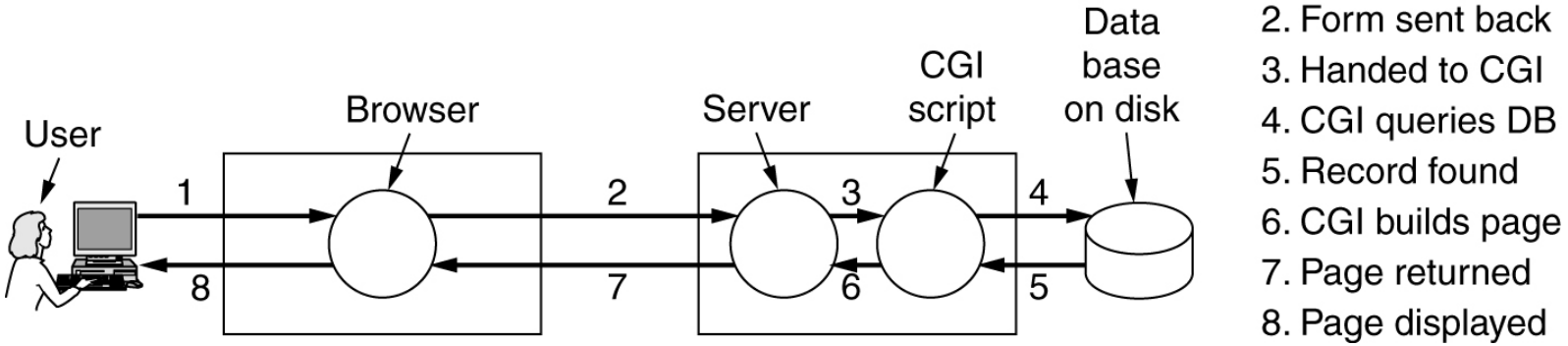
- Active document using Java applet



- Active document using client-site



Dynamic Web Documents



Steps in processing the information from an HTML form.

Dynamic Web Documents (2)

```
<html>
<body>

<h2> This is what I know about you </h2>
<?php echo $HTTP_USER_AGENT ?>

</body>
</html>
```

A sample HTML page with embedded PHP.

Dynamic Web Documents (3)

```
<html>
<body>
<form action="action.php" method="post">
<p> Please enter your name: <input type="text" name="name"> </p>
<p> Please enter your age: <input type="text" name="age"> </p>
<input type="submit">
</form>
</body>
</html>
```

(a)

```
<html>
<body>
<h1> Reply: </h1>
Hello <?php echo $name; ?>.
Prediction: next year you will be <?php echo $age + 1; ?>
</body>
</html>
```

(b)

```
<html>
<body>
<h1> Reply: </h1>
Hello Barbara.
Prediction: next year you will be 25
</body>
</html>
```

(c)

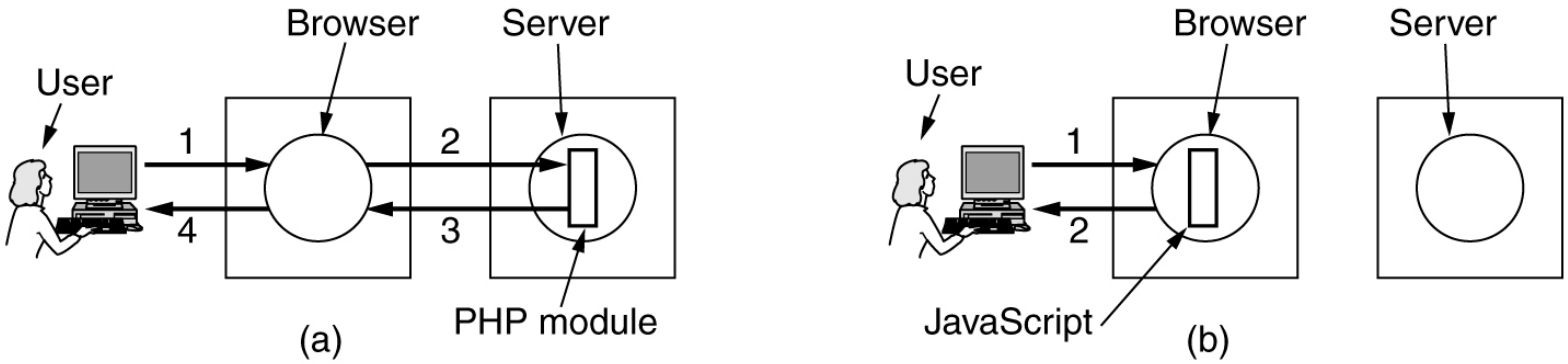
(a) A Web page containing a form. (b) A PHP script for handling the output of the form. (c) Output from the PHP script when the inputs are "Barbara" and 24 respectively.

Client-Side Dynamic Web Page Generation

Use of JavaScript for processing a form.

```
<head>
<script language="javascript" type="text/javascript">
function response(test  form) {
    var person = test  form.name.value;
    var years = eval(test  form.age.value) + 1;
    document.open();
    document.writeln("<html> <body>");
    document.writeln("Hello " + person + ".<br>");
    document.writeln("Prediction: next year you will be " + years + ".");
    document.writeln("</body> </html>");
    document.close();
}
</script>
</head>

<body>
<form>
Please enter your name: <input type="text" name="name">
<p>
Please enter your age: <input type="text" name="age">
<p>
<input type="button" value="submit" onclick="response(this.form)">
</form>
</body>
</html>
```

Client-Side Dynamic Web Page Generation (2)



(a) Server-side scripting with PHP.

(b) Client-side scripting with JavaScript.

Client-Side Dynamic Web Page Generation (3)

```
<html>
<head>
<script language="javascript" type="text/javascript">

function response(test_form) {
   function factorial(n) {if (n == 0) return 1; else return n * factorial(n – 1);}
   var r = eval(test_form.number.value);       //  r = typed in argument
   document.myform.mytext.value = "Here are the results.\n";
   for (var i = 1; i <= r; i++)                          // print one line from 1 to r
      document.myform.mytext.value += (i + "! =  " + factorial(i) + "\n");
}
</script>
</head>

<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute table of factorials" onclick="response(this.form)">
<p>
<textarea name="mytext" rows=25 cols=50> </textarea>
</form>
</body>
</html>
```

