Midterm 1

● Graded

Student

AKSHAJ KAMMARI

Total Points

49 / 72 pts

Question 1

1.a — **4** / 4 pts

- **− 2 pts** Incorrect (i)
- **− 2 pts** Incorrect (ii)
- ✔ **− 0 pts** Correct
- **− 1 pt** Partial (i)
- **− 1 pt** Partial (ii)

Question 2

1.b — **4** / 4 pts

- ✔ **− 0 pts** Correct
- **− 2 pts** 1 incorrect
- **− 1 pt** 1 explanation unclear
- **− 4 pts** Both incorrect
- **− 2 pts** 2 explanations unclear

Question 3

1.c — **0** / 2 pts

- **− 0 pts** Correct
- **− 1 pt** explanation unclear
- ✔ **− 2 pts** incorrect

Question 4

1.d **4** / 4 pts

- ✔ **− 0 pts** Correct
- **− 2 pts** Need to subtract from 100%
- **− 4 pts** incorrect
- **− 1 pt** arithmetic error

Question 5

2.a **5** / 10 pts

- **+ 5 pts** 2s for data
- ✔ **+ 5 pts** 400ms for handshake
- **− 2 pts** Arithmetic error
- **+ 0 pts** Incorrect
- **− 3 pts** incorrect prop delay

Question 6

2.b **6** / 10 pts

- **− 0 pts** Correct
- **− 4 pts** Partial
- ✔ **− 4 pts** Not represented as difference of points
- **− 10 pts** incorrect

Question 7

3.a **2** / 2 pts

- ✔ **− 0 pts** Correct
- **− 2 pts** Empty or incorrect answer

Question 8

3.b **2** / 2 pts

- ✔ **− 0 pts** Correct
- **− 1 pt** partial credit
- **− 2 pts** empty or incorrect answer

Question 9

3.c

2 / 2 pts

✔ **− 0 pts** Correct

− 1 pt patrial correct

− 2 pts empty or incorrect answer

Question 10

3.d

2 / 2 pts

✔ **− 0 pts** Correct

− 2 pts empty or incorrect answer

Question 11

4.a

0 / 3 pts

− 0 pts Correct

✔ **− 3 pts** Incorrect

− 1 pt Partially correct

Question 12

4.b

0 / 3 pts

− 0 pts Correct

− 1 pt Did not mention one of the steps
or the name is incorrect (did not add .edu and .net)

− 2 pts Invalid step or missed 2 steps

✔ **− 3 pts** Incorrect

Question 13

4.c

0 / 4 pts

− 0 pts Correct

− 1 pt Did not mention Princeton.edu

− 2 pts Invalid step

− 3 pts Just mention Princeton.edu

✔ **− 4 pts** Incorrect

Question 14

5.a

10 / 10 pts

✔ **− 0 pts** Correct

− 9 pts With slow start but no linear phase:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 8, 16.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 8 packets.
- In the fourth RTT, you can send 16 packets.
Since you need to send 25 packets, it would take 4 RTTs.

− 8 pts With slow start but no linear phase:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 8, 16.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 8 packets.
- In the fourth RTT, you can send 16 packets.
Since you need to send 25 packets, it would take 4 RTTs.

− 3 pts With slow start but no linear phase:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 8, 16.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 8 packets.
- In the fourth RTT, you can send 16 packets.
Since you need to send 25 packets, it would take 4 RTTs.

− 10 pts With slow start but no linear phase:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 8, 16.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 8 packets.
- In the fourth RTT, you can send 16 packets.
Since you need to send 25 packets, it would take 4 RTTs.

− 2 pts With slow start but no linear phase:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 8, 16.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 8 packets.
- In the fourth RTT, you can send 16 packets.
Since you need to send 25 packets, it would take 4 RTTs.

− 5 pts With slow start but no linear phase:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 8, 16.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 8 packets.

- In the fourth RTT, you can send 16 packets.
  Since you need to send 25 packets, it would take 4 RTTs.

Question 15

5.b **8 / 10 pts**

– 0 pts Correct

– 5 pts With slow start and linear phase with SS_THRESHOLD = 4 packets:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 5, 6, 7, 8.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 5 packets.
- In the fourth RTT, you can send 6 packets.
- In the fifth RTT, you can send 7 packets.
- In the sixth RTT, you can send 8 packets.

Since you need to send 25 packets, it would take 6 RTTs.

