COMPUTER NETWORKS

18CS46

UNIT 5

Congestion Control Algorithms

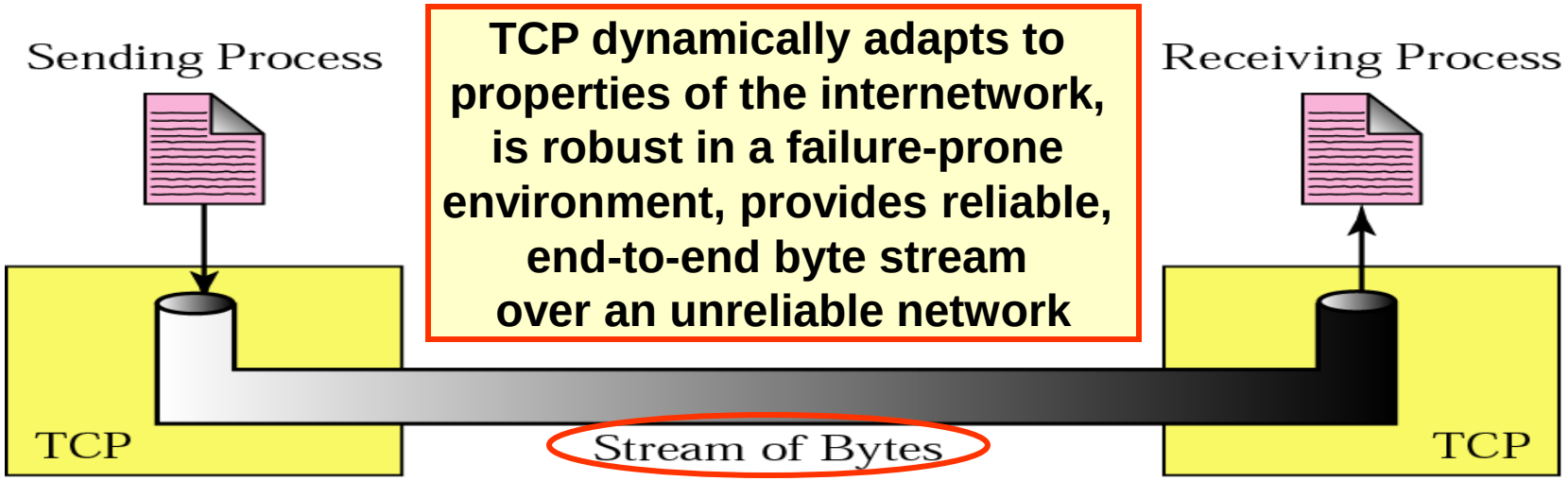
Dr. Minal Moharir

Computer Networks

Transport Layer (Transmission Control Protocol)

Internet Transport Protocols : TCP (RFC 1122 / 1323)

used for most Internet applications requiring *sequential* and *reliable* delivery called stream-connection oriented and reliable transport protocol



TCP is the main workhorse of the Internet

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

each machine supporting TCP has a TCP transport entity

what is a transport entity ?

- it is a library procedure
- a user process or
- a part of the OS kernel

what does a transport entity do ?

it manages streams and interfaces to IP layer

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

how does a TCP entity manage streams ?

- **accepts data streams from host processes**
- **breaks them to smaller units**
  (often 1460 bytes)
- **sends units as separate IP datagrams**
- **receiving TCP entity reconstructs arriving datagrams into original byte streams**

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

what the TCP must do ? :

- **retransmit data based on timing out, since, IP does not guarantee proper delivery**
- **reassemble datagrams (that may arrive in the wrong order) into messages inn the proper sequence**
- **furnish the reliability that most users want and that IP does not provide**

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP service model

TCP service is obtained by both sender and receiver

uses end points called sockets

socket contains { IP address + port number } (TSAP, 16-bits)

connection must be *explicitly* established between a socket on sending machine and a socket on receiving machine (#)

(#) Socket calls : SOCKET, BIND, LISTEN, ACCEPT, CONNECT, SEND, RECEIVE, CLOSE

Computer Networks
Transport Layer
Internet Transport Protocols : TCP

Well-known Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
67	BOOTP	Bootstrap Protocol
80	HTTP	Hypertext Transfer Protocol
110	POP-3	Remote e-mail access

More than 300 ports assigned ; details at *www.iana.org*

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP service model

features of a TCP connection :

- **full-duplex i.e. traffic can go in both directions at the same time**
- **point-to-point i.e. each connection has exactly two end points**
- **does not support multicasting or broadcasting**
- **is a byte stream, not a message stream**

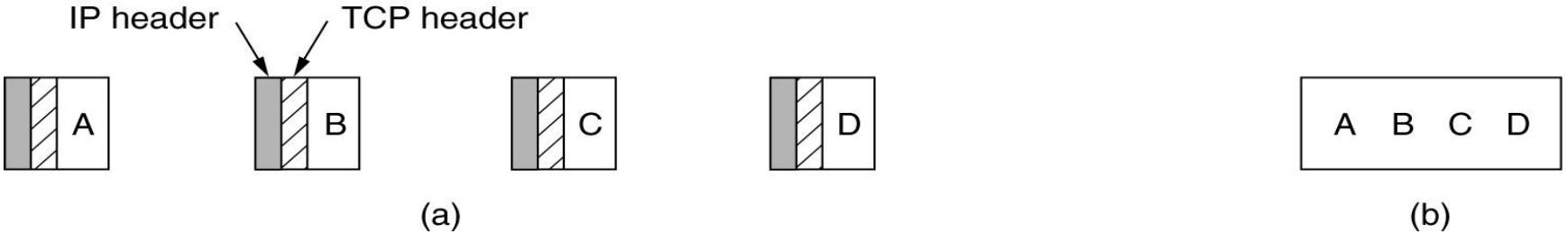
Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP service model

features of a TCP connection :



(a) sending process does four 512-byte writes to a TCP stream

(b) the data may be delivered to the receiving process as :

4 x 512-byte or 2 x 1024-byte or 1 x 2048-byte chunks

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP service model

when an application passes data to TCP, TCP may send immediately or buffer / aggregate it to collect a larger amount to send at once

applications use PUSH flag to inform TCP to send data immediately (not to delay sending), when necessary

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP service model

urgent data, say from an interactive user :

e.g. ctrl+c → application puts URGENT flag in the header → →

TCP stops accumulating data and transmits data meant for the connection immediately

at the destination, the receiving application is interrupted causing reading of urgent data

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP segment :

fixed 20-byte header (+ optional part)

followed by zero or more data bytes

TCP software :

decides on size of segment

can accumulate data from several writes to one segment

can split data from one write over many segments

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

two limits that restrict the size of data segment :

(i) each segment, including TCP header, must fit in the 65,515-byte IP payload

(ii) each network has a *M*aximum *T*ransfer *U*nit (*MTU*) - each segment should fit in the MTU ;

MTU is generally = Ethernet payload size = 1500 bytes

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Basic operation

basic protocol used by TCP entities is the sliding window protocol

each segment is assigned a 32-bit sequence number

when a sender transmits a segment, it also starts a timer

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

when a segment arrives at the destination, the receiving TCP entity sends back a segment with data (if any)(#) bearing an acknowledgement number equal to the next sequence number it expects to receive

if the sender's timer expires before the ack is received, the sender retransmits the segment

(#) segments without data can be acknowledgments or control messages

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Issues to be dealt with and solved in the face of network problems:

segments can arrive out of order

segments can be delayed in transit → triggering a retransmission due to time-out

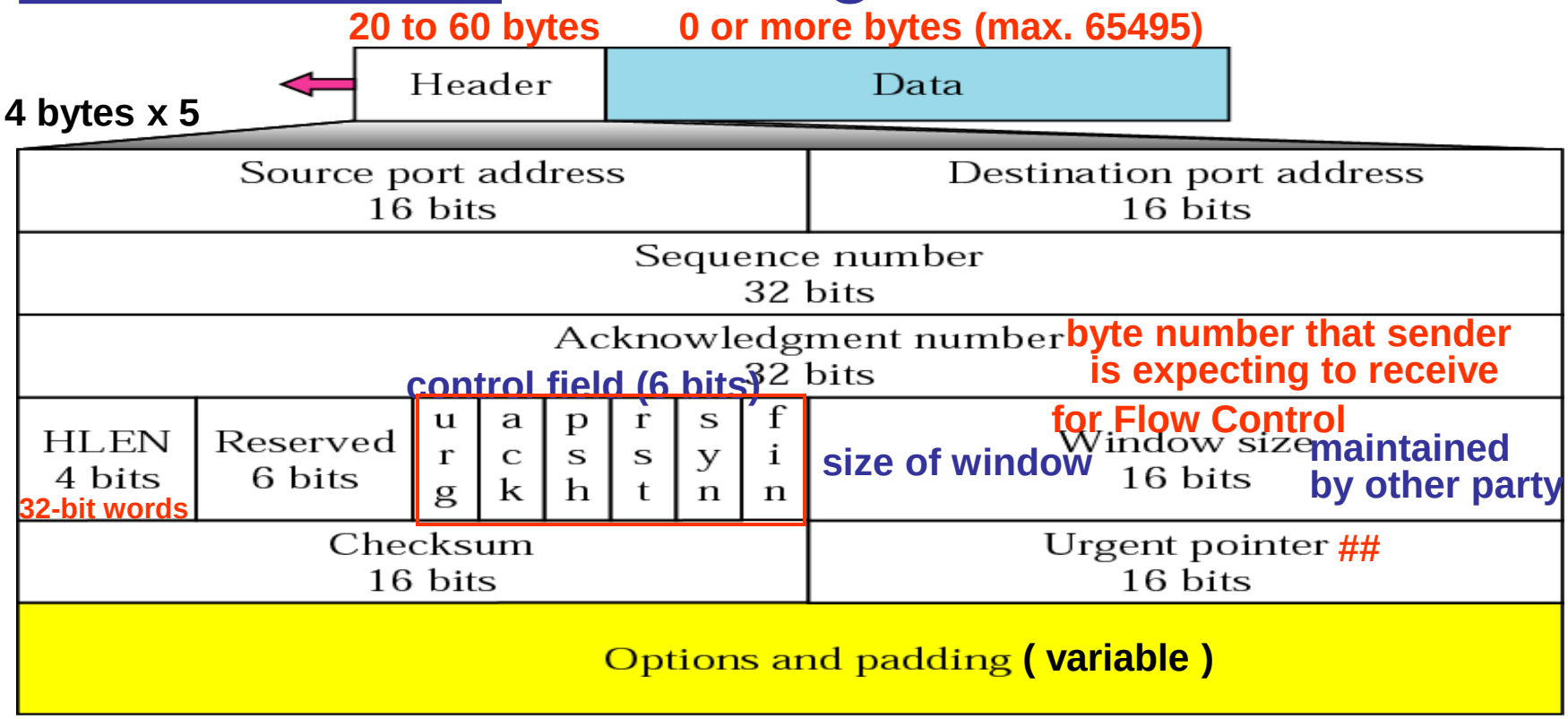
retransmissions may include different byte ranges than the original transmission - has to be kept track by TCP

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP Protocol : TCP segment format



byte offset from the current seq. no. at which urgent data is to be found

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP segment header

every segment begins with a fixed-format, 20-byte header

fixed header may be followed by header options

data, following the header, can vary from 0 to 65,495 bytes

segments without any data are valid and are commonly used for acknowledgement and control messages

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP segment header

how are connections identified ? :

by source and destination end points

what is an end point ? :

a port plus its host's IP(v4) address forms a 48-bit unique end point

16-bit *source port* and *destination port* fields in the segment header identify the local end points of a connection

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP segment header

every byte of data is numbered in a TCP stream

32-bit *sequence number* and *acknowledgement number* are used for the purpose

the acknowledgement number specifies the next byte expected

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

URG: Urgent pointer is valid	RST: Reset the connection
ACK: Acknowledgment is valid	SYN: Synchronize sequence numbers
PSH: Request for push	FIN: Terminate the connection

URG	ACK	PSH (#)	RST	SYN	FIN
		push the data	reset connection	sync seq. nos.	terminate connection

connection request → SYN = 1, ACK = 0

connection reply (accepted) → SYN = 1, ACK = 1

SYN and FIN segments have seq. nos. ; are processed in correct order

(#) signal to the receiver to deliver data immediately to application

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

URGENT flag in TCP header

in TCP, data is presented from the application program to TCP as a stream of bytes i.e. each byte has a position in the stream

sometimes, the application program may want to send *urgent* bytes

urgent data, say from an interactive user :

e.g. ctrl+c → application puts URGENT flag in the header → →

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

URGENT flag in TCP header

TCP stops accumulating data and transmits data meant for the connection immediately

at the destination, when the receiving TCP receives a segment with the URGENT bit set, it extracts urgent data (using the value of the urgent pointer) from the segment and delivers the data, out of order, to the receiving application program

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

PUSH flag in TCP header

an application program may want to send a keystroke to the receiver (without waiting for the window to be filled) and expect an immediate response

the sending side TCP sets the PSH bit to .. let the receiving side TCP know that the segment includes data that must be delivered to the receiving application program asap without waiting for more data to be received

Computer Networks

TCP Protocol : TCP segment format

window size field :

variable-sized sliding window is used to implement flow control in TCP

the *window size* fields informs the sender about how many bytes may be sent starting from the byte acknowledged

a value of 0 is valid

the receiver, when ready, can permit (sender to send) by transmitting a segment with an ack no. and a non-zero window size

Computer Networks

TCP Protocol : TCP segment format

checksum field : provided for extra reliability

it checksums the header, the data and the *pseudoheader* →

when performing checksum computation, the TCP checksum field is set to zero, ….

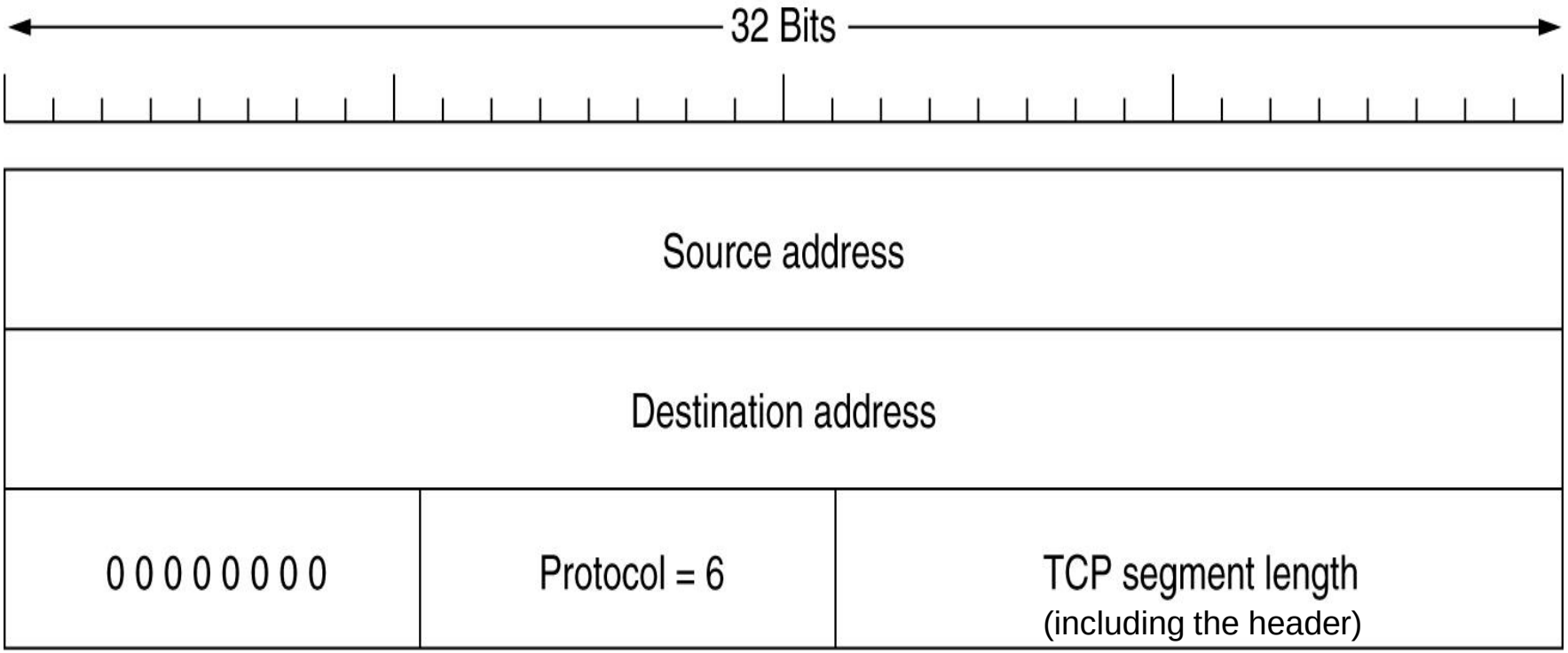
data field is padded out with one additional zero byte if its length is an odd number,….

16-bit checksum is calculated by adding the words in 1's complement and taking the complement of the sum

Computer Networks

TCP Protocol : TCP segment format

pseudoheader included in the TCP checksum



UDP (protocol field = 17) also uses the same pseudoheader for computing its checksum

Computer Networks

TCP Protocol : TCP segment format

pseudoheader included in the TCP checksum

pseudoheader contains :

(a) part of the header of the IP packet in which the user datagram is to be encapsulated

(b) eight 0s

(c) 8-bit protocol (6 for TCP)

(d) 16-bit TCP total length

Computer Networks

TCP Protocol : TCP segment format

TCP checksum : use of pseudoheader

used to ensure that the user datagram is not delivered to the wrong host in case IP header is corrupted

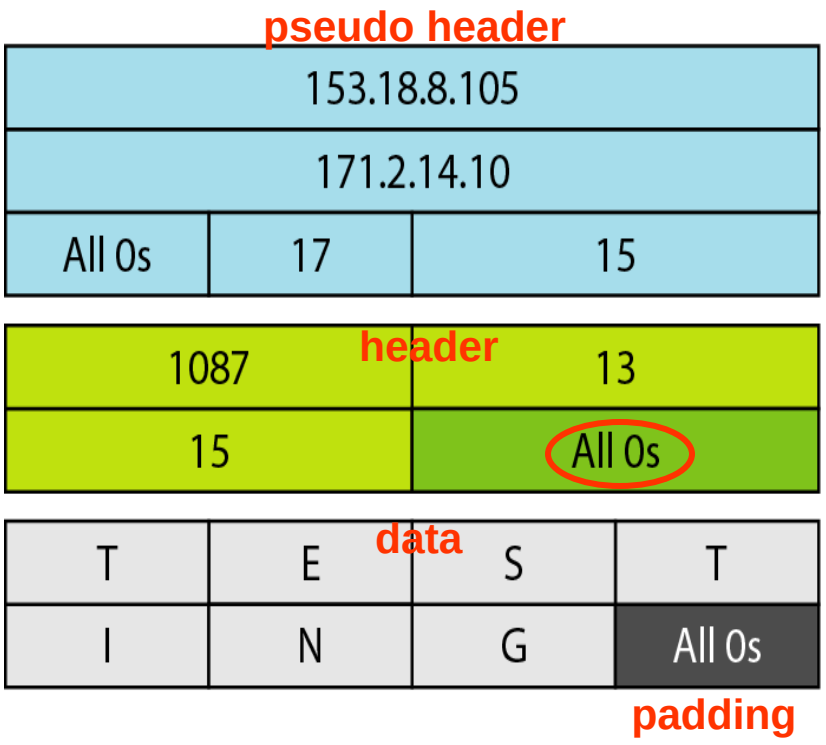
protocol field (value = 6) is added to ensure that the packet belongs to TCP

if the value changes during transmission, the checksum calculation at the receiver will detect it and the packet is dropped

ensures delivery to TCP / UDP protocol on the right host

Computer Networks

UDP Protocol : checksum computation example



10011001 00010010	→	153.18
00001000 01101001	→	8.105
10101011 00000010	→	171.2
00001110 00001010	→	14.10
00000000 00010001	→	0 and 17
00000000 00001111	→	15
00000100 00111111	→	1087
00000000 00001101	→	13
00000000 00001111	→	15
00000000 00000000	→	0 (checksum)
01010100 01000101	→	T and E
01010011 01010100	→	S and T
01001001 01001110	→	I and N
01000111 00000000	→	G and 0 (padding)
10010110 11101011	→	Sum
01101001 00010100	→	Checksum

Computer Networks

TCP Protocol : TCP segment format

Options field :

facilitates addition of extra facilities not covered by the basic header

(i) each host can specify maximum TCP payload it is willing to accept (default : 536 bytes)

(ii) RFC 1106 - use of selective repeat instead of go-back-n and NAKs to allow receiver to ask for specific segments

Computer Networks

TCP Protocol : TCP segment format : Example

The following is a partial dump of a TCP header in *hexadecimal* format :

05320017 00000001 00000000 500207FF 00000000

(i) What is the source port number?

(ii) What is the application being used?

(iii) What is the sequence number?

(iv) What is the ack number?

(v) What is the length of the header?

(vi) What is the type of segment?

(vii) What is the window size in bytes?

Computer Networks

TCP Protocol : TCP segment format : Example

(i) The source port number is 0x0532 (1330 in decimal).

(ii) The destination port number is 0x0017 (23 in decimal) → TELNET

(iii) The sequence number is 0x00000001 (1 in decimal).

(iv) The acknowledgment number is 0x00000000 (0 in decimal).

Computer Networks

TCP Protocol : TCP segment format : Example

(v) The header length is 0x5 (5 in decimal). There are 5 × 4 or 20 bytes of header.

(vi) The control field is 0x002. This indicates a SYN segment used for connection establishment.

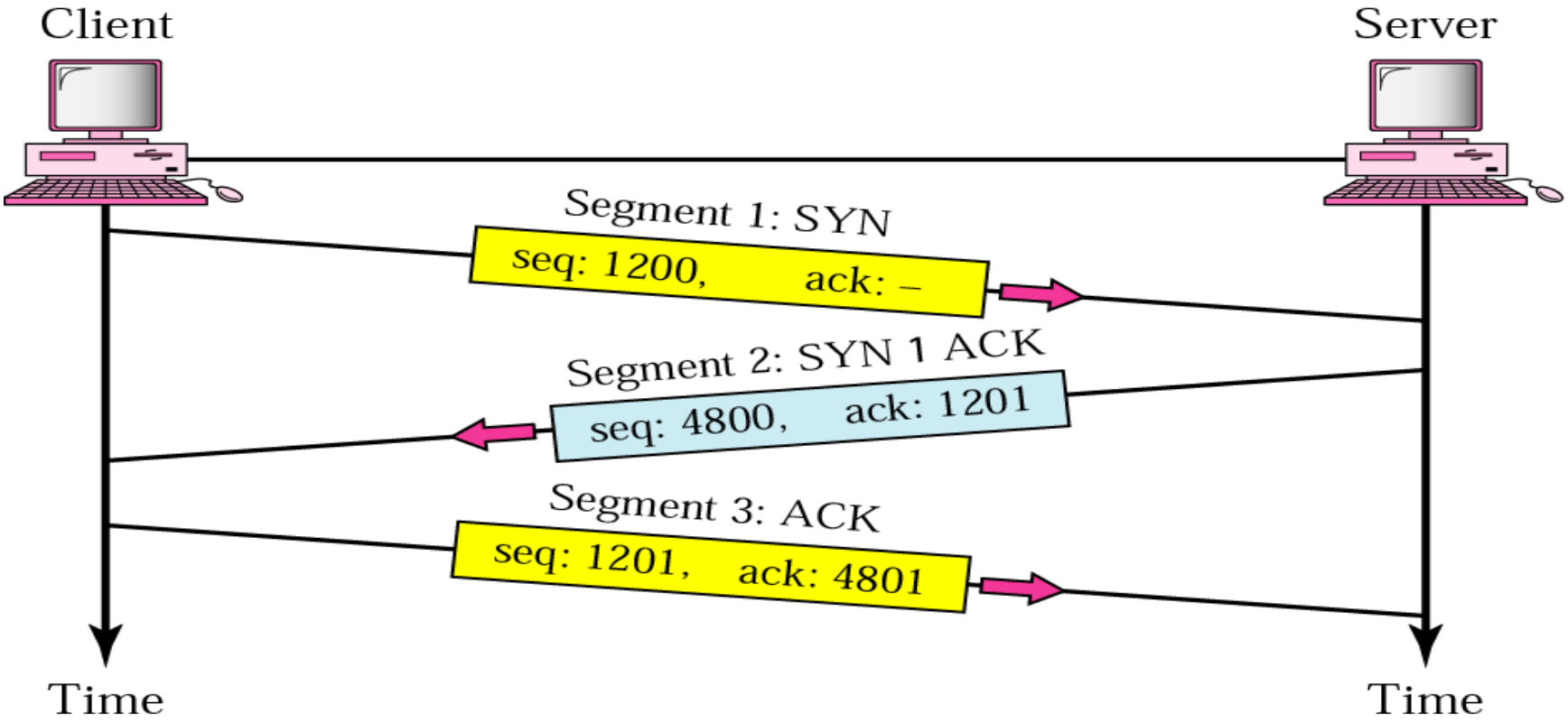
(vii) The window size field is 0x07FF (2047 in decimal).

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP connection establishment three-way handshake

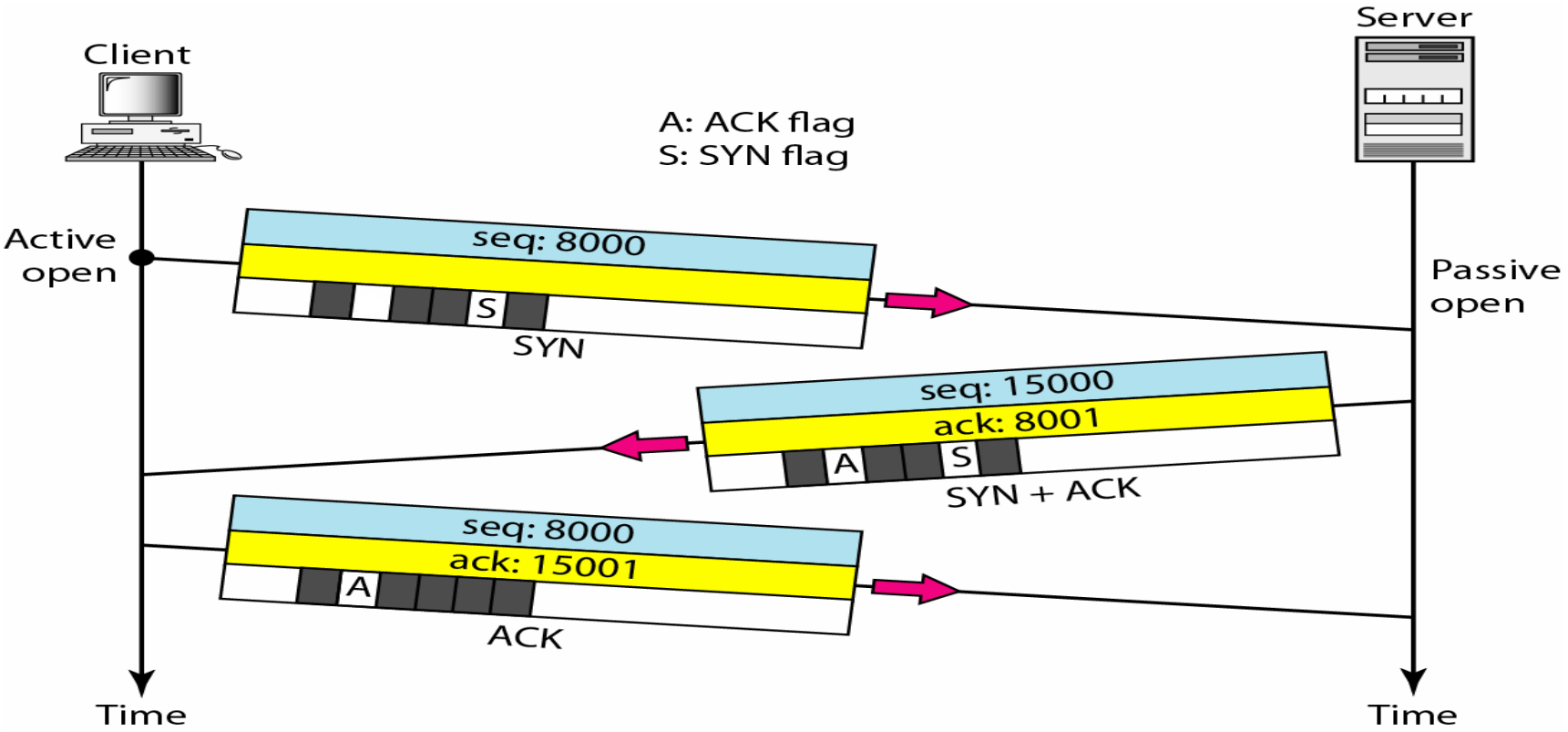


Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP connection establishment three-way handshake



Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP connection establishment

1. Client sends a SYN segment including :

(a) source and destination port numbers

(b) initialization sequence number (ISN) used for numbering the bytes of data sent

