Kev Sharma
NetID - kks107
Slot 1 of Final Exam

CS352 Final Examination

1. **Why are the internet protocols (such as IP stack and ISO/OSI Reference Model) arranged in layers?**

- The internet protocols are arranged in layers because this analogy is useful:
  - In explaining the service model of each layer
  - Providing separation of concerns between layers
  - And allowing layers to operate at different levels of abstraction

Let us assume the top layer is the application layer. The arrangement in layers is used to categorize the service provided by each layer. If you take any layer (let's say transport layer) from the stack, the cumulative services provided at that layer include transport layer services and all the services of the layers beneath it (network layer and so on).

This layering also serves as a way to modularize the functionality of any given layer in a plug-and-play like infrastructure. For example we could swap out IPv4 with IPv6 in the network layer as long as the functionality provided by IPv6 does not reduce the offered services of the network layer. The addition of new layers enhances the abilities provided by the immediate layer beneath it or adds new functionality that may be used by the layers above it. Example - TCP provides reliable data transfer (fixing unreliability of IP and giving reliable data transfer to application layer).

This layering of the IP stack helps higher layers operate at a level of abstraction greater than the layers beneath it. For example: the transport layer provides logical communication to processes whereas the network layer provides logical communication between different end systems.

2. List Two key differences between Clients and Servers in the Client-Server architecture.

- Clients initiate a connection to a Server to be serviced, whereas Servers wait for connections to service. More specifically, client processes are the ones who initiate contact with server processes.

- Client processes typically request data from Server processes whereas Server processes give data to client processes.

3. Could a network path with high bandwidth also have high propagation delay? Say YES or NO. Justify your answer.

YES, because d_transmission and d_propagation are not correlated.

D_trans (transmission delay) = L/R
Where L = the number of bits in a packet (bits)
And R = bandwidth of the link (bits per sec)

D_prop (propagation delay) = d/s
Where d = distance of the link and s = propagation speed.

It is possible for the following scenarios:

- Distance of the link `d` is very high
- Propagation speed is very low

Either of these cases drastically increases the propagation delay.

Note that neither the bandwidth of the link (bps) nor the transmission delay can alter the propagation delay because they cannot change the distance of the link nor the propagation speed. This leaves the propagation delay unaffected and uncorrelated to transmission rates and bandwidth speeds.

Therefore, it is possible for a network path to have high bandwidth and still have high propagation delay.

4. Why do we need the DNS protocol?

- DNS helps to provide logical communication between two hosts in two ways.
  DNS works like a distributed database:
  - In a computer network, not every node is able to keep track of the entire network topology. That means, that a node cannot keep track of the IP address (as used by IPv4 to provide logical communication between two different hosts) of every other node in the network. A system is therefore needed which provides a means to query the IP address of another node. The Domain Name System (DNS) Protocol provides a means to query for other nodes' IP addresses such that the network layer may transport datagrams from one end-system to another end-system whose IP address was previously unknown.

  DNS works like a translator:
  - If an application program wants information from "www.google.com", the network layer still requires an IP address to provide communication between two processes (need a [IP address, Port number] pair). The DNS serves to map hostnames (that are more memorable) to IP addresses (that the network layer needs but the application might not have).

I have chosen to explain the difference between Simple Mail Transfer Protocol and MAP rather than “Short Message Transfer Protocol'' and MAP.

5. List the difference between SMTP & MAP.

First, we note that the relationship between a mail server and a mail access client is as follows:

- A mail access client pushes a message to a mail server and expects that mail server to use SMTP to relay that message to another mail server.
- When that message is received in the other mail server, a mail access protocol (MAP) is used to retrieve the message from the mail server and allow a user to interact with it through a user client agent (like Gmail).
- The relationship between a user client agent and a mail server is like the Client-server architecture. In such an architecture, the server (mail server) is always on listening for messages that might be sent to it and sending messages to other mail servers. The client (user client agent) pulls messages that may have been received at the user’s mail server.

SMTP is used to transfer messages between mail servers. That is, a mail server uses the Simple Mail Transfer Protocol to send or receive a message to another mail server.

MAP is used by a user client agent (like Gmail) to retrieve the mail on that user’s mail server.

6. List two key differences between TCP and UDP.

   Within the transport layer:

   a. TCP provides a reliable data transfer service between end systems whereas UDP makes no guarantee that data will be transferred to the receiver.
   b. TCP provides a mechanism to throttle senders in order to reduce network congestion whereas UDP does not restrict the outflow of packets whatsoever.

