Computer Networks II

Curso 23/24 :: Exam 1

Escuela Superior de Informática

2025/04/01 18:31:03

This exam has a total of 20 points. For every 3 multiple-choice questions with 4 options or fewer answered incorrectly, 1 point will be deducted. Only one option is correct unless stated otherwise in the statement. The use of a calculator is not allowed. The exam duration is 40 minutes. ***Follow the instructions on the answer sheet****.*

1 [1p] Any TCP connection is identified by...

- ☐ **a)** One socket.
- ■ **b)** Two sockets.
- ☐ **c)** Four sockets.
- ☐ **d)** Two ISNs.
- ☐ **e)** Two PIDs (process ID).
- ☐ **f)** All are false.

2 [1p] Any TCP connection is always initiated...

- ☐ **a)** setting cwnd=MSS
- ☐ **b)** with a random ISN
- ☐ **c)** using triple handshake
- ☐ **d)** by the client
- ■ **e)** All are true

3 [1p] What is the specific purpose of the `bind()` system call in POSIX systems?

- ■ **a)** For TCP and UDP sockets, it associates a port with a process.
- ☐ **b)** Determines the maximum number of clients that can connect to the server.
- ☐ **c)** Blocks the process while waiting for an incoming connection request.
- ☐ **d)** Specifies the remote socket address to which a client wants to connect.

4 [1p] What differentiates a server from a client?

- ☐ **a)** The server is the one that serves the data.
- ■ **b)** The server listens on a known port.
- ☐ **c)** The server is more powerful.
- ☐ **d)** None of the above.

5 [1p] What does the following code do?

```
server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
server.bind(('', 3000))

while True:
    message, endpoint = server.recvfrom(1024)
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect(endpoint)
    client.send(message)
    client.close()
```

- ☐ **a)** It is a TCP server that sends itself the same requests it receives from a remote client.
- ☐ **b)** It is a TCP client that creates a new server each time it receives a response.
- ■ **c)** It is a kind of proxy that sends the data to same client node, but using a different protocol.
- ☐ **d)** It is an HTTP proxy that allows the client to decide the remote port for subsequent requests.

6 [1p] Given the following `tshark` capture, which method call is responsible for the first segment?

```
0.000000   10.10.10.1   -> 10.10.10.118 TCP 37804 > 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460
0.000304   10.10.10.118 -> 10.10.10.1   TCP 80 > 37804 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460
0.000314   10.10.10.1   -> 10.10.10.118 TCP 37804 > 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0
0.000337   10.10.10.1   -> 10.10.10.118 HTTP GET http://img.systemadmin.es/images/web/logo.gif HTTP/1.0
0.000754   10.10.10.118 -> 10.10.10.1   TCP 80 > 37804 [ACK] Seq=1 Ack=154 Win=6912 Len=0
```

■ **a)** connect() on the client. ☐ **b)** accept() on the client. ☐ **c)** bind() on the server. ☐ **d)** accept() on the server.

7 [1p] In UDP, data flow issues are found in this case:

- ☐ **a)** The client uses a UDP socket, while the server uses a TCP socket.
- ☐ **b)** The client uses a TCP socket, while the server uses a UDP socket.
- ■ **c)** When the sender's computer is significantly faster than the receiver's.
- ☐ **d)** Never

8 [1p] Why does *silly window syndrome* affect connection efficiency?

- ☐ **a)** Large segments negatively impact efficiency because they cause fragmentation.
- ■ **b)** Small segments negatively impact efficiency due to the relative overhead of headers.
- ☐ **c)** It complicates the management software that tracks unacknowledged (lost ACK) segments.
- ☐ **d)** Small segments negatively impact efficiency because they increase fragmentation costs.

9 [1p] Can there be a triple handshake for both connection establishment and disconnection in TCP?

- ☐ **a**) Yes, in cases with multiple RTOs.
- ■ **b**) Yes, when a peer sends FIN and the other has no more data to send.
- ☐ **c**) No, except when multiple RTOs occur for segments containing data.
- ☐ **d**) Obviously not; triple handshake is for connection establishment, while quadruple handshake is for disconnection.

10 [1p] Which TCP timer prevents the sender from remaining blocked when it never receives the window opening?

- ■ **a**) Persistence timer.
- ☐ **b**) Keep-alive timer.
- ☐ **c**) Retransmission timer.
- ☐ **d**) End-of-connection timer, also known as the Time-Wait timer.

11 [1p] The Selective Acknowledgment (SACK) functionality...

- ■ **a**) prevents unnecessary retransmissions.
- ☐ **b**) keeps a record of all ACKs lost in a transmission.
- ☐ **c**) selects the ACKs that are actually useful in a TCP transmission.
- ☐ **d**) is an optional flag in the TCP header that allows discarding duplicate ACKs.

