Scheme Of Evaluation
Internal Assessment Test 1 – September.2019

Sub:	Computer Networks							Code:	17CS52
Date:	09 / 09 / 2019	Duration:	90mins	Max Marks:	50	Sem:	VI	Branch:	ISE

Note: Answer Any Five Questions

Question #		Description	Marks Distribution		Max Marks
1	a)	**Compare and contrast Go-Back-N and Selective Repeat protocol**. • Go-Back-N • Selective Repeat.	5M 5M	10M	10 M
2	a)	**Discuss how an electronic mail is transferred over internet.** • Using SMTP protocol • Diagram. • Explanation	1M 2M 2M	5M	10 M
	b)	**Label DNS message.** • Message Format	5M	5M	
3	a)	**Compare and contrast POP and IMAP protocols.** • POP • IMAP	2.5M 2.5M	5M	10 M
	b)	**Describe circular DHT.** • Introduction to DHT • Circular DHT	1M 4M	5M	
4	a)	**Describe FTP protocol with example commands & responses.** • Working of FTP protocol • Example	5M 1M	6M	10 M
	b)	**Compare persistent with non-persistent connection.** • Persistent connection • Non-persistent connection	2M 2M	4M	

5	a)	**Describe Rdt 3.0 protocol.** • Diagram • Explanation	6M 4M	10M	10 M
6	a)	**Bob is sending a 16-bit 3 words to Alice using UDP, Alice has to test whether bit error is present in the received data or not. Sender side data:** **Data Sent: 1000101010111100** **0101011110001110** **1111000011100011. Identify whether error is there or not.** • Finding Checksum • Concluding no error	3M 2 M	5M	10 M
	b)	**Quote HTTP message format.** • Request message • Response message	2.5M 2.5M	5M	
7	a)	**Summarize the working of DNS.** • Diagram • Steps	6M 4M	10M	10 M
8	a)	**Consider your college is providing 15Mbps internet connectivity for public network and 100 Mbps speed within local area network; whether you are able to browse the web using this low speed, if yes explain.** • Answer: Yes • Explanation	1M 9M	10M	10 M

Answers

1 (a) Compare and contrast Go-Back-N and Selective Repeat protocol.

Pipelined protocols: overview

Go-back-N:

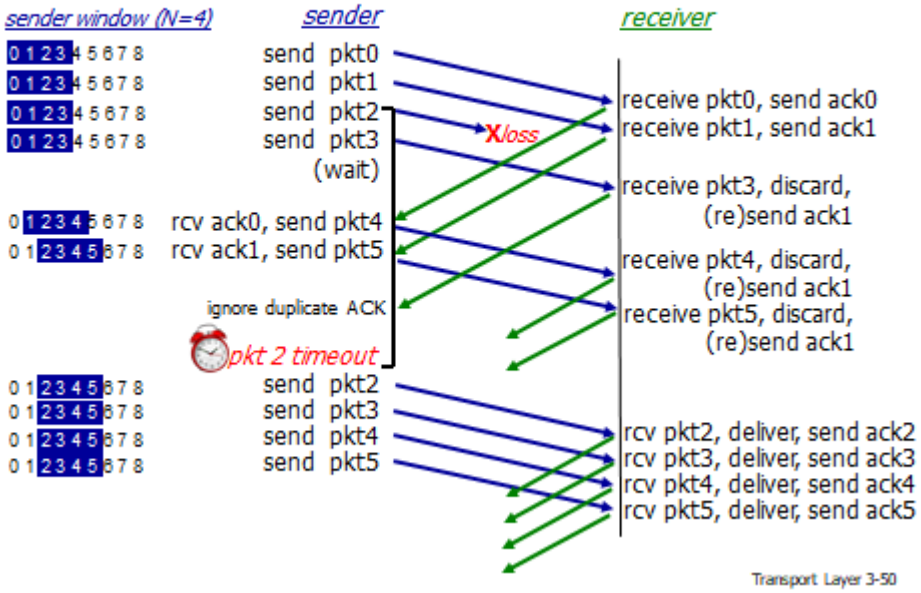
- sender can have up to N unacked packets in pipeline
- receiver only sends *cumulative ack*
  - doesn't ack packet if there's a gap
- sender has timer for oldest unacked packet
  - when timer expires, retransmit *all* unacked packets