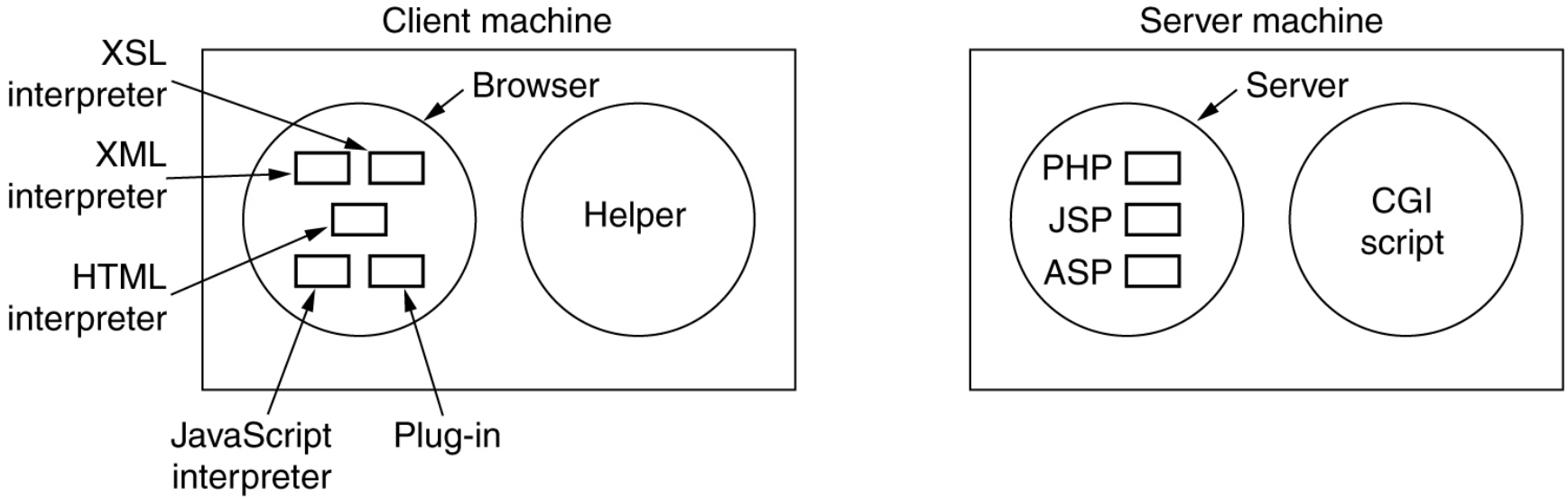
A JavaScript program for computing and printing factorials.

Client-Side Dynamic Web Page Generation (4)

```
<html>
<head>
<script language="javascript" type="text/javascript">
if (!document.myurl) document.myurl = new Array();
document.myurl[0] = "http://www.cs.vu.nl/~ast/im/kitten.jpg";
document.myurl[1] = "http://www.cs.vu.nl/~ast/im/puppy.jpg";
document.myurl[2] = "http://www.cs.vu.nl/~ast/im/bunny.jpg";
function pop(m) {
    var urx = "http://www.cs.vu.nl/~ast/im/cat.jpg";
    popupwin = window.open(document.myurl[m],"mywind","width=250,height=250");
}
</script>
</head>

<body>
<p> <a href="#" onMouseover="pop(0); return false;" > Kitten  </a> </p>
<p> <a href="#" onMouseover="pop(1); return false;" > Puppy  </a> </p>
<p> <a href="#" onMouseover="pop(2); return false;" > Bunny  </a> </p>
</body>
</html>
```

An interactive Web page that responds to mouse movement.

Client-Side Dynamic Web Page Generation (5)

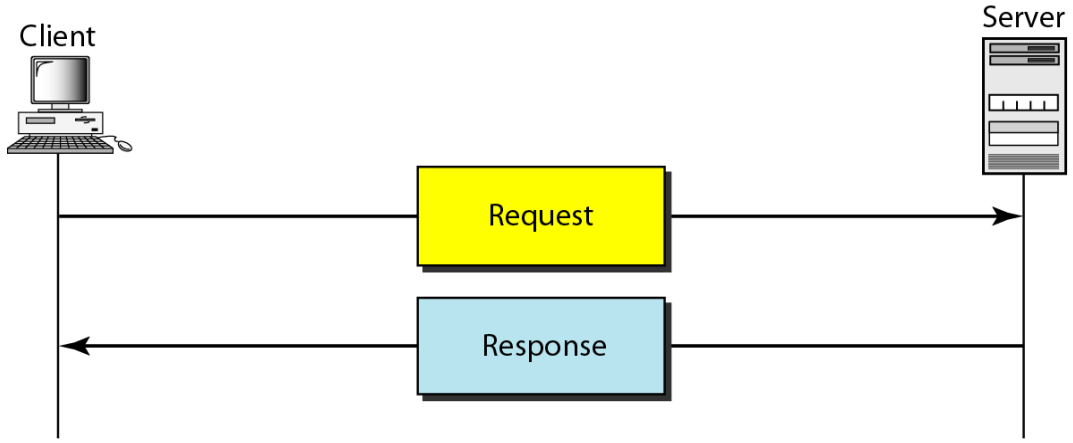


The various ways to generate and display content.

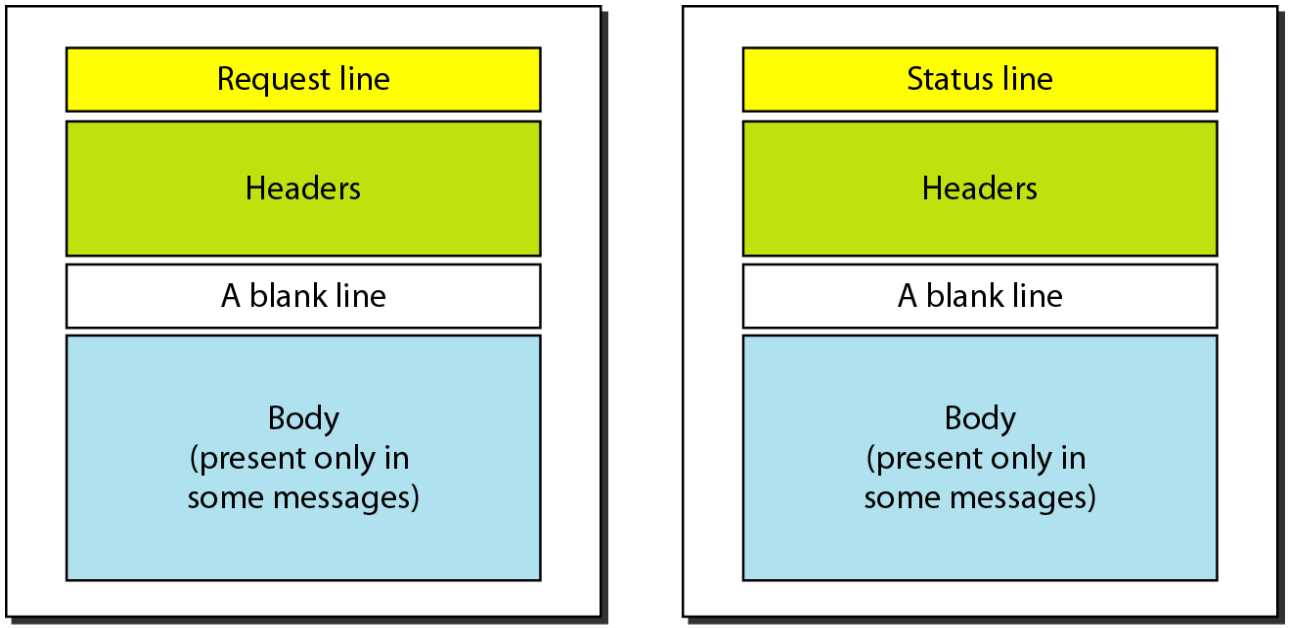
Hypertext Transfer Protocol (HTTP)