2. Server sends SYN and ACK

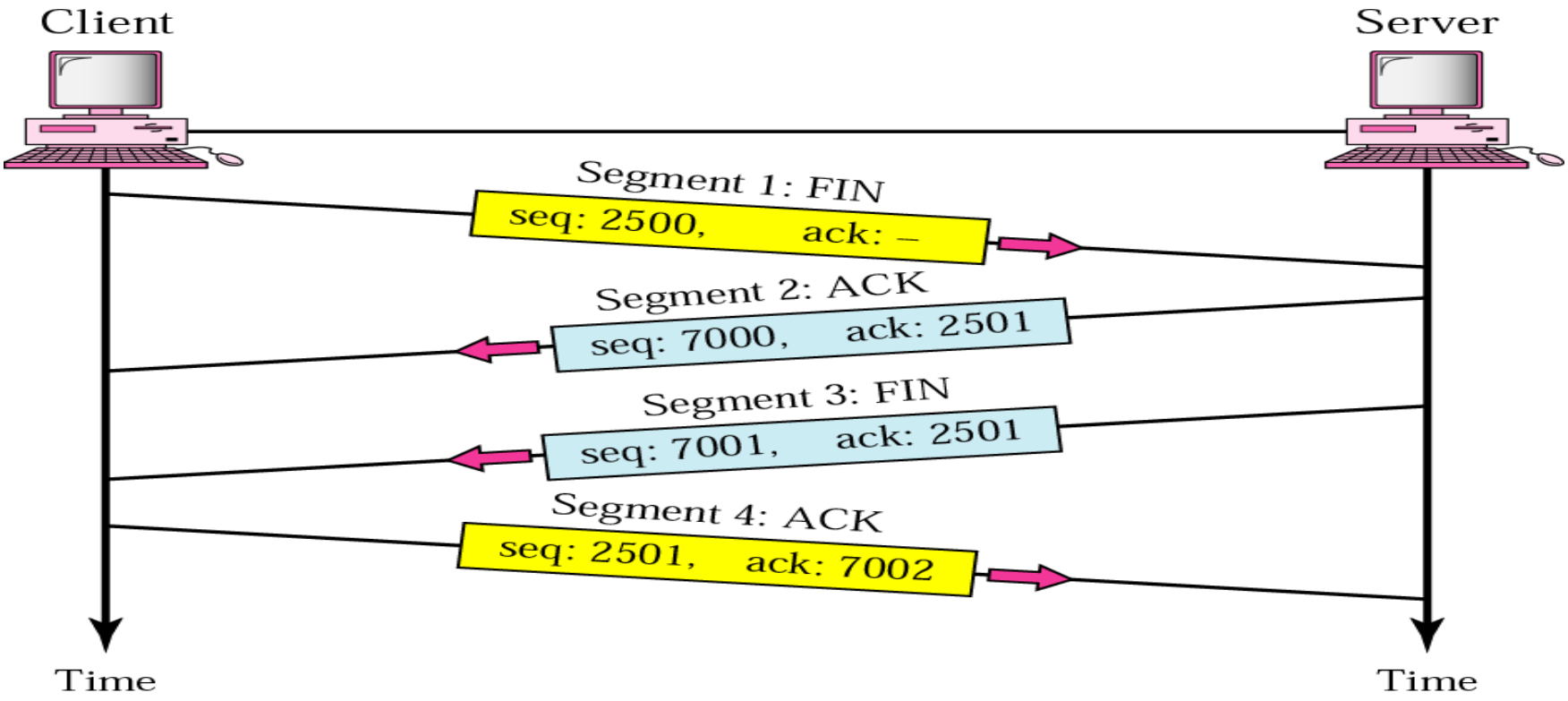
3. Client sends the 3rd segment - basically an acknowledgement of 2nd segment

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP connection release (four-step process)



Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP connection release (four-step process)

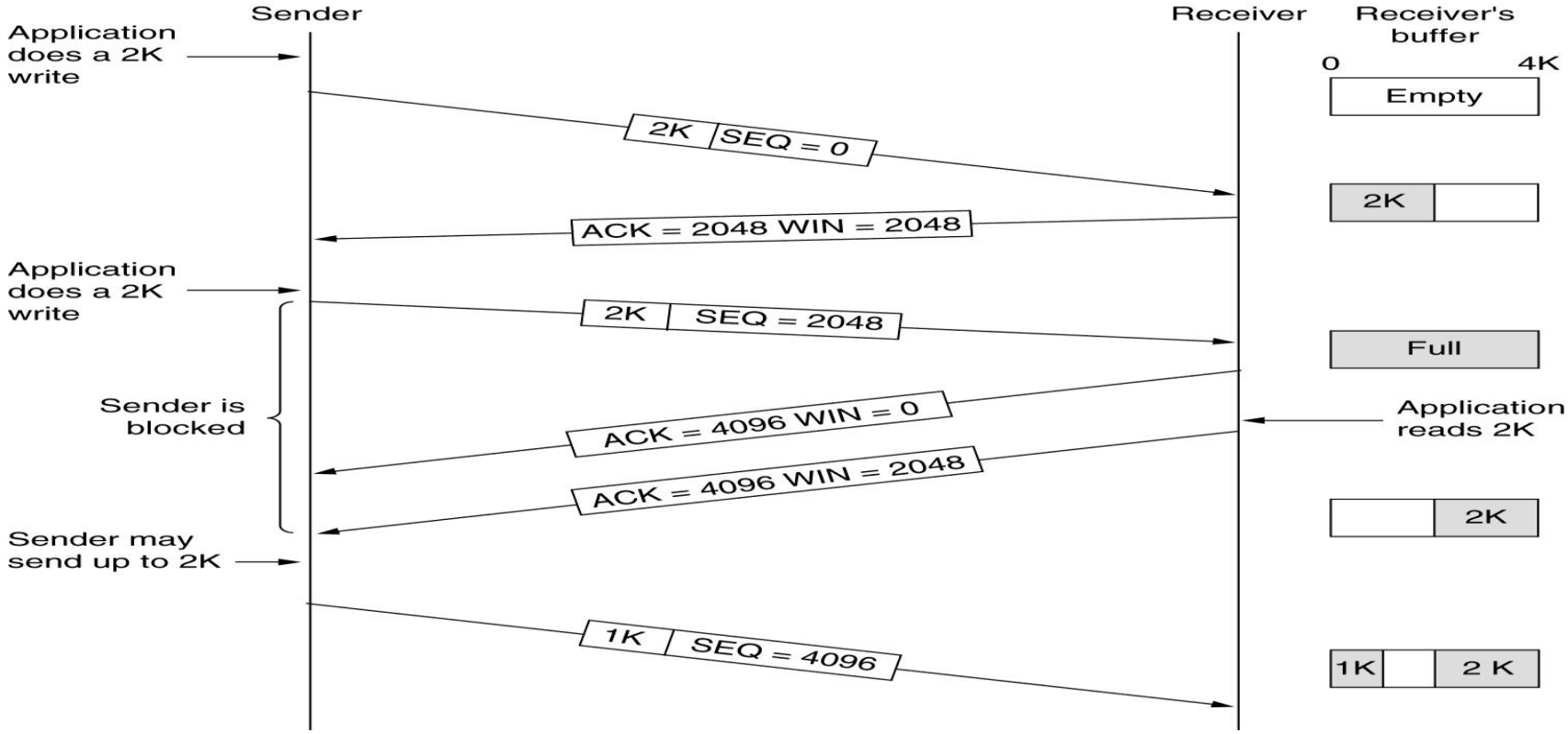
1. Client TCP sends FIN segment
2. Server TCP sends an ACK segment
3. Server TCP continues to send data ; when no more data is to be sent, sends a FIN segment
4. Client TCP sends ACK segment for receipt of FIN segment from Server

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy



Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

when the window is 0, sender cannot transmit, except as below :

- **urgent data may be sent to allow the user to kill the process running on the remote machine**
- **sender may send a 1-byte segment to trigger the receiver to re-announce the next byte expected and the window size**

to prevent deadlock if a window announcement gets lost

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy : telnet example

a telnet connection to an interactive editor that reacts on every key stroke

1. a character arrives at sending TCP entity

2. TCP creates a 21-byte TCP segment, gives it to IP to send a 41-byte IP datagram

3. at the receiving side, TCP sends a 40-byte ack (20-byte TCP header + 20-byte IP header)

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy : telnet example

4. later, when the editor has read the byte, TCP sends a window update, moving the window 1 byte to the right ; this is also a 40-byte packet
5. finally, when the editor has processed the character, it echoes the character as a 41-byte packet

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy : telnet example

total bandwidth used = 162 bytes

no. of segments sent for each character typed = 4

inefficient use of bandwidth → →

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy : telnet example

→ → inefficient use of bandwidth

one approach to optimize :

delay acks and window updates (steps 3 & 4) for 500 msec, expect some data to build up

assuming that editor echoes within, say, 500 msec, only one 41-byte packet needs to be sent back to the remote user

this method reduces the loading of the network by the receiver

but, sender still sends 1+40 bytes → →

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Nagle's algorithm

is a means of improving the efficiency of TCP/IP networks by …..

reducing the number of packets that need to be sent over the network

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Nagle's algorithm

specifies a method of dealing with what is called *the small packet problem*, created when an application generates data one byte at a time, causing the network to be overloaded with packets (a situation often referred to as *send-side silly window syndrome*)

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Nagle's algorithm

when data is to be sent 1 byte at a time -

(a) just send the 1st byte

(b) buffer all the rest till the outstanding byte is acknowledged

(c) send all the buffered characters in one TCP segment

(d) start buffering again till next ack

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Nagle's algorithm

(e) send an additional packet if enough data is available to fill-in a specified segment size

cannot be used in some applications like X-Windows involving mouse movements

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Nagle's algorithm

works by automatically concatenating a number of small buffered outgoing messages, and sending them all at once

as long as there is a sent packet for which the sender has received no ack, …….

the sender should keep buffering its output until it has a full packet's worth of output, …..

so that output can be sent all at once

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

another problem that can degrade TCP performance is :

Silly-window syndrome (receiving side)

data is passed to the sending TCP entity in large blocks ;

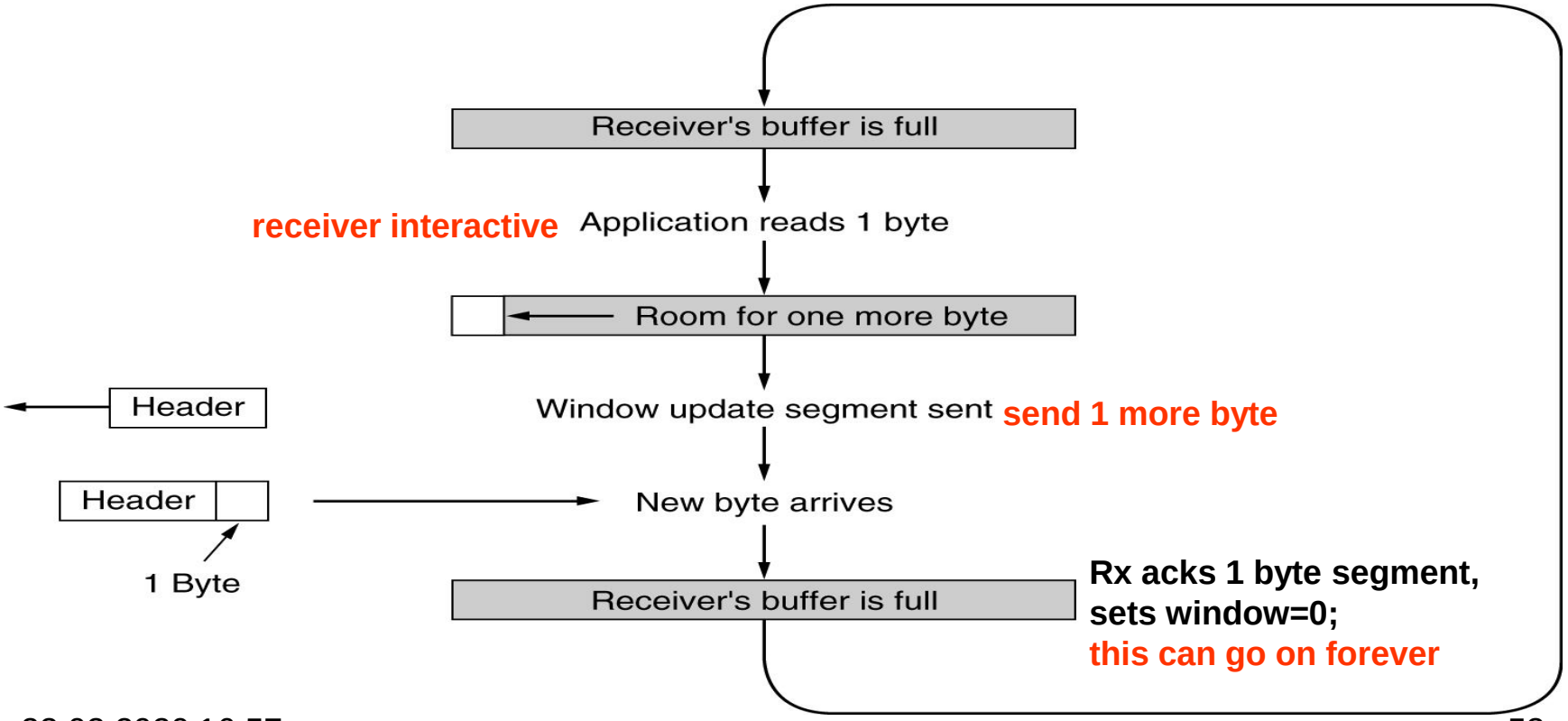
but an interactive application on the receiving side reads one byte at a time

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy : Silly-window syndrome



Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Silly-window syndrome (receiving side)

if a receiver is unable to process all incoming data, it requests that the sender *reduces* the amount of data it send at a time (the "window" setting on a TCP packet)

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Silly-window syndrome (receiving side)

if the receiver continues to be unable to process all incoming data, the window becomes smaller and smaller, sometimes to the point that the data transmitted is smaller than the packet header, making data transmission extremely inefficient

the name of this problem is due to the window size shrinking to a "silly" value

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Silly-window syndrome : Clark's solution

to prevent receiver sending a window update for 1 byte

1. wait till receiver has *decent amount* of space available and advertise that
2. send a window update only when advertised segment size is filled

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Nagle's algorithm and Clark's solution to the silly window syndrome are complementary

Nagle's algorithm :

solves the problem caused by the sending application delivering data to TCP one byte at a time

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Clark's solution :

solves the problem of the receiving application reading the data from TCP one byte at a time

both solutions are valid and can work together

the ultimate goal is :

sender not to send small segments; receiver not to ask for them

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP transmission policy

Another method for improvement of performance :

receiving TCP entity can also buffer data and not allow the application to read until it has a large chunk of data to provide

this results in reduction in the number of calls to TCP and hence, the overhead

non-interactive applications (efficiency more important than response) can tolerate this

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP Congestion Control

load offered to the network is more than it can handle → congestion builds up

use of "law of conservation of packets" : i.e. refrain from pushing a new packet into the network until the old one is delivered

TCP handles this by dynamically adjusting the window size

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP Congestion Control

first step in managing congestion is detecting it, generally based on time-outs

time-outs can be due to link problems or packet drops at congested router

with improvement in technology, links are becoming more and more reliable, so ….

most time-outs are due to congestion

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP Congestion Control

preventing congestion :

choose a suitable window size when connection is established

receiver can suggest window size based on its buffer size

if sender conforms to this window, no overflow at the receiver end

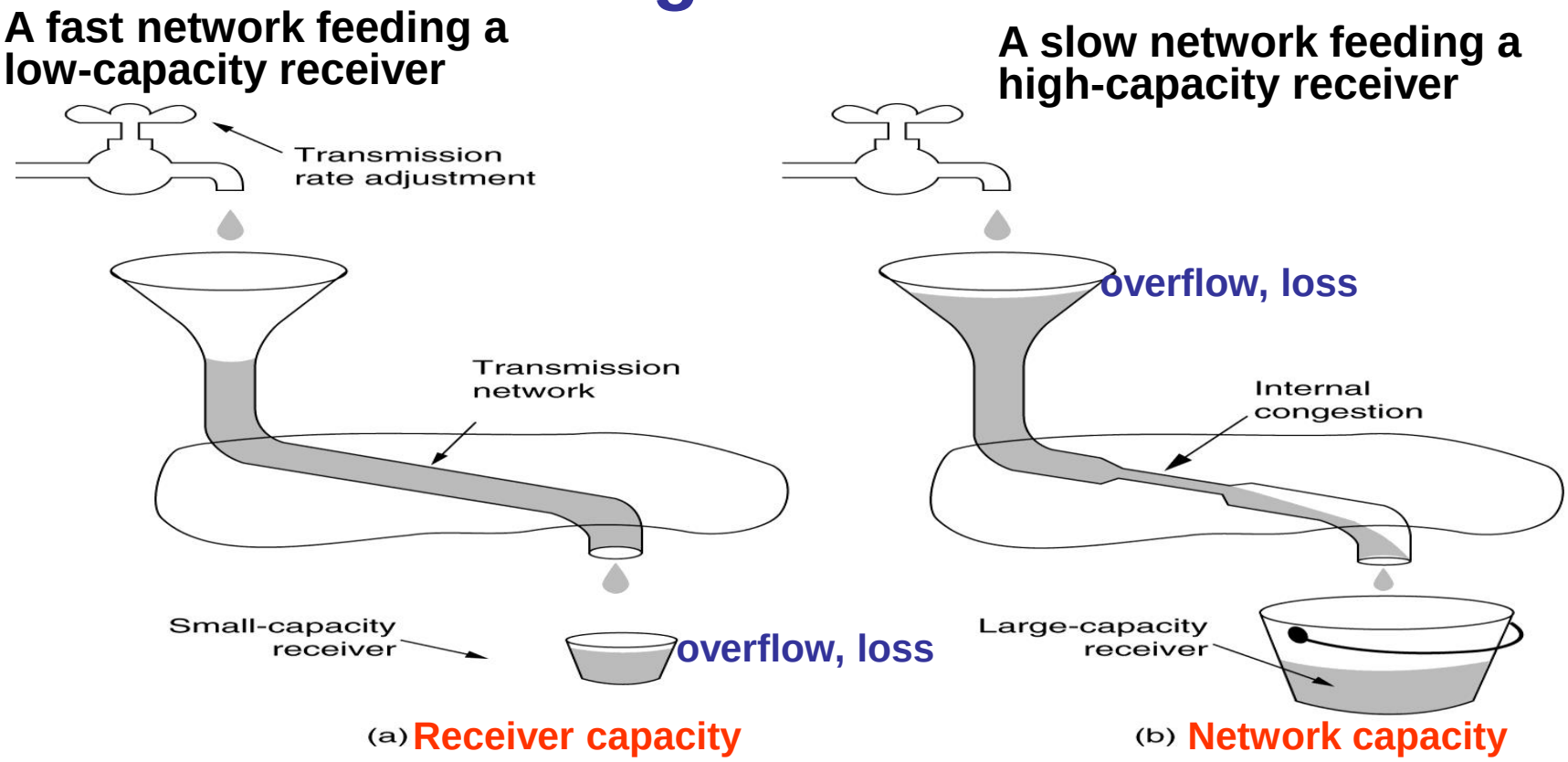
but problems may occur due to internal congestion

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP Congestion Control



Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP Congestion Control

1. each sender maintains two windows
   (a) window the receiver has granted
   (b) congestion window

2. no. of bytes sent is lesser of the two

Computer Networks

TCP Congestion Control

TCP maintains a new state variable for each connection, *CongestionWindow* used by the source to limit how much of data is allowed to be in transit at a given time

TCP's effective window is revised as

***MaxWindow* = MIN(*CongestionWindow*, *AdvertisedWindow*)**

source is allowed to send no faster than the slowest component - the network or the receiving host - can accommodate

Computer Networks
TCP Congestion Control

How is *CongestionWindow* set ?

source sets it based on its perceived level of congestion in the network

congestion window size is increased when level of congestion goes down and is reduced when level of congestion goes up

Computer Networks

TCP Congestion Control

Slow start (RFC 2001)

earlier design of TCP : source used to send as many packets, in a bursty manner, as the advertised window allowed

routers were not able to handle these bursts

Slow start was designed to handle this situation

the sender starts by transmitting one segment and waits for its ACK

Computer Networks

TCP Congestion Control

Slow start (RFC 2001)

when that ACK is received, the congestion window is incremented from one to two, and two segments can be sent

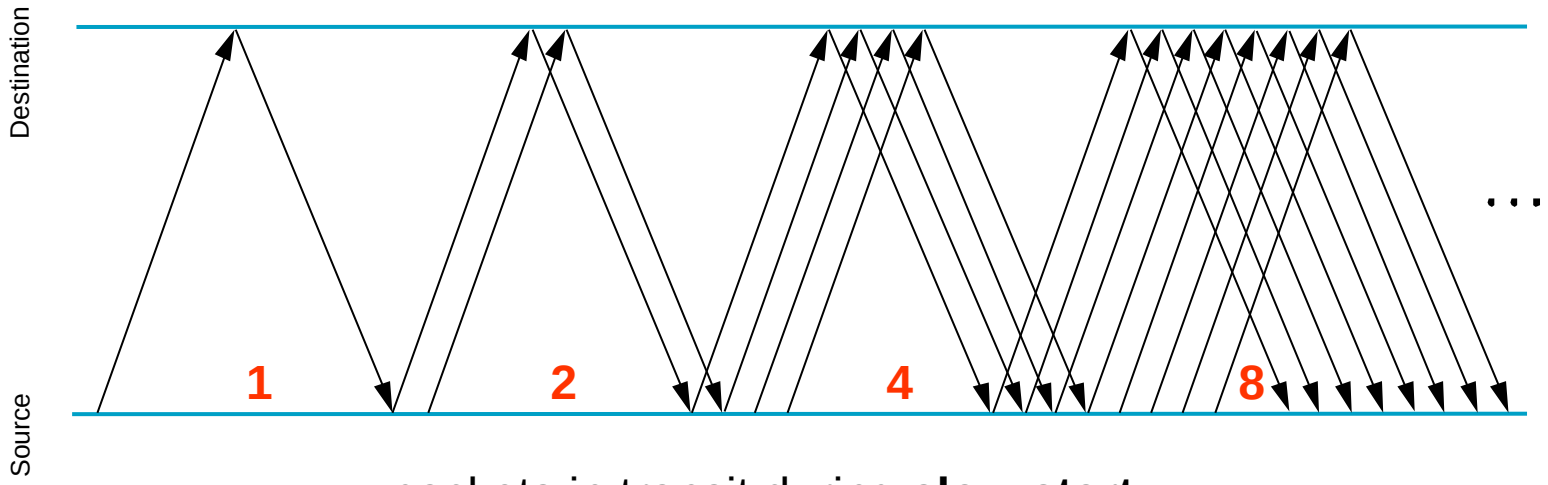
when each of those two segments is acknowledged, the congestion window is increased to four ……..

Computer Networks

TCP Congestion Control

Slow start

effectively increases the congestion window exponentially, rather than linearly



packets in transit during **slow start**

Computer Networks
TCP Congestion Control

Slow start

the number of packets in transit are effectively doubled every RTT

when there is packet loss, a timeout causes multiplicative decrease i.e. congestion window size is divided by 2

Packet-pair technique

send a pair of packets (probe) with no space between them
check the spacing of ACKs received for these to get an idea of the congestion

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP congestion control :

Internet congestion algorithm

uses a third parameter, *threshold*(#)

threshold is initially set to 64 KB

when time-out occurs, threshold is set to 32 KB and congestion window reset to 1 KB

congestion window then grows exponentially from 1 KB to 32 KB and linearly thereafter

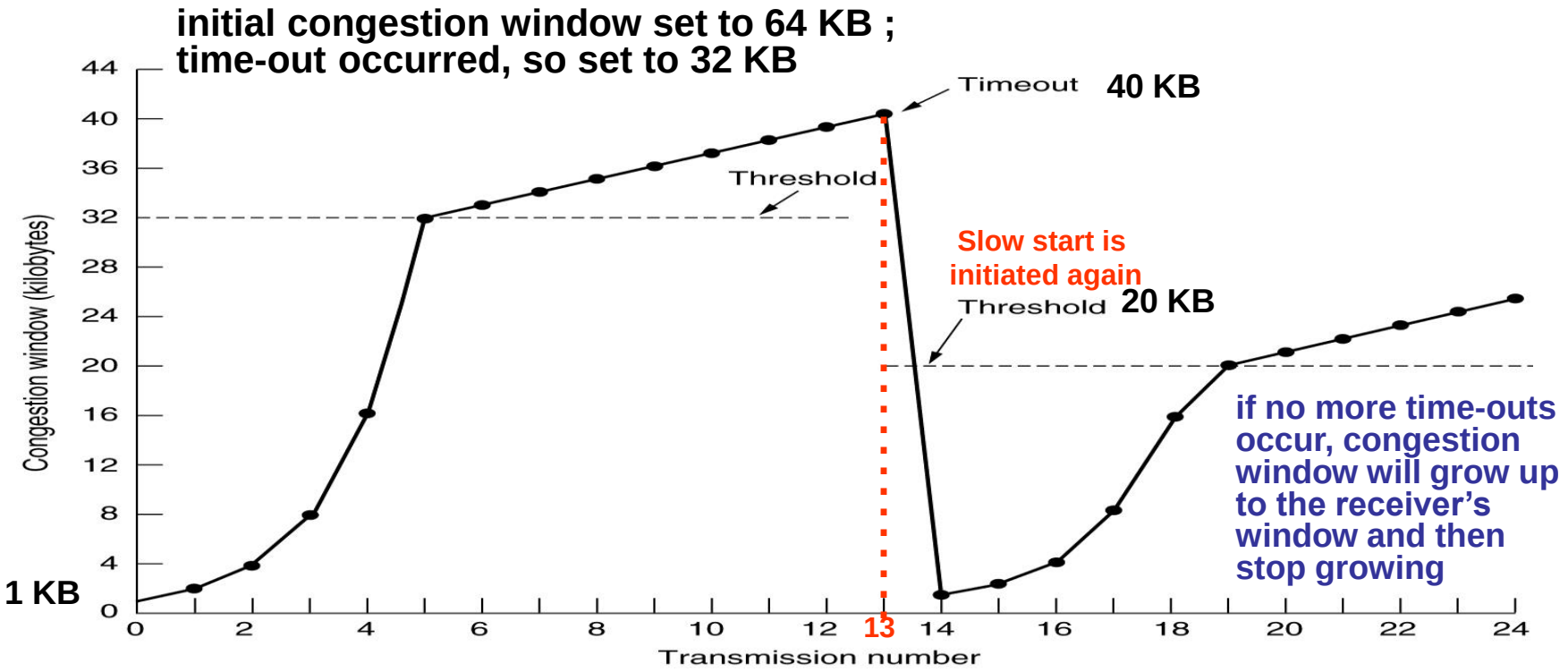
(#) : in addition to the receiver and congestion windows

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP congestion control : Internet congestion algorithm



Computer Networks
TCP Congestion Control

AIMD (Additive Increase Multiplicative Decrease) represents a *linear growth* of the congestion window, combined with an *exponential reduction* when a congestion takes place

the approach taken is to increase the transmission rate (window size), probing for usable bandwidth, until loss occurs

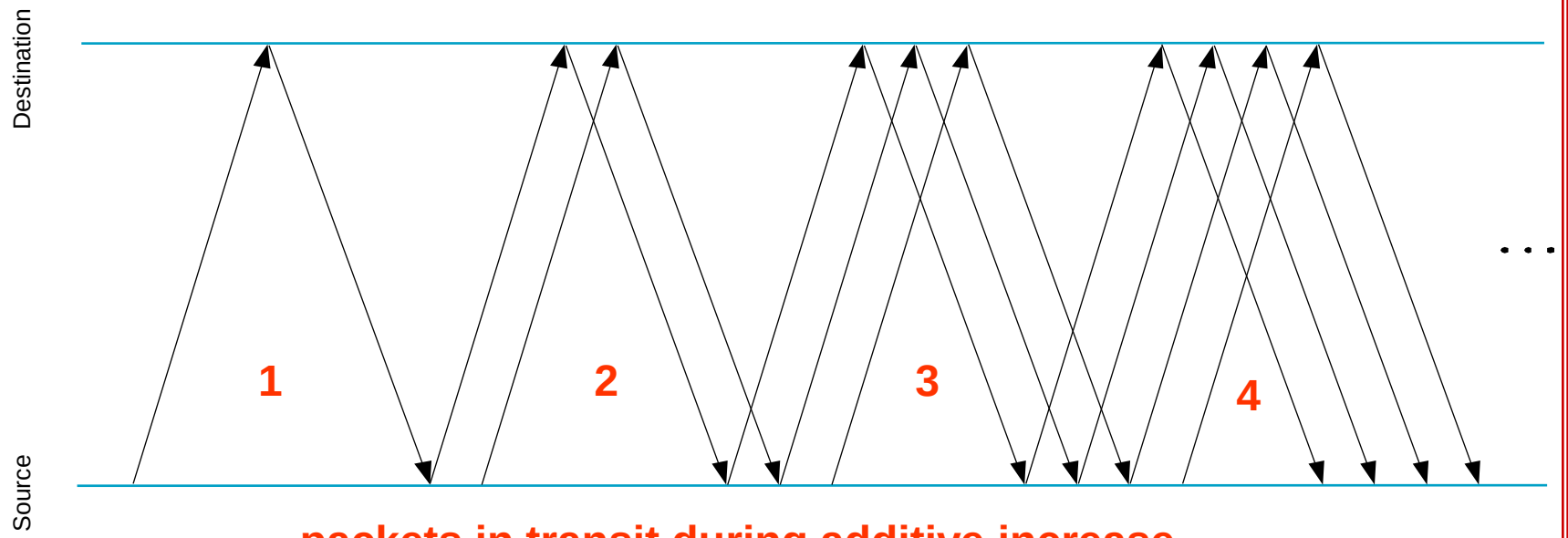
the policy of additive increase basically says : increase the congestion window by 1 MTU every RTT until a loss is detected

Computer Networks

TCP Congestion Control

AIMD

when loss is detected, the policy is changed to be one of multiplicative decrease which is to cut the congestion window in half after loss