Selective Repeat:

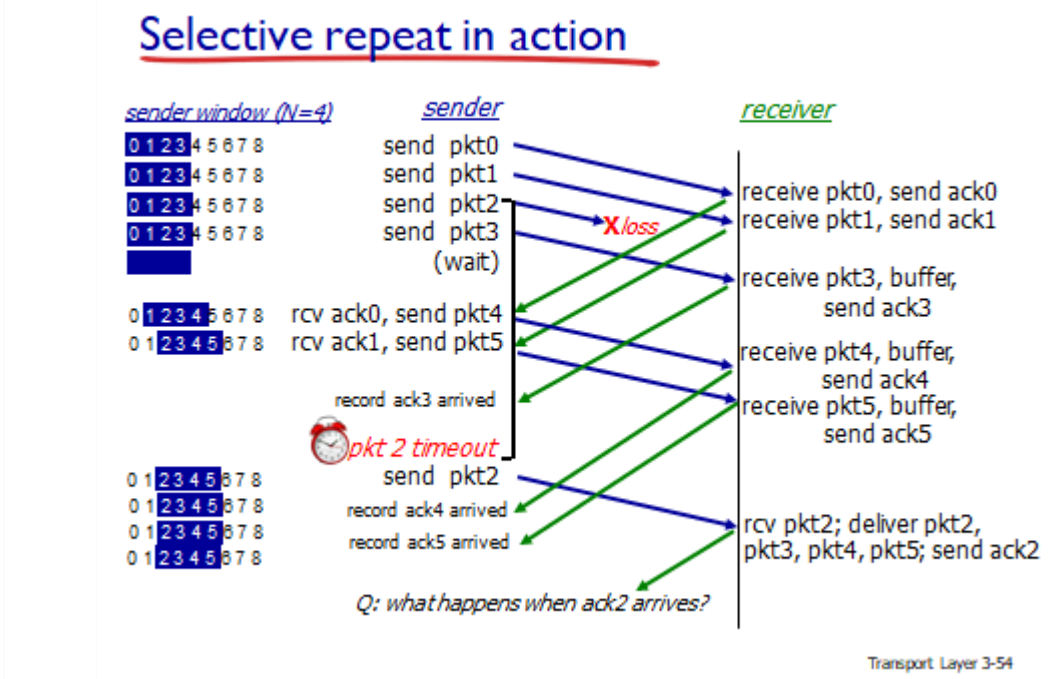
- sender can have up to N unack'ed packets in pipeline
- rcvr sends *individual ack* for each packet
- sender maintains timer for each unacked packet
  - when timer expires, retransmit only that unacked packet

