Internet Technology

08r. Exam 2 & Part of Exam 3 Review

Paul Krzyzanowski

Rutgers University

Spring 2016

Exam 3 – Question 1a

Suppose that Dijkstra's algorithm computed the following results for network costs starting from node *a*:

D(b), p(b)	*D(c), p(c)*	*D(d), p(d)*	*D(e), p(e)*	*D(f), p(f)*
2, *a*	1, *a*	2, *c*	4, *d*	3, *d*

(a) What is the full path from node *a* to node *f*? Express it as an ordered list of nodes starting from *a*.

D(v) = cost of least-cost path to node v

p(v) = the node before v on the least-cost path to v

- We have to work backwards
- Shortest distance from node *a* to node *f* is 3
  - Previous node on that path is *d*
- The previous node on the shortest path to *d* is *c*
- The previous node on the shortest path to *c* is *a*
- Shortest path from *a* to *f:* ***a c d f***

Exam 3 – Question 1b

Suppose that Dijkstra's algorithm computed the following results for network costs starting from node a:

D(b), p(b)	*D(c), p(c)*	*D(d), p(d)*	*D(e), p(e)*	*D(f), p(f)*
2, *a*	1, *a*	2, *c*	4, *d*	3, *d*

(b) Routing tables store first hops to a destination. What is the routing table at *a*? If some destination *n* is directly connected to *a*, write "*n*" for the next hop.

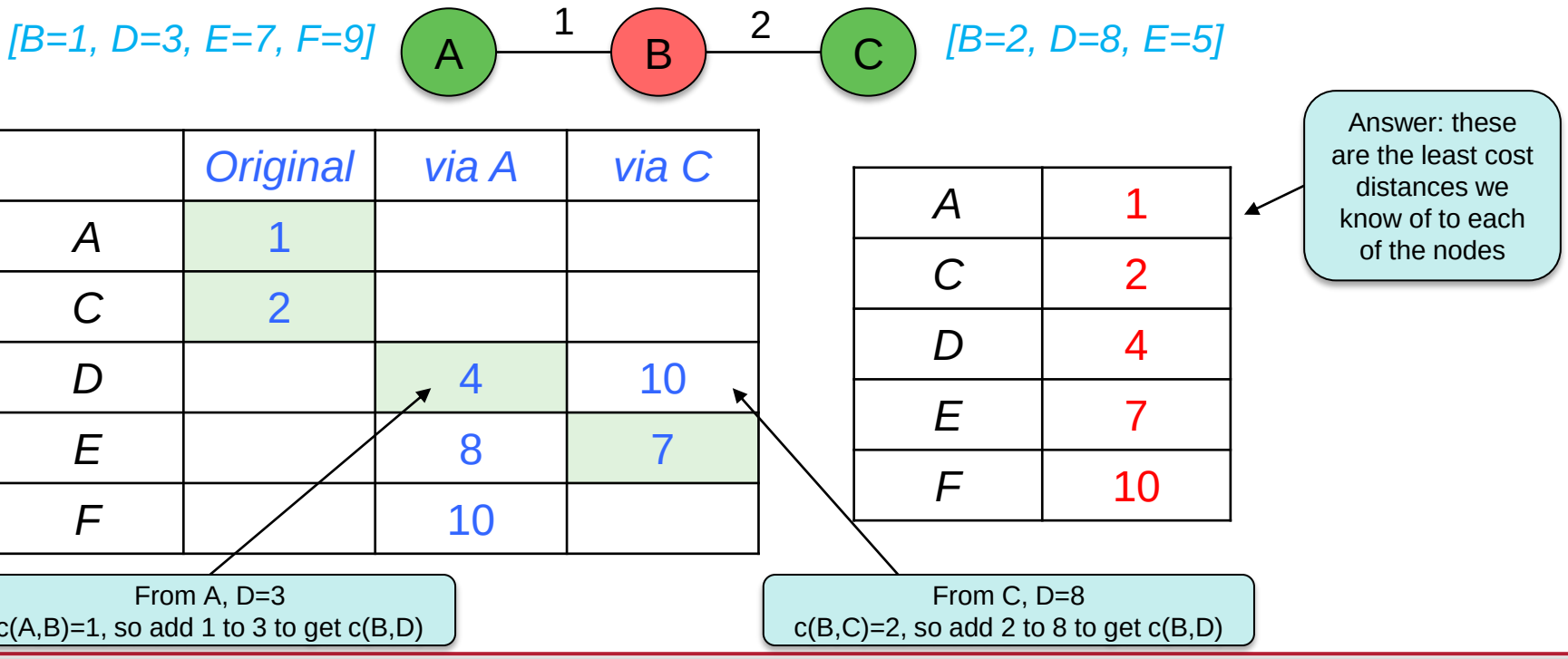
- Just as in part (a), work backwards
  - This time we don't need the entire path – just the node *n* where *p(n) = a: this is the first hop from a*