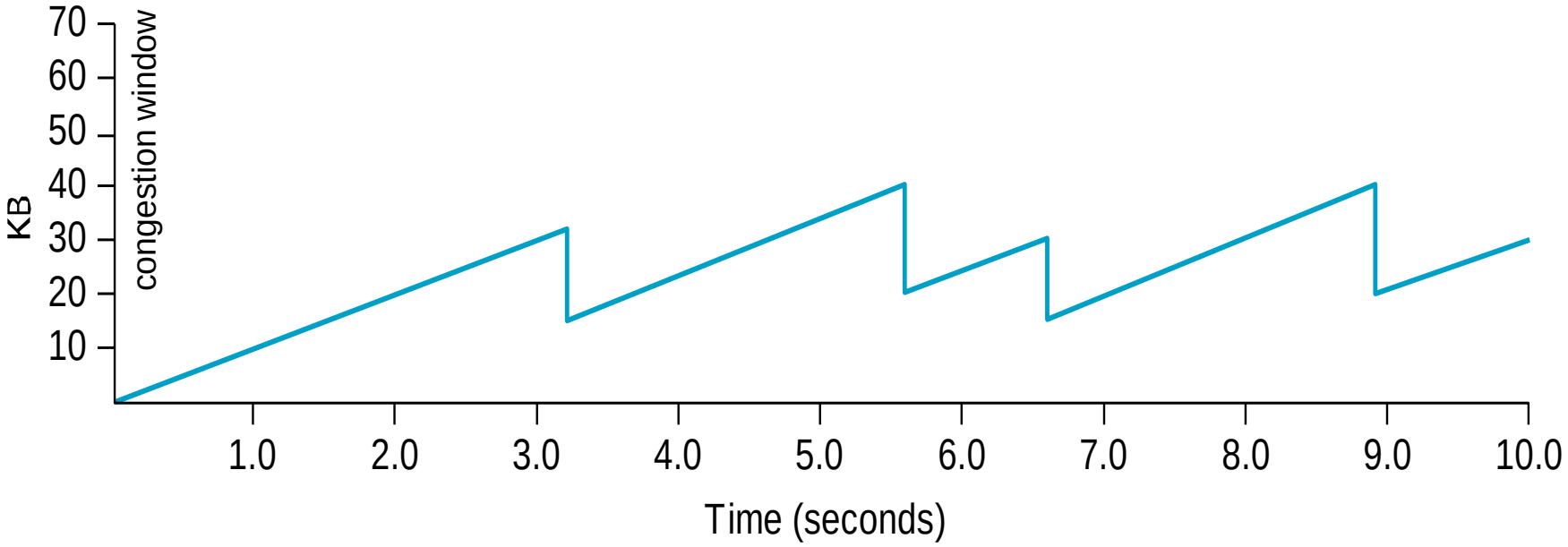


packets in transit during additive increase - one packet is added each RTT

Computer Networks

TCP Congestion Control

the pattern of continually increasing (conservatively) and decreasing the congestion window (aggressively) continues throughout the lifetime of the connection



Computer Networks

TCP Congestion Control

Slow start and AIMD

slow start is generally used at the beginning of a connection

or when a connection goes dead while waiting for timeout to occur and flow of data needs to be restarted

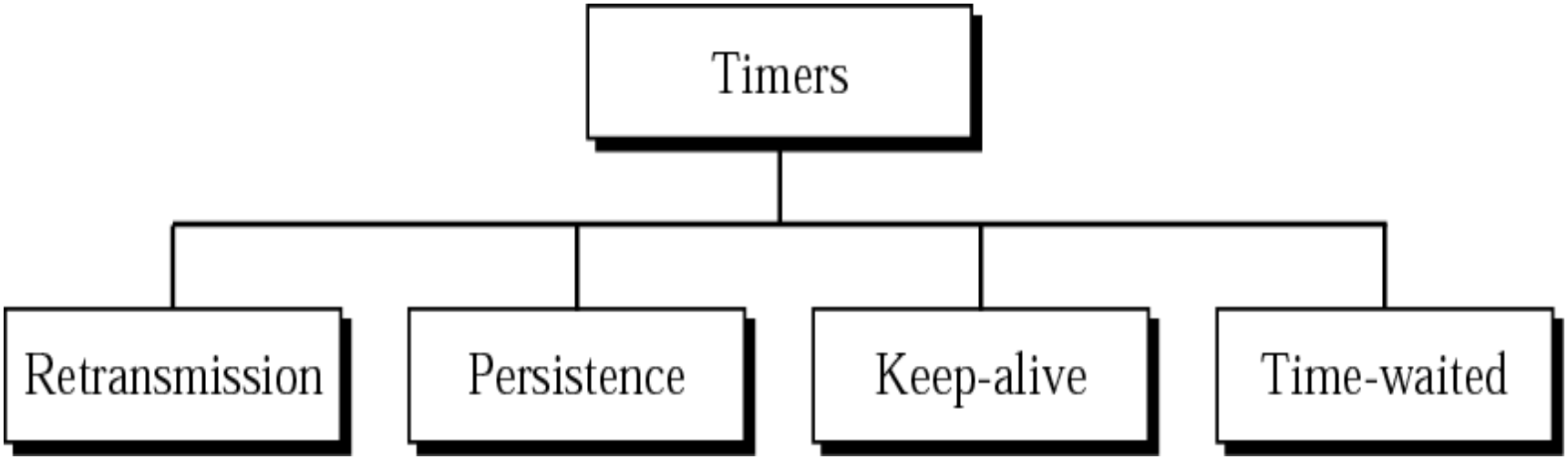
AIMD is the right approach to use when a connection is stable and the source is operating close to the available capacity of the network

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP Timer Management



Computer Networks

Transport Layer

Internet Transport Protocols : TCP

TCP Timer Management

Retransmission timer **:**

timer is started when a segment is sent

timer is stopped if acknowledgment for segment is received in time

if timer expires before receipt of ack → segment is re-transmitted and timer started again

Issue: How long should the timeout interval be?

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

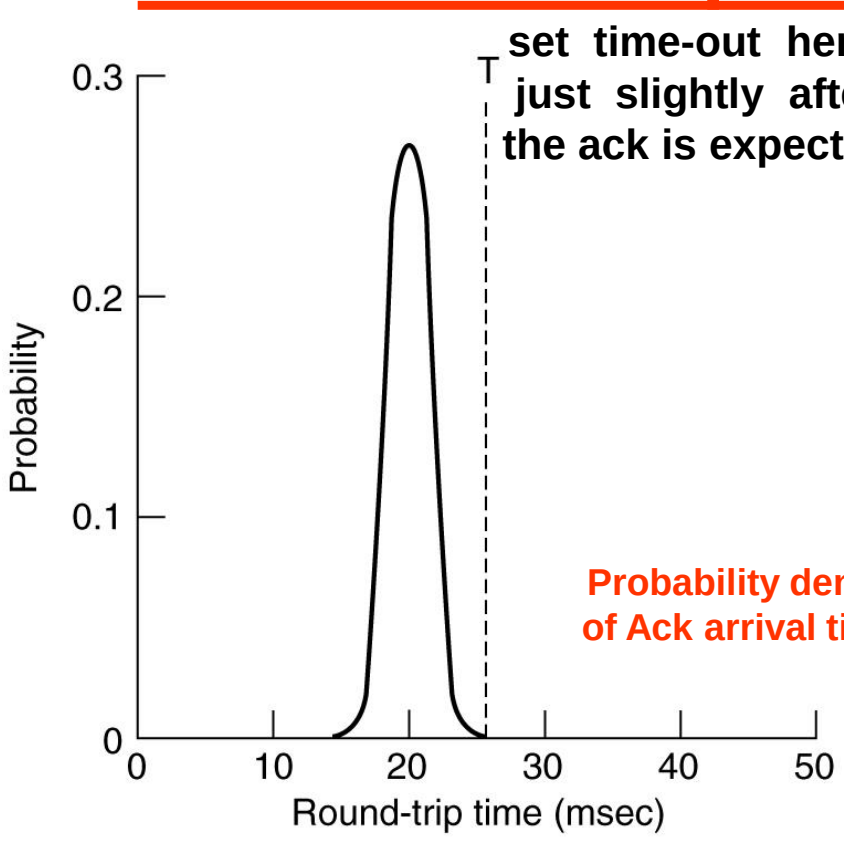
in data link layer, acks are rarely delayed since congestion does not play a part
absence of ack at the expected time means that either the frame or ack is lost

in TCP, the situation is radically different
probability densities in these layers →

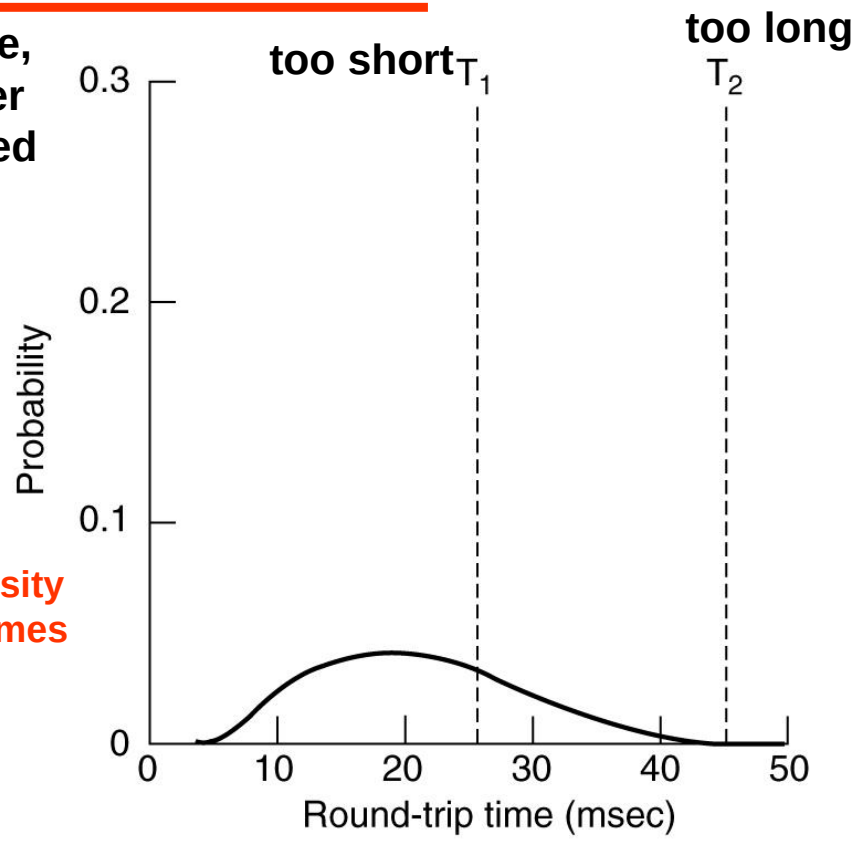
Computer Networks

Transport Layer

Internet Transport Protocols : TCP



Probability density of Ack arrival times



In data link layer

In transport layer

delay highly predictable : has low variance

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

in TCP, determining RTT to the destination and deciding on the time-out interval is tricky

time-out too short T_1 → unnecessary re-transmissions will occur → clogging

time-out too long T_2 → long retransmission delays when packet is lost → performance degradation

solution →

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Jacobson's algorithm

a ***dynamic algorithm*** to ***constantly adjust timeout interval based on continuous measurements*** of network performance

for each connection, TCP maintains a variable, ***RTT***, that is the ***best current estimate*** of the round-trip time to the destination connected

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

when a segment is sent, a timer is started and the time taken for receipt of acknowledgement is *measured* as *M*

(if time out period expires, retransmission is triggered)

***RTT* is updated according to a formula →**

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Jacobson's algorithm

(a) $RTT = \alpha RTT + (1 - \alpha)M$

α = smoothing factor, typically = (7 ÷ 8) or 90% ; determines how much weight is given to old value

M = measured value of time taken

(b) Deviation $D = \alpha D + (1 - \alpha) \left| RTT - M \right|$

(expected - observed)

(c) $Timeout = RTT + 4D$;

most TCP implementations use this time out value

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Example : AT 6.30

If the TCP RTT is currently 30 msec and

the following acknowledgements come in after 26, 32, 24 msecs respectively,

estimate the new RTT using Jacobson's algorithm.

State assumption made, if any.

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Example : AT 6.30

$$RTT = \alpha RTT + (1 - \alpha)M$$

(a) RTT = (0.9 x 30) + (0.1 x 26) = 29.600 msec

(b) RTT = (0.9 x 29.6) + (0.1 x 32) = 29.840 msec

(c) RTT = (0.9 x 29.84) + (0.1 x 24) = 29.256 msec

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Other Timers

Persistence timer **: for prevention of deadlock:**

(a) receiver sends ack with window size = 0, (to the transmitter), meaning, wait

(b) later receiver updates the window seeking transmission

(c) packet with (b) is lost

(d) both sender and receiver endlessly wait

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Other Timers

***Persistence timer* :**

(e) when the persistent timer expires, sender transmits probe to the receiver → receiver responds with window size

(f) if window size is still = 0, set persistent timer again and repeat cycle

(g) if window ≠ 0, send data

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Other Timers

Keepalive timer **: to terminate idle connections, one side checks whether the other side is still there;**
if no response, terminates connection

Timed Wait **:**
runs for twice the max packet life time to ensure that, when a connection is closed, all packets created by it have died off

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Wireless TCP and UDP

Wired networks assume :

(a) packets are (almost) never lost

(b) timeouts are (almost) solely due to congestion

so, TCP slows down and sends less vigorously (Jacobson's slow start algorithm) to reduce network load / congestion

not true in case of wireless :

wireless transmissions are highly unreliable; packets *are* lost

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Wireless TCP

wireless networks : sender should try harder and retransmit lost packets faster (recall : in wired networks sender should slow down)

in heterogeneous networks (say, 1000 Kms wired + 1 Km wireless last mile),…. correct decision making is difficult

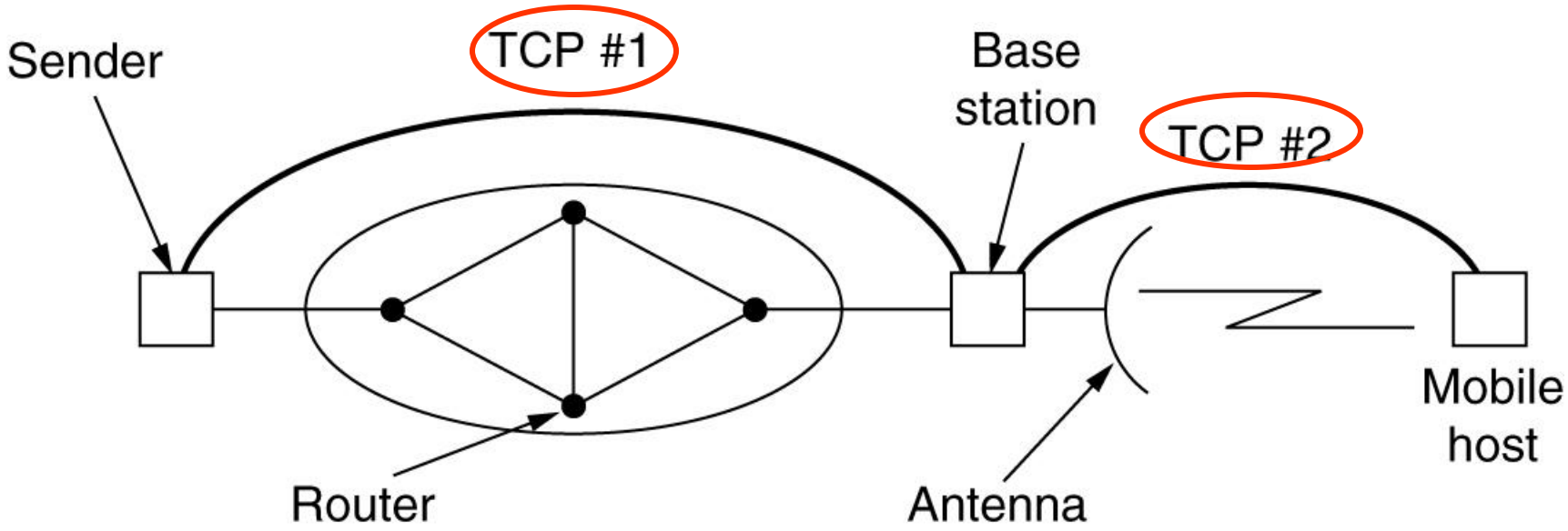
Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Wireless TCP : Indirect TCP

(Bakne & Badrinath)



splitting of connection into two **homogeneous connections**

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Wireless TCP : Indirect TCP

base station copies packets between the connections in both the directions

timeouts on the first connection can slow the sender down

timeouts on the second connection can speed the sender up

other parameters can also be tuned separately for each connection

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Wireless TCP : Indirect TCP

BS copies packets in both directions

Issue : receipt of acknowledgement at the sender means only BS has received, not really the receiver

Solution : Balakrishnan and others

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Wireless TCP : Indirect TCP

make small modifications to the network layer code in the base station

addition of a snooping agent at BS :

the agent observes and caches the TCP segments going out to and acknowledgements coming back from the mobile host

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Wireless TCP : Indirect TCP

if acknowledgement for a segment sent to the mobile host is not received before (relatively shorter) timer expires, BS just retransmits that segment

BS handles retransmissions without the knowledge of sender

Computer Networks

Transport Layer

Internet Transport Protocols : TCP

Wireless TCP : Indirect TCP

Disadvantage : sender may time out waiting for ack when BS-MH link is lossy and may invoke congestion control algorithm

so, in indirect TCP, congestion control algorithm is never initiated unless there is really a congestion in the wired part of the network

Computer Networks

Application Layer

layers below the application layer provide reliable transport

real working support for the users is provided by the Application layer enabling the user (human or software) to access the network

to allow the applications to function, there is a need for support protocols in the application layer also

Computer Networks

Application Layer

Applications in the application layer of the Internet model :

- **Electronic Mail (SMTP, …, …)**
- **File Transfer (FTP)**
- **Web Services (HTTP)**

Computer Networks

Electronic Mail

based on RFC 822

well known advantages (??) over *snail mail* (paper letters)

has its own (??!!) conventions, sloppily-written, full of jargons / smileys / emoticons

book giving over 650 smileys : Sanderson and Dougherty, 1993

Computer Networks

Electronic Mail

early days of the Internet :

- **e-mail messages were short**
- **consisted of text only**
- **used for quick exchange of memos**

current scenario :

- **messages include text, audio and video**
- **one message to be sent to multiple recipients**

Computer Networks

End of Class - 25
20.03.2015

Computer Networks
Electronic Mail

Architecture of e-mail : UA, MTA and MAA

- **user agent (UA) :**
  allows reading and sending of e-mail

  local program providing command-based, menu-based or GUI for interacting with the e-mail system

  expects addresses of the form :
  user@dns-address

Computer Networks
Electronic Mail

Architecture : UA, MTA and MAA

- **message transfer agent (MTA) :**

  moves messages from source to destination

  typically a daemon (background process)

Computer Networks
Electronic Mail

Architecture : UA, MTA and MAA

- **message access agent (MAA) :**

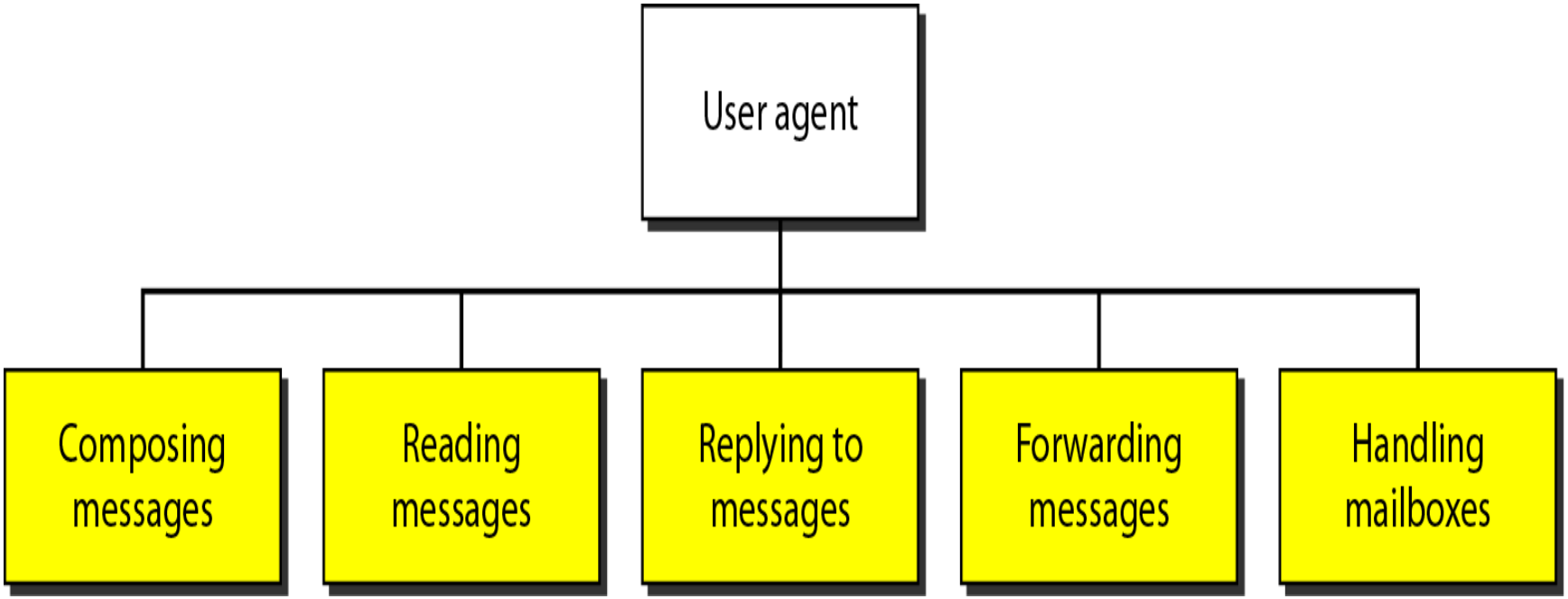
  allows UAs (on client PC) to contact the MTA on some server and …..

  copy the mail from the server to the user's system

Computer Networks

Electronic Mail

Architecture : user agent



user agent types : command - driven

GUI - based

Computer Networks

Electronic Mail

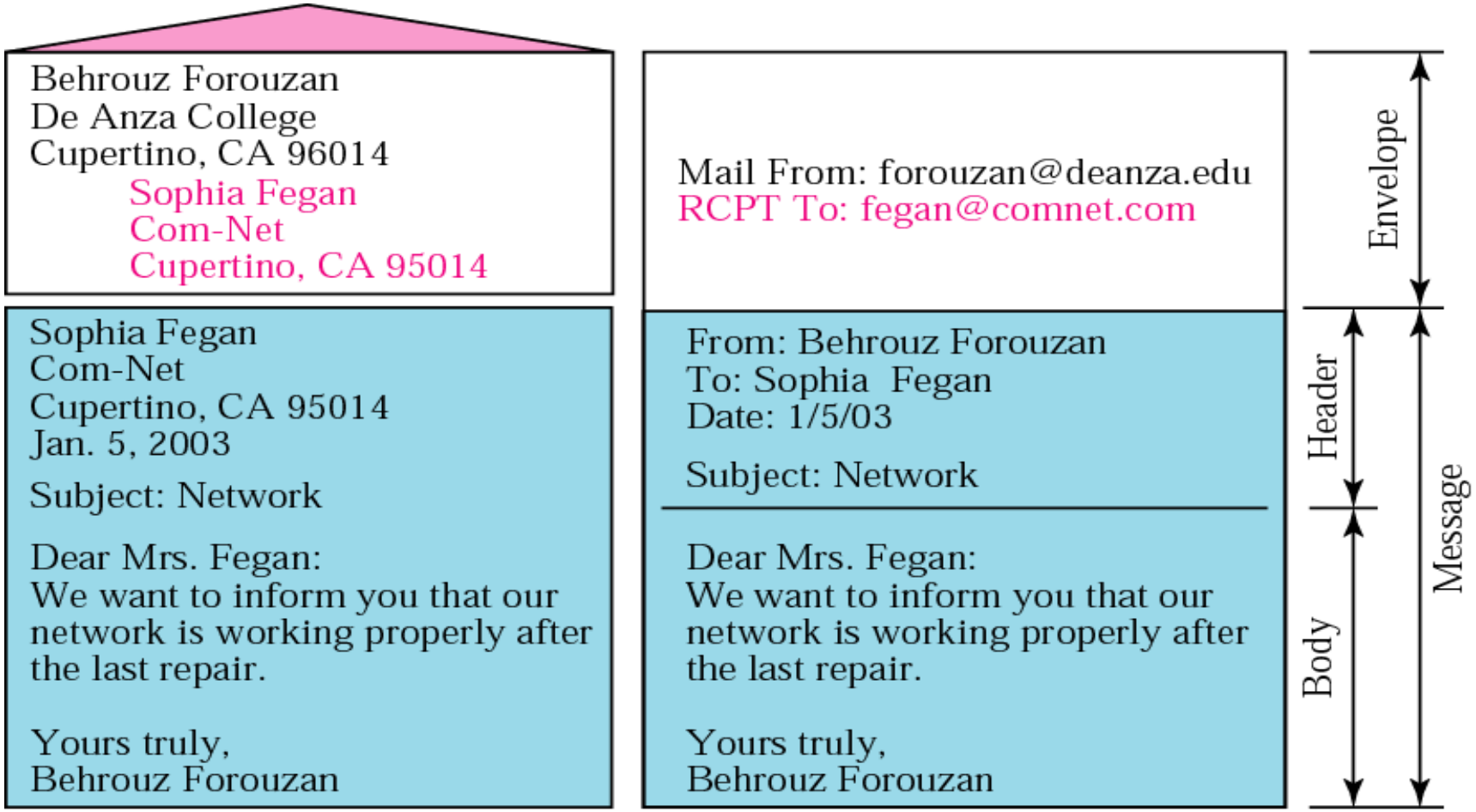
Architecture of e-mail : supports five basic functions

- composition - create messages / answers
- transfer - establish connection to destination or intermediary system, output message, release connection
- reporting - delivery status
- displaying
- disposition - delete before read, read and delete, re-read, forward, auto-forward, auto-reply, Cc, BCc, priority, encrypted,..