Transport Layer 3-46

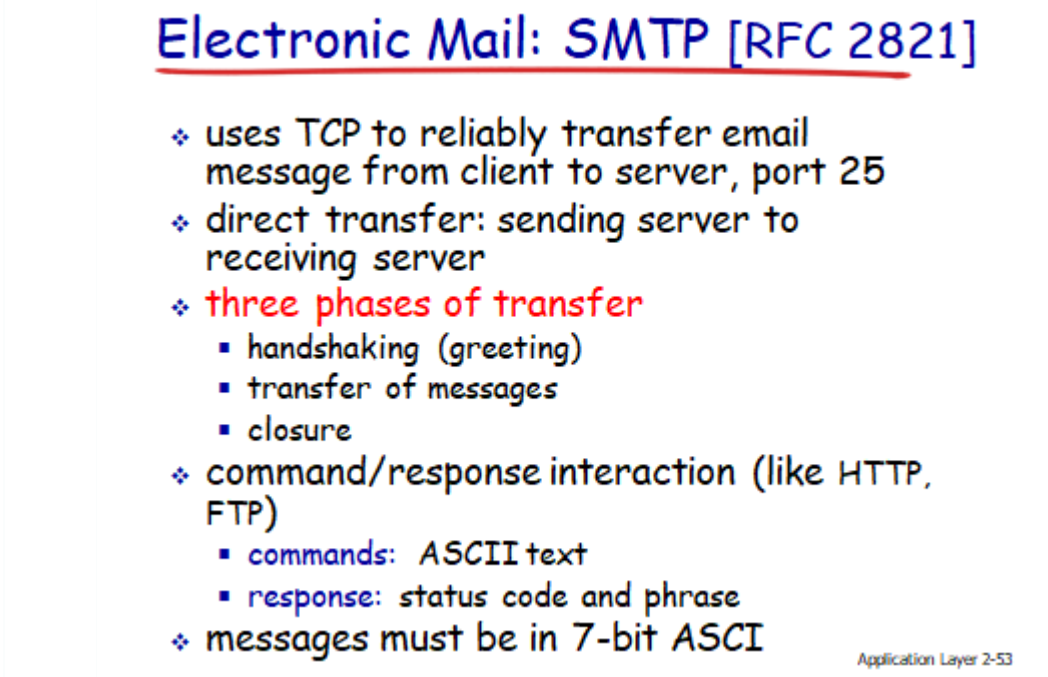
GBN in action



Transport Layer 3-50

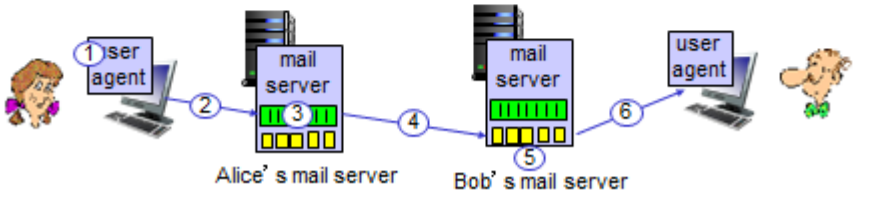


2 (a) Discuss how an electronic mail is transferred over internet.



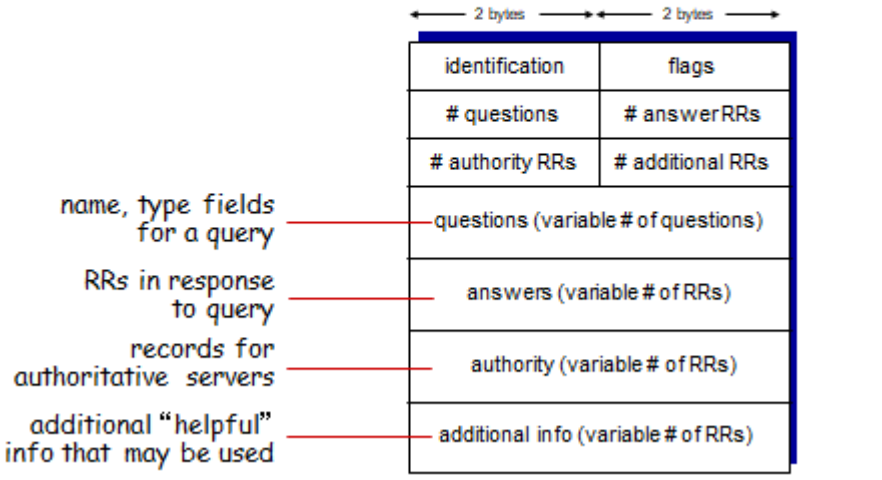
Scenario: Alice sends message to Bob

1) Alice uses UA to compose message "to" bob@someschool.edu
2) Alice's UA sends message to her mail server; message placed in message queue
3) client side of SMTP opens TCP connection with Bob's mail server
4) SMTP client sends Alice's message over the TCP connection
5) Bob's mail server places the message in Bob's mailbox
6) Bob invokes his user agent to read message



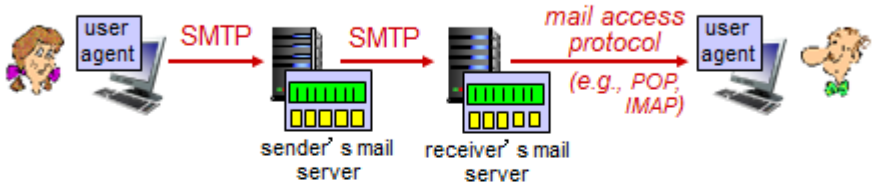
(b) Label DNS message.