HTTP is used mainly to access data on the WWW

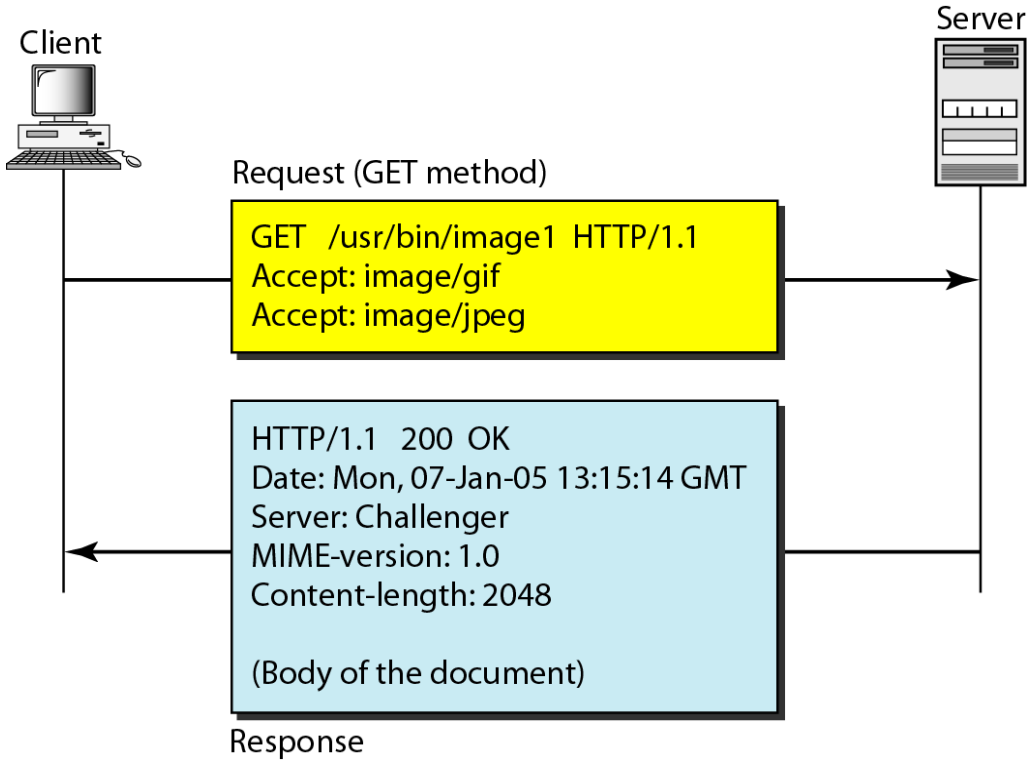
HTTP uses the services of TCP on well-known port 80



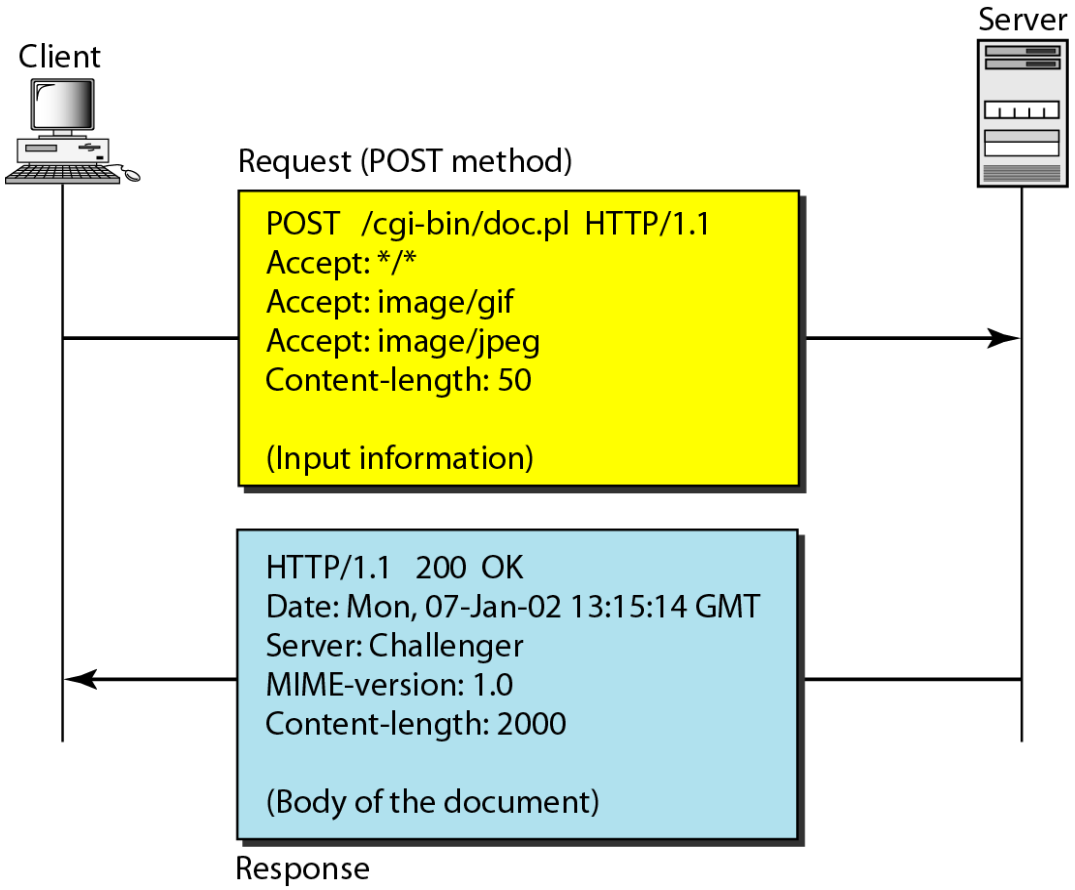
Request and Response Message



Example 1


Client
Server
Request (GET method)
GET /usr/bin/image1 HTTP/1.1
Accept: image/gif
Accept: image/jpeg
HTTP/1.1 200 OK
Date: Mon, 07-Jan-05 13:15:14 GMT
Server: Challenger
MIME-version: 1.0
Content-length: 2048
(Body of the document)
Response

Example 2


Client
Server
Request (POST method)
POST /cgi-bin/doc.pl HTTP/1.1
Accept: */*
Accept: image/gif
Accept: image/jpeg
Content-length: 50
(Input information)
HTTP/1.1 200 OK
Date: Mon, 07-Jan-02 13:15:14 GMT
Server: Challenger
MIME-version: 1.0
Content-length: 2000
(Body of the document)
Response

HTTP Methods

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

The built-in HTTP request methods.

HTTP Methods (2)

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

The status code response groups.

HTTP Message Headers

Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Cookie	Request	Sends a previously set cookie back to the server
Date	Both	Date and time the message was sent
Upgrade	Both	The protocol the sender wants to switch to
Server	Response	Information about the server
Content-Encoding	Response	How the content is encoded (e.g., gzip)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Last-Modified	Response	Time and date the page was last changed
Location	Response	A command to the client to send its request elsewhere
Accept-Ranges	Response	The server will accept byte range requests
Set-Cookie	Response	The server wants the client to save a cookie