Computer Networks

Electronic Mail

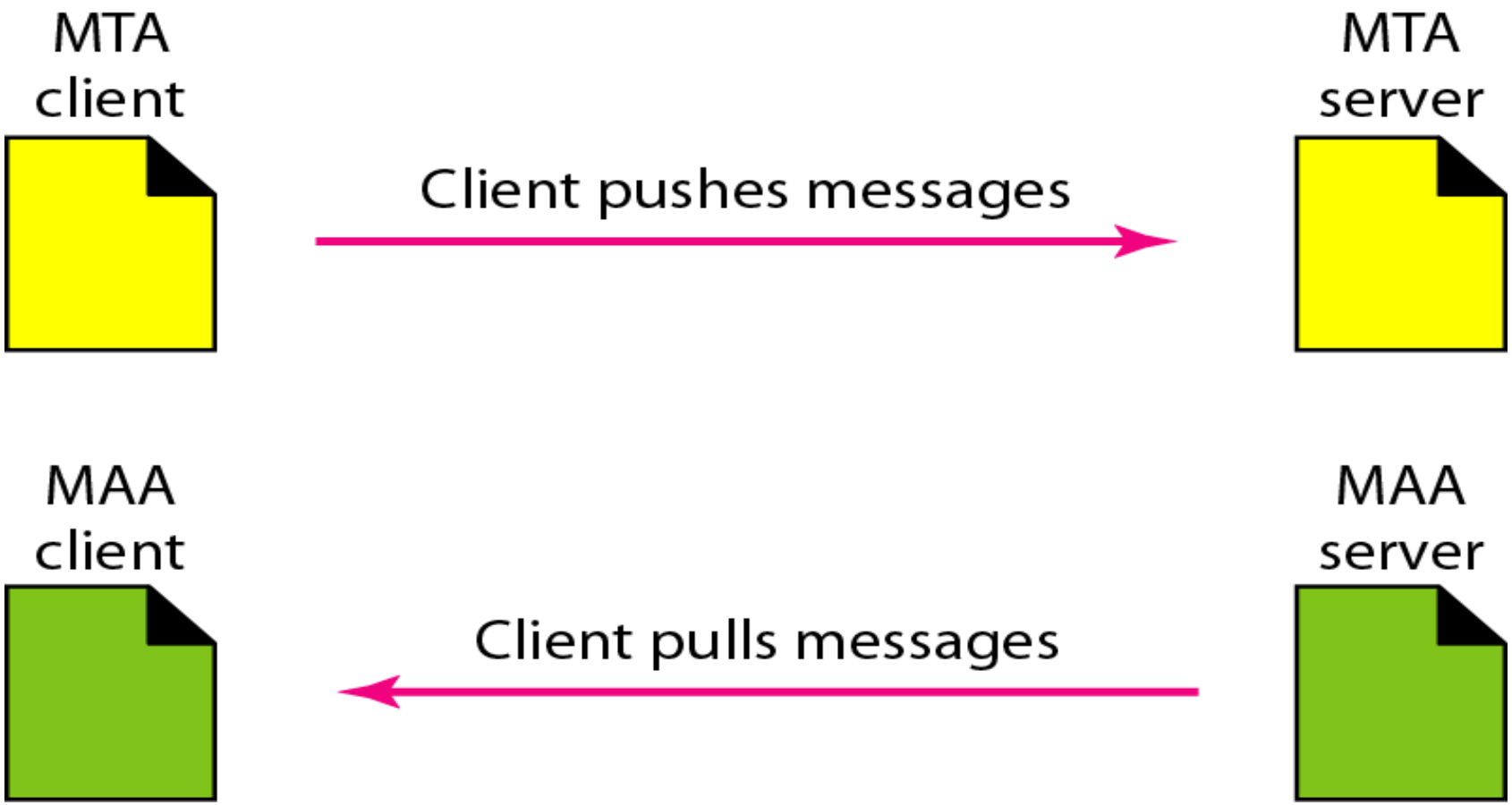
paper - mail and e-mail



Computer Networks

Electronic Mail

Push - Pull :



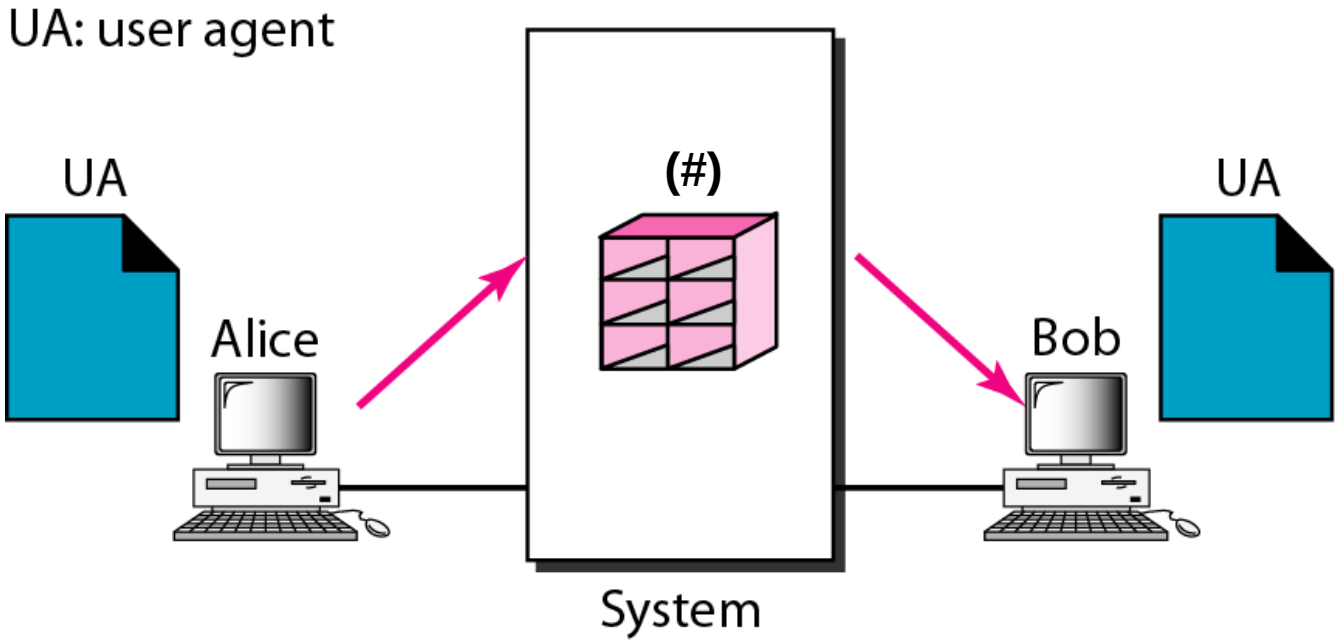
Computer Networks

Electronic Mail

First scenario :

sender & receiver on the same system

two user agents are required



 (#) : administrator has created one mail box for each user

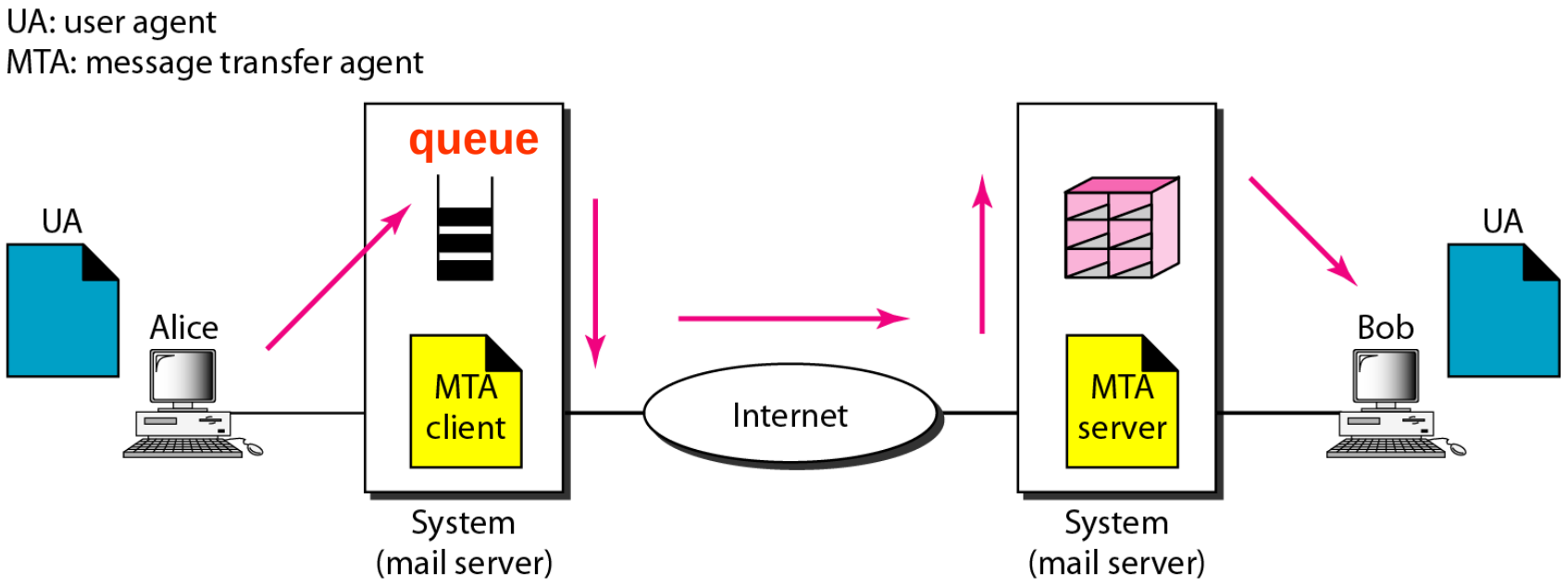
Computer Networks

Electronic Mail

Second scenario :

sender & receiver on the two different systems

two UAs and a pair of MTAs are required



Computer Networks

Electronic Mail

Third scenario :

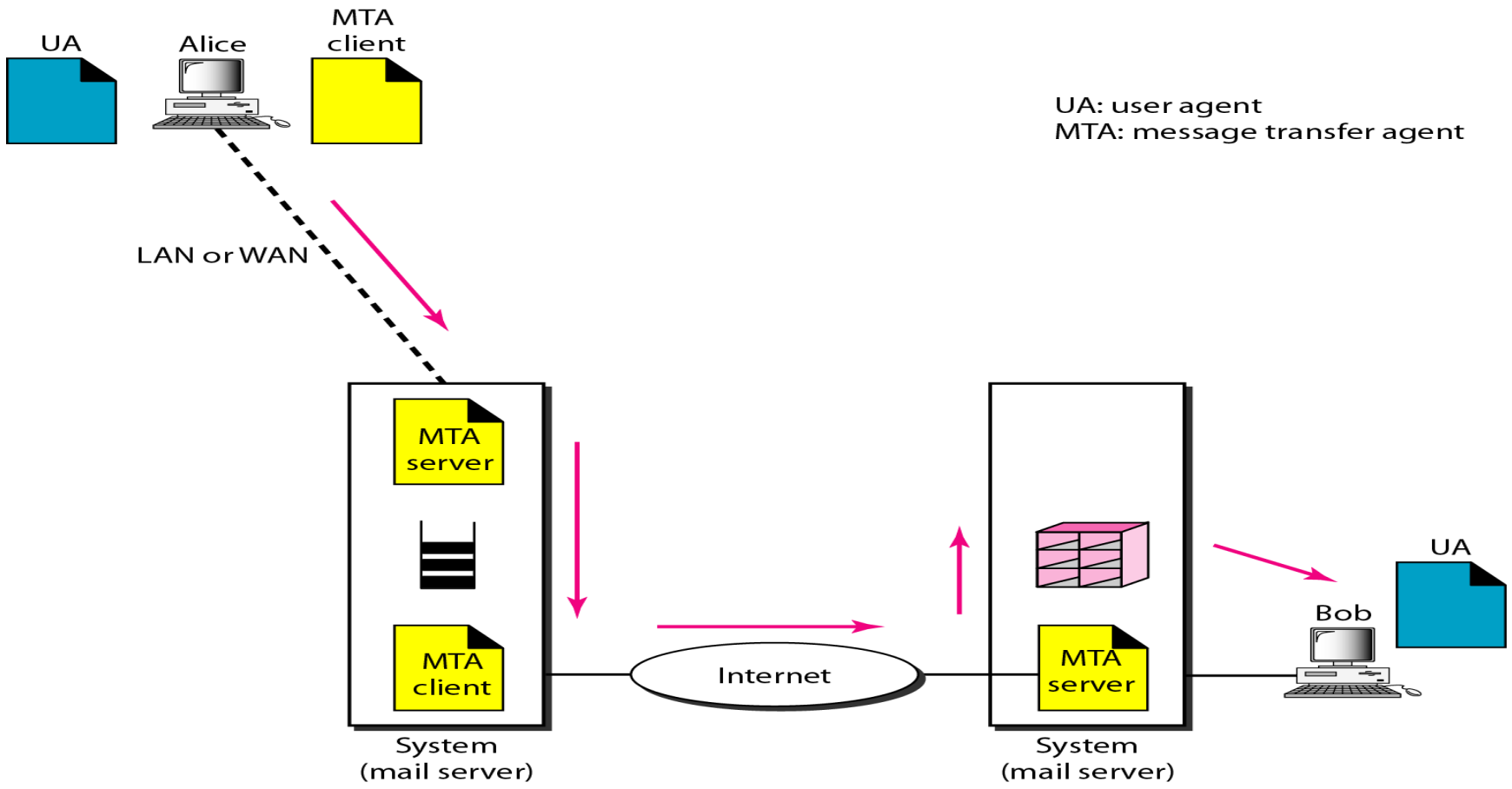
sender connected to mail server via LAN / WAN & receiver connected directly to system

two UAs and two pairs of MTAs (client & server) are required

Computer Networks

Electronic Mail

Third scenario :



Computer Networks

Electronic Mail

Fourth scenario : (most common)

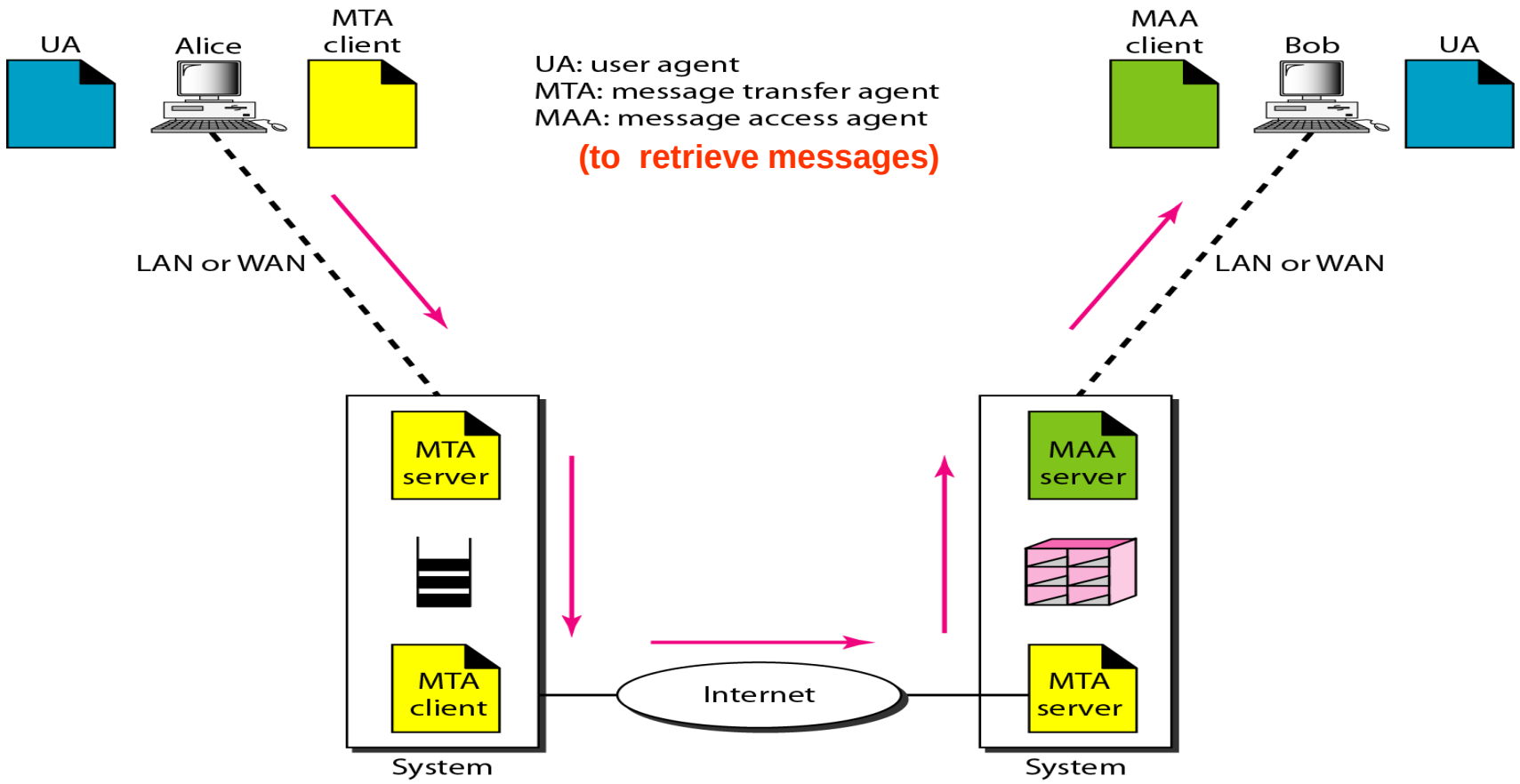
sender connected to mail server via LAN / WAN and receiver is also connected mail server via LAN / WAN

two UAs, two pairs of MTAs (client & server) and a pair of MAAs (client & server) are required

Computer Networks

Electronic Mail

Fourth scenario :



Computer Networks

Electronic Mail

ASCII e-mail message format : RFC 822 consists of a primitive envelope, header fields, a blank line and a body

basic header fields :

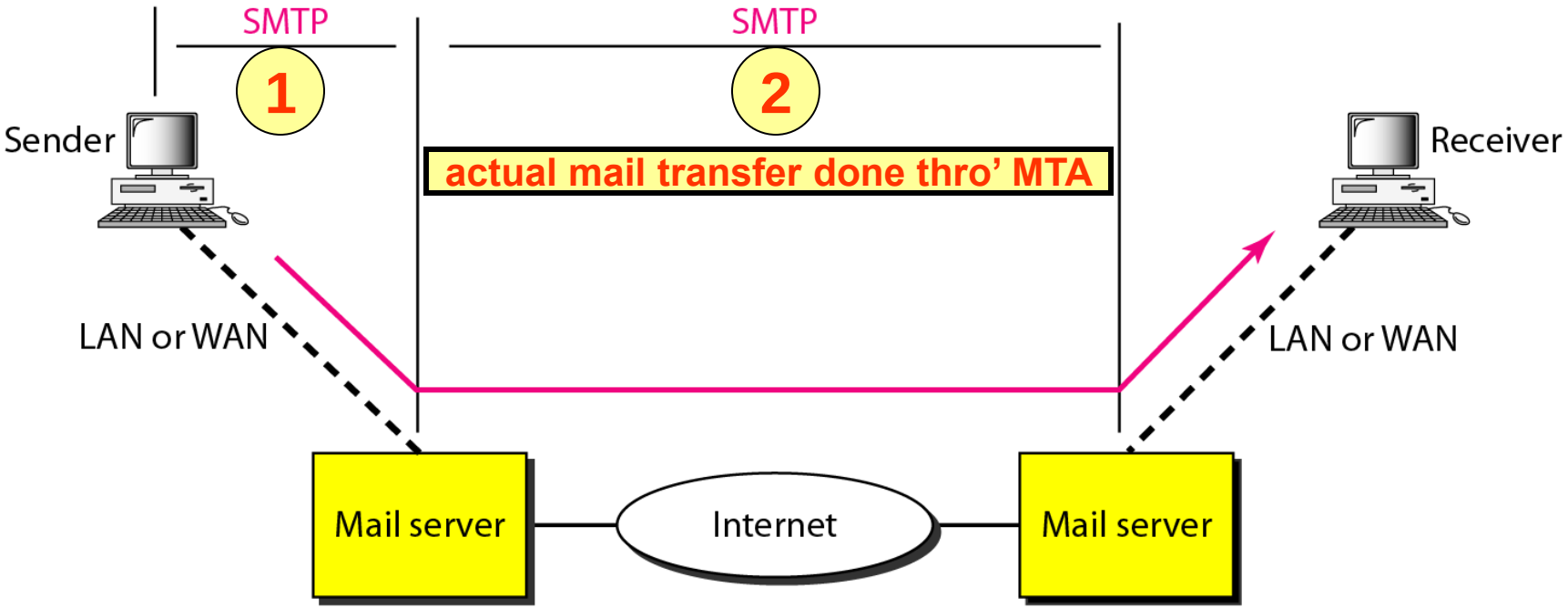
Header	Meaning
To:	E-mail address(es) of primary recipient(s)
Cc:	E-mail address(es) of secondary recipient(s)
Bcc:	E-mail address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender
Received:	Line added by each transfer agent along the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

Computer Networks

Message Transfer Agent : SMTP

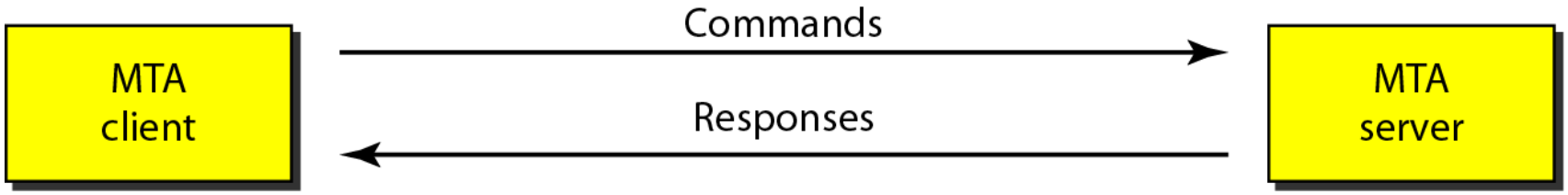
protocol that defines the MTA client (sender) and server (receiver) in the Internet is called the Simple Mail Transfer Protocol (SMTP)

uses *commands* and *responses*



Computer Networks

SMTP



Keyword: argument(s)

Command format

Computer Networks

SMTP

Commands : sent from client to server

Keyword	*Argument(s)*
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPT TO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VRFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name
SEND FROM	Intended recipient of the message
SMOL FROM	Intended recipient of the message
SMAL FROM	Intended recipient of the message

Computer Networks

SMTP

Responses : sent from server to client

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

Code	Description
	Permanent Negative Completion Reply
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

See Example 26.3 / Page 837 in Foruzan's book 4th Edition

Computer Networks

SMTP

Responses : sent from server to client

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

Code	Description
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

See Example 26.3 / Page 837 in Foruzan's book 4th Edition

Computer Networks

SMTP

mail transfer phases :

- **connection establishment**
- **mail transfer**
- **connection termination**

Computer Networks

SMTP

Example

$ telnet mail.adelphia.net 25
Trying 68.168.78.100 . . .
Connected to mail.adelphia.net (68.168.78.100).

```
==================  Connection Establishment  ================
  220 mta13.adelphia.net SMTP server ready Fri, 6 Aug 2004 . . .
HELO mail.adelphia.net
  250 mta13.adelphia.net
```

Computer Networks

SMTP

Example

```
==================        Mail Transfer        ================
MAIL FROM: forouzanb@adelphia.net
  250 Sender <forouzanb@adelphia.net> Ok
RCPT TO: forouzanb@adelphia.net
  250 Recipient <forouzanb@adelphia.net> Ok
DATA
  354 Ok Send data ending with <CRLF>.<CRLF>
From: Forouzan
TO: Forouzan

This is a test message
to show SMTP in action.
.
```

Computer Networks

SMTP

Example

```
==================== Connection Termination ================
    250 Message received: adelphia.net@mail.adelphia.net
QUIT
    221 mta13.adelphia.net SMTP server closing connection
Connection closed by foreign host.
```

Computer Networks

Electronic mail - MIME (RFCs 1341, 2045 - 2049)

SMTP can send messages only in 7-bit ASCII format -

cannot be used for languages that are not supported by 7-bit ASCII characters

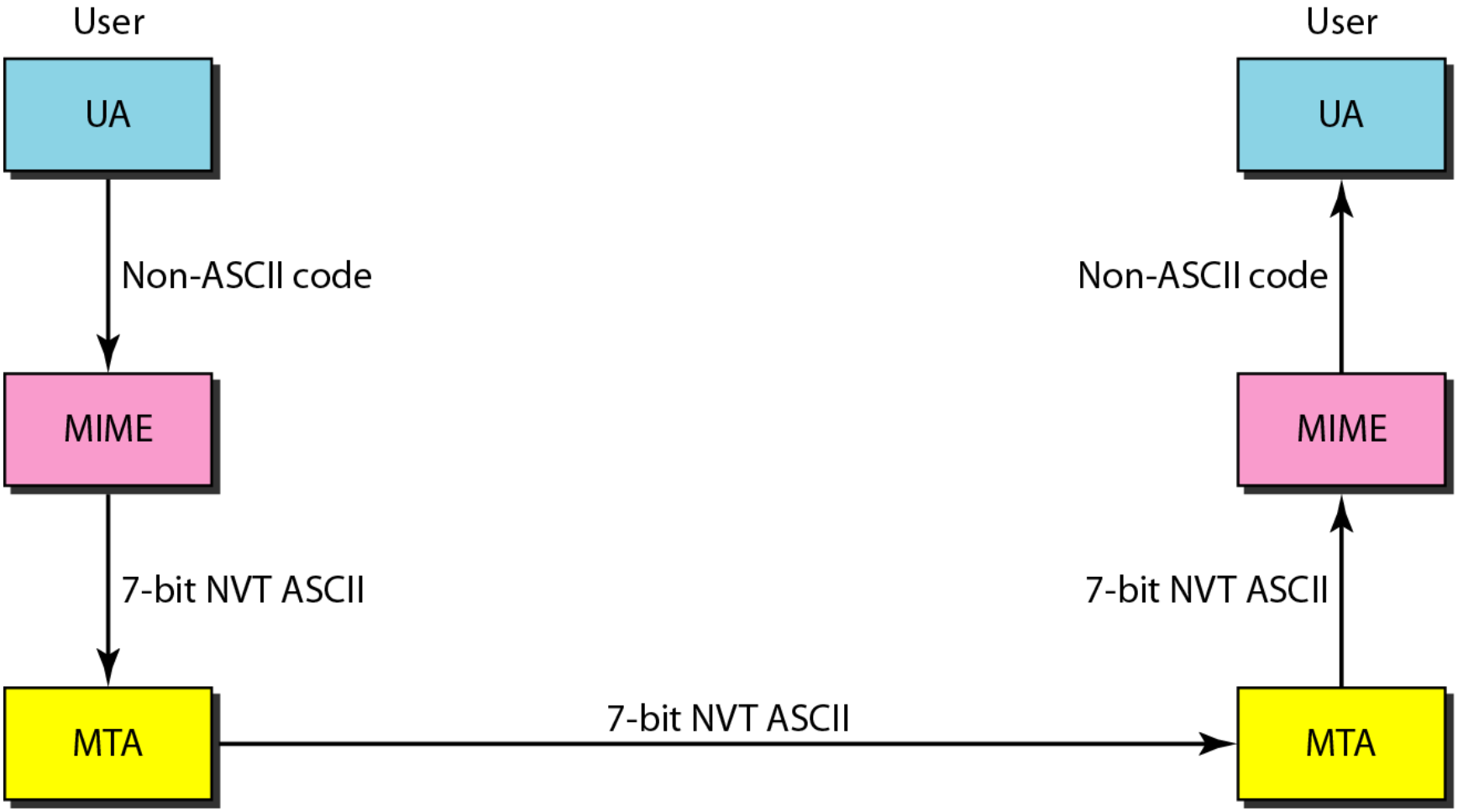
cannot be used to send messages not containing text (audio / video)

Multipurpose Internet Mail Extensions (MIME), an extension of SMTP, is a supplementary protocol

a set of software functions that transform non-ASCII data to ASCII data

Computer Networks

Electronic mail - MIME



Computer Networks

MIME

basic idea :

(a) continue to use of RFC 822 format
(b) add structure to the message body
(c) define encoding rules for non-ASCII messages

by not deviating from RFC 822 format, MIME messages can be sent using the existing mail programs and protocols

Computer Networks

MIME

defines five new message headers

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

Content-Description: defines whether the body is image, audio or video

Content-Id: uniquely identifies the whole message in a multiple-message environment

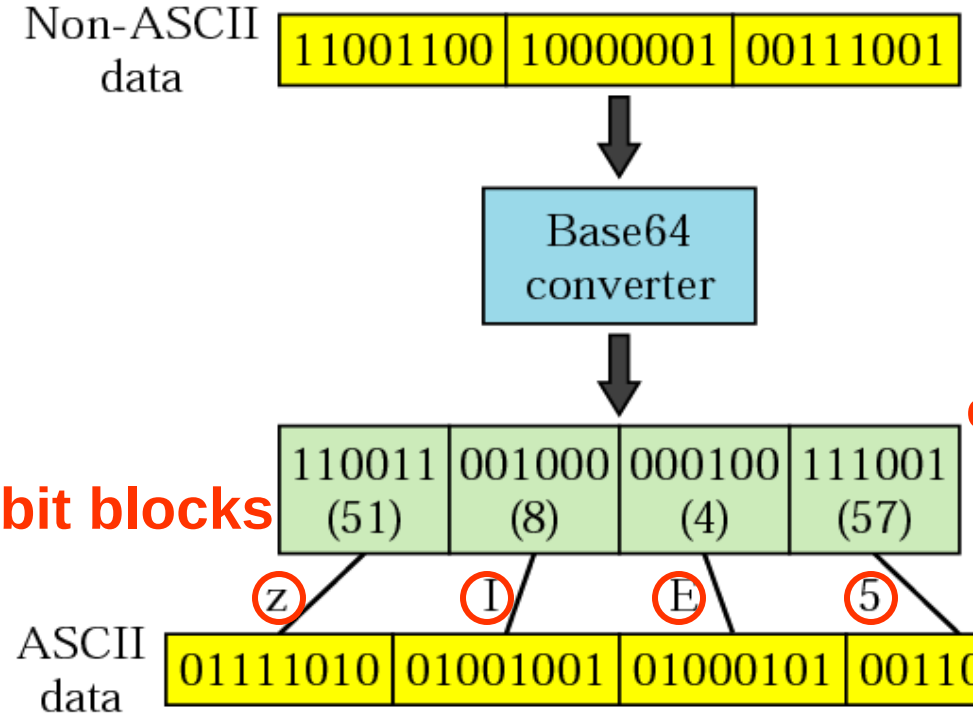
Content-Transfer-Encoding: tells how the body is wrapped for transmission thro' a network (5 types) → →

Computer Networks
MIME content transfer encoding

Type	Description
7bit	ASCII characters and short lines
8bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data are encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters are encoded as an equal sign followed by an ASCII code

Computer Networks
MIME

base64 encoding or ASCII armor for binary messages



binary stream grouped into 24-bit units

each 24-bit unit is broken into four 6-bit pieces

3-byte or 24-bit blocks

each 6-bit piece maps onto one of 64 valid ASCII characters → →

Computer Networks
MIME

base64

groups of 24 bits are broken up into four 6-bit units, with each unit sent as a character

Value	Code	Value	Code	Value	Code	Value	Code	Value	Code	Value	Code
0	A	11	L	22	W	33	h	44	s	55	3
1	B	12	M	23	X	34	i	45	t	56	4
2	C	13	N	24	Y	35	j	46	u	57	5
3	D	14	O	25	Z	36	k	47	v	58	6
4	E	15	P	26	a	37	l	48	w	59	7
5	F	16	Q	27	b	38	m	49	x	60	8
6	G	17	R	28	c	39	n	50	y	61	9
7	H	18	S	29	d	40	o	51	z	62	+
8	I	19	T	30	e	41	p	52	0	63	/
9	J	20	U	31	f	42	q	53	1		
10	K	21	V	32	g	43	r	54	2		

ASCII table

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Computer Networks
Electronic mail

Quoted-printable encoding

base64 encoding has an overhead ≈ 33%

if data consists mostly ASCII characters with a small non-ASCII portion → → *quoted-printable encoding* is used