DNS protocol, messages



3 (a) Compare and contrast POP and IMAP protocols.

Mail access protocols



- SMTP: delivery/storage to receiver's server
- mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]: authorization, download
  - IMAP: Internet Mail Access Protocol [RFC 1730]: more features, including manipulation of stored msgs on server
  - HTTP: gmail, Hotmail, Yahoo! Mail, etc.

POP3 (more) and IMAP

more about POP3

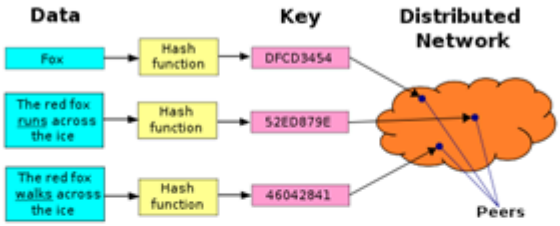
- previous example uses POP3 "download and delete" mode
  - Bob cannot re-read e-mail if he changes client
- POP3 "download-and-keep": copies of messages on different clients
- POP3 is stateless across sessions

IMAP

- keeps all messages in one place: at server
- allows user to organize messages in folders
- keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name

(b) Describe circular DHT.

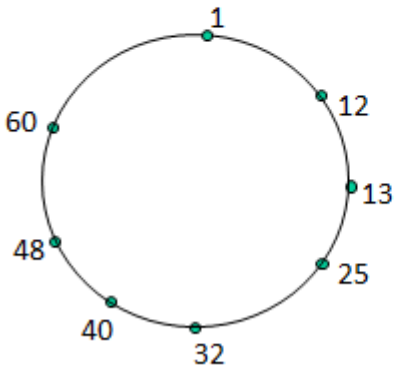
Distributed Hash Table (DHT)

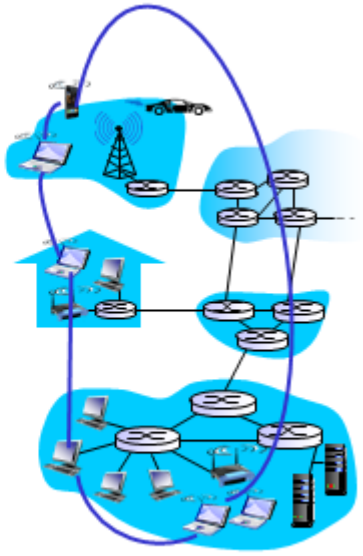


- Distribute (key, value) pairs over millions of peers
  - pairs are evenly distributed over peers
- Any peer can query database with a key
  - database returns value for the key
  - To resolve query, small number of messages exchanged among peers
- Each peer only knows about a small number of other peers
- Robust to peers coming and going (churn)