– 10 pts With slow start and linear phase with SS_THRESHOLD = 4 packets:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 5, 6, 7, 8.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 5 packets.
- In the fourth RTT, you can send 6 packets.
- In the fifth RTT, you can send 7 packets.
- In the sixth RTT, you can send 8 packets.

Since you need to send 25 packets, it would take 6 RTTs.

– 8 pts With slow start and linear phase with SS_THRESHOLD = 4 packets:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 5, 6, 7, 8.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 5 packets.
- In the fourth RTT, you can send 6 packets.
- In the fifth RTT, you can send 7 packets.
- In the sixth RTT, you can send 8 packets.

Since you need to send 25 packets, it would take 6 RTTs.

– 1 pt Your calculation is Correct, but in final statement it is wrong. It should be 6 RTT's instead of 8 RTT's.

– 7 pts With slow start and linear phase with SS_THRESHOLD = 4 packets:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 5, 6, 7, 8.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 5 packets.
- In the fourth RTT, you can send 6 packets.
- In the fifth RTT, you can send 7 packets.
- In the sixth RTT, you can send 8 packets.

Since you need to send 25 packets, it would take 6 RTTs.

– 9 pts With slow start and linear phase with SS_THRESHOLD = 4 packets:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 5, 6, 7, 8.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 5 packets.
- In the fourth RTT, you can send 6 packets.

- In the fifth RTT, you can send 7 packets.
- In the sixth RTT, you can send 8 packets.

Since you need to send 25 packets, it would take 6 RTTs.

✔ **– 2 pts** With slow start and linear phase with SS_THRESHOLD = 4 packets:
In this scenario, the congestion window starts at 2 packets and increases as follows: 2, 4, 5, 6, 7, 8.

- In the first RTT, you can send 2 packets.
- In the second RTT, you can send 4 packets.
- In the third RTT, you can send 5 packets.
- In the fourth RTT, you can send 6 packets.
- In the fifth RTT, you can send 7 packets.
- In the sixth RTT, you can send 8 packets.

Since you need to send 25 packets, it would take 6 RTTs.

CS 352 Fall 2023
Midterm 1 (80 Minutes)

On my honor, I have neither received nor given any unauthorized assistance on this examination (assignment):

First Name: Akshaj

Last Name: Kammari

NetID: ak1990

Question	Score
Problem 1 (14)	
Problem 2 (20)	
Problem 3 (8)	
Problem 4 (10)	
Problem 5 (20)	
Total (72)	

NetID: ak1090

OSI layer
high to low
application
presentation
session
transport
network
data link
physical

1) Intro to Networking (14 points)

a. **(4 points)** The concept of encapsulation is used widely in networking. Given that layer A can encapsulate packets from layer B:

i. Pick any 2 layers A and B from the OSI model that fit the above description, provide their names and justify how layer A encapsulates packets from B.

Lets take layer A as transport and layer B as session. From client side, as data moves down the OSI layer hierarchy, packets from session goes to transport and transport encapsulates these packets by adding headers and trailers.

ii. Can logic that operates on layer A modify the packet headers created by layer B? Justify your answer for full credit.

No, because packet headers change as data passes through layers. Even if it was possible it would lead to weaker encapsulation.

b. **(4 points)** Name the bottom 2 layers of the 7 layer OSI model. For each, provide the function the layer performs.

data link layer: creates frames of data and transfer between nodes

physical layer: contains the raw bit stream

c. **(2 points)** You are designing a voice-over-IP service. It is critical that your service provides low-latency audio. Which transport layer protocol would you use and why?

TCP, reliability is [illegible], as it has a fixed connection with a fixed destination, which has the best chance of avoiding loss of data.

NetID: ak1990

d. **(4 points)** A certain physical layer has a maximum transmission unit (MTU) of 100 bytes. Given the network stack below, calculate the maximum efficiency, or percentage of bandwidth, an application can receive. You can leave your answer in as an unreduced fraction:

Layer	Header size (bytes)	Trailer size (bytes)
Transport	9	0
Network	8	0
Link	5	2

100 - 9 - 8 - 5 - 2 = 76 bytes is the maximum amount of data that can be sent

$\boxed{\frac{76}{100}}$

2) Internet Properties (20 points)