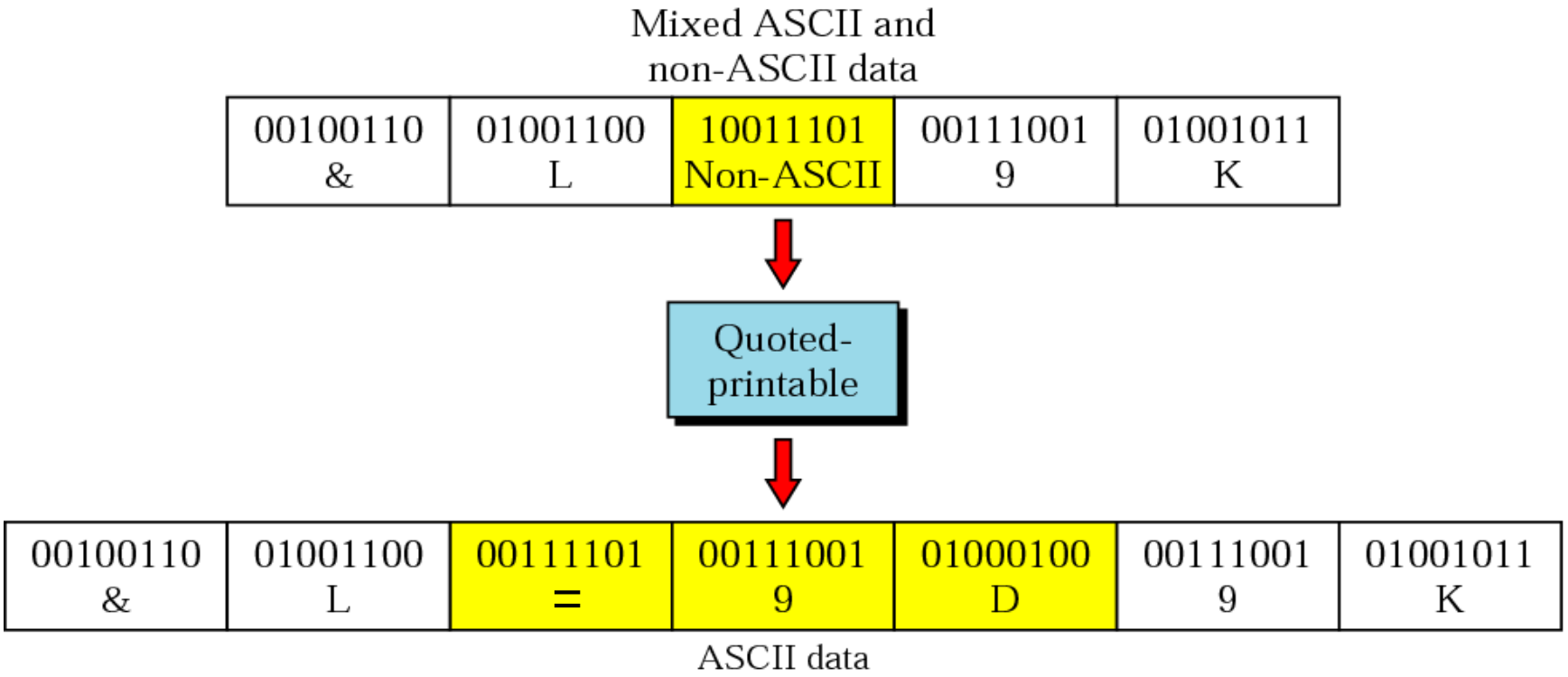
ASCII ? → sent as it is

non-ASCII ? → sent as three characters : = sign followed by

hexadecimal representation of the byte

Computer Networks
Electronic mail

Quoted-printable encoding



Computer Networks

MIME data content types and subtypes (RFC 2045)

Seven types

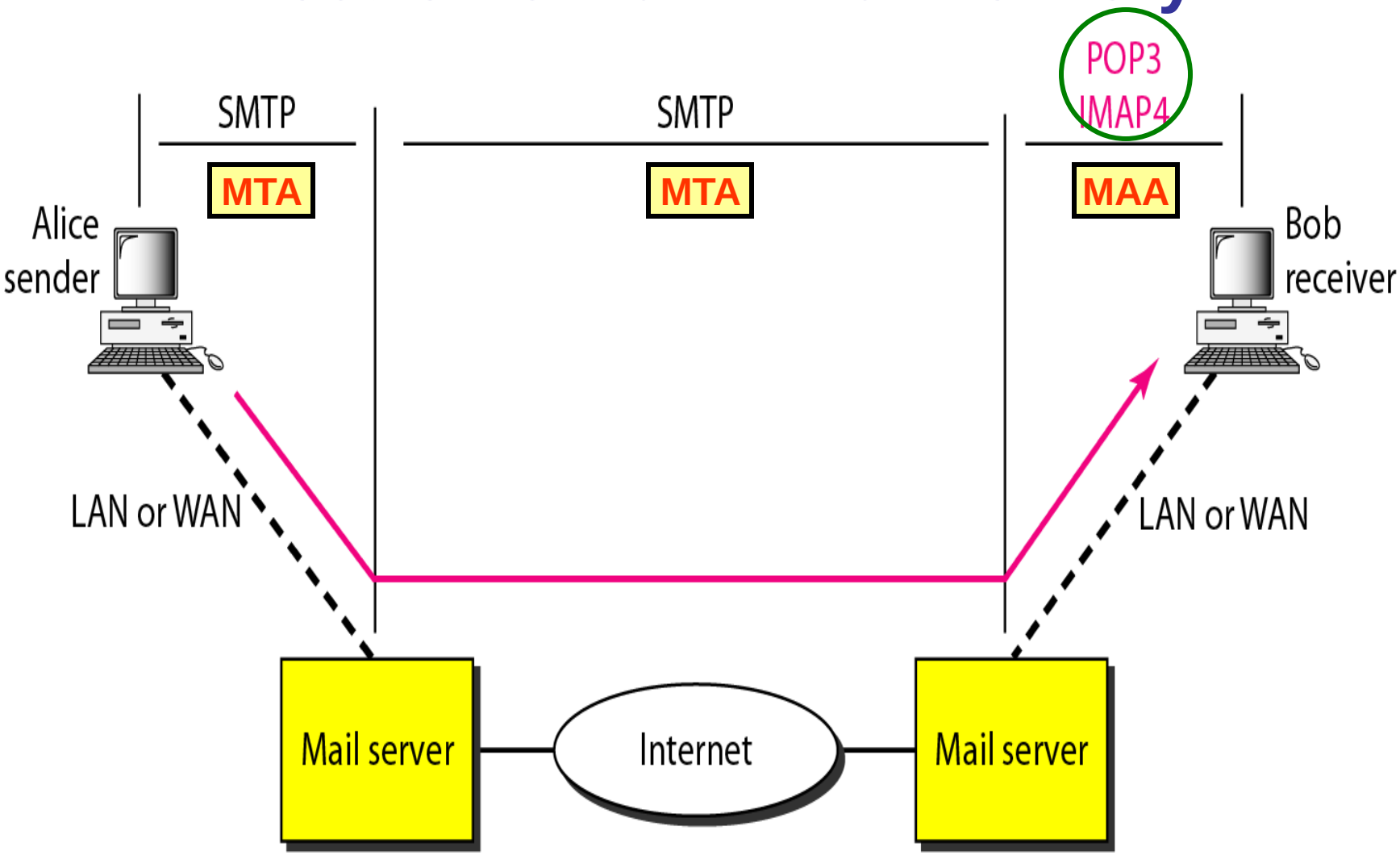
PP 598-601 Tanenbaum

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

Example → Content-Type: video/mpeg

Computer Networks

Electronic mail - final Delivery



Computer Networks

Electronic mail - final Delivery

sender has to establish on-line connection to send mail ; usually sends (***push***es) it to a server using SMTP

intended receiver need not be on-line to receive mail

receiver needs to ***pull*** the mail from the machine (mail server) to which the sender's mail has been delivered

required : a protocol which allows UAs on client PC to contact the MTA on some server and copy the mail from the server to the user's system

Computer Networks

Electronic mail - POP3 (Post Office Protocol)

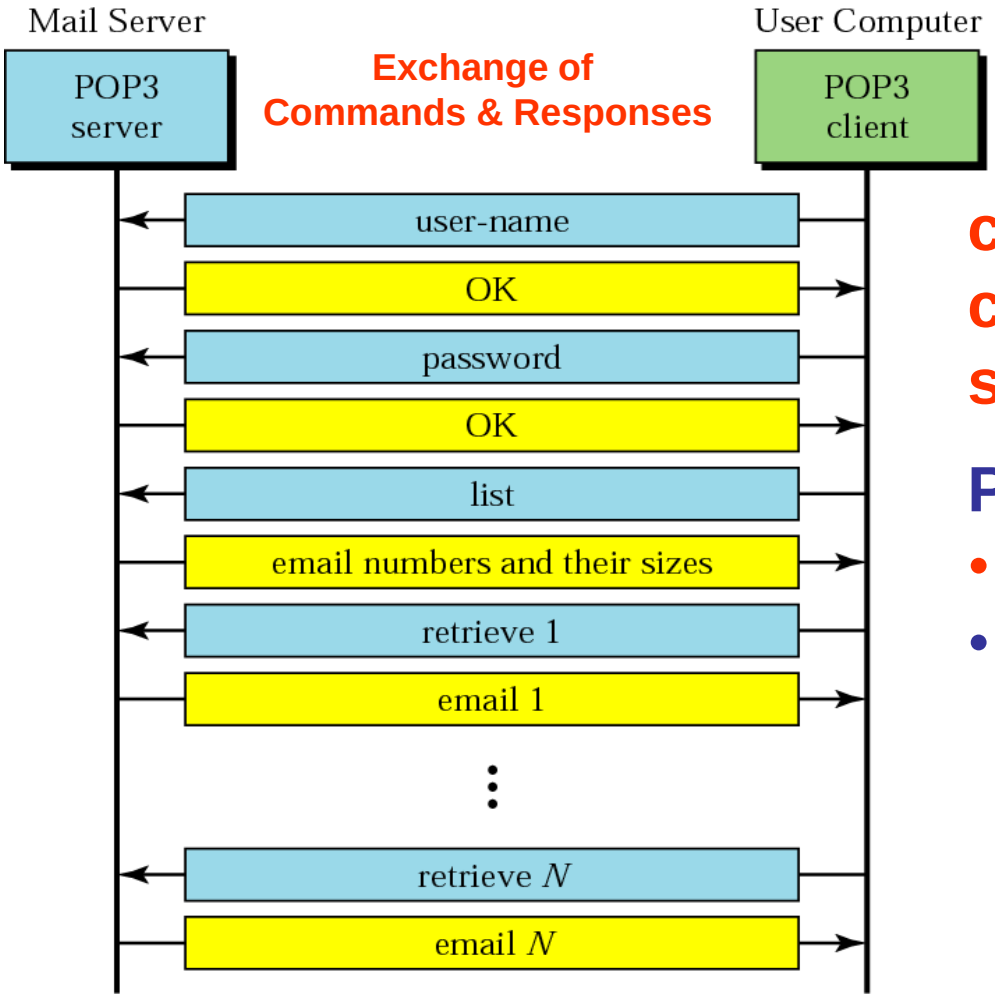
client agent opens a connection with the server TCP on port 110

three phases follow :

- authorization : user logs in
- transactions : user collects e-mails and marks them for deletion from the mail box
- update : deletion of e-mails

Computer Networks

Electronic mail - POP3 (Post Office Protocol version 3)



client agent opens a connection with the server TCP on port 110

POP3 has two modes :

- **delete mode**
- **keep mode**

Computer Networks

End of Class - 26
25.03.2015

Computer Networks

MIME

Example - 1

WS NSE 3e p173

The 24-bit raw bit sequence is :

00100011 01011100 10010001 or 235C91 (hex)

Express the above in base64 encoding

Computer Networks

MIME

Example - 1

WS NSE 3e p173

00100011 01011100 10010001

001000 110101 110010 010001

8 53 50 17

I 1 y R

49 31 79 52 (hex)

Computer Networks
MIME

Example - 2

AT 7.13

A binary file is 3072 bytes long.

What would be the overhead in % if the above file is encoded using base64 encoding.

Assume that a {CR + LF} pair is inserted after every 80 bytes sent and at the end.

Computer Networks
MIME

Example - 2 AT 7.13

file size of 3072 bytes is broken into 3-bytes blocks x 1024 by base64

each block is encoded into 4-bytes

i.e. a total of 1024 x4 = 4096 bytes

4096 bytes are broken into lines → →
(51 lines x 80 bytes) + (1 line x 16 bytes)

52 lines + 52 CRs + 52 LFs → →

total length = 4200 bytes → →

Overhead = 36.72 %

Computer Networks

Electronic mail despatch & receipt - Summary

1. Sender creates a message for using some e-mail program or UA
2. Sender initiates action to send
3. UA hands the message over to the MTA on the sender's host
4. MTA sees it is directed to : **receiver@dns-address**
5. MTA uses DNS to resolve the dns-address and gets the IP address

Computer Networks

Electronic mail despatch & receipt - Summary

6. MTA establishes a TCP connection to the SMTP server on port # 25 of this machine

7. Using an SMTP command sequence, MTA transfers message to receiver's mail box and breaks the TCP connection

8. Receiver's e-mail program or UA establishes a TCP connection on port # 110 of the mail server →

Computer Networks

Electronic mail despatch & receipt - Summary

9. **Receiver's UA runs the POP3 protocol to fetch the mail contents of the mail box to the local hard disk**

10. **Once all the mail has been transferred, the TCP connection is released**

- compose mail (using UA)
- initiate action to send
- use DNS / UDP to get IP address
- establish TCP connection on port #25
- transfer mail using MTA (SMTP) - ***Push*** operation
- release TCP connection

- establish TCP connection on port #110
- fetch mail using MAA (POP3) - ***Pull*** operation
- release TCP connection

Computer Networks

Electronic mail - IMAP4 (Internet Message Access Protocol)

disadvantages of POP3 :

- assumes that each time a client accesses the server, the entire mail box is downloaded
- is not convenient when user has to access mail box from different computers at different locations
- does not allow creation of folders on the server
- does not allow partial checking of mail before downloading

Computer Networks

Electronic mail - IMAP4 (Internet Message Access Protocol)

client-server protocol running over TCP - provides the following extra functions :

- **user can check e-mail header prior to downloading**
- **user can search (string-based) contents of e-mail prior to downloading**
- **user can create, delete, or rename mail boxes on the mail server**
- **user can create a hierarchy of mail boxes in a folder for e-mail storage**

Computer Networks
Electronic mail - POP3 vs IMAP4

Feature	POP3	IMAP
Where is protocol defined?	RFC 1939	RFC 2060
Which TCP port is 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

Computer Networks

Electronic mail - Web-based mail

mail transfer from sender's browser to the sender's mail server is done thro' *HTTP*

transfer of message from the sending mail server to the receiving mail server is done through *SMTP*

finally, message from the receiving server (web server) to receiver's browser is done thro' *HTTP*

Computer Networks

Electronic mail - Web-based mail

when receiver needs to receive mails, sends a message to the website →
website sends a form to be filled up →
login name & password match ? →
mail is transferred to the receiver's browser in *HTML* format

Examples of Webmail :

rediff, yahoo, hotmail, etc.

Computer Networks

File Transfer

need :

- **to transfer files from one computer to another in a networking or internetworking environment**

currently, the greatest volume of data exchange through the Internet is file transfer

solution :

file transfer protocol (FTP)

Computer Networks

File Transfer Protocol (FTP)

is the standard mechanism provided by TCP/IP for copying a file from one host to another

issues handled by FTP……

two systems may have :

- **different file naming conventions**
- **different ways to represent text and data**
- **different directory structures**

Computer Networks

File Transfer Protocol (FTP)

in terms of TCP connections, FTP differs from other client server applications

FTP establishes ***two*** connections :

- ***data*** connection, on port ***20***, for data transfer
- ***control*** connection, on port ***21***, for commands and responses

separation of commands and responses makes FTP more efficient

Computer Networks

File Transfer Protocol (FTP)

control connection uses simple rules of communications -- need to transfer only a line of command or response at a time

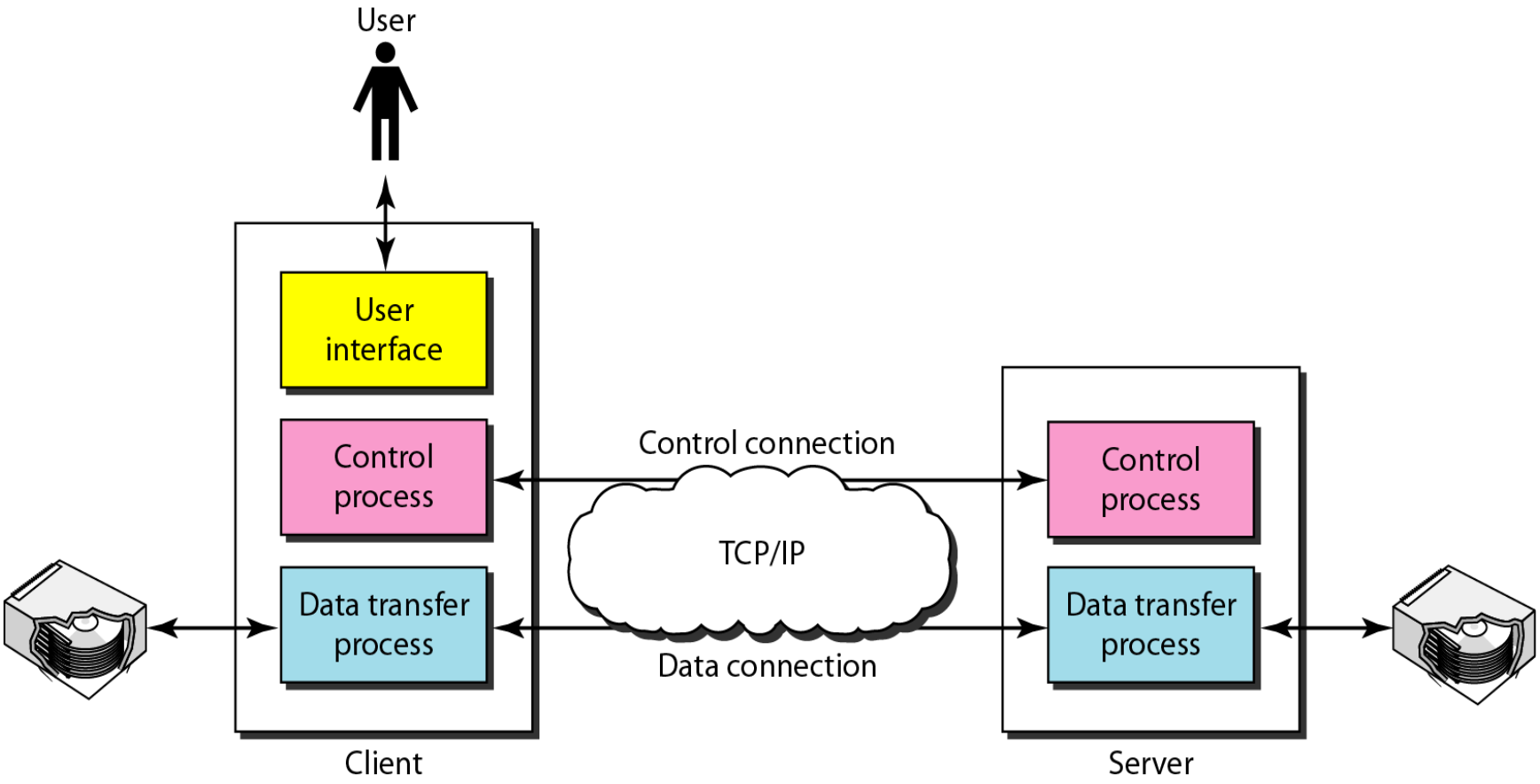
data connection needs more complex rules due to the variety of data types transferred

TCP treats both the connections the same way

Computer Networks

File Transfer Protocol (FTP)

Basic model of FTP



Computer Networks

File Transfer Protocol (FTP)

Basic model of FTP

control connection :

- **remains connected during the entire interactive FTP session**

data connection:

- **is opened and closed for each file transferred**
- **opens each time commands that involve file transfer are used**
- **closes when the file is transferred**

Computer Networks

File Transfer Protocol (FTP)

Communication over the control connection

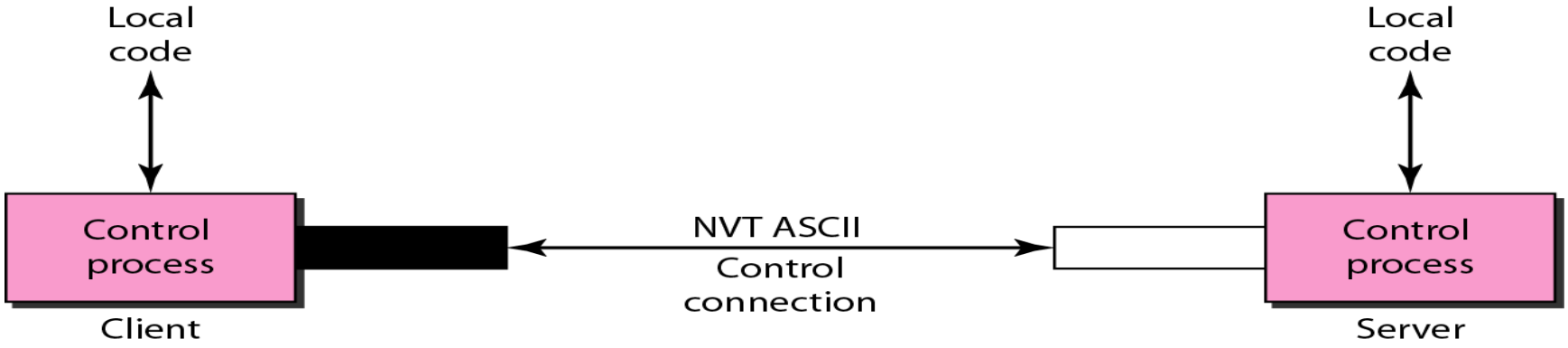
FTP uses 7-bit ASCII character set

communication is achieved through commands and responses ……

a short line, sent one at a time

no concern regarding file format / structure

each line is terminated with CR and LF



Computer Networks

File Transfer Protocol (FTP)

Communication over the data connection

file transfer happens over the data connection …..

under the control of commands sent over the control connection

file transfer in FTP involves :

(a) a file to be copied from server to client,
called *retrieving a file*
done under the supervision of the RETR command

Computer Networks

File Transfer Protocol (FTP)

Communication over the data connection

file transfer in FTP involves :

(b) a file to be copied from client to server, called *storing a file*
done under the supervision of the STOR command

(c) a list of directory or file names to be sent from the server to the client, done under the supervision of the LIST command

FTP treats a list of directory or file names as a file, which is sent over the data connection

Computer Networks

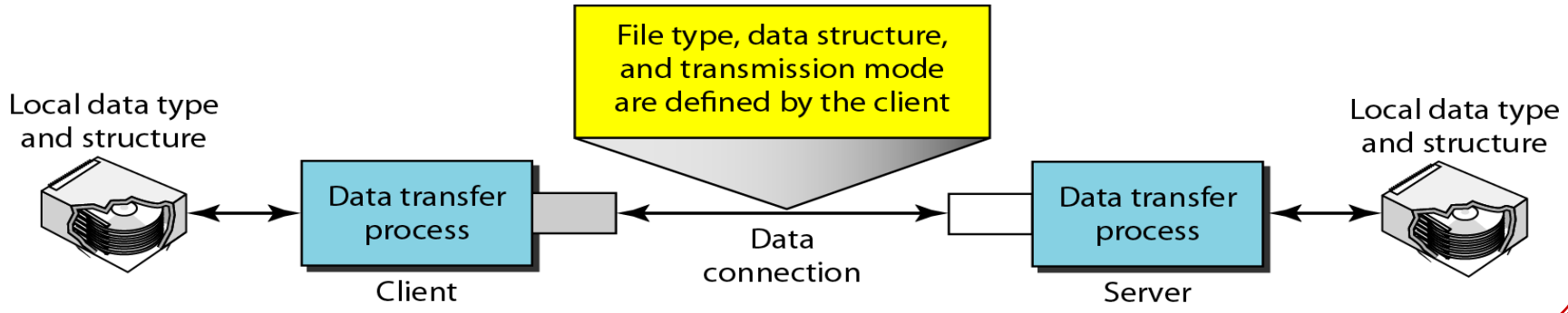
File Transfer Protocol (FTP)

Communication over the data connection

client must define the following attributes of communication :

- **type of file to be transferred** (ASCII, EBCDIC, image)
- **structure of the data** (file, record, page)
- **transmission mode** (stream, block, compressed)

the above process resolves the issue related to heterogeneity



Computer Networks

File Transfer Protocol (FTP)

Communication over the data connection

File Type

types of files that can be transferred across the data connection :

(a) ASCII :

default format for transferring text files

each character is encoded using 7-bit ASCII

sender and receiver transform the file from its own representation into ASCII and vice versa

Computer Networks

File Transfer Protocol (FTP)

Communication over the data connection

File Type

(b) EBCDIC :

the file format used by IBM

if one or both ends use EBCDIC encoding, the file can be transferred with this encoding

(c) Image :

default format for transferring binary files

sent as continuous stream of bits without any interpretation or encoding

Example : executable files

Computer Networks

File Transfer Protocol (FTP)

Communication over the data connection

Data structure

(a) file structure :

the file is a continuous stream of bytes

(b) record structure :

the file is divided into records

can be used with only text files

Computer Networks

File Transfer Protocol (FTP)

Communication over the data connection

Data structure

(c) page structure :

the file is divided into pages

each page has a page number and a page header

pages can be stored and accessed randomly or sequentially

Computer Networks

File Transfer Protocol (FTP)

Communication over the data connection

Transmission mode

(a) Stream mode :

this is the default mode

data is delivered from TCP to FTP as a continuous stream of bytes

if the data is file structure, no EOF is needed

closing of data connection by the sender defines the EOF

Computer Networks

File Transfer Protocol (FTP)

Communication over the data connection

Transmission mode

(a) Stream mode (contd..) :

if the data is record structure, each record will have a 1-byte EOR character and the end of file will have a 1-byte EOF

(b) Block mode :

data is delivered to TCP in blocks

each block is preceded by a 3-byte header

first byte is called the *block descriptor*

the next 2 bytes : define size of block in bytes

Computer Networks

File Transfer Protocol (FTP)

Communication over the data connection

Transmission mode

(c) Compressed mode :

if the file is big, data can be compressed

compression method used normally is run-length encoding -

consecutive appearances of data unit are replaced by one occurrence and the number of repetitions

text file : blanks are compressed

binary file : null characters are compressed

Computer Networks

File Transfer Protocol (FTP)

Example

black lines : commands sent by the client

```
$ ftp voyager.deanza.fhda.edu
Connected to voyager.deanza.fhda.edu.
220 (vsFTPd 1.2.1)
530 Please login with USER and PASS.
Name (voyager.deanza.fhda.edu:forouzan): forouzan
331 Please specify the password.
Password: ******
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls reports
227 Entering Passive Mode (153,18,17,11,238,169)
150 Here comes the directory listing.
drwxr-xr-x   2 3027      411          4096 Sep 24  2002 business
drwxr-xr-x   2 3027      411          4096 Sep 24  2002 personal
drwxr-xr-x   2 3027      411          4096 Sep 24  2002 school
226 Directory send OK.
ftp> quit
221 Goodbye.
```

service ready response sent over control connection

coloured lines : responses from the server control connection

to find the list of files on the directory named *'report'*

server opens the data connection

data transfer

server closes the data connection

server closes the control connection

Computer Networks

File Transfer Protocol (FTP)

Some ftp commands (Unix)

?	help about ftp commands
ascii	set the mode of file transfer to ASCII
binary	set the mode of file transfer to binary
bye	exit the ftp environment
quit	exit the ftp environment
close	terminate a connection with other computer

Computer Networks

File Transfer Protocol (FTP)

Some ftp commands (Unix)

open open a connection with other computer

get download a file

mget download multiple files (prompted)

mget* download all files

put upload a file

mput upload multiple files (prompted)

Computer Networks

File Transfer Protocol (FTP)

Some ftp commands (Unix)

?	*to request help or information about the FTP commands*	
ascii	*to set the mode of file transfer to ASCII* *(this is the default and transmits seven bits per character)*	
binary	*to set the mode of file transfer to binary* *(the binary mode transmits all eight bits per byte and thus provides less chance of a transmission error and must be used to transmit files other than ASCII files)*	
bye	*to exit the FTP environment (same as quit)*	
cd	*to change directory on the remote machine*	
close	*to terminate a connection with another computer*	
	close brubeck	closes the current FTP connection with brubeck, but still leaves you within the FTP environment.
delete	*to delete (remove) a file in the current remote directory (same as rm in UNIX)*	
get	*to copy one file from the remote machine to the local machine*	
	get ABC DEF	copies file ABC in the current remote directory to (or on top of) a file named DEF in your current local directory.
	get ABC	copies file ABC in the current remote directory to (or on top of) a file with the same name, ABC, in your current local directory.
help	*to request a list of all available FTP commands*	
lcd	*to change directory on your local machine (same as UNIX cd)*	
ls	*to list the names of the files in the current remote directory*	
mkdir	*to make a new directory within the current remote directory*	
mget	*to copy multiple files from the remote machine to the local machine;* *you are prompted for a y/n answer before transferring each file*	
	mget *	copies all the files in the current remote directory to your current local directory, using the same filenames. Notice the use of the wild card character, *.
mput	*to copy multiple files from the local machine to the remote machine;* *you are prompted for a y/n answer before transferring each file*	
open	*to open a connection with another computer*	
	open brubeck	opens a new FTP connection with brubeck; you must enter a username and password for a brubeck account (unless it is to be an anonymous connection).
put	*to copy one file from the local machine to the remote machine*	
pwd	*to find out the pathname of the current directory on the remote machine*	
quit	*to exit the FTP environment (same as bye)*	
rmdir	*to to remove (delete) a directory in the current remote directory*	

Computer Networks

Anonymous FTP

some sites allow use of files for public access , without the need for an FTP account i.e. user name and password

anonymous FTP facilitates this access

the user can use the following way :

user name ***anonymous***

password ***guest***

generally user access to resources is limited

Computer Networks

Anonymous FTP

Example

```
$ ftp internic.net
Connected to internic.net
220 Server ready
Name: anonymous
331 Guest login OK, send "guest" as password
Password: guest
```

Computer Networks

Anonymous FTP

Example (contd..)

```
ftp > pwd
257 '/' is current directory
ftp > ls
200 OK
150 Opening ASCII mode
bin
. . .
. . .
. . .
ftp > close
221 Goodbye
ftp > quit
```

Computer Networks

World Wide Web

a repository of information linked together

an architectural framework for accessing linked documents spread over a large number of machines all over the Internet

features : flexibility, portability, user-friendly

provides enormous wealth of information

began in 1989 at CERN (*Conseil Européen pour la Recherche Nucléaire*) -

European Council for Nuclear Research

Computer Networks

World Wide Web

CERN & MIT - agreement in 1994 to set up World Wide Web Consortium (W3C) - to develop the Web, standardise protocols, address interoperability issues

website : *www.w3.org*

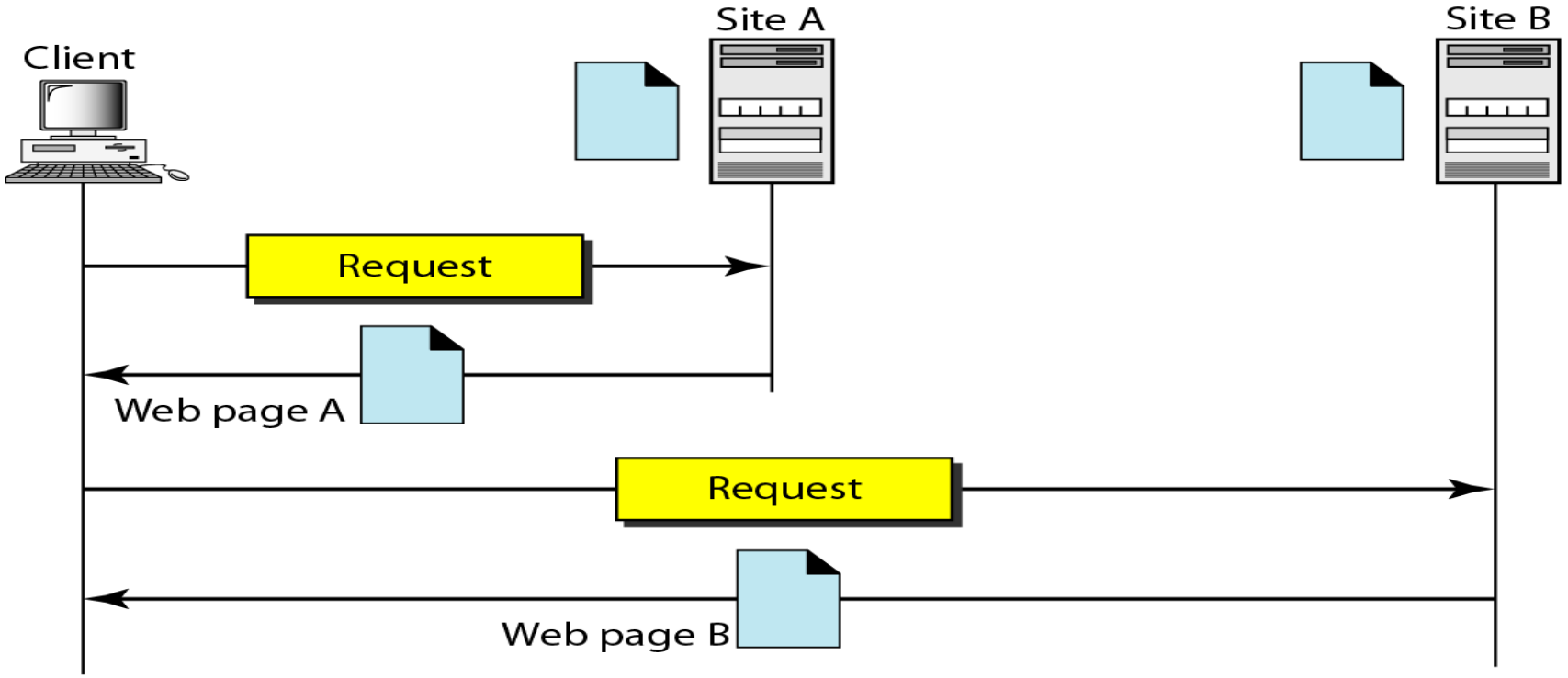
the best place to get up-to-date information about the Web is on the Web itself !