Circular DHT

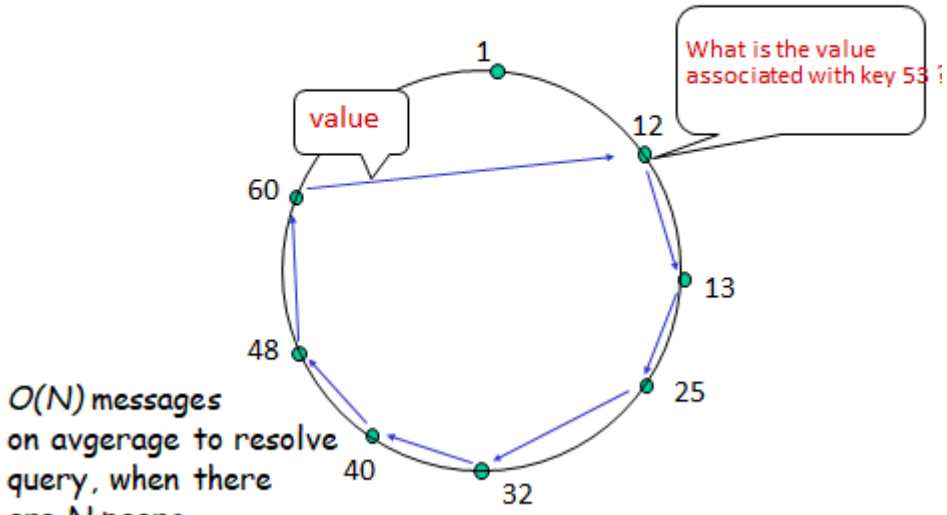
- each peer *only* aware of immediate successor and predecessor.





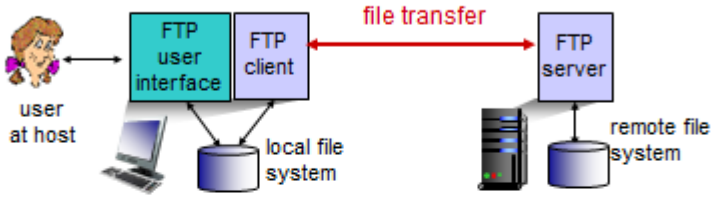
"overlay network"

Resolving a query



4 (a) Describe FTP protocol with example commands & responses.

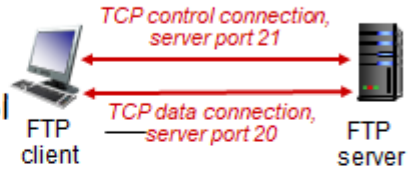
FTP: the file transfer protocol



- transfer file to/from remote host
- client/server model
  - *client:* side that initiates transfer (either to/from remote)
  - *server:* remote host
- ftp: RFC 959
- ftp server: port 21

FTP: separate control, data connections

- FTP client contacts FTP server at port 21, using TCP
- client authorized over control connection
- client browses remote directory, sends commands over control connection
- when server receives file transfer command, *server* opens 2nd TCP data connection (for file) to client
- after transferring one file, server closes data connection



- server opens another TCP data connection to transfer another file
- control connection: *"out of band"*
- FTP server maintains "state": current directory, earlier authentication

Example

```
[19:57:15] USER gavin
[19:57:15] 331 Please specify the password.
[19:57:15] PASS (密碼)
[19:57:15] 230 Login OK, used 22M of 700M.
[19:57:15] SYST
[19:57:15] 215 UNIX Type: L8


[19:59:56] 250 Directory successfully changed.
[19:59:56] PWD
[19:59:56] 257 "/public_html"
[19:59:56] PASV
[19:59:56] 227 Entering Passive Mode (202,38,64,11,143,20)
[19:59:56] LIST
[19:59:56] 150 Here comes the directory listing.
[19:59:56] 226 Directory send OK.
```

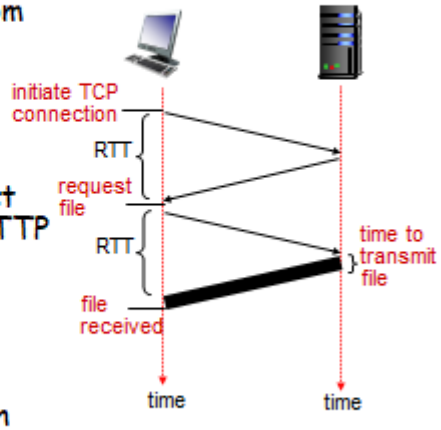
(b) Compare persistent with non-persistent connection.