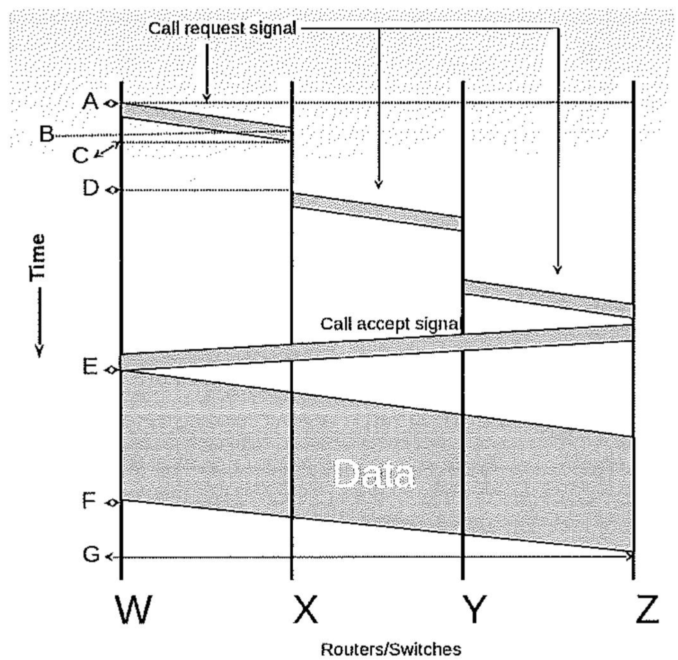
a. **(10 points)** Calculate the total time required for the receiver to get a 3000Kb file in the following: Assume a Round-Trip-Time (RTT) for 1 bit to be 200ms and an initial 2xRTT of handshaking in the protocol before any data can be sent. The bandwidth is 1.5Mbps, and the data is sent in one chunk, as in message switching. The total time to receive the message is (leave you answer in unreduced form):

$3000 \text{ KB} \cdot \frac{1000 \text{ b}}{1 \text{ KB}} \cdot \frac{8 \text{ bits}}{1 \text{ byte}} = 24\,000\,000$

$\frac{24\,000\,000}{1\,500\,000} + 2(200) = \boxed{\frac{24\,000\,000}{1\,500\,000} + 400 \text{ ms}}$

NetID: ak1798

b. **(10 points)** The image below shows an abstract model of a computer sending a message through a network. Increasing time is shown on the Y-axis and the route through the network is shown on the X-axis. Points in time are labeled on the Y-axis (e.g., Points A,B,...G). The time between 2 points A and B can be expressed as (B-A).



The efficiency of the network, that is, the fraction of time useful work is accomplished, is represented by what mathematical expression given the time points in the above diagram?

$$\boxed{\frac{EF}{AG}}$$

useful work = time data is actually being transferred

3) Application layer (8 points)

a. **(2 points)** HTTP is a stateless protocol, however, web sites need state for functions, for example, shopping carts. How was HTTP modified to enable stateful applications?

Through the addition of cookies

NetID: gk1990

b. **(2 points)** How does persistent HTTP (version 1.1) reduce the number of RTTs needed to make a series of requests compared to version 1.0?

Since at most one object is sent over a TCP connection in a persistent HTTP, The data must be bunched into a large circuit and sent at once instead of broken up into many small packets and sent multiple times

Consider the following text strings that were received by a web-server from a client:

```
      GET /cs352/index.html HTTP/1.1
      Host: www.cs.rutgers.edu
      User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0)
Firefox/80.0
      Accept:ext/xml,      application/xml,      application/xhtml+xml,
text/html;q=0.9,
      text/plain;q=0.8,image/png,*/*;q=0.5
      Accept-Language: en-us,en;q=0.5
      Accept-Encoding: zip,deflate
      Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
      Keep-Alive: 300
      Connection:keep-alive
```

c. **(2 points)** The URL of the document requested by the browser was: /cs352/index.html

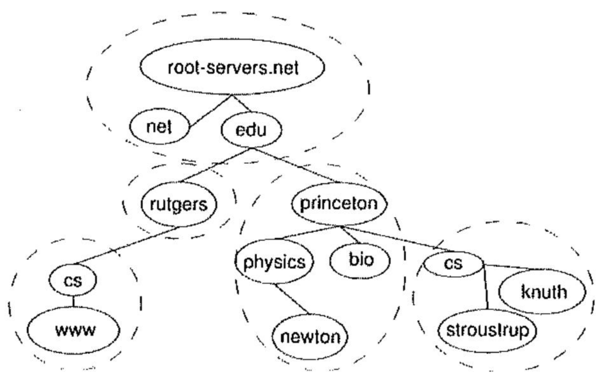
d. **(2 points)** The HTTP protocol version is : 1.1

NetID: [illegible]

4) DNS (10 points)

Consider the Internet in the figure below, in which zones are indicated by a dashed line. There is only one DNS server per zone, and it happens to have the same name as the highest node in each zone: *rootservers.net*, *rutgers.edu*, *cs.rutgers.edu*, *princeton.edu*, *cs.princeton.edu*. The only servers supporting recursive queries are *cs.rutgers.edu* and *cs.princeton.edu*.