b	*b*
c	c
d	c
e	c
f	c

Exam 3 – Question 2a

Node *B* has nodes *A* and *C* as neighbors. *B*'s initial distance vector is [*A*=1, *C*=2], identifying the cost of getting to *A* as 1 and the cost to *C* as 2. *B* receives a distance vector from *A* that contains: [*B*=1, *D*=3, *E*=7, F=9]. Shortly after that, *B* receives a distance vector from *C* that contains [*B*=2, *D*=8, *E*=5].

(a) After *B* receives these two vectors, what is its view of least-cost distances from *B* to the following nodes? Express the distance as a single number.



	Original	*via A*	*via C*
A	1		
C	2		
D		4	10
E		8	7
F		10	

A	1
C	2
D	4
E	7
F	10

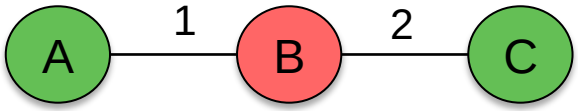
Answer: these are the least cost distances we know of to each of the nodes

From A, D=3
c(A,B)=1, so add 1 to 3 to get c(B,D)

From C, D=8
c(B,C)=2, so add 2 to 8 to get c(B,D)

Exam 3 – Question 2b

(b) After *B* updates its distance vector and sends it to *C*, what is *C*'s distance vector? Express the distance as a single number.



Answer: C chooses a route through B if the cost to a node is lower than it previously had

A	1
C	2
D	4
E	7
F	10

B's distance vector

Add c(B,A)=2

	Original	*from B*
A		3
B	2	
D	8	6
E	5	9
F		12

at node *C*

A	3
B	2
D	6
E	5
F	12

C's new distance vector

Exam 2 – Question 6

6. Why might using UDP instead of TCP be appealing?

(a) UDP supports much larger segment sizes. – not really: just 12 bytes

(b) UDP uses a checksum to ensure data integrity. – so does TCP

(c) UDP segments are sent immediately.

(d) UDP segments are sequenced to arrive in order – no, they're not

Question 7

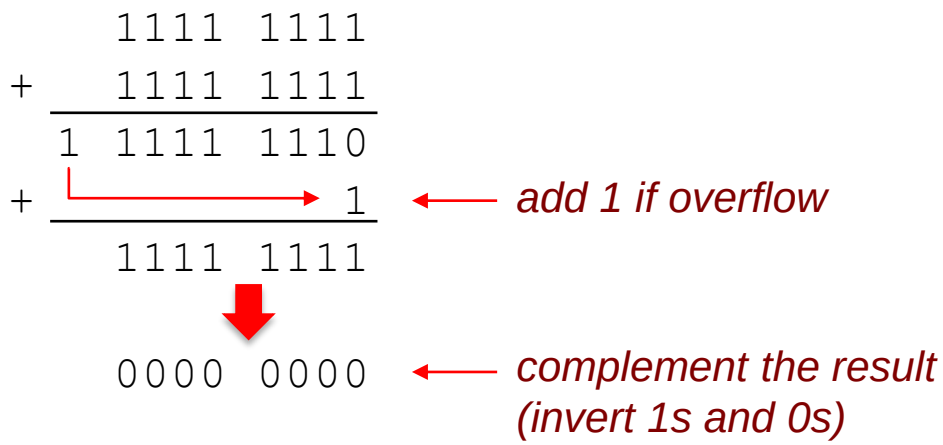
7. What is the eight-bit Internet checksum of the two bytes {0xff, 0xff}
   (in binary: 1111 1111, 1111 1111)?