Non-persistent HTTP: response time

RTT (definition): time for a small packet to travel from client to server and back

HTTP response time:

- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of HTTP response to return
- file transmission time
- non-persistent HTTP response time = 2RTT+ file transmission time



Persistent HTTP

non-persistent HTTP issues:

- requires 2 RTTs per object
- OS overhead for *each* TCP connection
- browsers often open parallel TCP connections to fetch referenced objects

persistent HTTP:

- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over open connection
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects

5 (a) Describe Rdt 3.0 protocol.

rdt3.0: channels with errors *and* loss

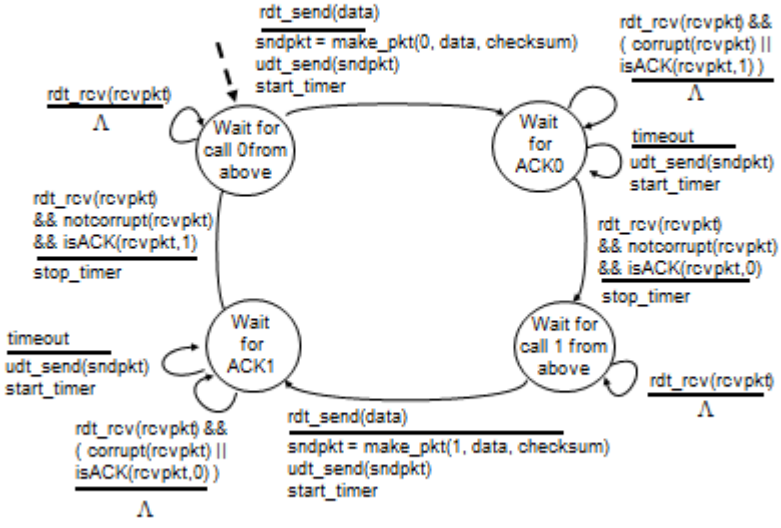
new assumption: underlying channel can also lose packets (data, ACKs)

- checksum, seq. #, ACKs, retransmissions will be of help … but not enough

approach: sender waits "reasonable" amount of time for ACK

- retransmits if no ACK received in this time
- if pkt (or ACK) just delayed (not lost):
  - retransmission will be duplicate, but seq. #'s already handles this
  - receiver must specify seq # of pkt being ACKed
- requires countdown timer

rdt3.0 sender



6 (a) Bob is sending a 16-bit 3 words to Alice using UDP, Alice has to test whether bit error is present in the received data or not. Sender side data:

Data Sent: 1000101010111100

0101011110001110

1111000011100011. Identify whether error is there or not.

Sum: 1101001100101110

Checksum: 0010110011010001

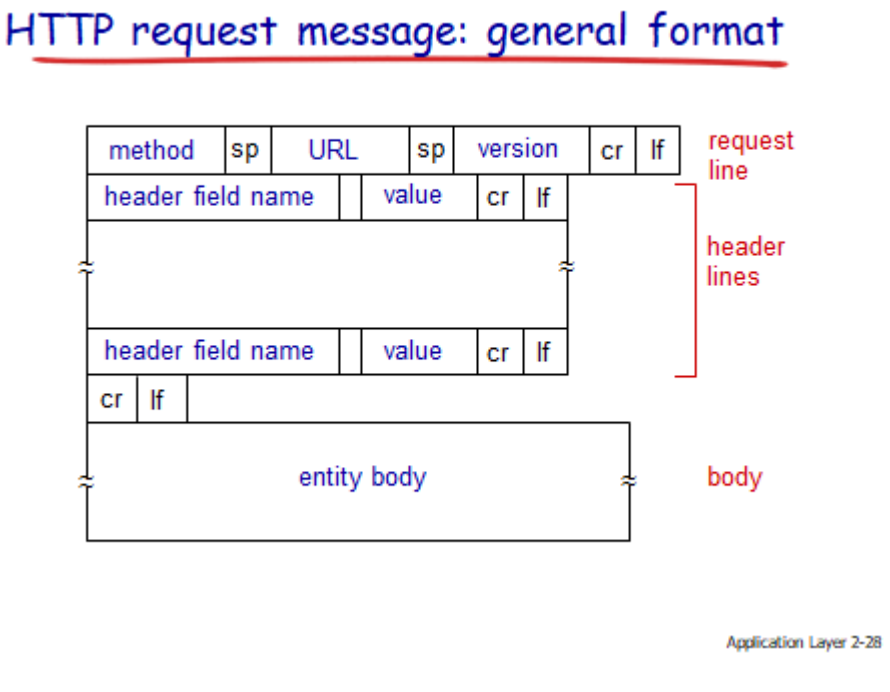
calculated in sender side.