For each of the queries below, list, in order, all the DNS servers contacted by the resolver (located in the OS of the machine running the query). Assume there is no caching performed at any level of the hierarchy.



a. **(3 points)** A user on *knuth.cs.princeton.edu* launches the query:

```
nslookup www.cs.rutgers.edu
```

knuth, cs, princeton, edu, rutgers, cs

b. **(3 points)** A machine *newton.physics.princeton.edu* launches a query:

```
nslookup remus.rutgers.edu
```

newton, physics, princeton, edu, rutgers

c. **(4 points)** The machine *eden.rutgers.edu* is physically moved into the same room as *www.cs.rutgers.edu*, but maintains the same IP address. A user on *bio.princeton.edu* launches the query:

```
nslookup eden.rutgers.edu
```

bio, princeton, edu, rutgers, cs

NetID: ak1990

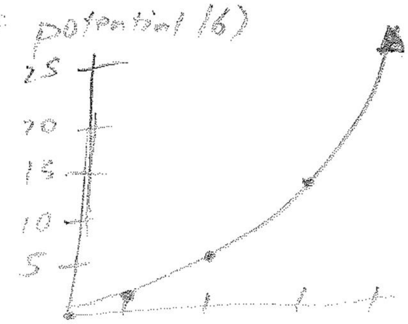
5) TCP (20 points)

A TCP connection has a flow control window of 40 packets and a minimum congestion window of 2 packets. How many RTTs are required to send 25 packets (with sequence numbers 1 through 25), assume no packet loss under the following conditions:

a. **(10 points)** With slow start but no linear phase, and no packet loss? Show your work, and/ or give a diagram for full credit:

RTT #	Sequence #
1	1, 2
2	3, 4, 5, 6
3	7, 8, 9, 10, 11, 12, 13, 14
4	15 - 25 (out of potential 16)

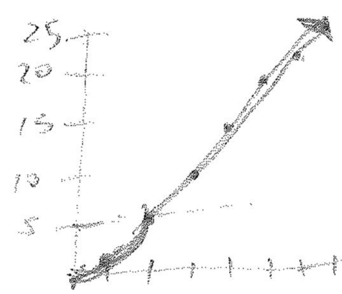
minimum congestion window of 2 packets otherwise # of RTT+1



$\boxed{\text{4 RTTs}}$

b. **(10 points)** Assume slow start and linear phase, with SS_THRESHOLD=4 packets. Show your work, and/ or give a diagram for full credit:

RTT #	Sequence #
1	1, 2
2	3, 4, 5, 6
3	7, 8, 9, 10
4	11, 12, 13, 14
5	15, 16, 17, 18
6	19, 20, 21, 22
7	23, 24, 25 (out of potential 4)



threshold

$\boxed{\text{7 RTTs}}$

TCP Slow Start: When connection begins, CongWin = 1 MSS, MSS=Max segment size, including data and header. Slow start doubles packet size on every succ ACK and resets if there is a timeout. To combat this, we have linear increase,Start the threshold at 64K, Slow star.Once the threshold is passed For each ack received, cwnd = cwnd + (mss*mss)/cwnd. TIMEOUT - reset window =1mss

TCP Message types- SYN(synch) SEQ()
Fast retransmit. Instead of timing out, TCP will resend the last packets rather than restarting the connection

Select a feature that UDP provides that TCP does not provide
Minimal processing delay

Transport layer multiplexing means the transport layer:
Routes packets to the correct process on the end host.

Suppose the UDP checksum in the UDP packet header does not equal the checksum computed by the receiver. The cause of this inequality could be because:
The physical layer at WiFi hub experienced interference and recorded burst of 1's when a long sequence of 0's was sent.
A C-language pointer bug in a router caused a program to write part of the header over the user-data portion of the UDP packet
A virus in the sender's operating system placed extra data in the UDP packet to control spam robots.

