

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. When prompted, it's required to check all correct options. The use of a calculator is not allowed. The exam duration is 50 min. **Follow answer sheet instructions.**

- 1** [1p] What is a common use of the *unspecified address* IP in sockets?
- ☐ a) When calling connect() for to a generic server (anycast). ☐ c) To indicate a disconnected master socket.
- ☒ b) A server that listens on all network interfaces. ☐ d) There is no common use for that address.

- 2** [1p] Why is a UDP server that does not use threads, subprocesses, etc., considered concurrent? Because...
- ☒ a) It can serve multiple clients.
- ☐ b) The OS provides automatic parallelism for UDP sockets.
- ☐ c) It is not possible to use threads or processes with UDP sockets.
- ☐ d) Several UDP clients can share the same connection and be served at the same time.

- 3** [1p] When a TCP connection is established, who can send data first?
- ☒ a) Either side.
- ☐ b) The client, because it initiates the connection.
- ☐ c) The server, because it accepts the connection.
- ☐ d) It is determined by the FIRST-DATA TCP option during connection.

- 4** [1p] How is server ISN determined?
- ☐ a) It is always zero.
- ☒ b) It is a random integer.
- ☐ c) It starts with a random integer and is incremented for each connection.
- ☐ d) It is set by the programmer.

- 5** [2p] Consider the following code snippet from a minimal HTTP server:

```

1  server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
2  server.bind(('', 80))
3  server.listen(5)
4  while 0:
5      server_child, addr = server.accept()
6      endpoint, data = server_child.recv(1024)
7      response = handle(data)
8      server.send(response)
9      server_child.close()
10 server.close()

```

Mark the lines that contain errors: (multiple answers)

- ☒ a) 1 ☐ c) 3 ☐ e) 5 ☐ g) 7 ☐ i) 9
- ☐ b) 2 ☒ d) 4 ☒ f) 6 ☒ h) 8 ☐ j) 10

- 6** [1p] Why does TCP continuously recalculate the retransmission timeout (RTO)?
- ☒ a) Network conditions may differ for each transmission.
- ☐ b) The RTT value depends on the size and content of each segment.
- ☐ c) It is only recalculated when the application performs a receive attempt (socket.recv).
- ☐ d) It is recalculated each time the peer changes the receive window size.

- 7** [1p] Which of the following TCP features are also present in UDP?
- ☐ a) Connection establishment and termination. ☐ c) Retransmission of lost data.
- ☐ b) Flow control. ☒ d) None of the above.

- 8** [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; to the TCP sender, both situations appear the same.
- ☐ 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.

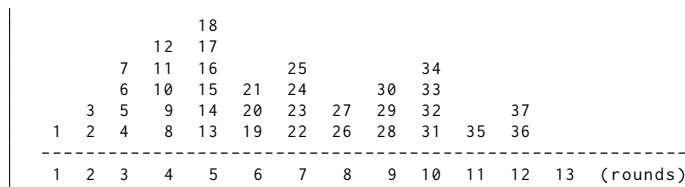
9 [2p] Which of the following differences between ACK and SACK is FALSE?

- ☐ a) SACK is more efficient than ACK when there are many packet losses.
- ☐ b) SACK is widely adopted but still not used in all transmissions.
- ☐ c) SACK avoids many unnecessary retransmissions, whereas ACK does not.
- ☒ d) SACK works over UDP, but not as efficiently as over TCP.

10 [2p] 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.

A [7p] Consider the following graph representing the congestion window of a TCP connection. The numbers indicate the order in which the segments are sent, but nothing about its content. Answer the following questions:



> **11** (1p) What is the value of the initial threshold (*ssthresh*) (measured in MSS)?

- ☐ a) 1
- ☐ b) 2
- ☐ c) 3
- ☒ d) 4
- ☐ e) 5
- ☐ f) 6

> **12** (1p) What happened in round 5?

- ☐ a) $cwnd > ssthresh$
- ☐ b) RTO
- ☒ c) 3 DUP ACK
- ☐ d) $win = 3$ MSS
- ☐ e) None

> **13** (1p) What happened in round 8?

- ☐ a) $cwnd > ssthresh$
- ☐ b) RTO
- ☐ c) 3 DUP ACK
- ☐ d) $win = 2$ MSS
- ☒ e) None

> **14** (1p) What happened in round 10?

- ☐ a) $cwnd > ssthresh$
- ☒ b) RTO
- ☐ c) 3 DUP ACK
- ☐ d) $win = 2$ MSS
- ☐ e) None

> **15** (1p) If there are still data to be sent and no other problem has occurred. What segments should be sent in round 13?

- ☒ a) 38
- ☒ b) 39
- ☒ c) 40
- ☐ d) 41
- ☐ e) 42
- ☐ f) 43

> **16** (2p) Mark the correct options. Each one indicates the phase and the rounds where it occurs. Also consider round 13.

SS stands for *slow start*, CA for *congestion avoidance*. Example: SS[20-22] means slow start occurred in rounds 20 through 22 inclusive.

- ☒ a) SS[1-3]
- ☐ b) SS[1-4]
- ☐ c) CA[3-5]
- ☒ d) CA[4-10]
- ☐ e) SS[8-9]
- ☒ f) SS[11-12]
- ☐ g) SS[11-13]
- ☒ h) CA[13]