Receiver side: Adding data with checksum

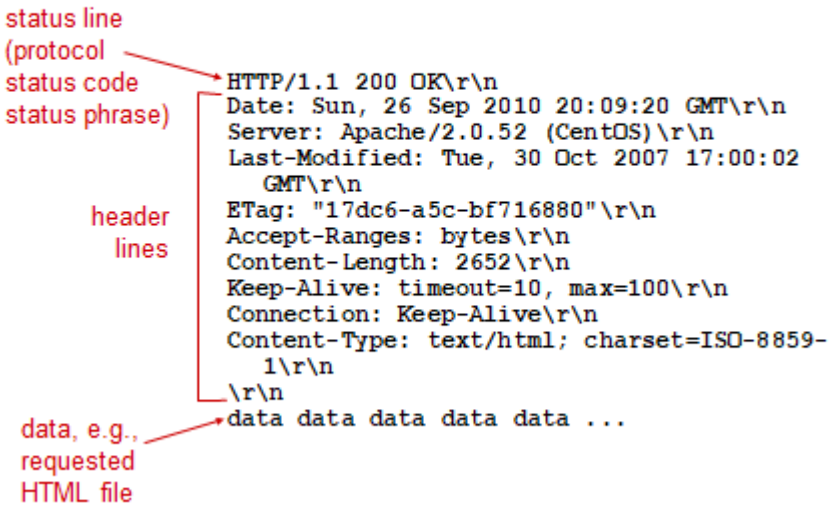
Result: All 1's

So there is no error in the received data.

(b) Quote HTTP message format.



HTTP response message



7 (a) Summarize the working of DNS.

DNS: domain name system

people: many identifiers:
- SSN, name, passport #

Internet hosts, routers:
- IP address (32 bit) - used for addressing datagrams
- "name", e.g., www.yahoo.com - used by humans

Q: how to map between IP address and name, and vice versa ?

Domain Name System:
- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol:* hosts, name servers communicate to *resolve* names (address/name translation)
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network's "edge"

DNS: services, structure

DNS services

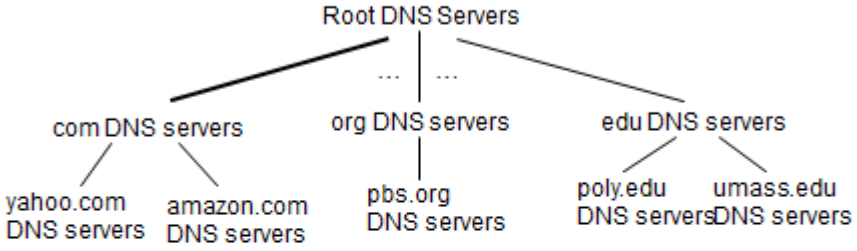
- hostname to IP address translation
- host aliasing
  - canonical, alias names
- mail server aliasing
- load distribution
  - replicated Web servers: many IP addresses correspond to one name

why not centralize DNS?

- single point of failure
- traffic volume
- distant centralized database
- maintenance

  A: doesn't scale!

