Network Systems (201600146/201600197), Test 3

March 22, 2019, 13:45–15:15

Answers

---

1. Addressing and IPv6

3 pt (a) B. B. C. C. A. A. C. B. C.

1 pt (b) A.

A tunneled packet contains not only the complete IPv6 packet, but also an IPv4 header of 20 bytes (assuming no header options). The total packet must not be larger than the MTU, so effectively 20 bytes fewer remain for the IPv6 packet.

1 pt (c) D.

1 pt (d) C.

1 pt (e) C.

In IPv4, routers do the fragmentation all by themselves; they don't need ICMP packets for that.
In IPv6, routers can't do fragmentation; instead, if they find a packet is too large, they send an ICMP message to the sender of that packet, so the sender can re-send it in smaller fragments. If ICMP packets are blocked, such a notification would never reach the sender, so the sender will never re-send the packet in appropriately sized fragments.

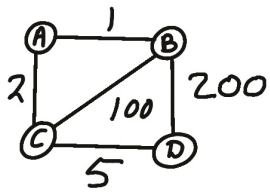
---

2. Routing

1 pt (a) (A, 0), (B, 1), (C, 2)

2 pt (b) (A, 0), (B, 1), (C, 2), (D, 7)

2 pt (c)



1 pt (d) 24.

First, A tells C that the cost is 5.
Then C tells A that the cost is 7.
Then A tells C that the cost is 9.
Then C tells A that the cost is 11.
... and so on ...
Then C tells A that the cost is 99.
Then A tells C that the cost is 101.
... and then C realizes its direct link to B, at cost 100, is cheaper.

The cost reported by C to A increases in steps of 4, from 7 to 99: that's a total of $\frac{99-7}{4}+1=24$ packets.

We also accepted 48, which would be correct if you asssume all nodes send updates every timeslot, rather than only if there's a change.

1 pt (e) D.

1 pt (f) B.

1 pt (g) A.

The statement is not true for LS simply because in LS *counting to infinity* does not occur (so there's nothing to be solved).
The statement is not true for DV either: *counting to infinity* does occur in DV but cannot be solved this way. It's more the other way around: the fact that a link cost becomes infinite (representing the failure of the link) *causes* or *starts* the problem.

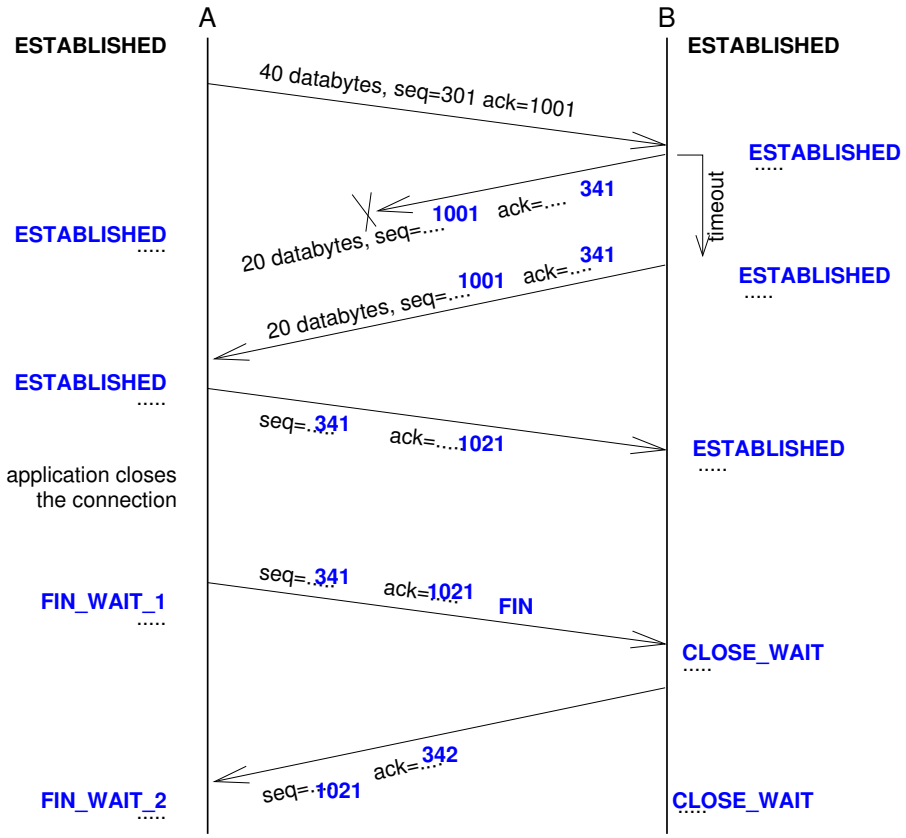
1 pt (h) C.

1 pt (i) C.

---

3. Transport layer protocols

5 pt (a)



Some notes:

- Until the application closes the connection, both sides are in the ESTABLISHED state. It was given that the connection was already set up before the diagram starts, so there's no need for SYN packets and associated states.
- Only the application at A closes the connection (this was given), so B does not yet send a FIN within this diagram.
- Be careful about how the seq and ack numbers change. Ack is always the next expected sequence number from the other side, so if the other side sends 40 bytes with sequence number 301, that packet contains bytes numbered 301...340, so next expected is 341.
- The FIN flag counts as if it takes 1 byte in the sequence number space, that's why the ack number in the last packet is 1 higher than the seq number in the previous packet (which had the FIN flag set).

1 pt (b) E.

1 pt (c) C.

This is a so-called "half open" connection.
Host A has already sent a FIN flag, so it cannot send any more new data.
Host B has not yet done so, so it can still send new data.

1 pt (d) C.

1 pt (e) B.

In such a URL, the port number on which the server is listening is specified explicitly, instead of using the default 80.

1 pt (f) C.

TCP always sets the ACK flag, except in the first SYN packet because then it does not yet know the other side's sequence number, so it cannot acknowledge anything.
Setting the ACK flag (and setting a correct number in the acknowledgement number field) even if there is no new data to be acknowledged, is useful in case an earlier packet which did acknowledge new data is lost.

1 pt (g) D.

Window scaling is needed to fully utilize a link when the bandwidth delay product is larger than 65535 (i.e., the largest possible receive window without window scaling).
PAWS is needed when more than 2^{32} bytes are transferred within the maximum time an IP packet can live, taken as 2 minutes.
So the situation where window scaling is useful without PAWS is when the bandwidth delay product is large, but the bandwidth is below 2^{32} bytes per 2 minutes.

1 pt (h) B.