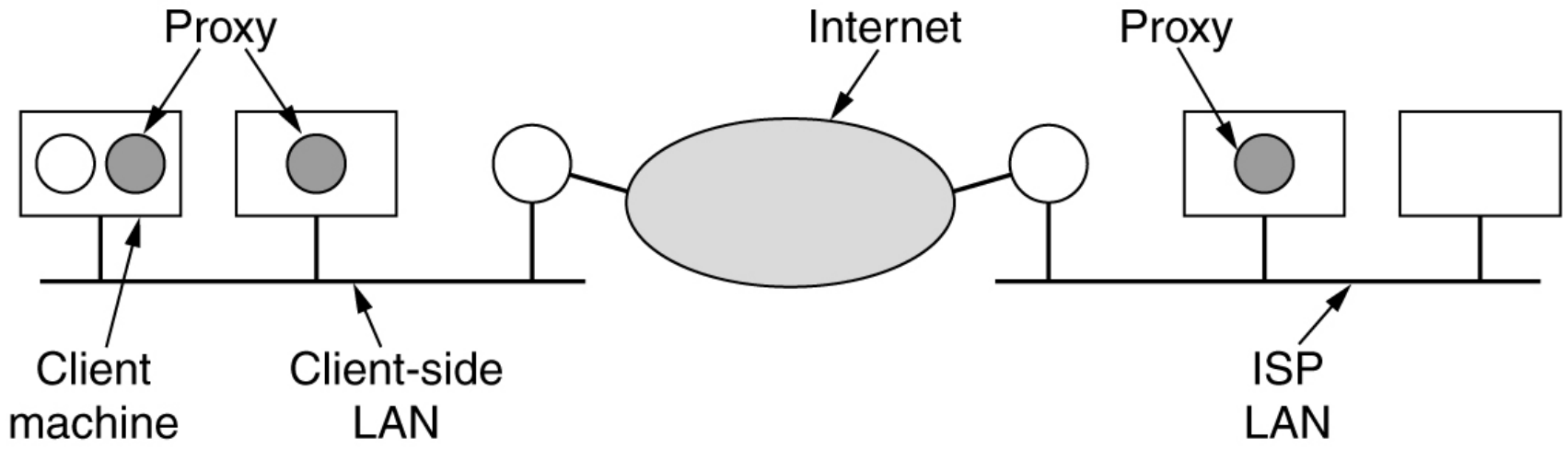
Some HTTP message headers.

Example HTTP Usage

The start of the output of *www.ietf.org/rfc.html*.

```
Trying 4.17.168.6...
Connected to www.ietf.org.
Escape character is '^]'.
HTTP/1.1 200 OK
Date: Wed, 08 May 2002 22:54:22 GMT
Server: Apache/1.3.20 (Unix) mod_ssl/2.8.4 OpenSSL/0.9.5a
Last-Modified: Mon, 11 Sep 2000 13:56:29 GMT
ETag: "2a79d-c8b-39bce48d"
Accept-Ranges: bytes
Content-Length: 3211
Content-Type: text/html
X-Pad: avoid browser bug

<html>
<head>
<title>IETF RFC Page</title>

<script language="javascript">
function url() {
 var x = document.form1.number.value
 if (x.length == 1) {x = "000" + x }
 if (x.length == 2) {x = "00" + x }
 if (x.length == 3) {x = "0" + x }
 document.form1.action = "/rfc/rfc" + x + ".txt"
 document.form1.submit
}
</script>

</head>
```

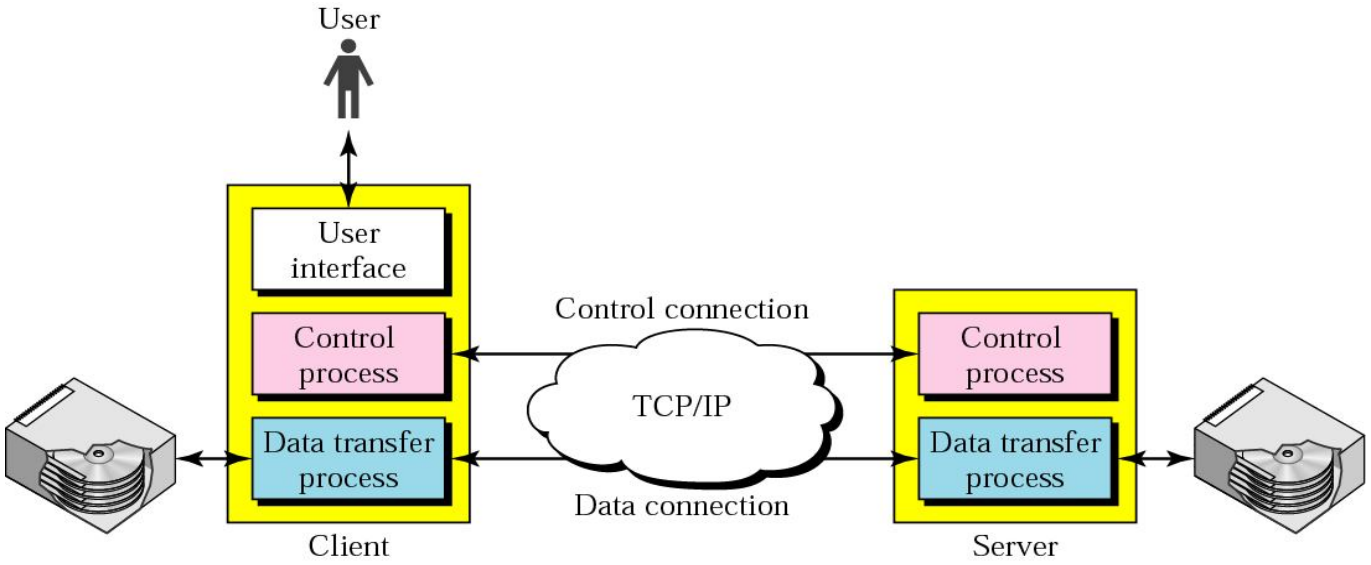
Caching



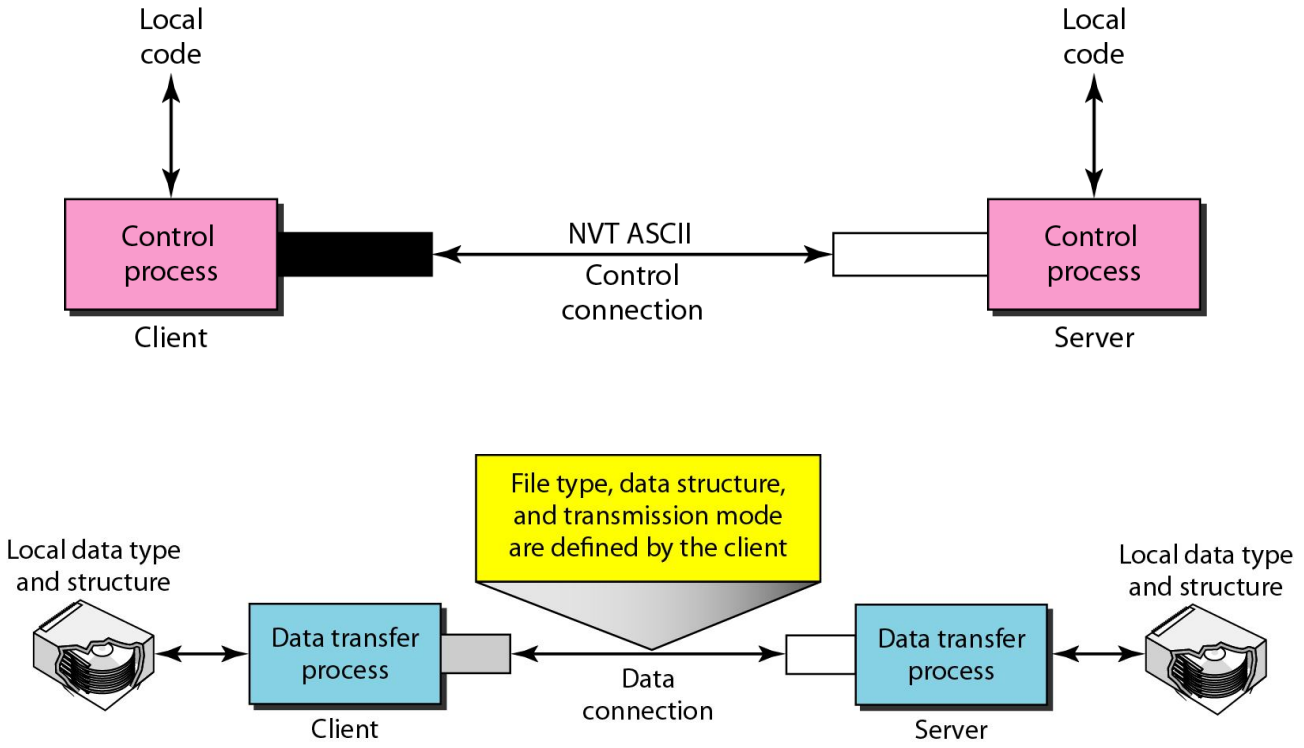
Hierarchical caching with three proxies.