(a) 0x00 (0000 0000)

(b) 0x01 (0000 0001)

(c) 0xfe (1111 1110)

(d) 0xff (1111 1111)



Question 8

8. Sequence numbers in a stop-and-wait protocol are useful for:

(a) Ensuring data is delivered in the correct order.

(b) Distinguishing a new message from a duplicate.

(c) Keeping track of the total number of messages sent.

(d) Matching the receive window size with the transmit window size.

The stop-and-wait protocol transmits one packet at a time
– *there is no out-of-order delivery!*

A sender will retransmit a packet if:

- the receiver got a corrupt packet (and sends the ACK for the previous seq #)
- The receiver's ACK got corrupt (can't make sense, so re-transmit)

Question 9

9. You have a 1 Gbps network. The packet size is 1,000 bits and there is a 10 ms round-trip time between two hosts.
What is the approximate network utilization with a stop-and-wait protocol?

(a) 0.01%

(b) 1%

(c) 10%

(d) Almost 100%

= 0.010 s = 1×10^{-2} s

Network utilization = (amount of traffic on the network) ÷
(amount of traffic that the network can support)

With stop-and-wait: one packet + ACK = 1 packet per 10 ms

How much can the network support in 10 ms?

1×10^9 b/s ÷ 1,000 b/packet = $1 \times 10^{9-3} = 1 \times 10^6$ packets/s

1×10^6 packets/s × 1×10^{-2} s = $1 \times 10^{6-2} = 10^4$ packets per 0.010 s

Network utilization = (1 packet) ÷ (10^4 packets) = 0.0001 = 0.01%

Question 10

10. With Go-Back-N, a segment can be deleted from the send buffer only when:

(a) It has been successfully received by the receiver.

(b) It and all prior segments have been successfully received by the receiver.

(c) All segments in the window have been successfully received by the receiver.

(d) The countdown timer expires after sending all segments in the window to the receiver

The question does *not* ask you about the protocol but about the behavior.

With GBN, a receiver *discards* out-of-sequence segments.

It sends cumulative ACKs back and a sender can delete segments once they have been *sequentially* received.

(a): No – might have been received out of sequence (& discarded)

(c): overkill

(d): that is an indication that segments were *not* received

Question 11

11. In a Go-Back-N protocol, a sender sends segments 5, 6, and 7. It gets back acknowledgements for segments 5 and 7. The acknowledgement for segment 6 does not arrive. What happens?

(a) The sender moves the base of its window to segment 8.

(b) The sender retransmits segment 6 upon getting the acknowledgement for segment 7.

(c) The sender retransmits segment 6 upon getting a timeout.

(d) The sender advances the base of its window to segment 6 and waits

This asks about the workings of the protocol.

GBN uses *cumulative ACKs.*

Getting ACK #7 means *all* segments up to and including 7 were received (the ACK for 6 must have been dropped or damaged).

The sender can slide the window past 7.

Question 12

12. Path MTU (Maximum Transmission Unit) discovery relies on:

(a) The ability to send a query to a router for its MTU size.

(b) The ability to turn off fragmentation.

(c) Committing to an unchanging end-to-end route.

(d) Negotiating with the receiver to pick the maximum MTU that both can support

The sender sets the DF (Don't Fragment) bit in the IP datagram.

If the size is too large for a router's outgoing link, it will not fragment the datagram.

Instead, it will send back an ICMP *Destination Unreachable* message type (3) with a code of *Fragmentation required and DF flag set* (4).

Question 13

13. TCP without the Selective Acknowledgement (SACK) option behaves like a Go-Back-N protocol except that:

(a) Only the earliest unacknowledged segment is sent upon timeout.