DNS: a distributed, hierarchical database



client wants IP for www.amazon.com; 1st approx:

- client queries root server to find com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

DNS: root name servers

- contacted by local name server that can not resolve name
- root name server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server

TLD, authoritative servers

top-level domain (TLD) servers:

- responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp
- Network Solutions maintains servers for .com TLD
- Educause for .edu TLD

authoritative DNS servers:

- organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts
- can be maintained by organization or service provider

Local DNS name server

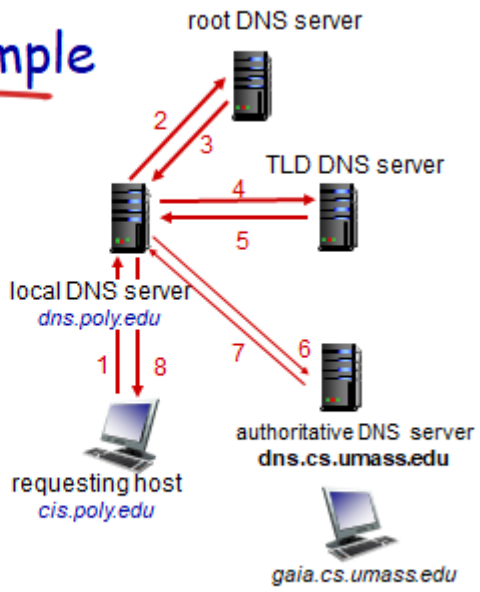
- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one
  - also called "default name server"
- when host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts as proxy, forwards query into hierarchy

DNS name resolution example

- host at cis.poly.edu wants IP address for gaia.cs.umass.edu

iterated query:

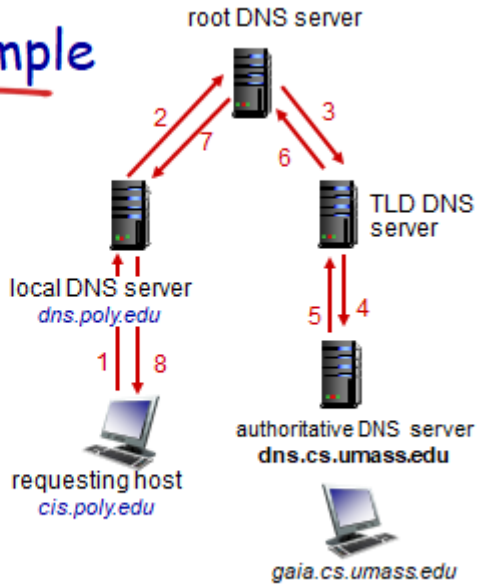
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS name resolution example

recursive query:

- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?



Application Layer 2-71

8 (a) Consider your college is providing 15Mbps internet connectivity for public network and 100 Mbps speed within local area network; whether you are able to browse the web using this low speed, if yes explain.

Yes, it is possible by using cache server.

Caching example: install local cache

assumptions:

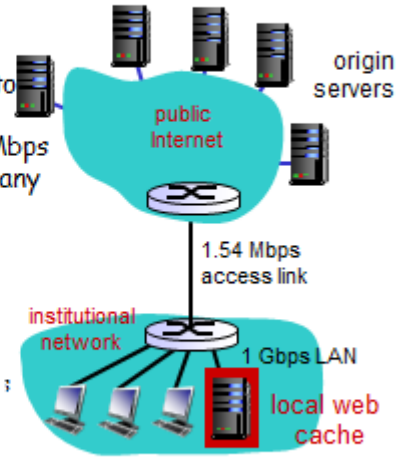
- avg object size: 100K bits
- avg request rate from browsers to origin servers:15/sec
- avg data rate to browsers: 1.50 Mbps
- RTT from institutional router to any origin server: 2 sec
- access link rate: 1.54 Mbps

consequences:

- LAN utilization: 15%
- access link utilization?
- total delay =?

How to compute link utilization, delay?

Cost: web cache (cheap!)



Application Layer 2-42

Caching example: install local cache

Calculating access link utilization, delay with cache:

- suppose cache hit rate is 0.4
  - 40% requests satisfied at cache, 60% requests satisfied at origin
- access link utilization:
  - 60% of requests use access link
- data rate to browsers over access link = 0.6*1.50 Mbps = .9 Mbps
  - utilization = 0.9/1.54 = .58
- total delay
  - = 0.6 * (delay from origin servers) +0.4 * (delay when satisfied at cache)
  - = 0.6 (2.01) + 0.4 (~msecs)
  - = ~1.2 secs
  - less than with 154 Mbps link (and cheaper too!)