File Transfer

FTP uses the services of TCP. It needs two TCP connections. The well-known port 21 is used for the control connection, and the well-known port 20 is used for the data connection.

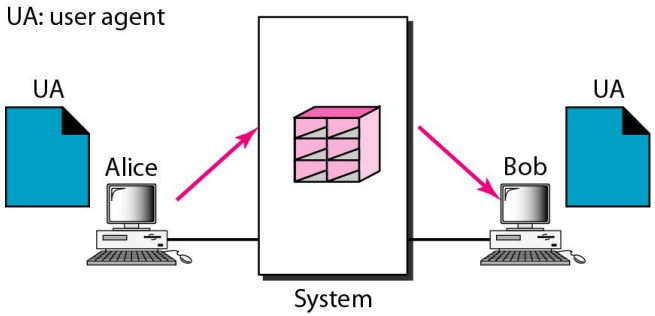


Communication over Two Connections

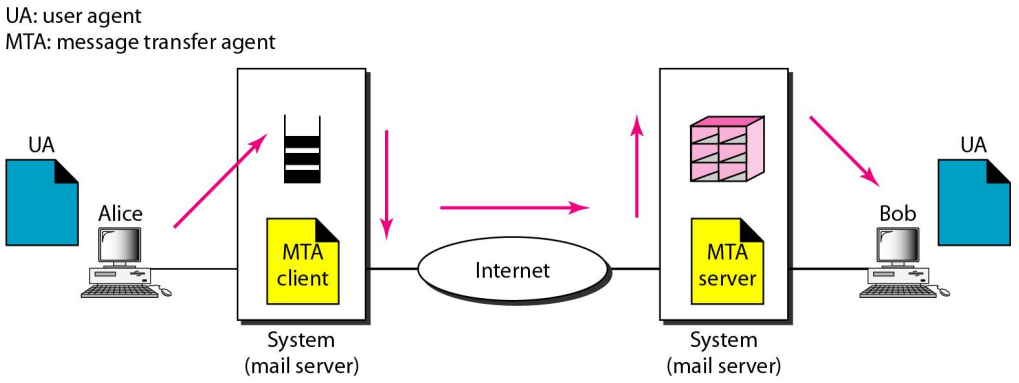


Electronic Mail

When the sender and the receiver of an e-mail are on the same system, we need only two user agents.

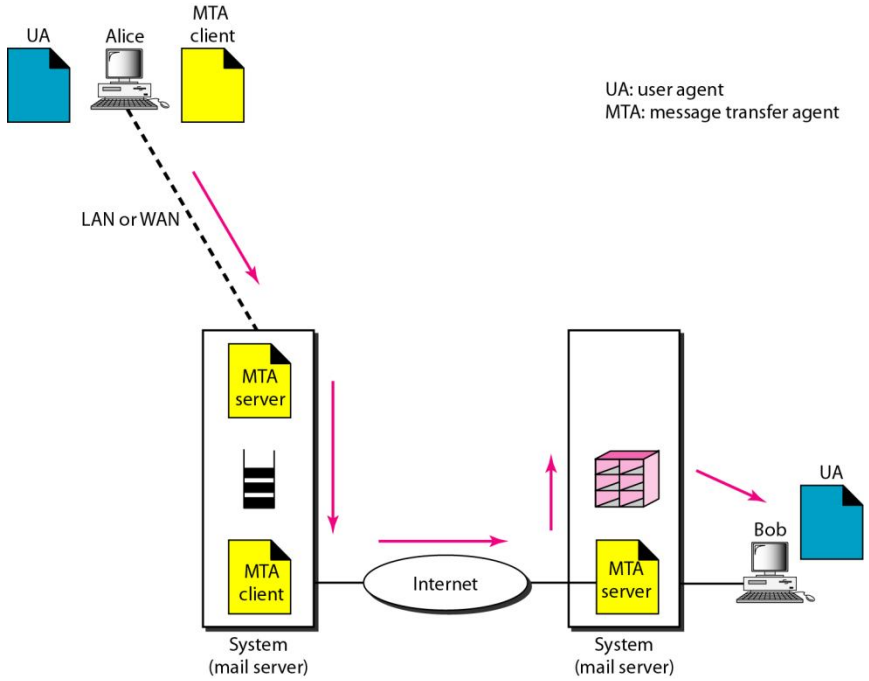


- When the sender and the receiver of an e-mail are on different systems, we need two UAs and a pair of MTAs (client and server).



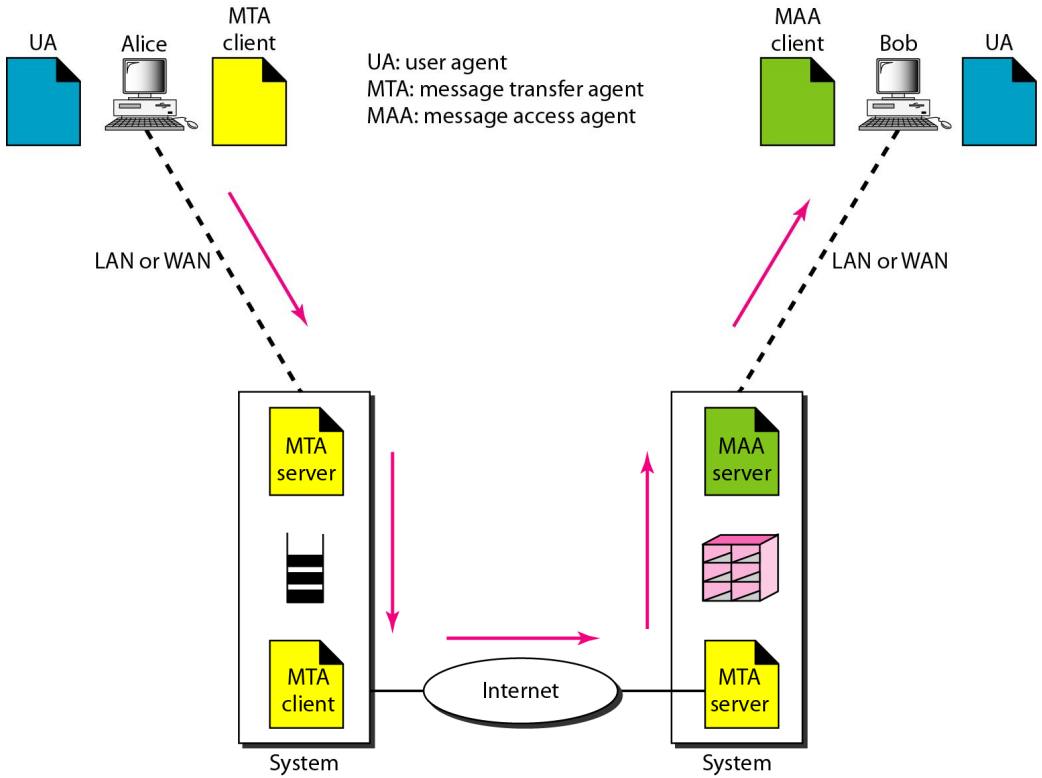
Electronic Mail

When the sender is connected to the mail server via a LAN or a WAN, we need two UAs and two pairs of MTAs (client and server).