7. UDP offers no real guarantees on data reliability or ordering to applications using it as a transport protocol. So why do some applications still use UDP?

   Unlike TCP, UDP senders are not throttled nor are they affected by the slowdown caused by TCP handshaking procedures. UDP is faster and therefore application developers who want better performance with less delays may prefer UDP over TCP.

   Certain applications may also not be affected by packet loss in the same way that others are. For example, file transfer may need reliable data delivery; but a gaming application may have a higher tolerance for dropped packets for which UDP is more suitable.

   A proprietary application protocol can then reserve the choice to use UDP for its speed and build the data reliability and ordering in the application program if it needs to.

8. Why do we need a retransmit timeout (RTO) in a TCP sender?

Packet loss occurs when a packet switch has a full buffer and an incoming packet may not be buffered in that packet switch. A congested network implies that many such packet switch buffers are nearing or at capacity. In a congested network then, many incoming packets may be dropped.

When a TCP sender does not receive an acknowledgement for a packet it sent before acknowledgement timeout, it may infer the underlying cause to be packet loss. Because one of TCP's functions is to provide a mechanism for reducing network congestion, TCP will need to throttle the sender and prevent it from sending packets into the network as fast it currently is (to reduce network congestion).

TCP throttles a sender by enforcing a retransmit timeout (RTO) in the event of many such missing acknowledgements. When many senders in a network are throttled for some time, the network gets less congested as the packet switches have more time to transmit the buffered packets and reduce the number of buffered packets.

9. **Why do we need Selective Repeat given we have Go-Back N already?**

   To assess the motivation of using Selective Repeat, let us use an example from Go-Back N:

   - In Go-Back N assume that a sender sends packet 1,2,3,4,5 to the receiver. For some reason, packet 2 is lost in transmission. When packet 1 then 3,4,5 are received at the receiver, the receiver will discard packet 3,4,5 entirely because packet 2 is missing.
   - Here we note that well received packets were dropped because one was missing. This means that those dropped packets will need to be retransmitted unnecessarily (wasting bandwidth and being impolite to network).

Selective Repeat will acknowledge packets 3,4,5 despite them being out of order and send acknowledgements for 3,4,5. The sender, after incurring timeout while waiting for ack for packet 2, will retransmit only packet 2. After the sender gets the ack for packet 2, it can move the window start to 6.

In selective repeat, unlike with Go back N, the receiver had actually buffered packets 3,4,5 and after receiving packet 2 flushed packets 2,3,4,5 up to the transport layer buffer for consumption by the application process.

With this motivating example, we see how Selective Repeat prevented the sender from resending correctly received packets (and reducing unnecessary network traffic in comparison to Go-Back N).

10. List Three Kinds of Routing algorithms and their application scenarios

Three kinds of routing algorithms are:

- Link State algorithm
- Distance Vector algorithm
- Hierarchical Routing algorithm

The Link State algorithm may be used, in the context of networking, for determining the best route from one node to another within an Autonomous system (AS). The Link State algorithm is facilitated by having each node in an AS send information about its neighbors to every other node such that all nodes then get the complete topology of the AS. This is used in OSPF - Open Shortest Path First.

To set forwarding tables where we do not have access to the entire topology, Distance Vector algorithms can be used. The means in which this can be facilitated is by having nodes propagate information about their distance vector estimates to their neighbors (upon these dv estimates changing).

Where it is unfeasible for different autonomous systems to send information such that all nodes in every AS can derive the complete topology of the internet, we can use Hierarchical routing algorithms.

11. List the key Difference between TDMA and FDMA

Both fall under the umbrella of circuit-switched networks.

In frequency-division multiple access, a link’s frequency spectrum is divided into different frequencies and those divisions service different connections.

In time-division multiple access, a channel is divided into time frames and each frame that is sent over the link is divided into multiple time slots where each time slot is allocated to a different connection.

12. List two different IP support protocols and their Key Functions:

Two IP protocols and their key functions are:

DHCP - The Dynamic Host Configuration Protocol is used to dynamically allocate an IP address to a host looking to join a subnet. A DHCP server typically exists on the subnet which will offer an IP address to a host that requests it through a broadcast request.

RARP - RARP is used to determine an IP address (at the network layer) for a given MAC address (at the link layer).