Given that HTTP can already transfer arbitrarily long streams of bytes, so what are MIME types for? Identifying the format of the data

Consider an HTTP client that wants to retrieve a Web document at a given URL. The IP address of the HTTP server is initially unknown. What transport and application-layer protocols besides HTTP are needed in this scenario? TCP and DNS

Because HTTP is stateless, if client requests an identical web URL it previously requested within the last 1 second, the server will: Resend the message.

At what layer is the DNS protocol? App layer

If you contacted the 'rutgers.edu' name server to find out the IP address of 'cs.rutgers.edu' was, what type of name server would you be talking to? Authoritative

An authoritative DNS server: Holds type A (address) records for its domain.

SMTP is what kind of protocol? In band

Recall in the FTP protocol, the server remembers the current working directory of the client between LIST directory commands, and the CD (change directory) command changes the current working directory. This design choice means the FTP protocol: Is a stateful protocol

FTP is an out-of-band protocol because: It uses one socket to send commands, and a second socket to send data

Select a feature that UDP provides that TCP does not provide: minimal processing delay

If TCP used the arithmetic mean of all packets ACKs' to compute the round-trip time (RTT) it would mean: The observations of the network in the past have the same influence on the TCP timeout value as recently received packets.

What is put in the TCP header in order to accomplish flow control? The number of bytes the target host can accept

A straightforward implementation of a TCP would use what data structure to hold packets, and why? A queue, to provide in order delivery

A port scan is: sending packets and waiting for responses to different ports in order to map which applications are listening on which ports.

Waiting for an acknowledgement for every packet before sending the next one causes the time to send a long message to become proportional to the round-trip- time. What transport strategy mitigates the impact of long round-trip times (RTTs)? Pipelining packets

Why does TCP perform a retransmit immediately after three duplicate ACKs? It is likely that a packet was lost.

If the network is probably congested, is it better to: Reduce the amount of unacknowledged data.

Functions $\text{Timeout} = \beta * \text{SRTT}$

switching time

$\text{SRTT} = \alpha\ \text{SRTT} + (1-\alpha)\ \text{RTT}_{\text{arriving ACK}}$

RTT -> Return Time to Sender (time to send + return ack)

Jacobson / Karels Algorithm

Error = RTT - SRTT
SRTT = SRTT + ($\alpha \times$ Error)
Dev = Dev + [$\delta \times$ (|Error| - Dev)]
Timeout = SRTT + ($\beta \times$ Dev)

Single link delay

Number of switches (in-between stations)	S
Time to set up circuit:	C
Size of the packet:	P
Size of the header:	h

Circuit

$$C + \frac{N}{B}$$

Internet: Network of networks. Best effort. Entire array arrives in tact or not at all. No guarantees.

Encapsulation: add information needed by current layer around higher layers data, Treat the neighboring layer's information as a "black box", can't look inside or break message

TCP socket identification: (IP address, port number) for both source and destination, stream sockets. fixed connection with fixed destination.
Congestion control, Flow control

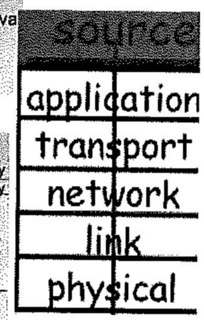
UDP: connectionless Internet transport protocol. No congestion control, optional **checksum** for error detection, datagram sockets. Can be lost

Use layers to hide complexity, Layer N uses service provided by layer N-1. layer N-1 provides a service to layer N

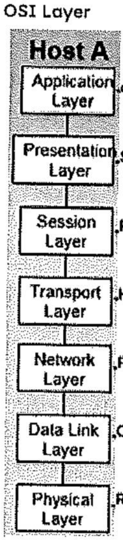
Queuing delay

- R=link bandwidth (bps)
- L=packet length (bits)
- a=average packet arriva

traffic intensity = La/R



OSI Layer



Application Layer: communication between two applications of the same type (FTP / HTTP)

Presentation Layer: Specific functions such as encryption, conversion

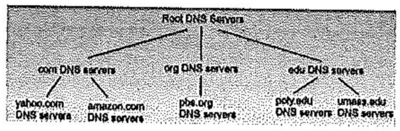
Session Layer: Perform sync between apps

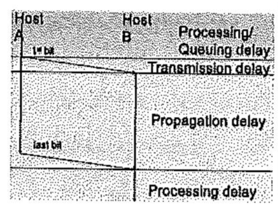
Transport Layer: Hides network detail from session. Provides reliable end-to-end communication / flow control, retransmission

Network Layer: Routing decisions, congestion control

Data Link Layer: Creates frames of data and transfer between nodes

Physical Layer: Raw bit stream





La/R ~ 0: average queueing delay small
La/R → 1: delays become large
La/R > 1: more "work" arriving than can be serviced, average delay infinite!