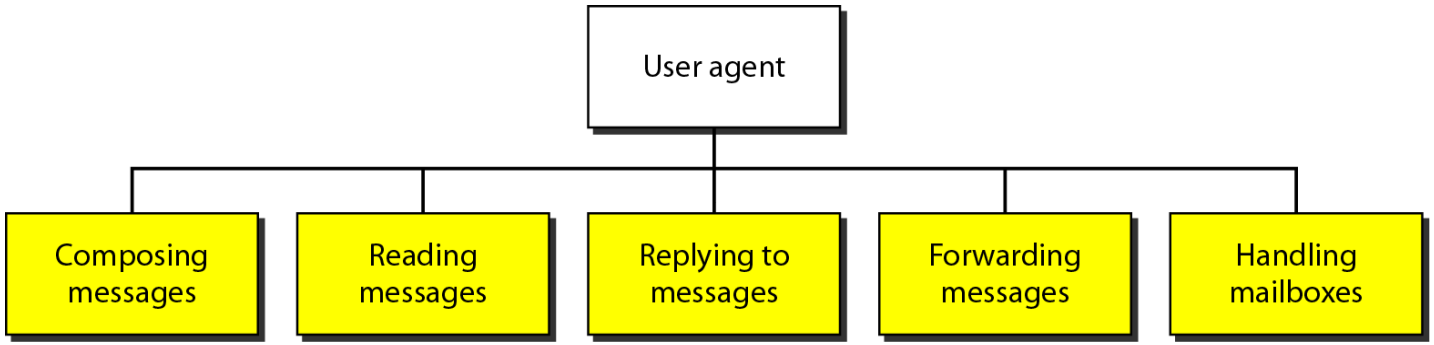
Electronic Mail

When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two UAs, two pairs of MTAs and a pair of MAAs. This is the most common situation today.



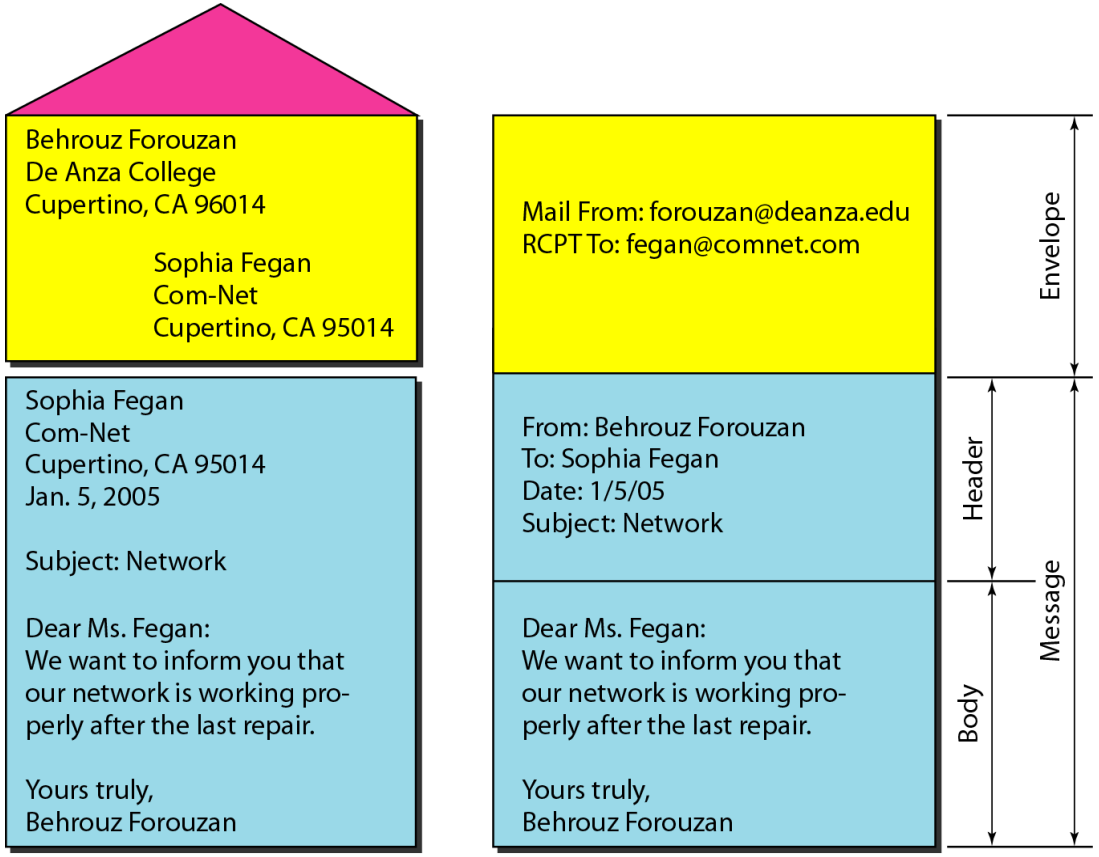
User Agent

Services provided by a user agent



- Command-driven user agents: mail, pine, and elm.
- GUI-based user agents: Eudora, Outlook, and Netscape.