13. Suppose a TCP sender transmits a packet with a starting sequence number 1001 and data size 1500 bytes. What is the ACK number on the receiver’s (positive) acknowledgement for this packet?

- Assume that a previous segment has been sent since TCP uses sequence numbers in reliable data transfer to indicate the offset of the file/message that it wants to send (in pieces).
- The acknowledgement number sent back is the sequence number of the next byte that the receiver needs.

So Receiver already has 1000 bytes and the sender is sending it data starting at byte offset 1001 with 1500 bytes following

Note Receiver only has until offset 1000. It doesn’t have that 1 pseudo byte that it presumably sent in the ack previously.

1000 bytes + 1500 received = 2500 bytes.
The next byte the receiver needs is 2501.

Therefore the receiver sends 2501 as the ACK number for the positive acknowledgement of this received packet.

14. List Three Key Differences between MAC and IP addresses.

- An IPv4 address is 32 bits whereas a MAC address is usually 48 bits.

- A MAC address, unlike IP address, does not depend upon the subnet in which it is connected. That is, a MAC address is portable while an IP address is not.

- An IP address is used to provide logical communication between hosts in the network layer whereas a MAC address is used to provide communication between machines in the link layer. That is, an IP address is used for network layer (3) forwarding whereas a MAC address is used to link layer forwarding (layer below network layer).

15. List both one difference and one similarly between a Switch and a Router.

One similarity is that both a switch and a router are considered packet switches. As such, they both perform forwarding functions for packets.

One difference is that a switch only operates at the Link Layer whereas a Router may operate at the Link Layer and the Network Layer.

16. List the Key Difference between a Home Agent and a Foreign Agent:

Within the context of mobility, a mobile host has a home network. This home network (the permanent home to which the mobile host belongs) contains a home agent. Different mobile networks that the mobile device may be within are called visited networks and those networks contain a foreign agent akin to the home agent.

Since the mobile device's IP address may change (care of address), **the home agent is the first point of contact** to be used in indirect routing to relay a message to a foreign agent which will then relay it to the mobile device.

The other key distinction is that a foreign agent is an agent not within the mobile device's home network. Foreign agents register with home agents when a mobile device enters the foreign agent's home network.

17. Why does CSMA/CD perform poorly in wireless networks?

Wireless networks use CSMA/CA (collision avoidance) instead of CD (collision detection) because retransmitting a larger frame is costly (after detecting collision) and collision detection while transmitting in wireless interfaces is not as easy as in physical interfaces. And so it is preferable to use a locking mechanism scheme like Collision Avoidance to prevent collisions altogether.

18. List the key differences between (i) 2G and 3G as well as (ii) 3G and 4G LTE Cellular Network

i) 3G supports transfer of DATA and Voice whereas 2G only supports the transfer of Voice.

ii) GPRS stands for General Packet Radio Service and 3G uses Serving GPRS nodes and Gateway GPRS nodes to operate the DATA transfer parallel (but separate) to Voice. 4G LTE does not have any distinction between DATA and Voice and so it doesn’t operate those two separately.

19. Why do we need a Public Key Encryption Algorithm?

My public key K^+ can be distributed to Alice who wants to securely send me a message. Since Alice and I do not already have an established Symmetric key to encrypt/decrypt data, how can we communicate with confidentiality?

Premise 1: I have a public key K^+.
Premise 2: I have a private key K^- such that $K^-(K^+(m)) = m$
Premise 3: Alice knows m

In order for Alice to send me a message m (in this case a Symmetric key we'll use later), she will need to use my public key to encrypt that message such that only I may be able to decrypt it (premise 2).

Alice requires the use of a Public Key Encryption Algorithm such that she can take my public key K^+ and apply the Encryption algorithm using K^+ on m so that only I may be able to decrypt it using my secret key. This Encryption Algorithm facilitates the first step in Alice being able to confidently tell me message m. This algorithm needs to be standard such that everyone who uses it to encrypt message m2 using my public key can share the output with me and that K^-(that output) is the same each time.

20. What are the Network Support Approaches for Multimedia Content?

There are three network support approaches for Multimedia Content:
Approach 1 - Predicting how much link capacity to deploy

- In this approach, we try to solve the problem of providing good performance by upgrading hardware to improve the bandwidth.