(b) All unacknowledged segments are sent upon timeout.

(c) All segments in the current window are sent upon timeout. *– this is GBN*

(d) Only the last unacknowledged segment is sent upon timeout. *– no, first*

Note that (b) and (c) are essentially the same. ACKs are cumulative, so the window will slide past a segment upon receiving its ACK. What's left in the window are unacknowledged segments, some of which perhaps were not transmitted yet.

Question 14

14. A TCP header does not have this field:

(a) Source address

(b) Destination port

(c) Sequence number

(d) Checksum

The source address is part of the IP header (network layer, not transport layer).

Question 15

15. Host A sends the following consecutive TCP segments to host B:

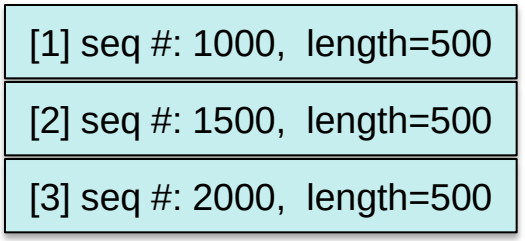
(1) sequence #1000, 500 bytes;

(2) 500 bytes;

(3) 500 bytes.

Host B receives only segments #1 and #3.

What ACK does it send upon receiving segment #3?



(a) 1000

(b) 1500

(c) 2000

(d) 2500

TCP uses cumulative ACKs. ACK# = earliest byte # that the receiver needs

After receiving (1), the receiver sends ACK = 1500 (needs byte 1500)

After receiving (3), the receiver still needs byte 1500

It stores seq # 2000 but sends back the cumulative ack, ACK=1500

Question 16

16. Suppose host B now receives segment #2 (after it receives segment #3). What ACK does it send back to host A?

(a) 1000

(b) 1500

(c) 2000

(d) 2500

[1] seq #: 1000, length=500
[2] seq #: 1500, length=500
[3] seq #: 2000, length=500

TCP caches received segments (unless there is no more room)

After question #15, it had

bytes 1000…1499 (which it could deliver to the application)

bytes 2000…2499 (which is being held in the receive buffer)

When it gets #2, it gets bytes 1500…1999. Now it has bytes through 2499.

It sends back the cumulative ACK: 2500 is the next byte it needs.

Question 17

17. TCP's Slow Start:

(a) Provides an additive (linear) increase in transmission rate.

(b) Delays data transmission until a TCP connection is established.

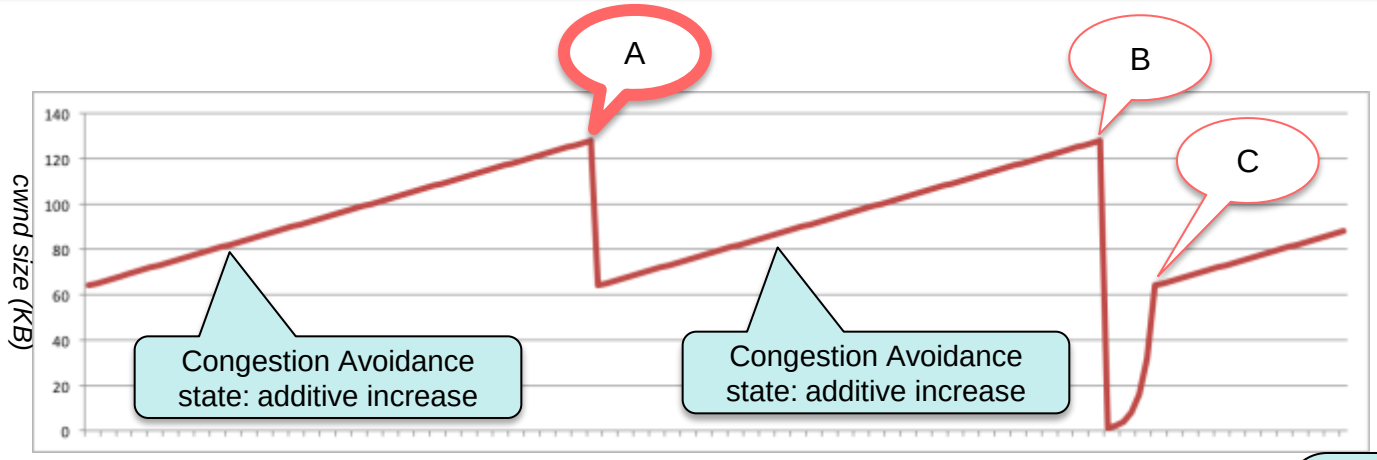
(c) Provides a multiplicative decrease in the transmission rate.