Computer Networks

World Wide Web

Architecture

client-server service distributed over many locations called (*web*) *sites*



Computer Networks
World Wide Web
Architecture

Client (browser)

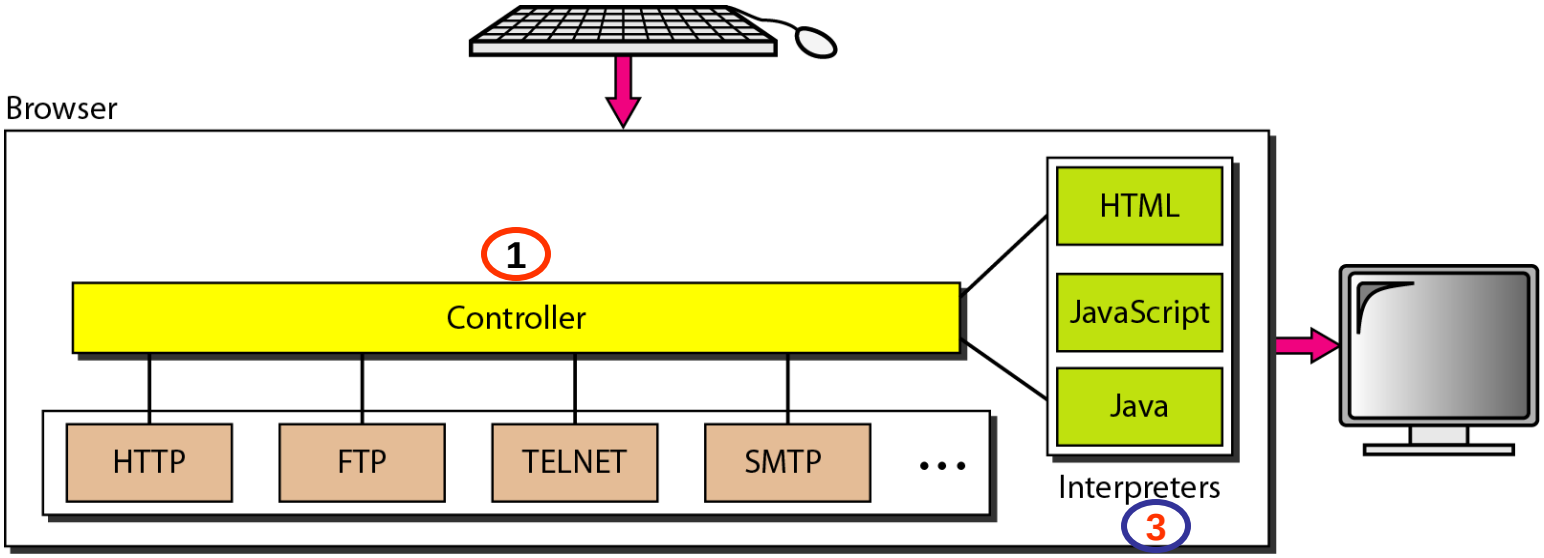
web pages **can be retrieved and viewed using a program called a *browser***

a web browser :

- **is a (client) software application**
- **enables a user to retrieve, display and interact with text, images, and other information……. typically located on a web page at a website on the world wide web or on a LAN**

Computer Networks
World Wide Web

Browser



3 **Interpreter for displaying documents (depends on type of document)**

- **controller**
- **client protocol**
- **interpreter**

Computer Networks
World Wide Web

Client (browser)

Controller :

- **receives input from keyboard or the mouse**
- **uses client programs to access documents**
- **uses one of the interpreters to display document on the screen**

Web Server :

- **stores / caches web pages**
- **sends documents in response to clients' requests**
- **may use multithreading or multiprocessing**

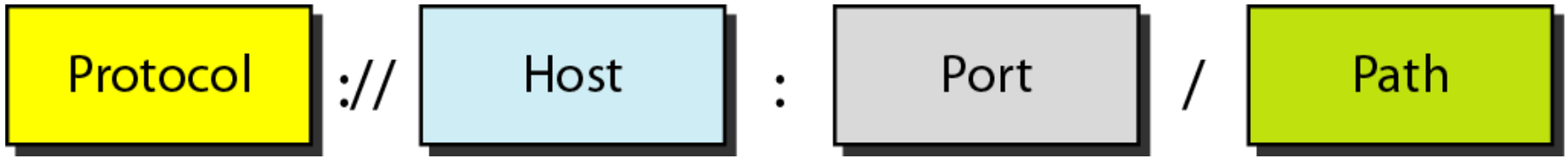
Computer Networks

World Wide Web

Uniform Resource Locator (URL)

the page's world wide name

HTTP protocol, used to retrieve documents,
uses locators to facilitate a client to ….
access web pages / documents located throughout the world

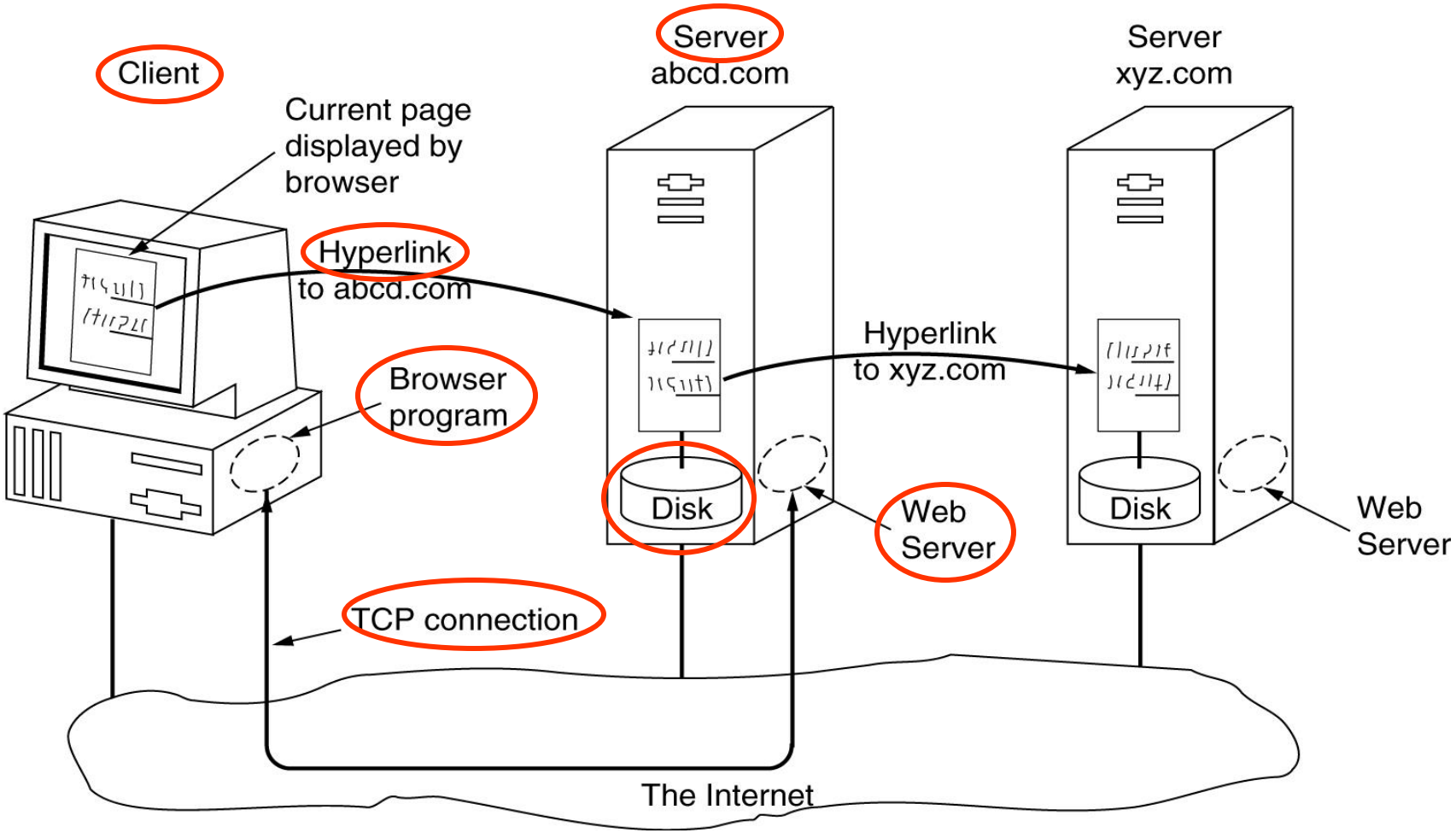


how ? **where ?**

http **www.abcd.com** **products.html**

Computer Networks
World Wide Web

Web model



Computer Networks
World Wide Web

The Client Side

Fetching and display of web page : steps

1. browser determines the URL entered or selected (clicked) by user
2. browser queries **DNS** for the IP address of the host (server) name in the URL
3. DNS replies with the IP address
4. browser makes a TCP connection to port # 80 on the (now known) server IP address
5. server accepts the TCP connection

Computer Networks
World Wide Web
Fetching and display of web page : steps

6. browser sends over a request asking for the file with the name contained in the URL
7. server gets the name of the file asked for, gets it from disk and sends the file
8. TCP connection is released
9. browser displays text contained in the file
10. browser displays images contained in the file

Computer Networks

World Wide Web

Browser is basically an HTML interpreter

additional buttons / features :

- **navigation : ← ; → ; home page**
- **book marks for pages for easy revisiting**

Web page contents :

- **ordinary text, hyperlinked text (HTML)**
- **icons, drawings, photographs etc. with or without hyperlinks**
  (PDF document, GIF / MPEG, MP3, JPEG, …….etc.)

Computer Networks

World Wide Web

how are these handled in the design of browsers leaving flexibility for handling existing and future file types ?

- **keep the browser simple - to display text pages (of type *text/html*) directly**
- **use plug-ins / helpers to handle others**

when server returns a page, it also returns additional information about that page i.e. MIME type of the page → →

Computer Networks

MIME data content types and subtypes (RFC 2045)

PP 598-601 Tanenbaum

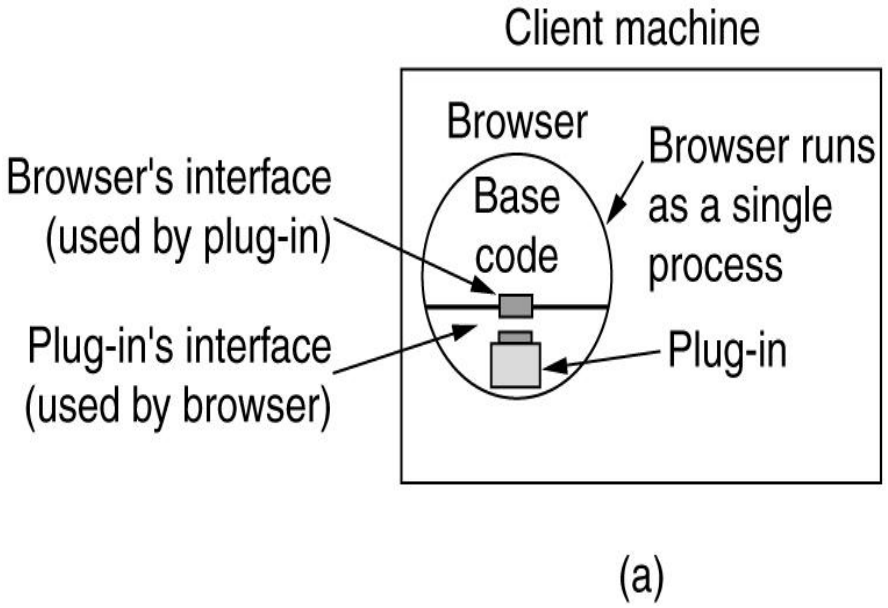
Type	Subtype	Description
Text	Plain	Unformatted text
	Enriched	Text including simple formatting commands
Image	Gif	Still picture in GIF format
	Jpeg	Still picture in JPEG format
Audio	Basic	Audible sound
Video	Mpeg	Movie in MPEG format
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	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

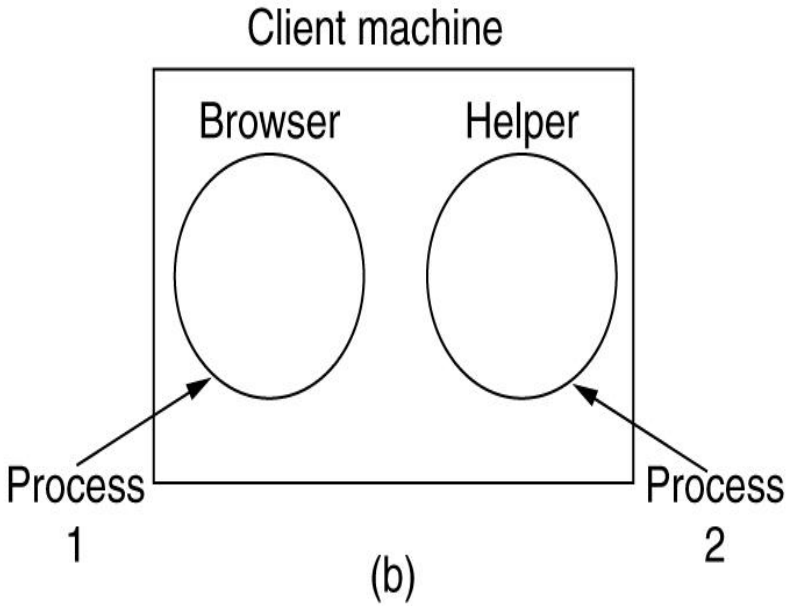
Example → Content-Type: video/mpeg

Computer Networks

Web browser :

Plug-ins and Helper Applications





Plug-in **Helper application**

Computer Networks

Web browser : plug-in

A *plug-in* is a program that interacts with a host application (e.g. browser) to provide a certain, usually very specific, function *on demand*

Applications support plug-ins for :

- **enabling third party developers to create capabilities to extend an application**
- **to support features yet unforeseen**
- **to reduce the size of an application**

Computer Networks

Web browser : plug-in

- is a code module that the browser fetches from the disk and installs as an extension to itself
- must implement a set of procedures as laid down by the browser
- runs inside the browser, so it can have access to the current page
- is removed from the memory after the job is done i.e. the user quits the page
- **Eg.** Adobe Flash Player

Computer Networks

Web browser : helper application

A ***helper application*** **is an external viewer program launched to display content retrieved using a browser**

unlike a plug-in (whose full code is included into browser code), …….

a small line is added to the browser code to tell it to open a certain helper application in case it encounters a certain file format

Computer Networks

Web browser : helper application

- is a complete program, runs as a separate process
- accepts the name of a file where content has been stored, opens file and displays contents
- exists independent of the browser, ex. Acrobat Reader, MS Word, …

a URL can point directly to a PDF or Word file and upon clicking by the user, Acrobat or Word is automatically started

Computer Networks
WWW : Cookies

Web was originally designed as a *stateless* entity,
i.e. server does not keep track of clients - (no log-in)
client requests page, server returns page, relationship is forgotten

however, in the current situation, the Web has many other functions → →

Computer Networks

WWW : Cookies

Functional requirements of today's Web :

(i) access to registered clients only

(ii) websites used as electronic stores – users can browse, select required items, save them in cart

(iii) websites used as portals

(iv) advertising

to facilitate the above, → →

Computer Networks
WWW : Cookies

server needs to know users' details

identifying / *remembering* user by IP address is not feasible due to :

- **sharing of computers by users**
- **use of NAT by ISPs**

so, cookies are created by the servers and sent to clients

Computer Networks

WWW : Cookies (RFC 2109)

a cookie is a message, no more than a 4 KB file / string

the main purpose of cookies is to identify / keep track of users and possibly prepare customized web pages for them & send the same to users during subsequent visits

contents of the cookie :

information the server has gathered about the client such as ……

name, registration number, etc.

Computer Networks

WWW : Cookies

Creation and storage of Cookies

implementation - dependant

basic principles :

(a) when a server receives a request from client, it stores client information in a file or string

information stored includes :

- **domain name of client**
- **contents of the cookie**
- **time stamp**
- **implementation-dependent information**

Computer Networks
WWW : Cookies

Creation and storage of Cookies (contd…)

(b) the server includes the cookie in the response that it sends to the client

(c) when the client receives the response, the browser stores the cookie in the cookie directory, which is stored by the domain server name

Computer Networks
WWW : Cookies

How are Cookies used ?

- client sends a request to server
- browser checks in the cookie directory for a cookie, if sent by the server earlier
- if found, the cookie is included in the request
- when server receives the request, it concludes that the client is an old one

the contents of the cookie are never read by the browser or disclosed to the user

a cookie is read only by the server which created it

Computer Networks

WWW : Cookies

How are Cookies used ?

(i) the site, that restricts access to registered clients only, ….

sends a cookie to the client when the client registers for the first time

for any repeated access ….

only those clients that send the appropriate cookie are allowed

Computer Networks

WWW : Cookies

How are Cookies used ?

(ii) an e-commerce site can use cookie for its client shoppers

when a client selects an item and adds it to the cart, ……

a cookie that contains info about the item (part no., description, unit price) is sent to the browser

Computer Networks
WWW : Cookies

How are Cookies used ?

(ii) contd…..

this is repeated for every item selected and the cookie is updated with the new selection information

when the client finishes shopping and wants to check out …..

the last cookie is retrieved and the total payment due is calculated for further processing

Computer Networks

WWW : Cookies

How are Cookies used ?

(iii) a web portal example :

when a user selects his / her favourite pages …..

a cookie is made and sent

during subsequent accesses, ….

the cookie is sent to the server to show what the client is looking for

Computer Networks

WWW : Cookies

Issue **: concern for Internet privacy, since they can be used for tracking browsing behavior**

modern browsers allow users to decide whether to accept cookies (stored on the client's hard disk in a cookie directory); but rejection makes some websites unusable

e.g. : shopping baskets implemented using cookies do not work if cookies are rejected

Computer Networks
WWW : Cookies

four components are involved :

- a cookie header line in the HTTP response message
- a cookie header line in the HTTP request message
- cookie file kept on the user's end system and managed by the user's browser
- back- end database at the website

Computer Networks

WWW : Cookies

Steps involved :

1. User sends a request to web server that uses cookies

2. Web site creates a unique identification number and creates, in its back-end data base, an entry indexed by unique *id no.*

3. Server responds to user's browser, including in the HTTP response a Set-cookie: header, which contains the unique *id no.* ; e.g. Set-cookie: 1234567

Computer Networks

WWW : Cookies

Steps involved :

4. When user's browser receives this message & sees Set-cookie: header, it appends a line to the special cookie file that the browser manages ; line includes the host name of server and the *id no.*

5. Each time the user requests for a web page, the browser consults the cookie file, extracts the *id no.* and puts (in the request) a cookie header line that includes the *id no.*

Computer Networks

WWW : Cookies

website is able to know which page the user with the id no. has visited and the time of visit

cookies can be used to create a user session layer on top of stateless HTTP

cookies simplify the Internet shopping experience

but, they remain highly controversial as they can also be viewed as an infringement on the user's privacy

Computer Networks

WWW : Cookies

:\chandrasekhars\Local Settings\Temporary Internet Files

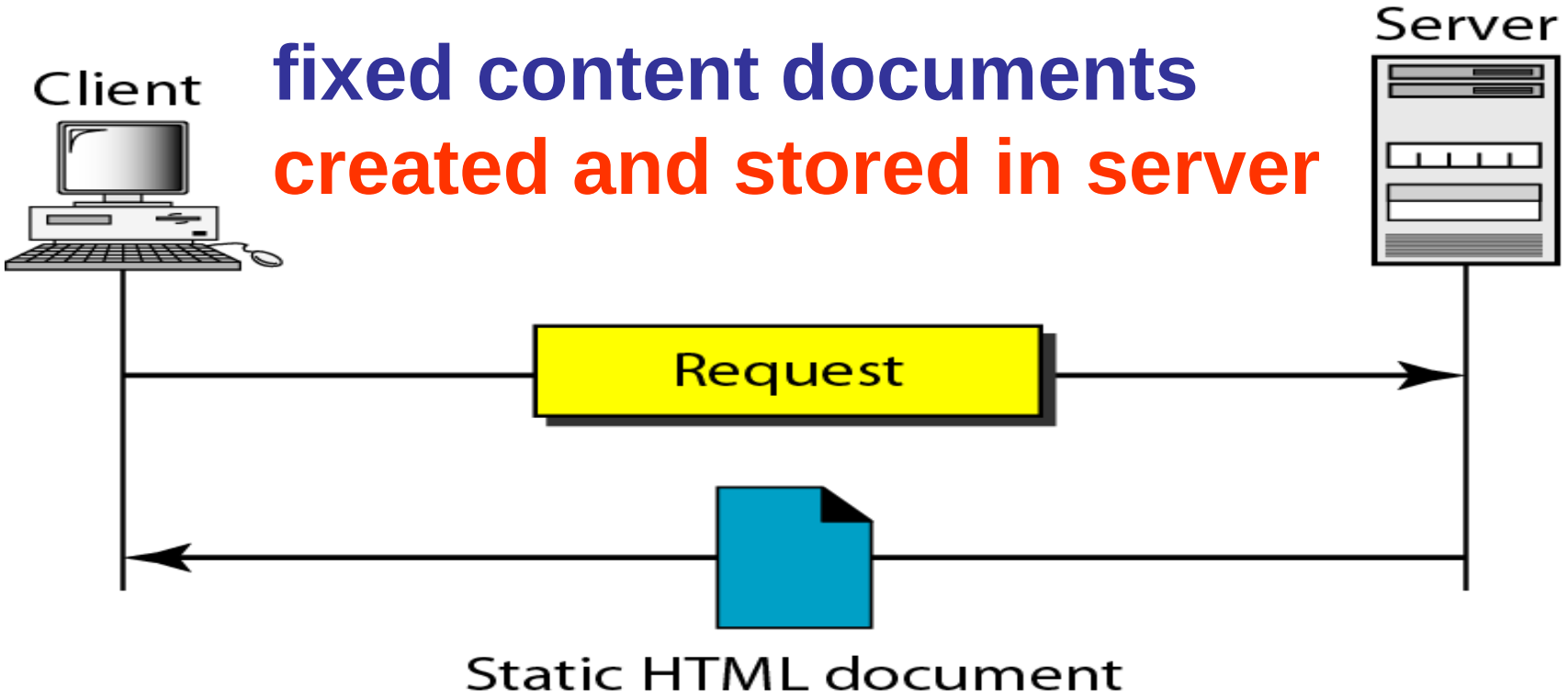
Go

Name	Internet Address	Type	Size	Expires	Last Modified	Last Access
cookie:chandr...	Cookie:chandrasekhars@www.insuranceday.com/	Text Document	1KB	4/2/2015 4:24 PM	4/2/2013 4:24 PM	4/2/2013 4:
cookie:chandr...	Cookie:chandrasekhars@www.jobsholl.com/	Text Document	2KB	7/24/2015 9:28 AM	7/24/2013 9:28 AM	7/24/2013 9
cookie:chandr...	Cookie:chandrasekhars@www.kent.edu/	Text Document	1KB	2/4/2044 11:10 AM	2/11/2014 11:09 ...	2/11/2014 1
cookie:chandr...	Cookie:chandrasekhars@www.linkedin.com/	Text Document	1KB	6/4/2015 9:38 PM	3/26/2014 3:57 PM	3/26/2014 3
cookie:chandr...	Cookie:chandrasekhars@www.madanapalas.com/	Text Document	1KB	2/25/2016 10:21 ...	2/25/2014 10:21 ...	2/25/2014 1
cookie:chandr...	Cookie:chandrasekhars@www.makemytrip.com/	Text Document	1KB	None	4/1/2013 2:29 PM	4/1/2013 2:
cookie:chandr...	Cookie:chandrasekhars@www.manipal.edu/	Text Document	1KB	11/12/2015 4:08 ...	2/15/2013 4:08 PM	2/15/2013 4
cookie:chandr...	Cookie:chandrasekhars@www.mathgoodies.com/	Text Document	1KB	11/7/2015 9:36 AM	11/7/2013 9:36 AM	11/7/2013 9
cookie:chandr...	Cookie:chandrasekhars@www.merriam-webster.com/	Text Document	1KB	12/14/2013 1:24 ...	12/14/2013 1:13 ...	12/14/2013
cookie:chandr...	Cookie:chandrasekhars@www.microsoft.com/	Text Document	1KB	11/21/2023 10:4...	11/23/2013 10:4...	11/23/2013
cookie:chandr...	Cookie:chandrasekhars@www.moneycontrol.com/	Text Document	1KB	7/29/2015 9:23 AM	2/22/2014 12:22 ...	2/22/2014 1
cookie:chandr...	Cookie:chandrasekhars@www.ndtv.com/	Text Document	1KB	3/30/2013 3:26 PM	2/28/2013 3:26 PM	2/28/2013 3
cookie:chandr...	Cookie:chandrasekhars@www.nethradhama.org/	Text Document	1KB	7/4/2015 12:19 PM	7/4/2013 12:19 PM	7/4/2013 12
cookie:chandr...	Cookie:chandrasekhars@www.nokia.com/	Text Document	1KB	8/29/2013 4:48 PM	8/29/2013 4:18 PM	8/29/2013 4
cookie:chandr...	Cookie:chandrasekhars@www.online.citibank.co.in/	Text Document	1KB	1/1/2020 5:30 AM	3/14/2014 3:04 PM	3/14/2014 4
cookie:chandr...	Cookie:chandrasekhars@www.online-convert.com/	Text Document	1KB	4/8/2013 9:04 AM	3/9/2013 9:04 AM	3/9/2013 9:
cookie:chandr...	Cookie:chandrasekhars@www.purplemath.com/	Text Document	1KB	10/7/2014 12:13 ...	10/7/2013 12:12 ...	10/7/2013 3
cookie:chandr...	Cookie:chandrasekhars@www.qsstats.com/	Text Document	1KB	1/18/2024 10:24 ...	1/20/2014 10:24 ...	1/20/2014 1
cookie:chandr...	Cookie:chandrasekhars@www.random.org/	Text Document	1KB	11/7/2015 9:35 AM	11/7/2013 11:25 ...	11/7/2013 1
cookie:chandr...	Cookie:chandrasekhars@www.rcuk.ac.uk/	Text Document	1KB	2/6/2014 10:24 AM	2/6/2013 10:24 AM	2/6/2013 10
cookie:chandr...	Cookie:chandrasekhars@www.sankaraeye.com/	Text Document	1KB	7/4/2014 12:03 PM	7/4/2013 12:03 PM	7/4/2013 1:
cookie:chandr...	Cookie:chandrasekhars@www.sbi.co.in/	Text Document	1KB	9/26/2014 8:09 AM	8/5/2013 4:09 PM	8/5/2013 4:
cookie:chandr...	Cookie:chandrasekhars@www.schools9.info/	Text Document	2KB	7/30/2014 1:16 PM	7/30/2013 1:16 PM	7/30/2013 1
cookie:chandr...	Cookie:chandrasekhars@www.scrumalliance.org/	Text Document	1KB	12/27/2063 10:0...	12/27/2013 10:0...	12/27/2013
cookie:chandr...	Cookie:chandrasekhars@www.sony.co.in/	Text Document	1KB	8/29/2014 3:08 PM	8/29/2013 3:08 PM	8/29/2013 3
cookie:chandr...	Cookie:chandrasekhars@www.speedguide.net/	Text Document	1KB	9/2/2018 2:01 PM	9/3/2013 2:00 PM	9/3/2013 2:
cookie:chandr...	Cookie:chandrasekhars@www.statebankofmysore...	Text Document	1KB	3/2/2015 9:37 AM	3/12/2014 9:38 AM	3/12/2014 1
cookie:chandr...	Cookie:chandrasekhars@www.subnet-calculator.c...	Text Document	1KB	3/3/2013 11:16 PM	3/1/2013 11:12 AM	3/1/2013 11