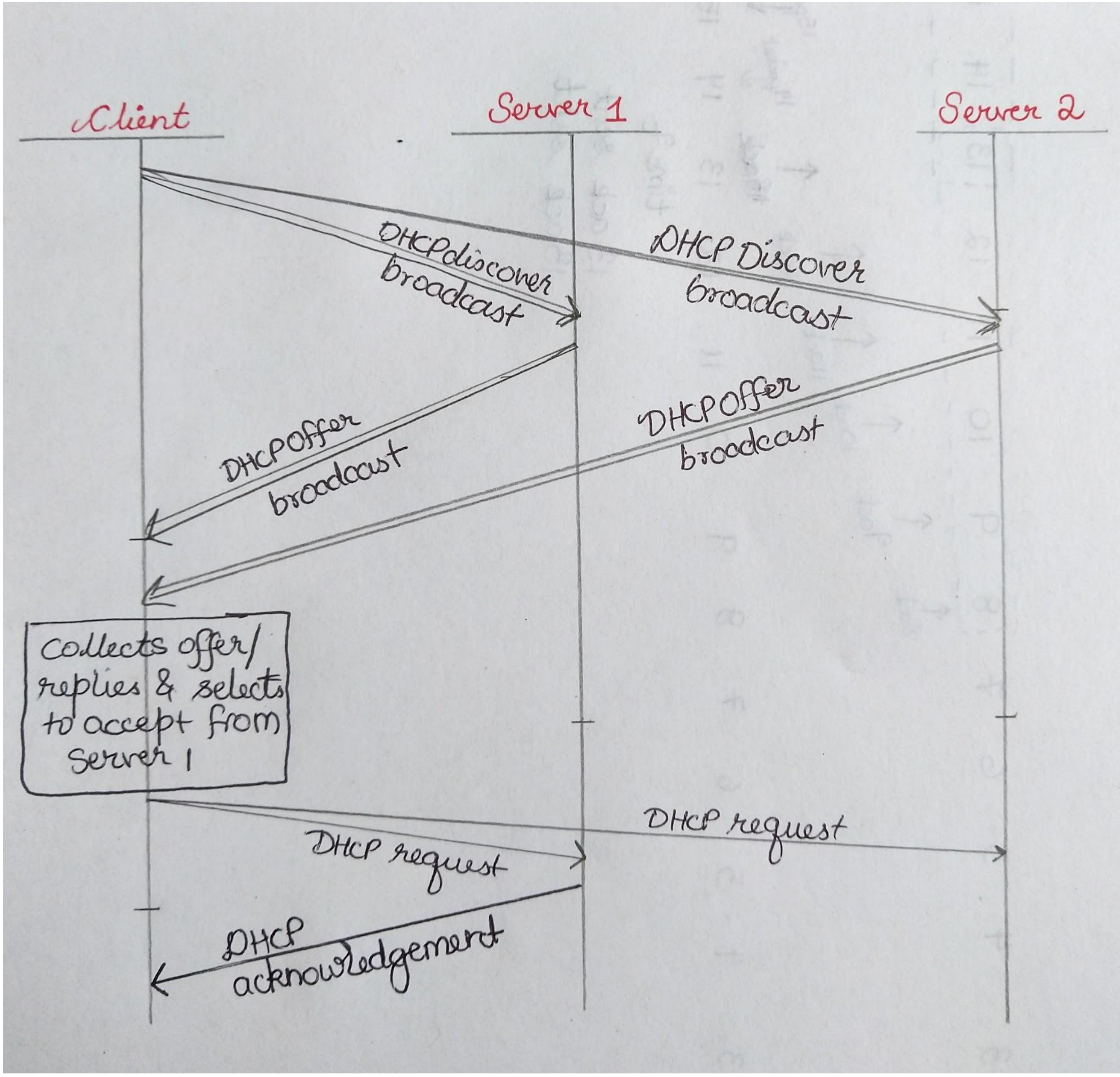
Approach 2 - Treating one class of traffic differently than another

- In this approach we give certain classes of service like Video or Audio more preference of HTTP requests since HTTP requests may drown out Audio.

Approach 3 - Providing guarantees of connection's quality of service

- We do as ^ suggests by reserving resources for a connection after we accept its request for a certain quality of service. The way we may do this is through circuit switching (TDMA for example).

21. DHCP IP address allocation, Client chose Server 1



22.

i) A is transmitting to B.