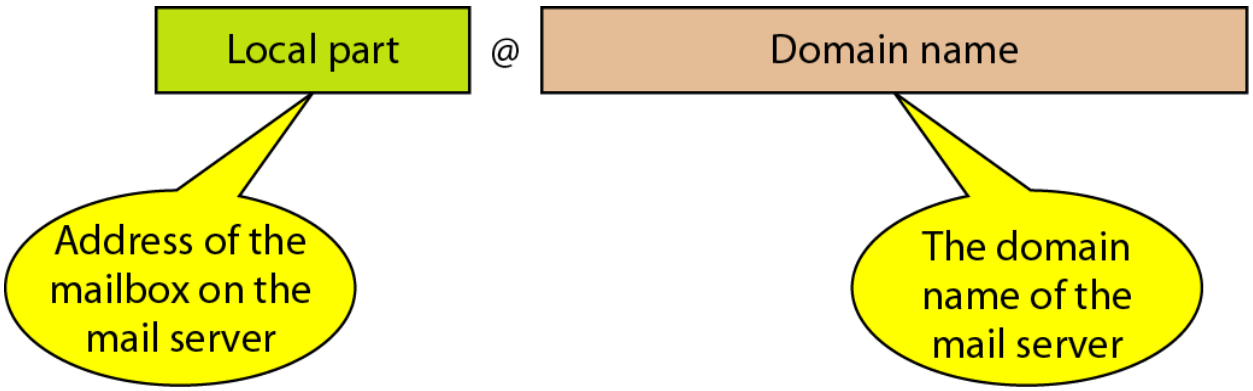
Format of an E-mail



a. Postal mail

b. Electronic mail

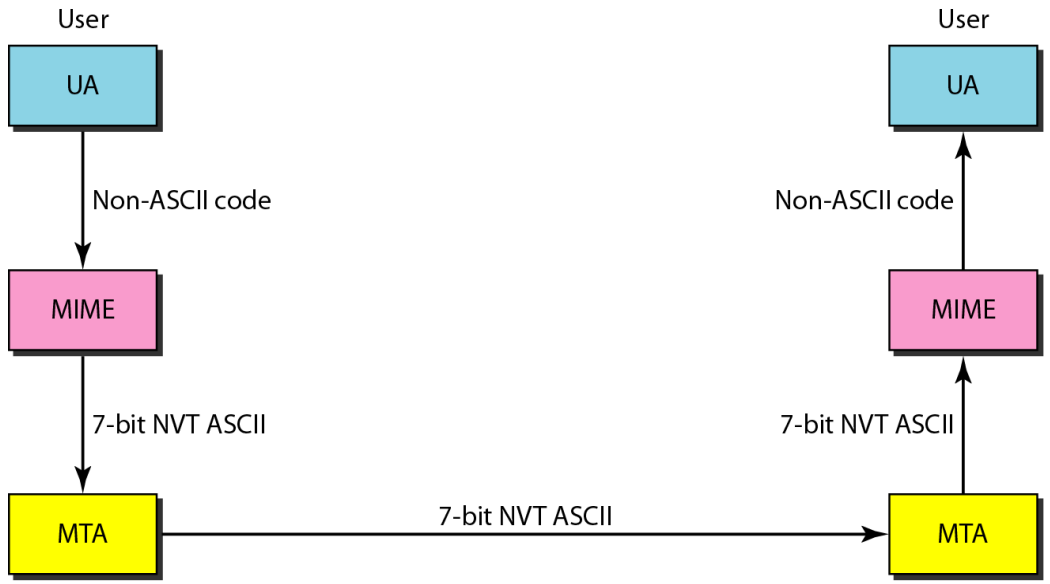
Email address



MIME

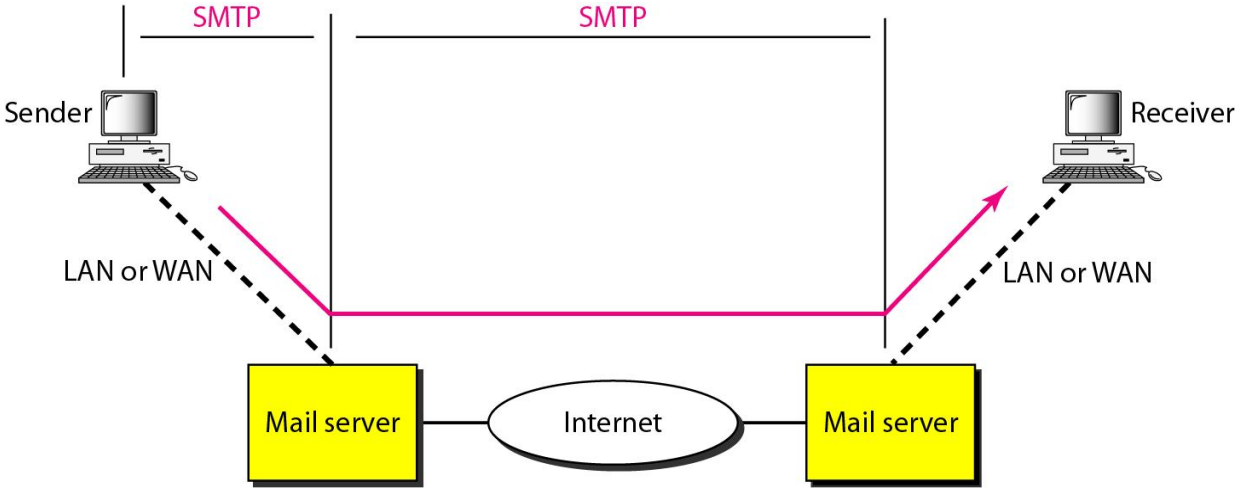
Multipurpose Internet mail Extensions (MIME)

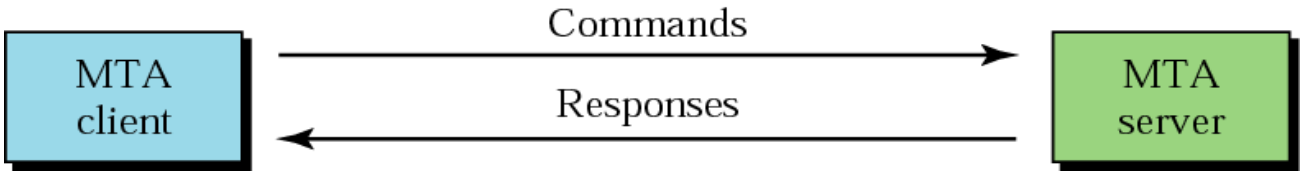
Supplementary protocol that allows non-ASCII data to be sent through SMTP



Mail Transfer Agent (MTA): SMTP

The actual mail transfer is done through MTA





SMTP Functioning

Three phases of transfer:

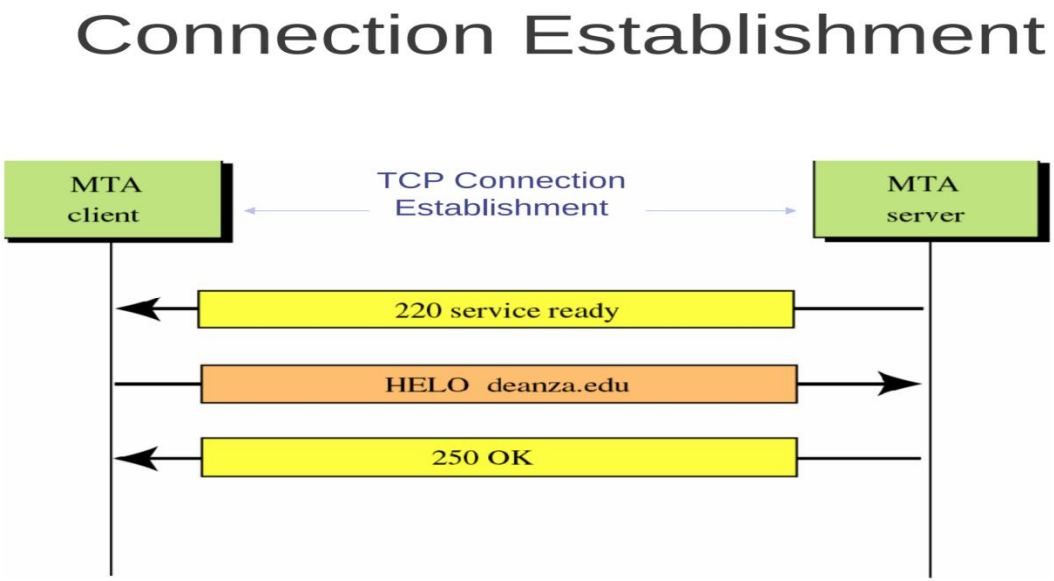
- handshaking: Connection establishment
- transfer: direct transfer from sending server (client) to receiving server (server); pushbased: client sends data instead of server
- closure: Connection termination

SMTP

SMTP clients and servers have two main components

- User Agents – Prepares the message, encloses it in an envelope.
- Mail Transfer Agent (MTA) – Transfers the mail across the internet
- To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA

.