Computer Networks

World Wide Web

Web documents : static

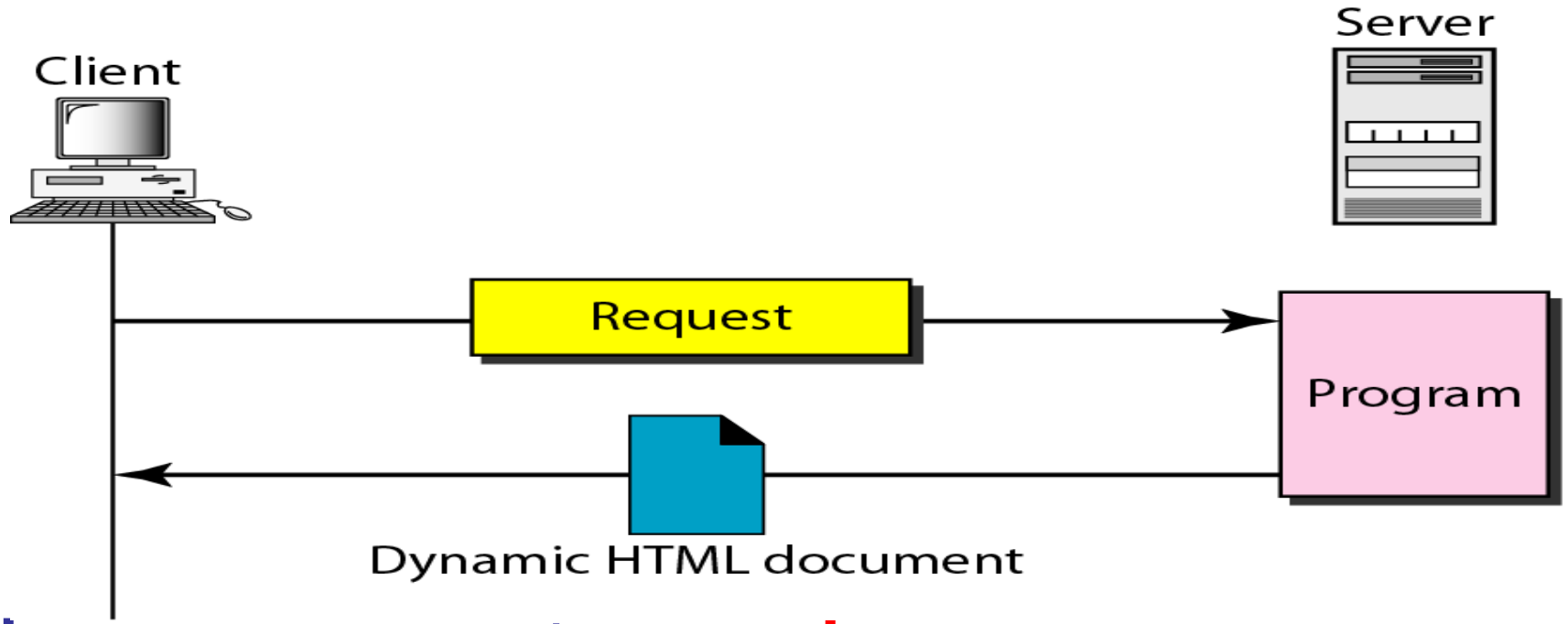


fixed content documents

created and stored in server

user can get only a copy of the document, cannot change the contents, can only display

Computer Networks
World Wide Web
Web documents : dynamic



browser requests → web server runs an application program or a script to create the dynamic document → returns the output

Computer Networks

World Wide Web

Web documents : dynamic

a fresh document is created for each request

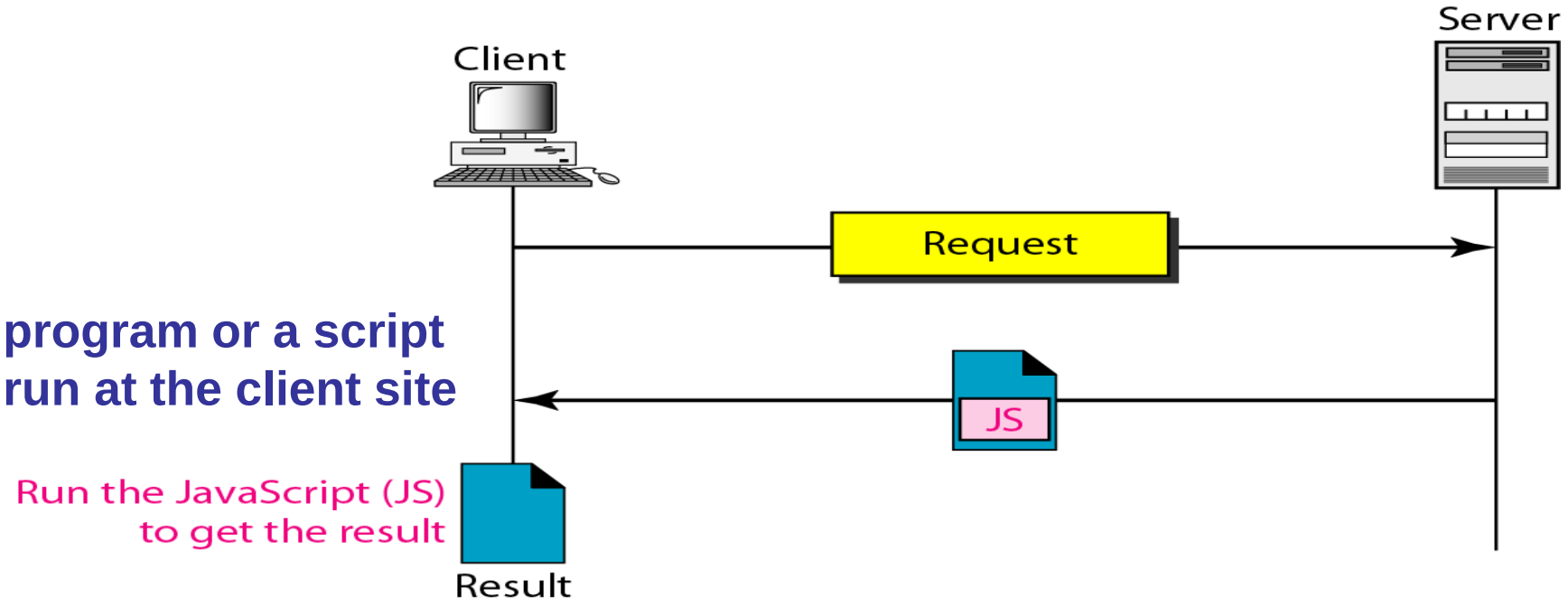
so, the contents of a dynamic document can vary from one request to another

e.g. retrieval of the time and date from the server

client can ask the server to run a program, say, the date program in Unix and send the result to the client

Computer Networks
World Wide Web

Web documents : active



browser requests → web server sends a copy of the document or script → script runs on the client

Computer Networks

World Wide Web

*H*yper*T*ext *M*arkup *L*anguage (HTML)

- allows users to produce web pages that includes text, graphics and pointers to other web pages
- is a markup language used for describing how documents are to be formatted
- contains explicit commands for formatting

Computer Networks
World Wide Web
*H*yper*T*ext *M*arkup *L*anguage (HTML)

background of the word *markup* :
used in the publishing industry - by copy editors on the manuscripts - put marks - to instruct type setters / printers how to format text

advantage of markup language :
straight forward, browser simply has to understand the markup commands
the markup commands are embedded within the HTML file → easy for browser

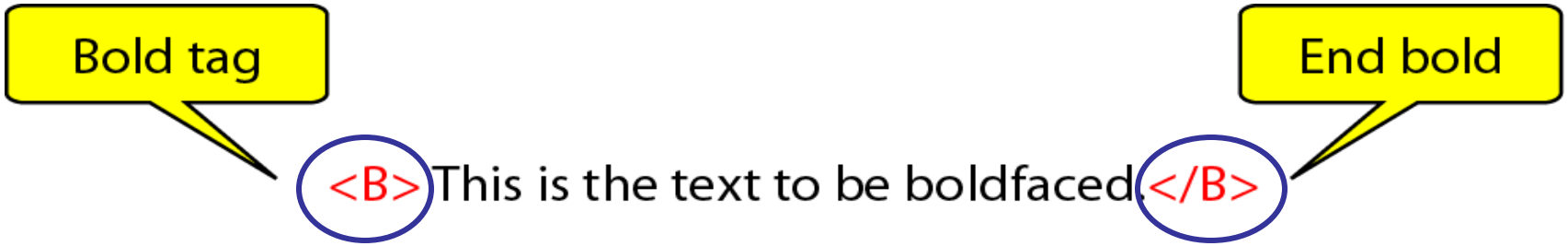
Computer Networks
World Wide Web

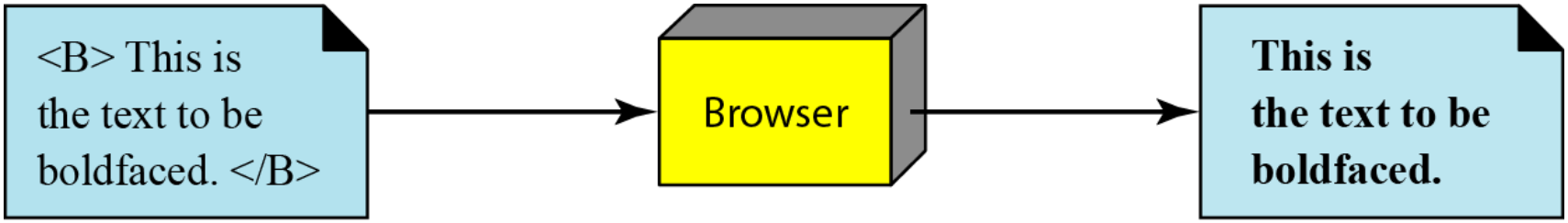
HTML

is a language used for creating web pages

***tag* is a formatting command ; example**

(instructions for the browser)





Computer Networks
World Wide Web

HTML

allows use of only ASCII characters for *both* the main text and for (embedding) formatting instructions in the file itself →

ensures compatibility across different types of software platforms

reformatting of page is possible →

a page produced on a higher resolution window can be received and rendered on a lower resolution window

Computer Networks
World Wide Web

HTML

web page consists of 2 parts : head, body:

head contains title of the page and other information that the browser will use

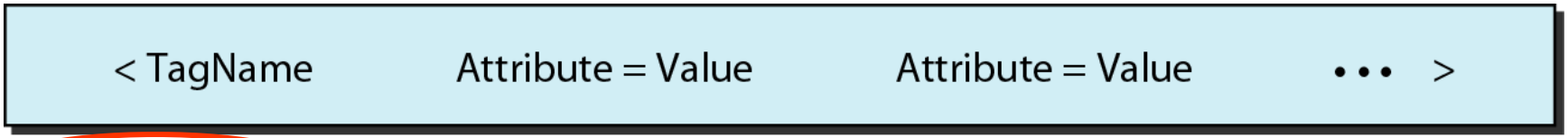
body contains the actual contents

Computer Networks
World Wide Web

HTML tags

every HTML *tag* (a mark that is embedded into the text) is a name followed by an optional list of attributes,

enclosed between symbols < and >



a. Beginning tag

b. Ending tag

Computer Networks
World Wide Web
HTML : common tags

Beginning Tag	Ending Tag	Meaning
Skeletal Tags		
<html>	</html>	Defines an HTML document
<head>	</head>	Defines the head of the document
<body>	</body>	Defines the body of the document
Title and Header Tags		
<title>	</title>	Defines the title of the document
<hn>	</hn>	Delimits a level n heading

Computer Networks
World Wide Web

HTML : common tags

Beginning Tag	Ending Tag	Meaning
Text Formatting Tags		
		Boldface
		Italic
<u>	</u>	Underlined
_		Subscript
[	]	Superscript
Data Flow Tag		
		Centered
 	</br>	Line break

Computer Networks
World Wide Web

HTML : common tags

Beginning Tag	Ending Tag	Meaning
List Tags		
		Ordered (numbered) list
		Unordered (bulletted) list
		An item in a list
Image Tag		
<img>		Defines an image
Hyperlink Tag (anchor)		
<a>		Defines an address (hyperlink)
Executable Contents		
<applet>	</applet>	The document is an applet

Computer Networks
World Wide Web

HTML : common tags

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
...	Set ... in boldface
...	Set ... in italics
...	Center ... on the page horizontally
...	Brackets an unordered (bulleted) list
...	Brackets a numbered list
	Starts a list item (there is no)
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a Horizontal rule
<img src="...">	Displays an image here
 ... 	Defines a hyperlink

Computer Networks
World Wide Web

HTML tags

image tag **is used to point to the file of a photo or image ; it defines :**

source and alignment

<img src"/bin/images/image1.gif" align=middle>

anchor **is a mechanism used to create hyperlinks ; defined by tags :**

**<a ….. > and **

**Author **

the word *Author* appears in the text ; the user can click on the same to go to the web page

HTML tags : Example - 1

This example shows how tags are used to let the browser format the appearance of the text.

```
<HTML>
    <HEAD>
        <TITLE> First Sample Document </TITLE>
    </HEAD>
    <BODY>
        <CENTER>
        <H1><B>  ATTENTION </B></H1>
        </CENTER>
        You can get a copy of this document by:
        <UL>
        <LI> Writing to the publisher
        <LI> Ordering online
        <LI> Ordering through a bookstore
        </UL>
    </BODY>
</HTML>
```

HTML tags : Example - 2

This example shows how tags are used to import an image and insert it into the text.

```
<HTML>
    <HEAD>
        <TITLE> Second Sample Document </TITLE>
    </HEAD>
    <BODY>
        This is the picture of a book:
        <IMG SRC="Pictures/book1.gif"   ALIGN=MIDDLE>
    </BODY>
</HTML>
```

Computer Networks
World Wide Web

HTML Table (from ver 3.0)

consists of one or more rows

each row consists of one or more *cells*

cells can contain :

text, figures, icons, photographs, other tables

tables are started by <table> tag

other tags :

<caption>, <tr>, <th>, <td>

Computer Networks
World Wide Web
HTML Table : example

Some Differences between HTML Versions

Item	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0
Hyperlinks	X	X	X	X
Images	X	X	X	X
Lists	X	X	X	X
Active Maps and Images		X	X	X
Forms		X	X	X
Equations			X	X
Toolbars			X	X
Tables			X	X
Accessibility features				X
Object embedding				X
Scripting				X

Computer Networks
World Wide Web

HTML Table : example

```
<html>
<head> <title> A sample page with a table </title> </head>
<body>
<table border=1 rules=all>
<caption> Some Differences between HTML Versions </caption>
<col align=left>
<col align=center>
<col align=center>
<col align=center>
<col align=center>
<tr> <th>Item  <th>HTML 1.0  <th>HTML 2.0  <th>HTML 3.0 <th>HTML 4.0 </tr>
<tr> <th> Hyperlinks <td> x <td> x <td> x <td> x </tr>
<tr> <th> Images <td> x <td> x <td> x <td> x </tr>
<tr> <th> Lists <td> x <td> x <td> x <td> x </tr>
<tr> <th> Active Maps and Images <td>   <td> x <td> x <td> x </tr>
<tr> <th> Forms <td>   <td> x <td> x <td> x </tr>
<tr> <th> Equations <td>   <td>   <td> x <td> x </tr>
<tr> <th> Toolbars <td>   <td>   <td> x <td> x </tr>
<tr> <th> Tables <td>   <td>   <td> x <td> x </tr>
<tr> <th> Accessibility features <td>   <td>   <td>   <td> x </tr>
<tr> <th> Object embedding <td>   <td>   <td>   <td> x </tr>
<tr> <th> Scripting <td>   <td>   <td>   <td> x </tr>
</table>
</body>
</html>
```

Computer Networks
World Wide Web

HTML Forms

need to support two-way traffic viz. companies want customers to fill out details for carrying out transactions

implemented from HTML version 2.0↑

contain boxes or buttons that allow users to fill info or make choices and then send the info back to the page's owner

<input> tag is used for the purpose

variety of parameters are used for defining size, nature of box displayed,….

Computer Networks
World Wide Web

HTML Forms

Widget Order Form

46 characters wide; expects user to type in a string; stored in variable "customer"

Name

Street address

City State Country

type = radio

Credit card # Expires M/C Visa

Widget size Big Little Ship by express courier

type = radio type = checkbox

Submit order

Thank you for ordering an AWI widget, the best widget money can buy!

Computer Networks
World Wide Web

HTML Forms

```
<html>
<head> <title> AWI CUSTOMER ORDERING FORM </title> </head>
<body>
<h1> Widget Order Form </h1>
<form ACTION="http://widget.com/cgi-bin/widgetorder" method=POST>
<p> Name <input name="customer" size=46> </p>
<p> Street Address <input name="address" size=40> </p>
<p> City <input name="city" size=20> State <input name="state" size =4>
Country <input name="country" size=10> </p>
<p> Credit card # <input name="cardno" size=10>
Expires <input name="expires" size=4>
M/C <input name="cc" type=radio value="mastercard">
VISA <input name="cc" type=radio value="visacard"> </p>
<p> Widget size Big <input name="product" type=radio value="expensive">
Little <input name="product" type=radio value="cheap">
Ship by express courier <input name="express" type=checkbox> </p>
<p><input type=submit value="submit order"> </p>
Thank you for ordering an AWI widget, the best widget money can buy!
</form>
</body>
</html>
```

(a)

Computer Networks

World Wide Web

Dynamic Web Documents

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 and …

returns the output of the program or script as a response to the browser that requested the document

e.g. retrieval of time and date from a server

Computer Networks
World Wide Web

Common Gateway Interface (CGI)

is a technology that creates and handles dynamic web documents

allows programmers to use any of several languages viz. C, C++, C Shell, Perl, Tcl

defines a set of rules the programmer must follow

is a set of standards that defines how :

- **a dynamic document is written**
- **data are input to the program**
- **the output result is used**

Computer Networks

World Wide Web

Common Gateway Interface (CGI)

common **: indicates that the standard defines a set of rules common to any language or platform**

gateway **: means that a CGI program can be used to access other resources such as databases, graphical packages,…….**

interface **: means that there is a set of predefined terms, variables, calls,…. that can be used in any CGI program**

Computer Networks

World Wide Web

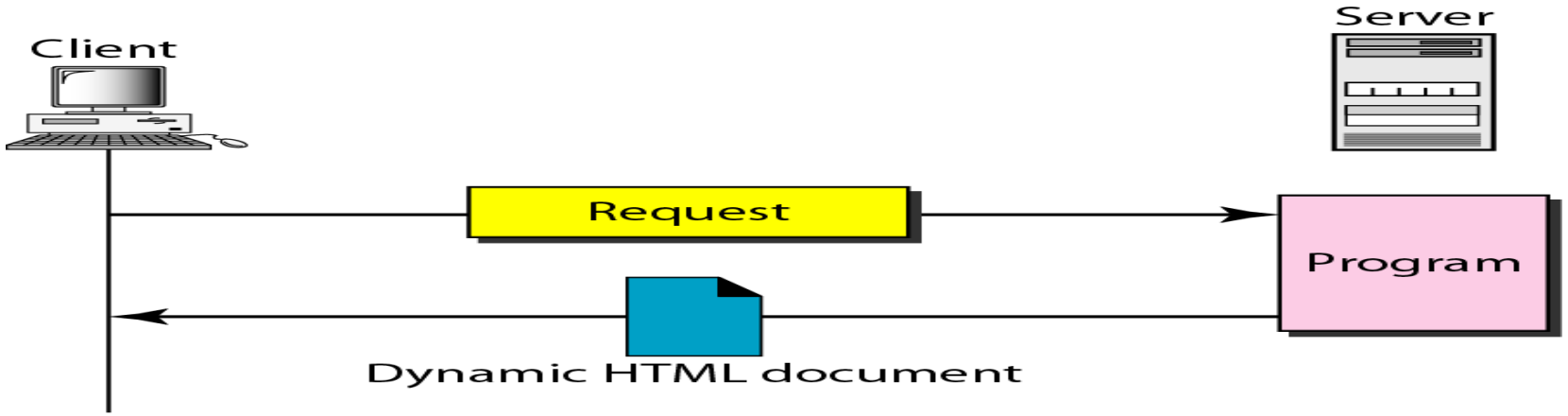
Common Gateway Interface (CGI)

Input

from a browser to a server is sent using a *form*

information in a form can be appended to the URL after a " ? " symbol ; example

http://www.server1.rvce.edu.in/cgi-bin/prog1.pl?53



Computer Networks
World Wide Web

Common Gateway Interface (CGI)

Output

execute a program at the server site and send the output to the client (browser)

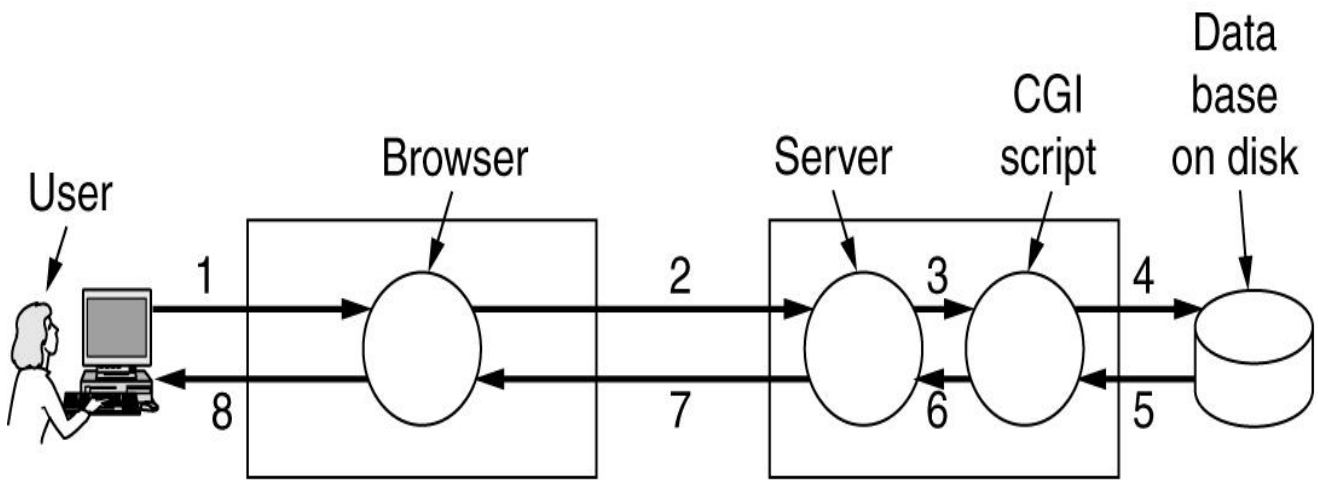
output can be :

text, text with HTML structures, graphics data, status code, instructions to the browser, ……

output of CGI program :

header (used to interpret the body), blank line, body

Computer Networks
World Wide Web
Server side dynamic web page generation



1. User fills in form
2. Form sent back
3. Handed to CGI
4. CGI queries DB
5. Record found
6. CGI builds page
7. Page returned
8. Page displayed

2. when user submits form, contents of the form is sent to the server

4. user-supplied information is used to process

6. custom HTML page is built

Computer Networks

World Wide Web

Scripting technologies for dynamic documents

if part of the dynamic document has fixed content and the other part has varying content → →

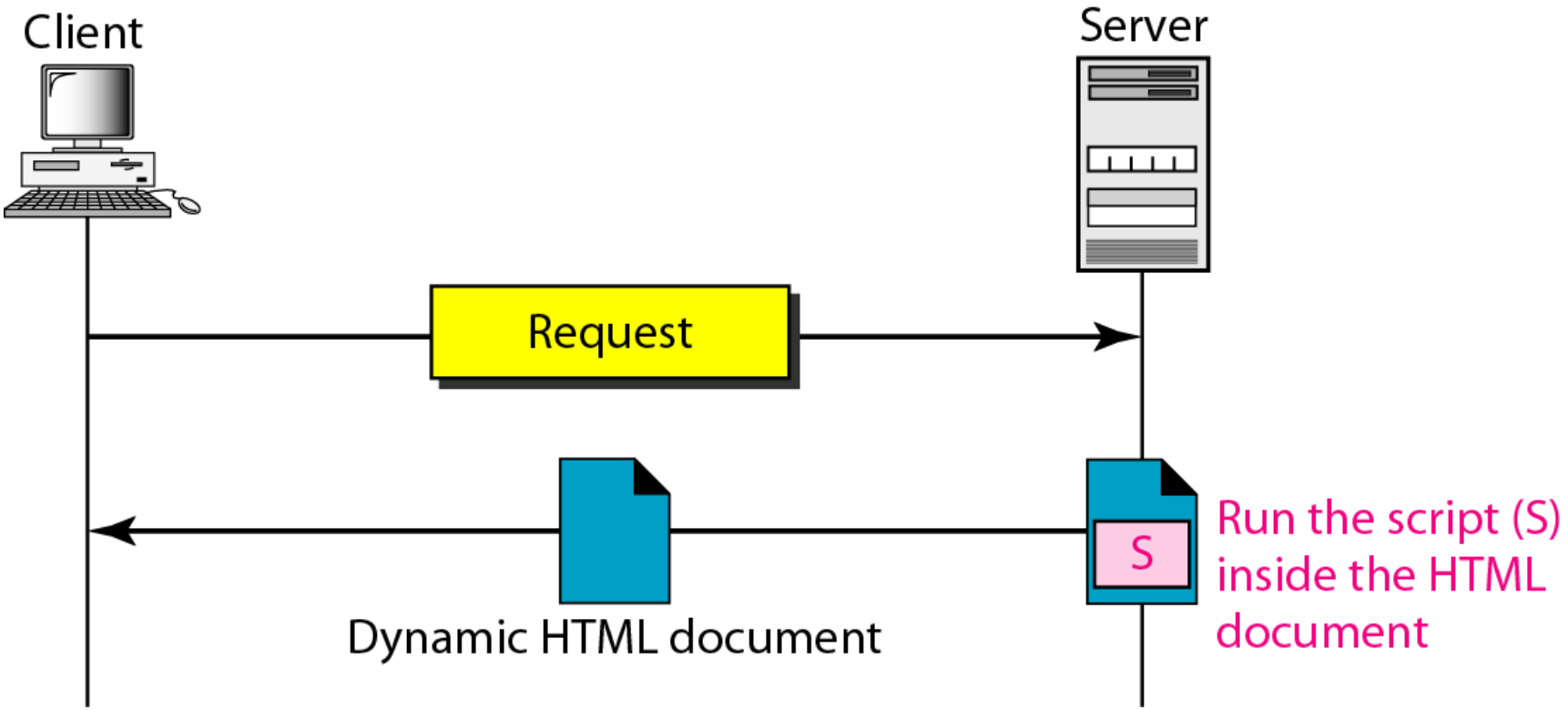
creating an entire document every time a request is made is inefficient → →

- **create a file containing the fixed part of the document using HTML**
- **embed a script (source code) that can be run by server to handle varying content**

Computer Networks

World Wide Web

Scripting technologies for dynamic documents



Computer Networks

World Wide Web

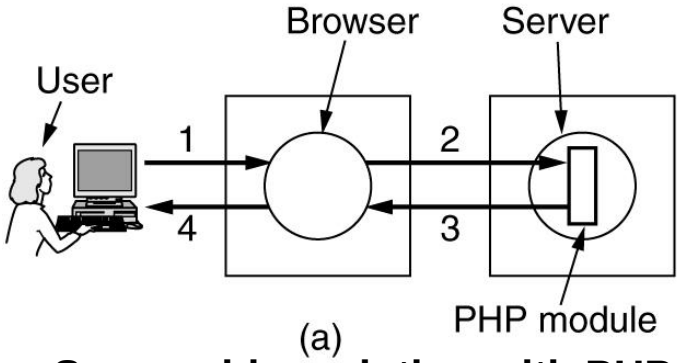
Scripting technologies for creating server side dynamic documents using scripts :

1. **Hypertext Preprocessor (PHP) - uses Perl**
2. **Java Server Pages (JSP) - uses Java**
3. **Active Server Pages (ASP) - a Microsoft product, uses VB**
4. **ColdFusion, an Adobe product, embeds SQL database queries in HTML documents**