1) D cannot hear B’s CTS, if D transmits -> collision at B. D is a hidden terminal
2) C can hear B’s CTS. So C won’t transmit. C is none.

ii) B is transmitting to A.

1) D can’t hear B, D can’t affect A. D is neither.
2) C can hear B but can’t reach A. So if C doesn’t transmit to D if it can, it is waiting needlessly. C is an exposed terminal.

iii) A is transmitting to C.

1) B hears RTS and CTS. B is none.
2) D doesn’t hear A but can create collisions at C, D is a hidden terminal.

iv) D is transmitting to B

1) C hears RTS and CTS. C is none.
2) A doesn’t hear D but can create collisions at B, A is a hidden terminal.

v) C is transmitting to B

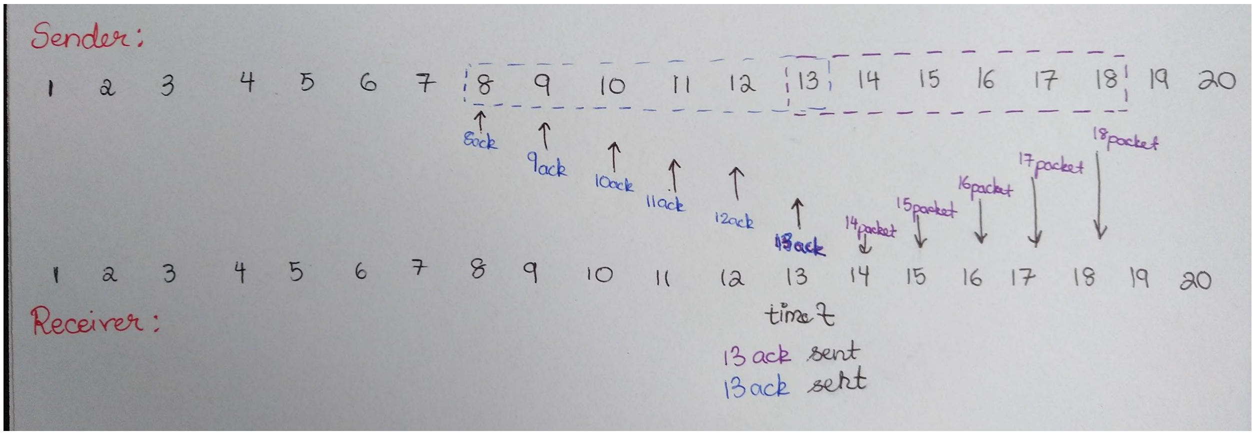
1) D hears C but not B. If D sends -> collision at B. D is a hidden terminal.
2) A hears CTS from B, so it doesn’t transmit. A is none.

vi) B is transmitting to C

1) A hears B but not C. If A sends -> collision at C. A is a hidden terminal.
2) D hears CTS from C, so it doesn’t transmit. D is none.

23. Go Back N protocol question:

Diagram to pictorially represent my reasoning:



We need a range [min, max] of sequence number of packets that could be in the sender's window at time t.

In my diagram I have two cases, one in blue and one and purple. They are entirely different scenarios and are uncorrelated.

In go-back N protocol, the acknowledgements are cumulative.

Blue Case to Find minimum:

- At time t it could be the case that the cumulative ack for packet 8 is not yet received at the sender. Thus the minimum window sequence number can be 8. This can happen through events where the window moved forward by 6 after receiving cumulative ack for 7 (and 7 was the last packet sequence in the previous sender window).

Purple Case to Find Maximum:

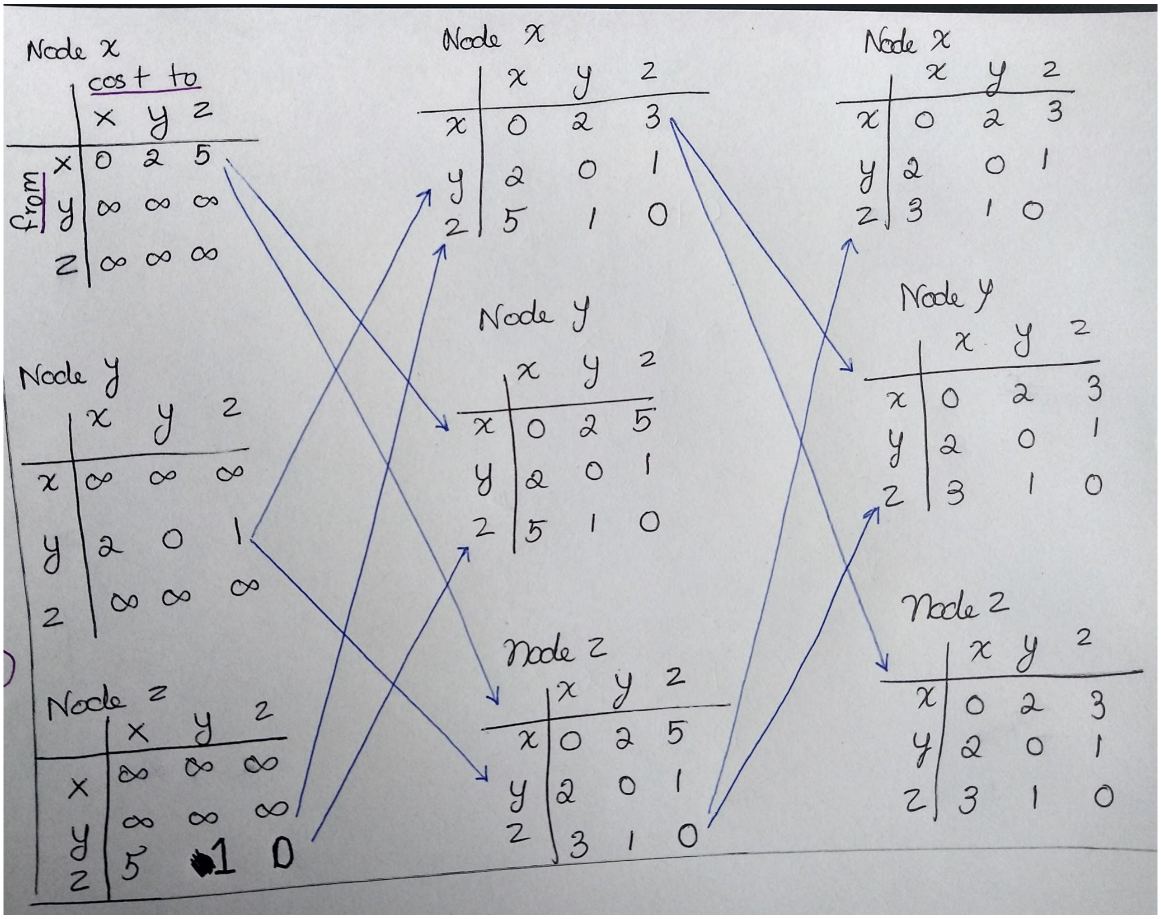
- Before time t, let us assume that cumulative ack was received for 12. Then the sender transmitted packet 13,14,15,16,17,18 from its new window that moved forward by 6. At time t, the cumulative ack for 13 is sent because that packet arrived at the receiver before 14,15,16,17,18 packets.

The range of possible sequence number of packets in the sender's window at time t therefore fall between 8 and 18 inclusive. That is, [8, 18].
From this range we may select any contiguous window of size 6 (for example: [9,10,11,12,13,14] or [12,13,14,15,16,17]).

24. Dijkstra’s shortest path algorithm

N’	D(B) p(B)	D(C) p(C)	D(D) p(D)	D(E) p(E)	D(F) p(F)
A	5, A	8, C	13, A	∞	34, F
AB		7, B	13, A	∞	34, F
ABC			13, A	8, C	34, F
ABCE			10, E		17, E
ABCED					17, E
ABCEDF					

25. Distance Vector Algorithm



Node x

cost to

from

	x	y	z
x	0	2	5
y	∞	∞	∞
z	∞	∞	∞

Node y

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

Node z

	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	5	1	0

Node x

	x	y	z
x	0	2	3
y	2	0	1
z	5	1	0

Node y

	x	y	z
x	0	2	5
y	2	0	1
z	5	1	0

Node z

	x	y	z
x	0	2	5
y	2	0	1
z	3	1	0

Node x

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

Node y

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

Node z

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

26. TCP connection slow start problem

Let CW be the congestion window.

- CW = 1; packet sent: 1
- CW = 2; packet sent: 2, 3
- CW = 4; packet sent: 4, 5, 6, 7
- CW = 6; packet sent: 8, 9, 10, 11, 12, 13
- Retransmit 6 and 7
- CW = 1; packet sent: 14
- CW = 2; packet sent: 15, 16
- CW = 4; packet sent: 17, 18, 19, 20
- CW = 8; packet sent: 21, 22, 23, 24, 25, 26, 27, 28
- CW = 16; packet sent: 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40

10RTTs are required as seen by 10 bullet points.