(d) Provides an exponential increase in transmission rate.

TCP Slow Start *prevents* this slow ramp at startup by providing an *exponential* increase in cwnd size.

The congestion window starts at one MSS and increases by 1 MSS with each received ACK, doubling each RTT.

Question 18



N.B.: The chart labels were a little bit off and the in-class instructions were to ignore them and look at the shape of the graph.

18. The chart above shows TCP's congestion window over time. What happened at (A)?

(a) A retransmission timer expired.

(b) Three duplicate ACKs were received.

(c) The maximum window size was reached.

(d) The ssthresh value was reached

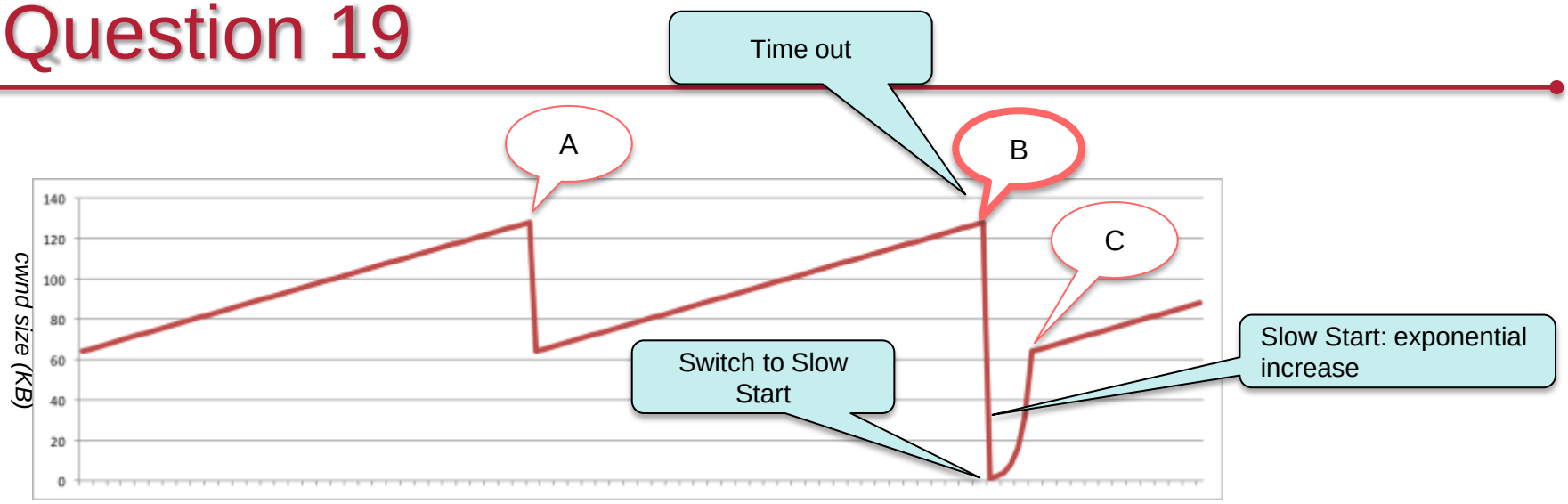
3 duplicate ACKs: assume that a segment was lost → *Fast Recovery state*

Cut cwnd in half.