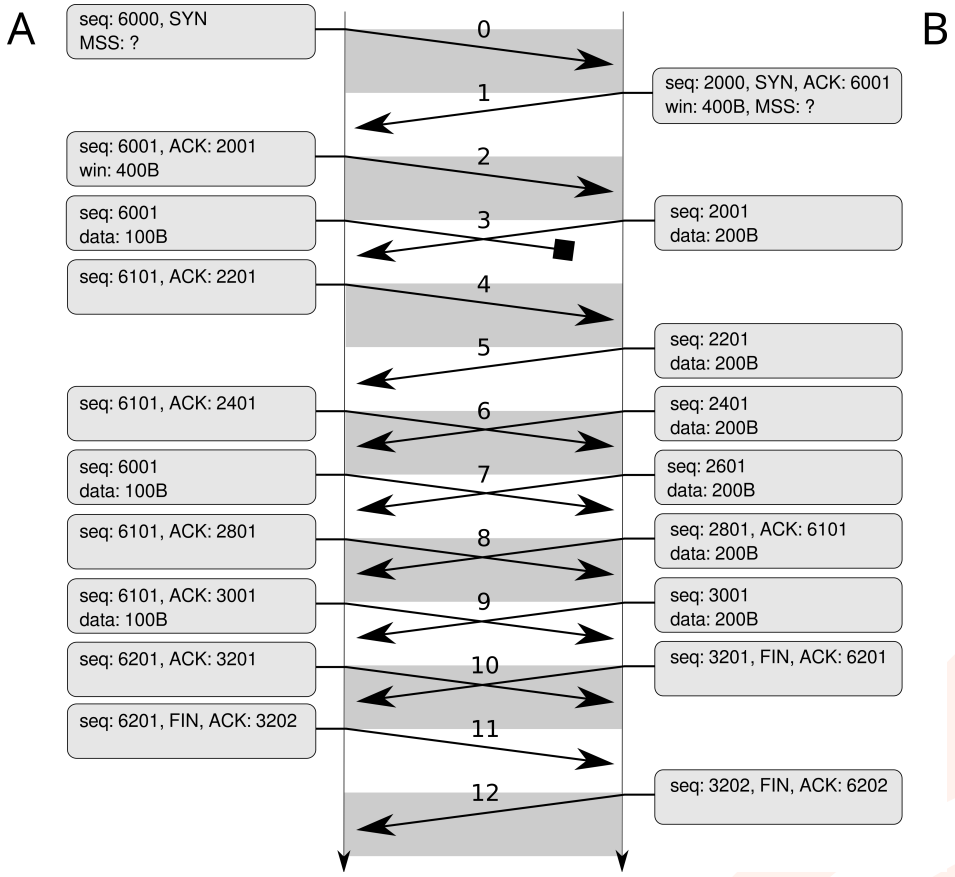
12 [1p] How does a TCP sender determine whether a data segment or its corresponding ACK has been lost?

- ☐ **a**) It depends on the sequence number.
- ☐ **b**) It cannot; for the TCP sender, both situations appear identical.
- ☐ **c**) The receiver explicitly reports lost ACKs in subsequent messages.
- ■ **d**) If a data segment is lost, duplicate ACKs may arrive, which does not happen when an ACK is lost.

13 [1p] Which of the following is NOT a reason for TCP to modify the *sequence number* field in a header?

- ☐ **a**) When the FIN flag is set.
- ☐ **b**) When the SYN flag is set.
- ☐ **c**) When the segment contains data.
- ■ **d**) For *pure* ACK segments (without data or other flags).

A [7p] Based on the TCP connection represented in the figure, answer the questions, considering that A and B will send data in sync with a clock tick. The initial ssthresh value is 64 KiB.



> **14** (1p) Does the message pattern indicate that congestion control is being used?

- ■ **a)** yes
- ☐ **b)** no
- ☐ **c)** Only during the handshake
- ☐ **d)** Only during the first 4 data segments

> **15** (1p) How many RTT rounds does B perform?

☐ **a)** 1 ☐ **b)** 2 ■ **c)** 3 ☐ **d)** 4 ☐ **e)** 5 ☐ **f)** Not applicable

> **16** (1p) What appear to be the MSS values? (Select one for A and one for B)

- ☐ **a)** B:100
- ■ **b)** B:200
- ☐ **c)** B:300
- ☐ **d)** B:400
- ■ **e)** A:100
- ☐ **f)** A:200
- ☐ **g)** A:300
- ☐ **h)** A:400

> **17** (1p) What message from A is missing at tick 5?

☐ **a)** a retransmission ☐ **b)** an ACK ☐ **c)** a data segment ■ **d)** nothing is missing

> **18** (1p) What appears to be A's retransmission timeout?

☐ **a)** 1 ☐ **b)** 2 ☐ **c)** 3 ■ **d)** 4 ☐ **e)** 5 ☐ **f)** 6

> **19** (1p) How much effective data (excluding RTX) is sent by each end? (Select one for A and one for B)

- ☐ **a)** B:800
- ☐ **b)** B:1000
- ☐ **c)** B:1100
- ■ **d)** B:1200
- ☐ **e)** A:100
- ■ **f)** A:200
- ☐ **g)** A:400

> **20** (1p) Why does B send no data at tick 4?

- ☐ **a)** Due to A's lost message at tick 3.
- ■ **b)** It would exceed swnd.
- ☐ **c)** rwnd is full.
- ☐ **d)** swnd is empty.