Total delay = Internet delay + access delay + LAN delay

Initial SRTT = 1.50
$\alpha = 0.875$, $\beta = 4.0$

RTT Meas.	SRTT	Timeout
1.5 s	= 1.50	= $\beta \times 1.50 = 6.00$
1.0 s	= $1.50 \times \alpha + 1.0 \times (1-\alpha) = 1.44$	= $\beta \times 1.44 = 5.76$
2.2 s	= $1.44 \times \alpha + 2.2 \times (1-\alpha) = 1.54$	= $\beta \times 1.54 = 6.16$
1.0 s	= $1.54 \times \alpha + 1.0 \times (1-\alpha) = 1.47$	= $\beta \times 1.47 = 5.88$
0.8 s	= $1.47 \times \alpha + 0.8 \times (1-\alpha) = 1.39$	= $\beta \times 1.39 = 5.56$
3.1 s		
2.0 s		

Packet Switching time

Processing delay	T1
Message Size (bits, bytes)	N
Link bandwidth (bps)	B
Transmission delay (seconds)	T2

T1+N/B+T2

$$(S+1) * \frac{(p+h)}{B} + (\lceil \frac{N}{p} \rceil - 1) * \frac{(p+h)}{B}$$

Packet Delay

Node processing: execute protocol code, check bit errors, determine output link

Queuing: wait time at output link for transmission, depends on congestion level at router

Transmission delay: time to get bits on wires, packet length/link bandwidth

Propagation delay: time for bits to move across the wires, length of link/propagation speed

Nodal delay = processing delay(plan to go) + queueing delay(waiting to go) + transmission delay(getting out the door) + propagation delay(travel time)

App Archs

Client-Server: Server, always on, permanent ip. Clients, Communicate with server, may intermittently connect May have dynamic IPaddresses. Don't communicate directly with each other

Pure P2P Architecture: No always-on server, end systems directly communicate. Peers are intermittently connected and change IPaddresses. Highly scalable but difficult to manage

Hybrid of Client-Server and P2P: **Skype:** Voice-over-IP P2P application, Centralized server: finding address of remote party, Client-client connection: direct (not through server)

HTTP

Uses TCP, HTTP is "stateless"Server maintains no info about past client requests

Client initiates TCP connection (creates socket) to server, port 80. Server accepts TCP connection from client. HTTP messages (applicationlayer protocol messages) exchanged between browser (HTTP client) and Web server(HTTP server). TCP connection closed

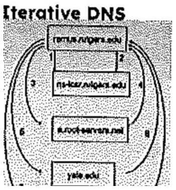
Nonpersistent HTTP: At most one object is sent over a TCP connection.

Persistent HTTP: Multiple objects can be sent over single TCP connection between client and server.

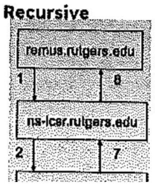
FTP

transfer file to/from remote host. client/server model- client: side that initiates transfer (either to/from remote), server: remote host

Iterative DNS



Recursive

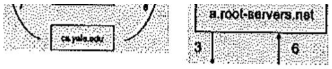


14. Suppose a sender is using a stop-and-wait protocol on a cross country gigabit link (10^9/bits/sec). A packet is 1000 bytes (8000 bits), and the propagation delay is 30 msec. Ignoring the transmission delay for the acknowledgements, the effective bandwidth is ___ Mb/s and the link utilization is ___%.

Answer: Recall in a stop and wait protocol, every packet transmission must wait for an acknowledgement. Compute the time the sender is transmitting over the total time transmitting and waiting foe the ack to return. See the book, figure 3.17 for an example.

- Effective bandwidth: the time it actually takes to send a bit

$= L/T$



$$= L'/(t_{trans} + t_{\text{wait-rec}} + t_{ack})$$
$$= L/(t_{trans} + t_{propa})$$
$$= 8000/(8000/10^9 + 30 * 10^{-3})$$