Do a Fast Retransmit (don't wait for RTO timer)

cwnd = congestion window size # bytes (⇒# packets) of unacknowledged packets that may be in transit at one time

Question 19



19. What happened at (B)?

(a) A retransmission timer expired.

(b) Three duplicate ACKs were received.

(c) The maximum window size was reached.

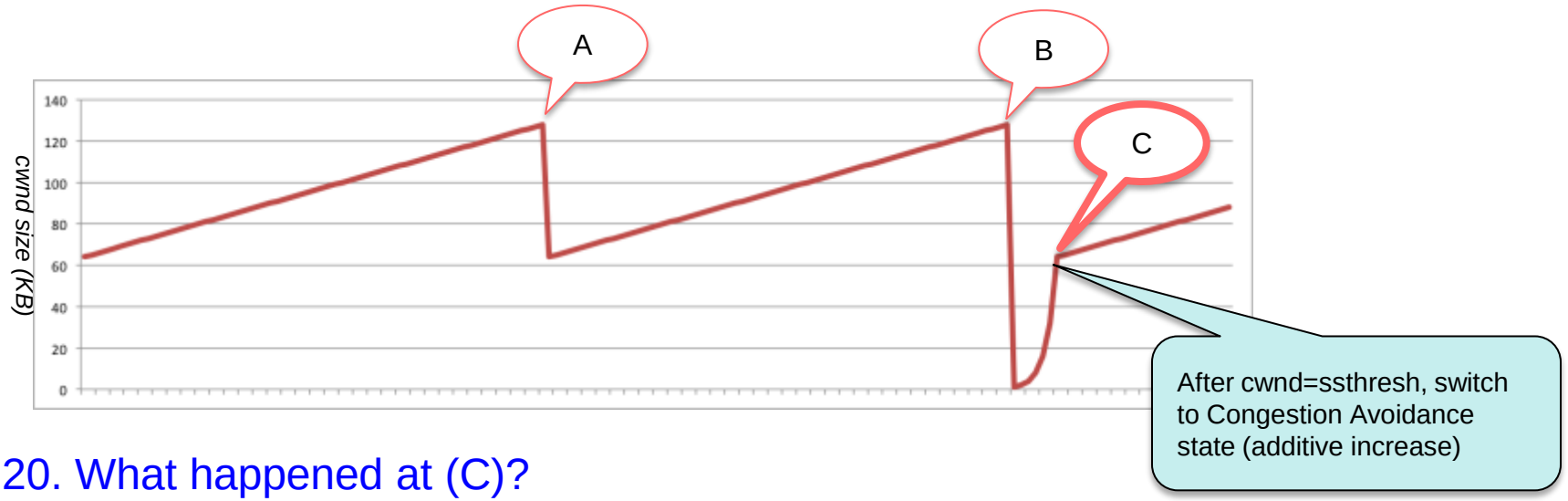
(d) The ssthresh value was reached

An RTO timer expiring means that we go back to the Slow Start state.

ssthresh = “slow start threshold”

ssthresh is set to 50% of the current cwnd and cwnd is set to 1.

Question 20



20. What happened at (C)?

(a) A retransmission timer expired.

(b) Three duplicate ACKs were received.

(c) The maximum window size was reached.

(d) The ssthresh value was reached

An RTO timer expiring means that we go back to the Slow Start state.

ssthresh is set to 50% of the current cwnd and cwnd is set to 1.

Question 21

21. TCP Fast Retransmit transmits:

(a) One segment without waiting for a timeout.

(b) Every unacknowledged segment if it detects the loss of one segment.

(c) A segment immediately upon getting a negative ACK from the sender.

(d) Retransmits a segment upon getting a retransmission timeout

Three duplicate ACKs = assume the segment with that seq # was lost

Do not wait for RTO to expire

Send the segment corresponding to the ACK number

Question 22

22. Which router architecture does not require replicating the forwarding table?

(a) Conventional shared memory.

(b) Shared memory with distributed CPUs.

(c) Shared bus, no shared memory.

(d) Switched data path.

Architectures (b) – (d) have a CPU in each line card rather than use a central CPU for forwarding datagrams.