Mail Access Agent: POP and IMAP

The third stage: pull protocol (SMTP is a push protocol for the first/second stages)

Two mail access protocols

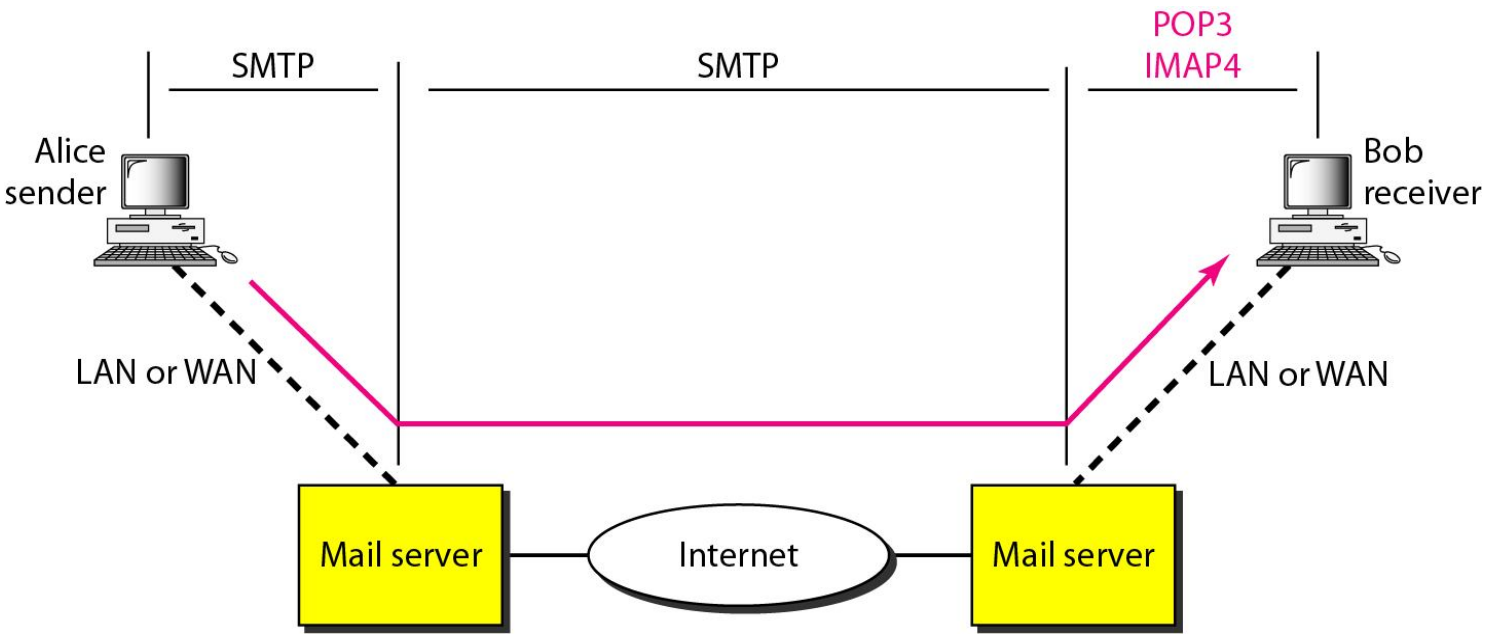
- Post Office Protocol, version 3 (POP3)
- Internet Mail Access Protocol, version 4 (IMAP4)

POP3 is simple and limited in functionality

IMAP4 is similar to POP3, but has more features with extra functions

- A user can check the email header prior to downloading
- A user can search the contents of the email for a specific string of characters prior to downloading
- A user can create, delete, or rename mailboxes on the mail server
- A user can create a hierarchy of mailboxes in a folder for email storage

POP3 and IMAP4

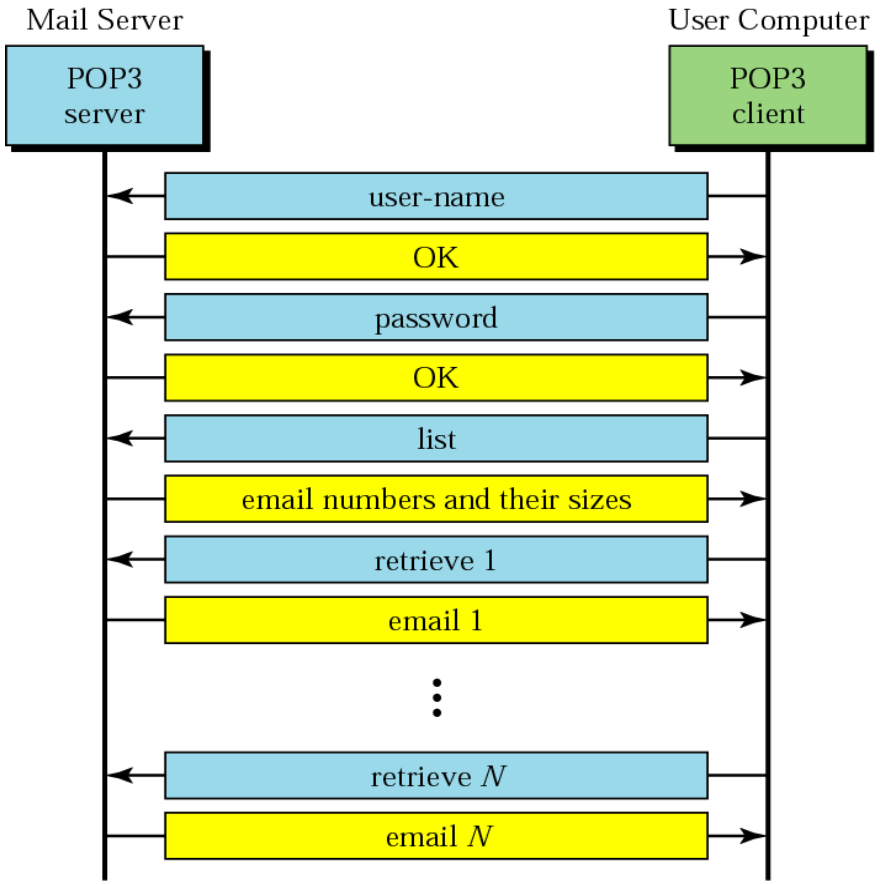


POP3

Post Office Protocol (POP3)

- 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 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.

POP3


Mail Server
User Computer
POP3 server
POP3 client
user-name
OK
password
OK
list
email numbers and their sizes
retrieve 1
email 1
retrieve N
email N

IMAP

IMAP

- Developed after POP and attempts to fix POP deficiencies
- allows keeping all mail on the server
- allows mail categorization via folder metaphor
- mail is easily flagged (answered, draft, deleted, seen, recent); this isn’t the same on all servers
- provides for multiple connections to the server

POP3 vs IMAP4

- With IMAP4, all your mail stays on the server in multiple folders, some of which you have created
- With POP3 you only have one folder, the Inbox folder. When you open your mailbox, new mail is moved from the host server and saved on your computer. If you want to be able to see your old mail messages, you have to go back to the computer where you last opened your mail
- With POP3 "leave mail on server" only your email messages are on the server, but with IMAP your email folders are also on the server