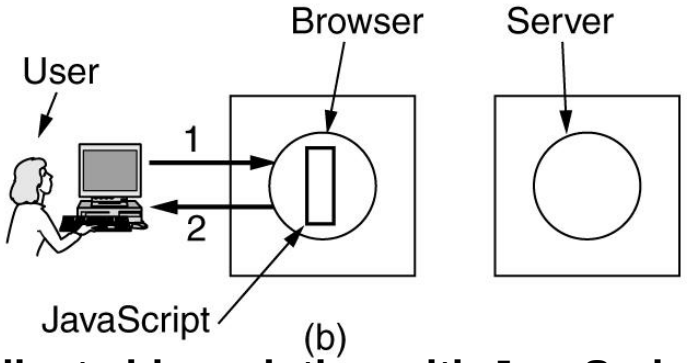
Computer Networks

World Wide Web

Server and Client side scripting



Server side scripting with PHP



Client side scripting with JavaScript

used when interaction with a remote database is needed

used when interaction is with the user at the client computer

Computer Networks

World Wide Web

HyperText Transfer Protocol (HTTP)

protocol used to access data on the web

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

uses one TCP connection

HTTP functions as a combination of FTP and SMTP

similar to FTP :

because it transfers files and uses the services of TCP

Computer Networks

World Wide Web

HyperText Transfer Protocol (HTTP)

similar to SMTP :

because data transferred between the client and server look like SMTP messages

the format of the messages is controlled by MIME-like headers

SMTP messages are read by e-mail users

HTTP messages are read and interpreted by the server and client (browser)

SMTP messages are stored and forwarded

HTTP messages are delivered immediately

Computer Networks

World Wide Web

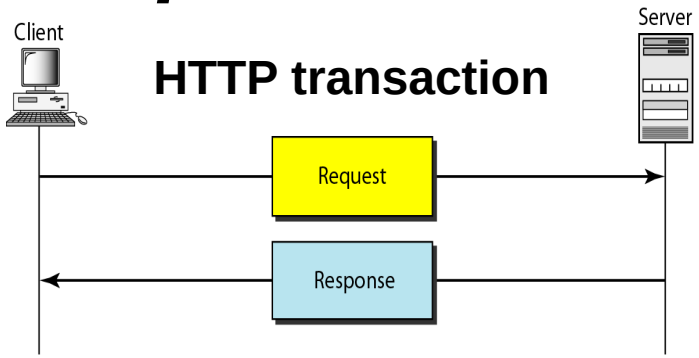
HyperText Transfer Protocol (HTTP)

commands from client to server :

embedded in ***request*** message

commands from server to client:

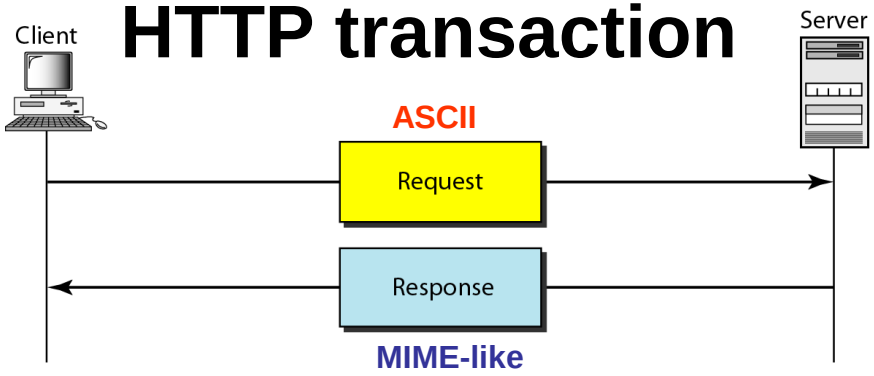
embedded in ***response*** message



response message includes the contents of the transferred file and other information

Computer Networks
World Wide Web
HyperText Transfer Protocol (HTTP) - RFC2616

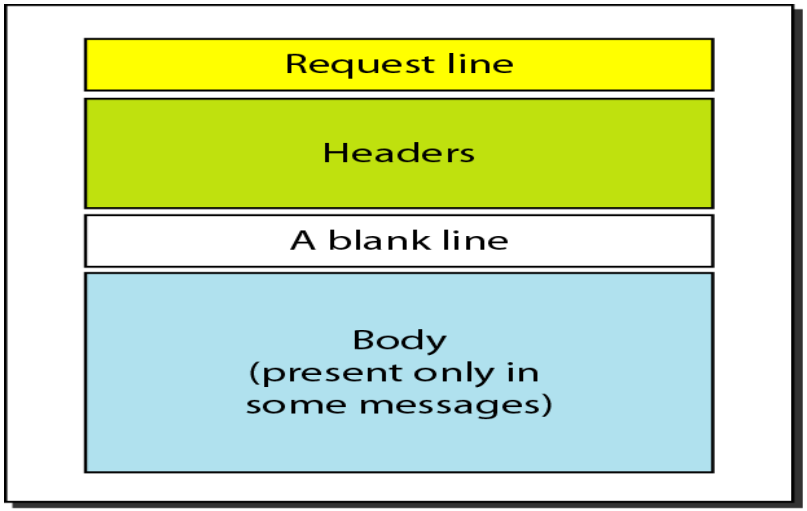
HTTP transaction



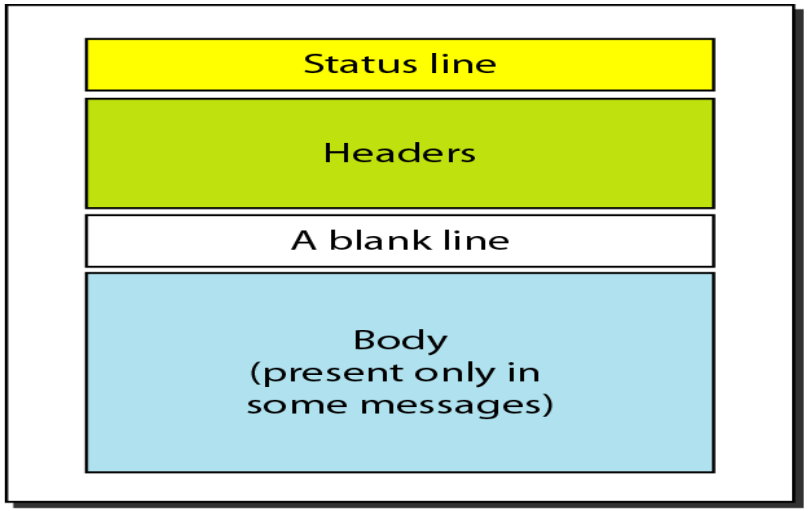
HTTP is a stateless protocol

client sends request

server responds



Request message



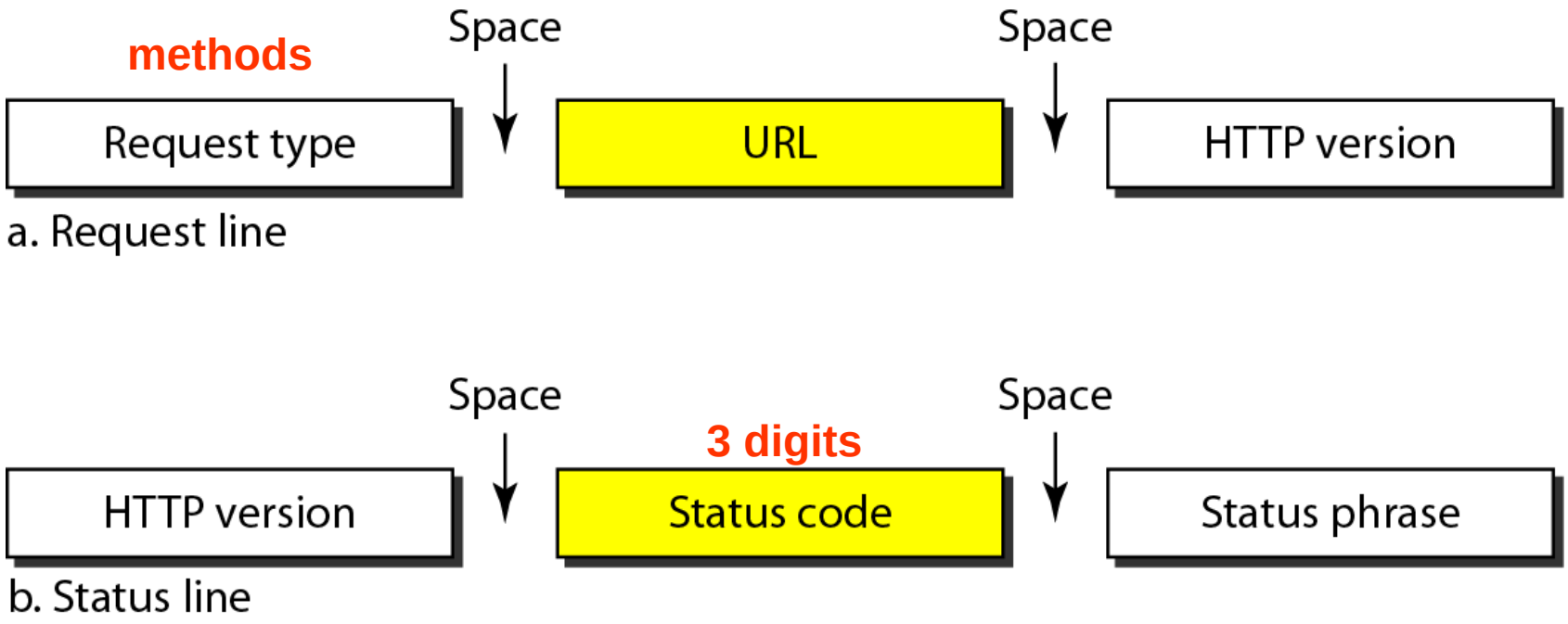
Response message

Computer Networks

World Wide Web

HyperText Transfer Protocol (HTTP)

HTTP request and status lines



Computer Networks

World Wide Web

HyperText Transfer Protocol (HTTP)

Request type : categorized into *methods :*

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

Computer Networks

World Wide Web

HyperText Transfer Protocol (HTTP)

Status code : response groups

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

Computer Networks
World Wide Web
HyperText Transfer Protocol (HTTP)
Status codes

Code	*Phrase*	*Description*
	Informational	
100	Continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
	Success	
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.

Computer Networks
World Wide Web
HyperText Transfer Protocol (HTTP)
Status codes

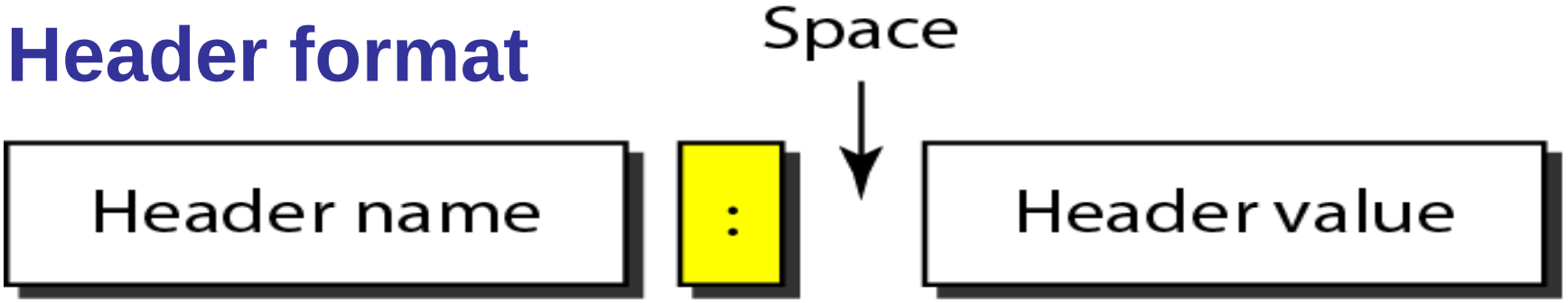
Code	*Phrase*	*Description*
	Redirection	
301	Moved permanently	The requested URL is no longer used by the server.
302	Moved temporarily	The requested URL has moved temporarily.
304	Not modified	The document has not been modified.
	Client Error	
400	Bad request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in this URL.
406	Not acceptable	The format requested is not acceptable.
	Server Error	
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future.

Computer Networks

World Wide Web

HyperText Transfer Protocol (HTTP)

Header format



the header exchanges additional information between client and server in one or more lines

Examples :

- **client can request document to be sent in a special format**
- **server can send extra info about document**

Computer Networks
World Wide Web
HyperText Transfer Protocol (HTTP)

the header can consist of one or more header lines

a header line belongs to one of four categories :

- **general headers**
- **request headers (only in request message)**
- **response headers (only in response message)**
- **entity headers**

Computer Networks
World Wide Web
HyperText Transfer Protocol (HTTP)

General headers give general info about the message

Header	*Description*
Cache-control	Specifies information about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

Computer Networks

World Wide Web

HyperText Transfer Protocol (HTTP)

Request headers

specify client's configuration and client's preferred document format

Header	*Description*
Accept	Shows the medium format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
From	Shows the e-mail address of the user
Host	Shows the host and port number of the server
If-modified-since	Sends the document if newer than specified date
If-match	Sends the document only if it matches given tag
If-non-match	Sends the document only if it does not match given tag
If-range	Sends only the portion of the document that is missing
If-unmodified-since	Sends the document if not changed since specified date
Referrer	Specifies the URL of the linked document
User-agent	Identifies the client program

Computer Networks

World Wide Web

HyperText Transfer Protocol (HTTP)

Response headers

specify server's configuration and special info about the request

Header	*Description*
Accept-range	Shows if server accepts the range requested by client
Age	Shows the age of the document
Public	Shows the supported list of methods
Retry-after	Specifies the date after which the server is available
Server	Shows the server name and version number

Computer Networks
World Wide Web
HyperText Transfer Protocol (HTTP)
Entity headers give info about the body of the document

Header	*Description*
Allow	Lists valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the medium type
Etag	Gives an entity tag
Expires	Gives the date and time when contents may change
Last-modified	Gives the date and time of the last change
Location	Specifies the location of the created or moved document

Computer Networks
World Wide Web
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

Computer Networks

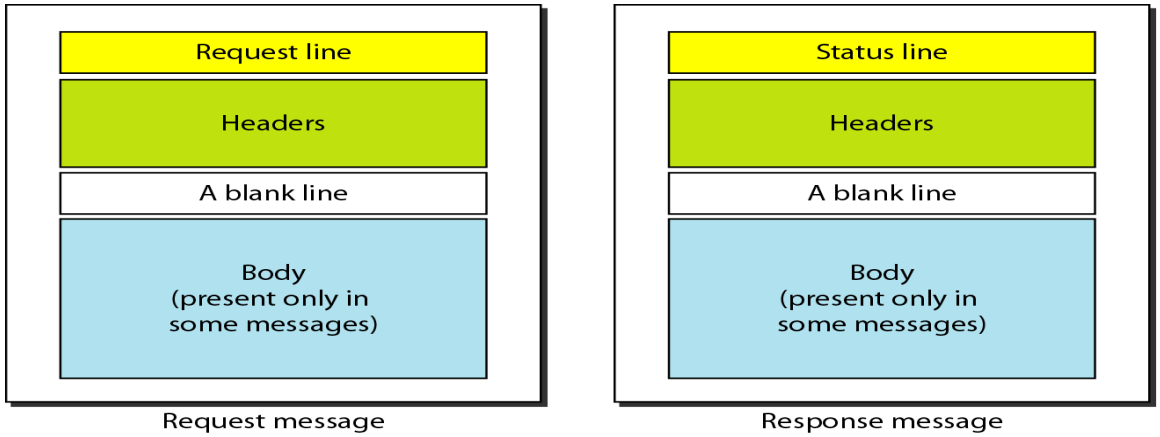
World Wide Web

HyperText Transfer Protocol (HTTP)

Body

can be present in a request or response message

usually contains the document to be sent or received



Computer Networks

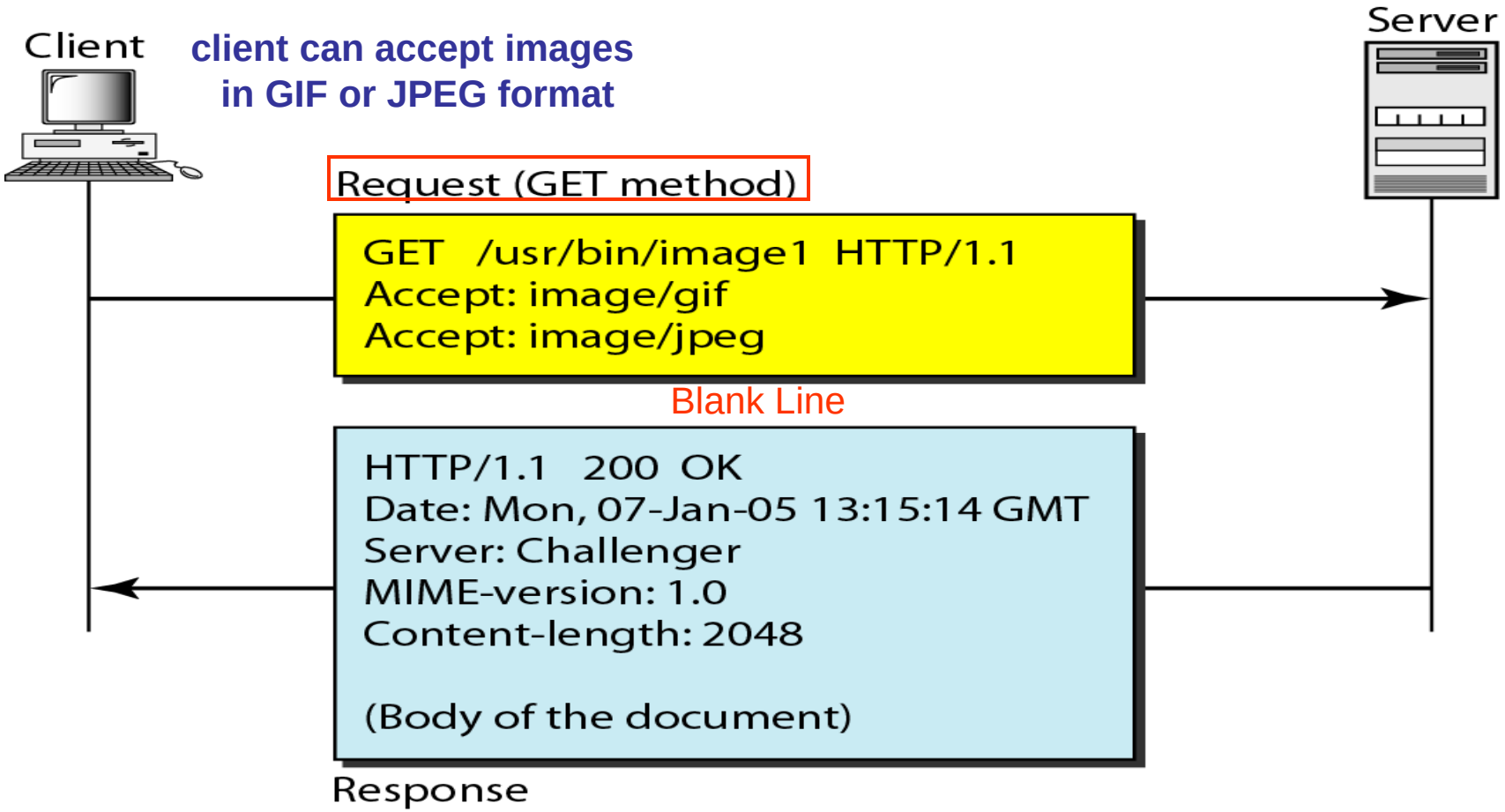
World Wide Web

HyperText Transfer Protocol (HTTP)

Request type : categorized into *methods :*

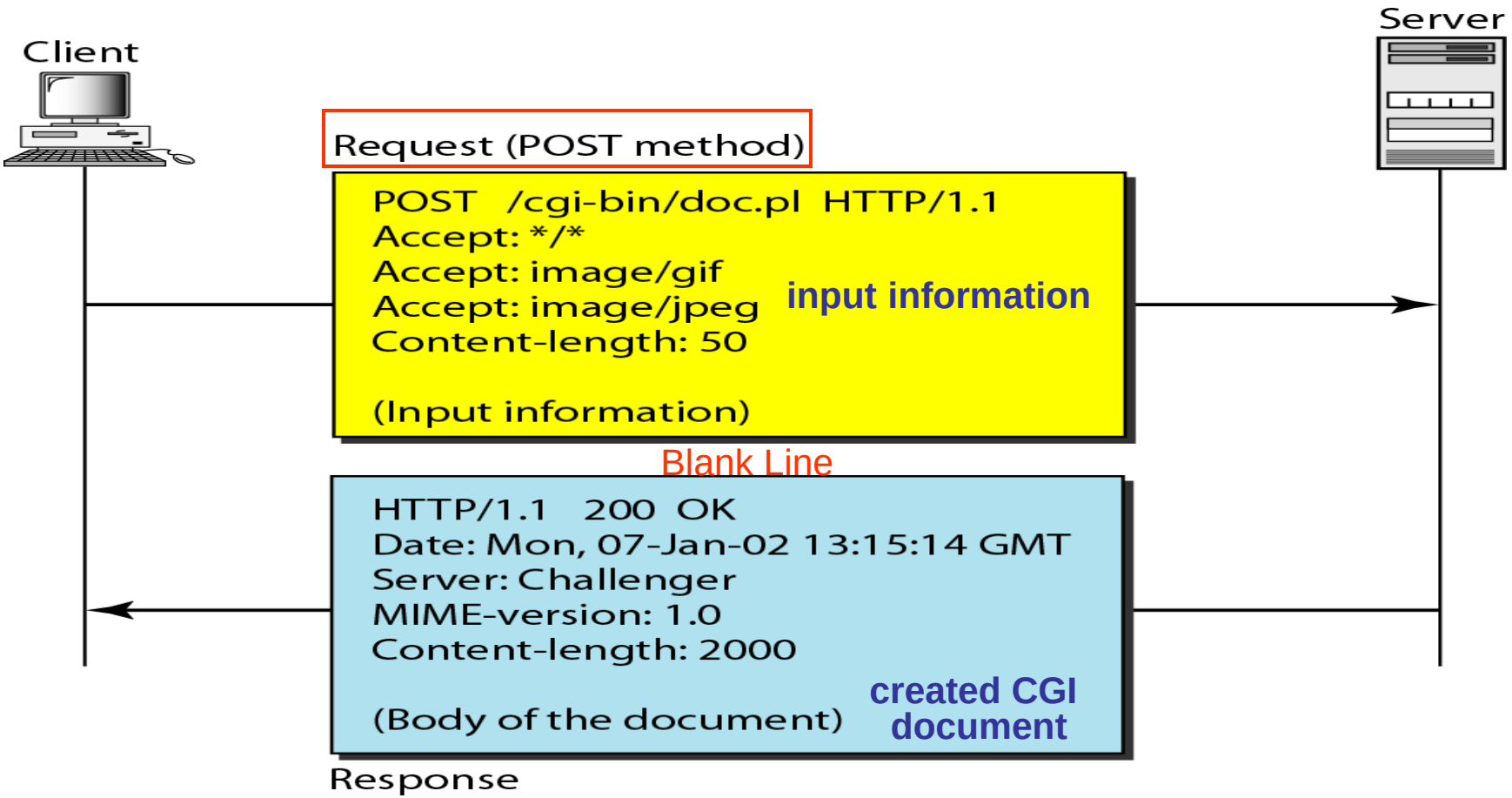
Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires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

Computer Networks
World Wide Web
HTTP : example of document retrieval



Computer Networks
World Wide Web

HTTP : example of client sending data to server



Computer Networks

World Wide Web : TCP connections

Non-Persistent connections (HTTP ver 1.0)

1. client opens connection, sends request
2. server sends response, closes connection
3. client reads data until end-of-file marker and then closes connection

Persistent connections (HTTP ver 1.1↑)

server leaves connection open to receive additional requests and sends responses till client requests to close connection or till time out period

Computer Networks

World Wide Web : TCP connections

Non-Persistent connections (HTTP ver 1.0)

1. client opens connection, sends request
2. server sends response, closes connection
3. client reads data until end-of-file marker and then closes connection

Persistent connections (HTTP ver 1.1↑)

server leaves connection open to receive additional requests and sends responses till client requests to close connection or till time out period

Computer Networks

World Wide Web

HyperText Transfer Protocol (HTTP)

Proxy server

HTTP supports proxy servers

proxy server is a system that keeps copies of response to recent requests

proxy server reduces the load on the real server, decreases traffic, and improves latency

Computer Networks

World Wide Web

Proxy server and Caching

saving of pages that have been requested, for reuse, if required

usually, a *process* called *proxy* maintains the cache

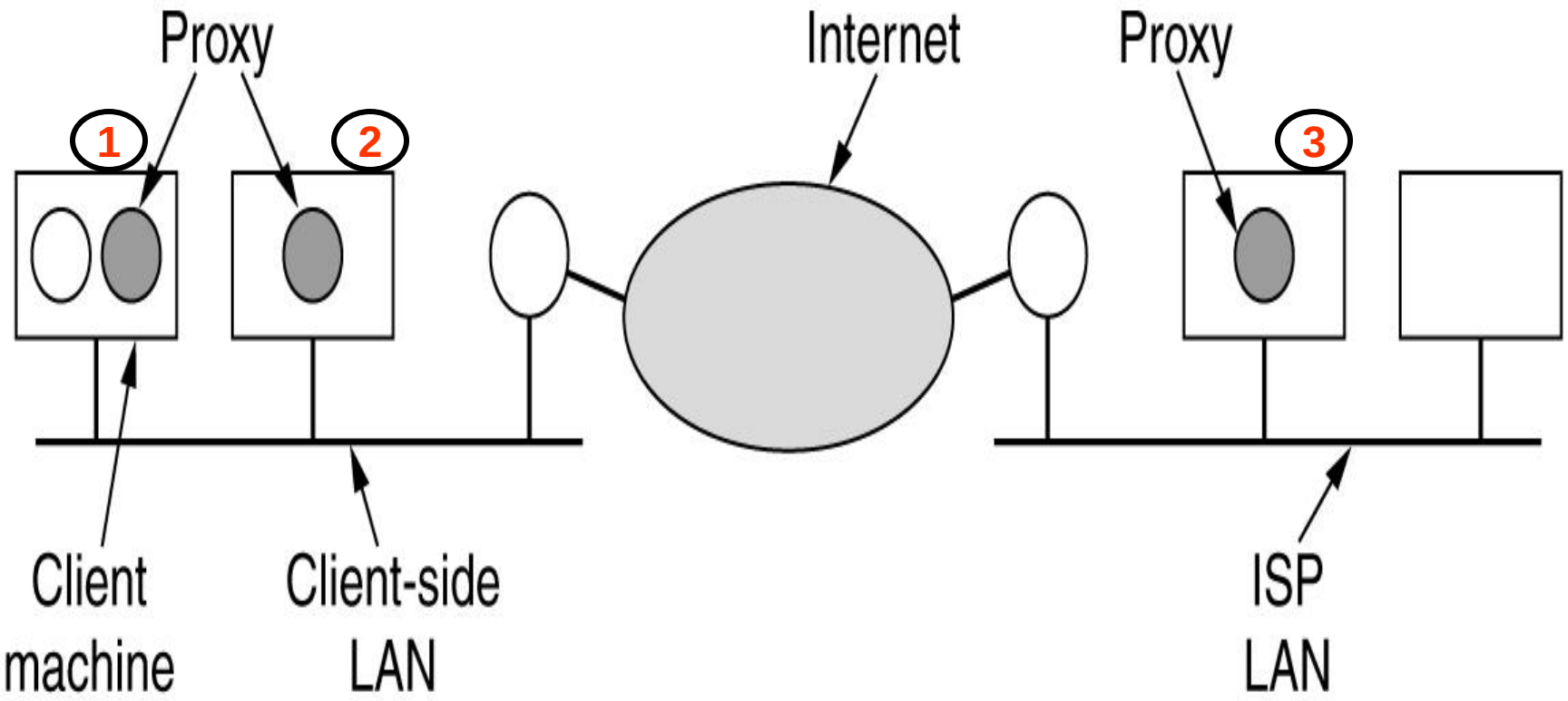
- **browser can be configured to make all page requests to a proxy instead of making request the real server of the page**
- **if the proxy has the page, it returns immediately ; else**
- **it fetches page from the real server, adds it to cache, returns it to requesting client**

Computer Networks

World Wide Web

Proxy server and Caching

hierarchical caching



Computer Networks

World Wide Web

Proxy server and Caching

who should cache ? **how long ? → →**

↓

any of the following systems running proxy:

1. individual PCs

2. a machine shared on the LAN

3. a machine at the ISP

these can be tried successively, in the above order

if step 3 fails, situation is resolved by contacting a higher-level cache or from the real server itself

Computer Networks

World Wide Web

Proxy server and Caching

how long ?

time duration to cache may vary wildly - example of share market data

cache duration too long → high hit rate, but may return stale pages

cache duration too short → low hit rate (not very effective), but may rarely return stale pages

Computer Networks

World Wide Web

Proxy server and Caching

how long ?

1.Use of *Last-modified* header :
set the holding time in cache equal to (*current-time* minus *last-modified time*)(#)

method often works well in practice, but does return stale pages from time to time

(#) an HTTP header

Computer Networks

World Wide Web

Proxy server and Caching

how long ?

2. Using special features of RFC 2616 :
request header " *If-modified-since* " is sent by proxy to the server
specifies the page the proxy wants and the time the cached page was last modified
if the page has not been modified since then, the server sends back status code 304 (*Not modified*) → proxy uses cached page

Computer Networks
World Wide Web
Proxy server and Caching

how long ?

2. Using special features of RFC 2616 :
if the page has been modified since then
→ the new page is returned

combination of methods 1 & 2, is also used

Computer Networks
World Wide Web
Proxy server and Caching

how long ?

3. Proactive caching

when proxy fetches a page from server, it looks for hyperlinks, if any → can issue requests to relevant servers and preload pages into cache

- **improves response time**
- **may flood lines with pages that may never be needed**

Computer Networks

World Wide Web

Proxy server and Caching

how long ?

4. Dynamic pages (eg. generated by PHP script)

- should never be cached

server instructs all proxies along the path back to the client not use the current page without verifying its freshness

(more cache control mechanisms in RFC 2616)

Computer Networks

World Wide Web

Proxy server and Caching

how long ?

4. Dynamic pages (e.g. generated by PHP script)

- should never be cached

server instructs all proxies along the path back to the client not use the current page without verifying its freshness

(more cache control mechanisms in RFC 2616)