Each CPU needs a copy of the forwarding table to avoid contention in accessing one in shared memory.

Question 23

23. Which field is NOT in an IPv4 header?

(a) Header length.

(b) Time to live (hop limit).

(c) ACK number.

(d) Header checksum.

IP does not provide reliable delivery.

Hence, there is no need for acknowledgements.

Question 24

22. The minimum condition for an IP receiver to know the size of a reassembled IP datagram is when:

(a) It has received all fragments of the datagram.

(b) It has received the last fragment of a datagram, but not necessarily all fragments.

(c) It has received the first fragment of a datagram.

(d) When it has received any fragment since all fragments contain the full datagram length

Each datagram tell us:

- Are there more fragments (MF flag)?
- Offset of the fragment (÷8)
- Total length of the datagram

If we know the offset of the last fragment and the length of the datagram, we know the length of the full datagram.

total data size = (offset*8) + (total_length - header_length)

Question 25

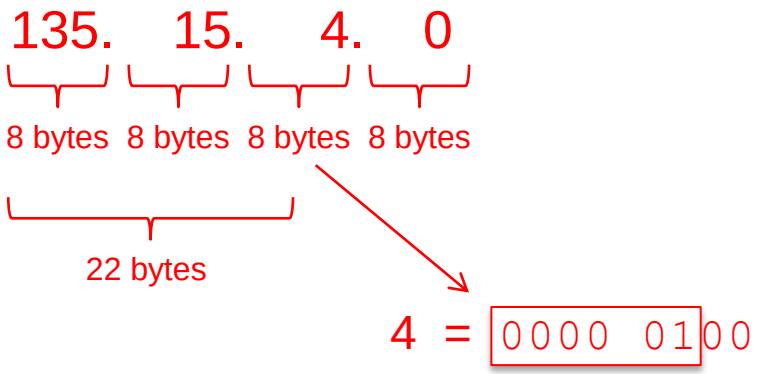
25. Which is a valid IP address on a 132.15.4.0/22 network?

(a) 132.15.10.7

(b) 132.15.9.8

(c) 132.15.8.9

(d) 132.15.7.10



We need a third digit whose bits match = `000001xx`

(a) 10 = 0000 1010 – no

(b) 9 = 0000 1001 – no

(c) 8 = 0000 1000 – no

(d) 7 = 0000 0111 – yes!

Question 26

26. In DHCP, the Dynamic Host Configuration Protocol, the DHCP server communicates back to the client via a(an):

(a) UDP message to the client DHCP process.

(b) UDP broadcast to all machines on the LAN

(c) TCP message to the client DHCP process.

(d) ICMP message to the client machine

At this point, the client does not have an IP address. Only a broadcast message can reach it.

Question 27

27. NAT, Network Address Translation, requires that all the machines in the internal network:

(a) Have private addresses that are not valid on the wide-area Internet.

(b) Share the same IP address.

(c) Contact a DHCP server to get an on-demand address assignment.

(d) Contact a NAT server to get an address assignment.

NAT converts between private internal addresses and one or more external addresses.

Question 28

28. The Internet Control Message Protocol (ICMP) is likely to be used to:

(a) Get a unique IP address.

(b) Choose one of several available routes.

(c) Find out that the time-to-live expired on an IP packet.

(d) Define a route for an IP datagram

(a) No. That would be DHCP (Dynamic Host Configuration Protocol).

(b) No. ICMP does not allow a host to choose a route.

(c) Yes. ICMP type = *Time Exceeded* (11), code = *TTL expired in transit* (0)

(d) No. ICMP does not allow a host to specify a route.

Question 29

29. Which header does not have a checksum in it?

(a) IPv4

(b) IPv6

(c) TCP

(d) UDP

(a) IPv4: header checksum

(c) TCP: TCP header + IP pseudo header checksum

(d) UDP: UDP header + IP pseudo header checksum

IPv6 designers decided to omit a checksum in the IPv6 header, arguing that it is already computed at the link layer (e.g., ethernet) and key fields of the IP header are used in UDP and TCP checksums

Question 30

30. IPv6 does not have which of the following IPv4 headers (same or similar names)?

(a) Protocol version. – *identifies the version of the protocol (6)*

(b) Time to live (hop limit). – *decremented by 1 at each router; kills loops*

(c) Source IP address. – *identifies who sent the datagram*

(d) Fragment offset.