27. 13 objects RTT Calculation:

- Let RTT_0 refer to the round trip time between web browser and web server.
- I explain my reasoning in the curly braces {}

(i). Using persistent HTTP connection without pipelining:

Total Time = RTT1 + RTT2 + RTT3 + RTT4 + RTT5 + RTT6 {dns query for web server IP}

+ $1*RTT_0$ {open persistent TCP connection}
+ $1*RTT_0$ {request & receive HTML file}

+ $1*RTT_0$ {request & receive 1 small object} * 13 {thirteen objects}

= RTT1 + … + RTT6 + $1RTT_0$ + $1RTT_0$ + $13RTT_0$

Total time = RTT1 + … + RTT6 + $15*RTT_0$

(ii). Using non-persistent HTTP with 5 parallel connections:

Total Time = RTT1 + RTT2 + RTT3 + RTT4 + RTT5 + RTT6 {dns query for web server IP}

+ $1*RTT_0$ {open non-persistent TCP connection}
+ $1*RTT_0$ {request & receive HTML file}

+ $1*RTT_0$ {open 5 non-persistent TCP connections in parallel}
+ $1*RTT_0$ {request & receive 5 small objects in parallel, have 5}

+ $1*RTT_0$ {open 5 non-persistent TCP connections in parallel}
+ $1*RTT_0$ {request & receive 5 small objects in parallel, have 10}

+ $1*RTT_0$ {open 3 non-persistent TCP connections in parallel}
+ $1*RTT_0$ {request & receive 3 small objects in parallel, have 13}

Total time = RTT1 + … + RTT6 + $8RTT_0$

28.

Note: 1RTT = 100 milliseconds

(i) How long (in ms) does it take for the first ACK to arrive at the TCP sender after its first packet was transmitted?

Time = ½ RTT {sending the packet across link from Sender to rec}

+ 10^3 bits / 10^6 bps {transmission delay at Receiver to transmit ACK packet where L = 10^3 bits and R = 10^6 bps}

+ ½ RTT {sending ACK packet across link from Rec to sender}

= 1 RTT + 10^{-3} seconds = 200 milliseconds = 2RTT

(ii). Suppose the TCP sender transmits 10 packets, unacknowledged, before the first ACK is received. What is the window size in KBits?

- We know the window size in packets = 10.
- 10 packets * 1 KB per packet. = 10 KBits

(iii). What is the throughput of the TCP sender above, in KByte/s?

- Throughput = File size / Time Receive = 1 KBits / (time to receive first packet)
- Throughput = 1KB / (transmit: 10^3 bits / 10^6 bps + 1/2RTT)
- = 1KB / (1 ms + 50ms)

29. 1,2,3 = i dont know , ran out of time.

30. Fragmentation

i) I assume that the Transmission rate at B and C is equal and that propagation delay = 0.

LINK	LENGTH	ID	FLGS	OFFSET
AB	1420	1234	0	0
BC	700	1234	0, DF=0, MF=1	0
CD	300	1234	0, DF=0, MF=1	0
CD	300	1234	0, DF=0, MF=1	280
CD	140	1234	0, DF=0, MF=1	560
BC	700	1234	0, DF=0, MF=1	680
CD	300	1234	0, DF=0, MF=1	680
CD	300	1234	0, DF=0, MF=1	960
CD	140	1234	0, DF=0, MF=1	1240
BC	60	1234	0, DF=0, MF=0	1360
CD	60	1234	0, DF=0, MF=0	1360

ii) The fragments are reassembled at the destination end system. They should not be reassembled anywhere else.

Reassembly at a receiver indicates that the receiver wants a whole datagram. With a whole datagram, the transport layer contents (segment) inside of it may be extracted.

> But no receiver other than the destination end system needs access to those segments because that receiver is simply a packet switch in the network that needs to route the datagram (even if it is fragmented) forward to the next packet switch.
>
> That is, intermediary packet switches do not need to read transport layer contents by piecing together fragments into one whole datagram and thus should not reassemble in the first place.

Reassembly at every receiver would result in redundancy through a series of fragmentations and reassemblies.