Network Systems (201600146/201600197), Test 2

March 10, 2017, 13:45–15:15

Answers

1. Distance-vector routing

(a) {(A,0), (C,1), (D,200), (F,1)}

(or any other clear notation, e.g., a table)

(b) {(A,0), (B,5), (C,1), (D,101), (E,11), (F,1)}

(c) {(A,0), (B,5), (C,1), (D,21), (E,11), (F,1)}

(d) {(A,0), (D,21), (E,11), (F,1)}

(e) A and C will do a count-to-infinity, in each iteration increasing their estimate of the cost to reach F by 1, until it exceeds 10+10+100 (i.e., the cost of going to F via D and E). So about 120 iterations.

(f) Answer A.

(g) Answer B.

2. Addressing and routing

(a) Answers C and D.

Note that E has too many colon-separated blocks to be a correct IPv6 address notation.

(b) Answer C.

This can be found either by computing the HD ratios using the formula; or simply by recalling that with a larger network, a larger fraction of addresses is effectively left unused at the same HD ratio.

(c) Answer A.

The one with the higher HD ratio, since that's a measure for how troublesome address assignment is.
Unfortunately and unintentionally, I picked the numbers such that the HD ratios of both networks are already above the threshold for convenient assignment (about 0.87). Still, the one with the highest HD ratio is most in trouble.

(d) Answer D.

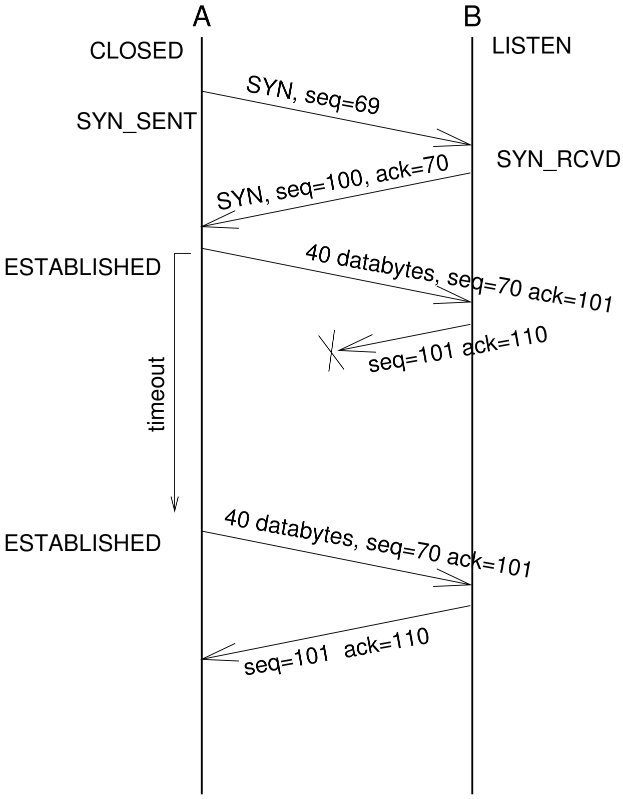
(e) Answer B.

3. Transport layer protocols

(a) Answer A.

(b) Answer B.

(c)



Note that the last packet from A is a retransmission (since it comes at the end of a timeout), not new data, hence the same sequence number.

The initial state of A can also be LISTEN according to the state-transition diagram, but this is rather rare: it would be the situation where the application first decides to listen for an incoming connection, and then changes its mind and decides to open the connection itself.

(d) Answer D and E.

Note that A is wrong: the TCP header has a field for the window *advertised by the receiver*, but the *effective* window that determines what the sender transmits next may well be less, depending on how much it has already transmitted. See the book.

(e) From all states except CLOSED and LISTENING, an arrow should be drawn to CLOSED with the tag "RST" (i.e., upon receipt of a segment with the RESET flag set, and transmitting nothing in response).

A packet with RST is sent by TCP if it receives a packet which it can't make sense of; e.g., after a host crashes and reboots, it may get a packet from a TCP connection that was active before the crash.

So if we receive a RST, we should simply forget about the connection entirely, i.e., go to the CLOSED state. Sending an acknowledgement doesn't make sense, since the other side doesn't understand this connection.

Some students tried to indicate when we should *send* and RST, but the question only asked for *receiving*.

Some students didn't seem to know that the diagram on page 3 of the exam (or p. 405 of the book) is the state-transition diagram. The diagram of question (c) is something totally different, variously called a timeline diagram, a packet exchange diagram, or a time-sequence diagram.

(f) No, that wouldn't make sense, since UDP doesn't have sequence numbers that can overflow, windows that can be too small, etc.