Unlike IPv4, routers never fragment IPv6 datagrams.

The router will simply drop the packet and send back an ICMP error message

Nodes are expected to perform path MTU discovery (minimum transmission unit for the entire route).

The IP layer at the sender has the ability to fragment packets using optional (extra) headers in IPv6 that enable fragmentation

Question 31

31. ARQ (Automatic Request) protocols avoid the need to use ACKs by having a server automatically resend each segment.

True False

False.

This statement doesn't make sense since the sender needs some way of determining that a segment needs to be resent.

Question 32

32. A countdown timer is not needed in a protocol that uses ACKs and sequence numbers.

True **False**

False.

How will you know that a packet that you sent was never received at the destination?

With TCP, the ACK is a sequence # stating the next byte that we want expect to receive. If we get two ACKs with the same number, we assume that some segment was lost. In this case we did not need a timer but the mechanism only worked because:
(a) we had additional data to transmit
(b) the receiver received that additional data

Question 31-34

33. UDP allows different processes to send messages to the same socket at a server.

True False

True.

All UDP messages sent to a specific port number will be delivered to the socket that is reading messages from that port.

This differs from TCP, where a dedicated socket is created ONLY for listening – getting incoming connections. Once a TCP connection is accepted, a new socket is created that is dedicated only to that connection.

Question 34

34. Increasing the congestion window size increases TCP's transmission rate.

True False

True.

If we send more packets out over the network before waiting for acknowledgements,
We increase our network utilization and increase our effective transmission rate.

Question 35

35. The Congestion Avoidance state provides an additive increase to the transmission rate.

True False

True.

TCP uses the AIMD (Additive Increase, Multiplicative Decrease) principle
As long as messages are being delivered successfully, the window size increases by 1 MSS (maximum segment size) every RTT (round-trip time) interval

Question 36

36. TCP requires a Virtual Circuit network since it requires connection service at the network layer.

True **False**

False.

IP-based protocols do not use virtual circuit networks (such as ATM).

Virtual circuit networks have connection-aware routers, where a path is precomputed during connection setup and each message has a virtual circuit identifier to use that specific path.

With TCP, all knowledge of the “connection” is maintained at TCP driver in the client and server. IP datagrams are used for transmission.

Question 37

37. A router has to recompute the IP header checksum.

True False

True.

The header checksum includes a time-to-live (TTL) value, which is a decrementing hop count. Each time a datagram goes through a router, the TTL field is decremented. Once it reaches 0, the packet is discarded.

Since a field in the IP header changes each time the datagram goes through a router, the router will have to recompute the checksum with the updated TTL value.

Also - in the case of fragmentation, the router will create new datagrams and compute checksums for them.

Question 38

38. The control plane of a router is responsible for forwarding datagrams between ports.

True False

False.

The data plane of a router handles all the work of forwarding datagrams.

The control plane is the supervisory layer that runs the user interface, computes forwarding tables, and other things not related to the continuous flow of datagrams.

Question 39

39. Under NAT, two different computers in an organization may appear to an outsider as having the same IP address.

True False

True.

NAT permits multiple machines to share a single external IP address.

Internally, the machines will have distinct private addresses. A NAT router will translate the source address (and port) on each outgoing packet to the public address. It will translate the public destination address on incoming packets to the private address based on either forwarding rules or return messages from previously sent packets.

Question 38-40

40. A router that fragments an IP datagram that carries TCP data must replicate the TCP header onto each fragment.

True **False**

False.

IP is ignorant of higher layers: TCP & UDP headers are just data – part of the payload.

A fragmented IP packet needs to be reassembled before it can be passed to the higher layer (e.g., TCP).

The end