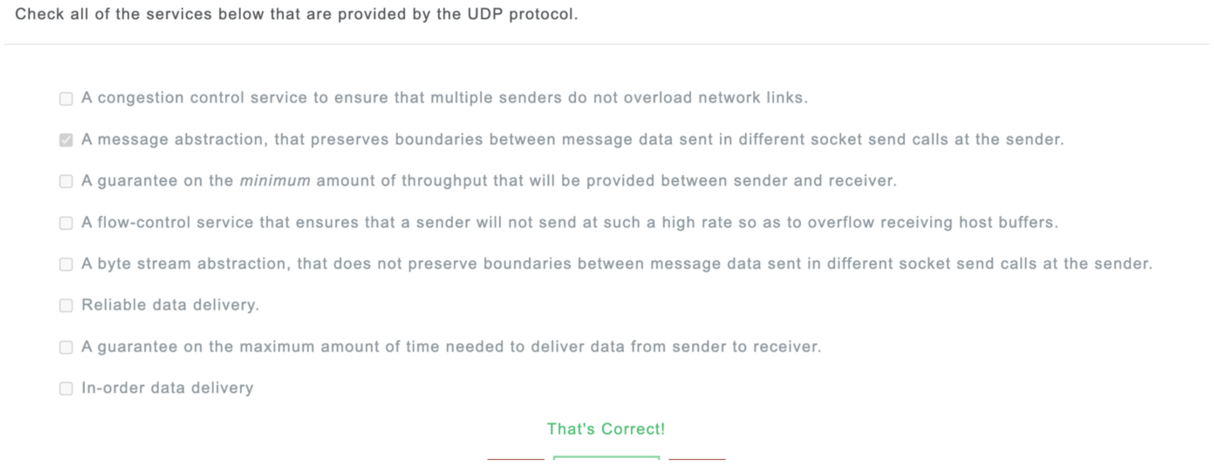
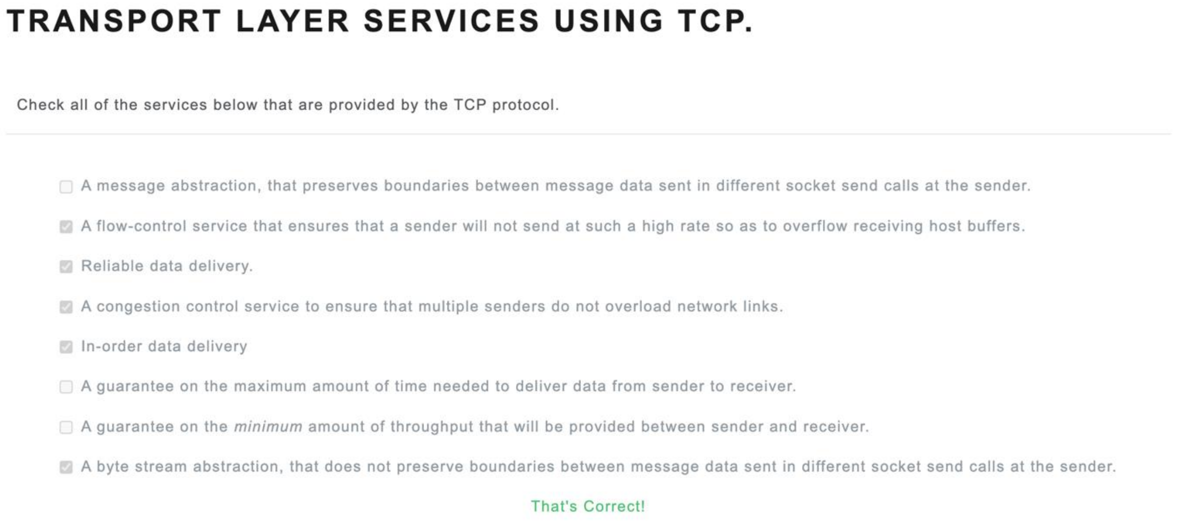
1.



Check all of the services below that are provided by the UDP protocol.

- ☐ A congestion control service to ensure that multiple senders do not overload network links.
- ☑ A message abstraction, that preserves boundaries between message data sent in different socket send calls at the sender.
- ☐ A guarantee on the *minimum* amount of throughput that will be provided between sender and receiver.
- ☐ A flow-control service that ensures that a sender will not send at such a high rate so as to overflow receiving host buffers.
- ☐ A byte stream abstraction, that does not preserve boundaries between message data sent in different socket send calls at the sender.
- ☐ Reliable data delivery.
- ☐ A guarantee on the maximum amount of time needed to deliver data from sender to receiver.
- ☐ In-order data delivery

That's Correct!



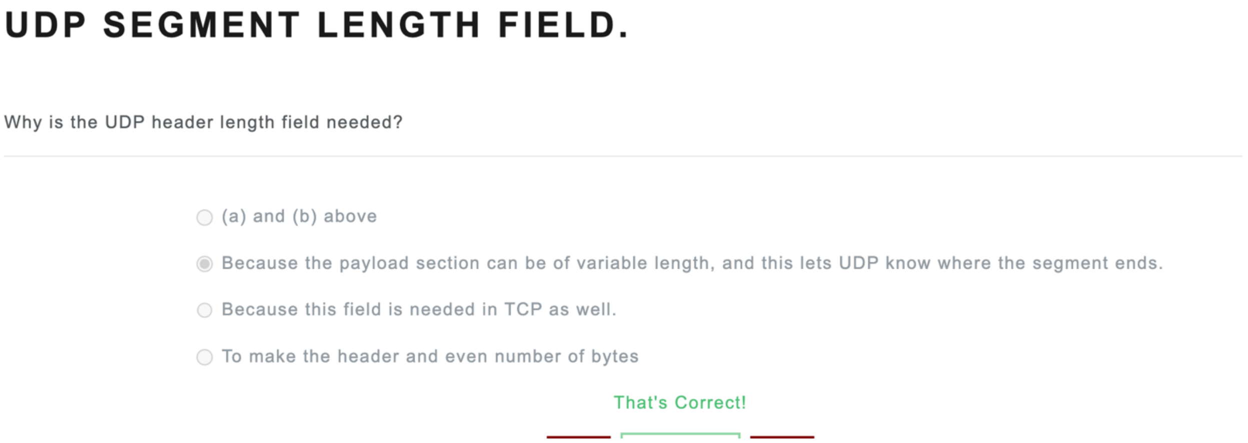
TRANSPORT LAYER SERVICES USING TCP.

Check all of the services below that are provided by the TCP protocol.

- ☐ A message abstraction, that preserves boundaries between message data sent in different socket send calls at the sender.
- ☑ A flow-control service that ensures that a sender will not send at such a high rate so as to overflow receiving host buffers.
- ☑ Reliable data delivery.
- ☑ A congestion control service to ensure that multiple senders do not overload network links.
- ☑ In-order data delivery
- ☐ A guarantee on the maximum amount of time needed to deliver data from sender to receiver.
- ☐ A guarantee on the *minimum* amount of throughput that will be provided between sender and receiver.
- ☑ A byte stream abstraction, that does not preserve boundaries between message data sent in different socket send calls at the sender.

That's Correct!

2.



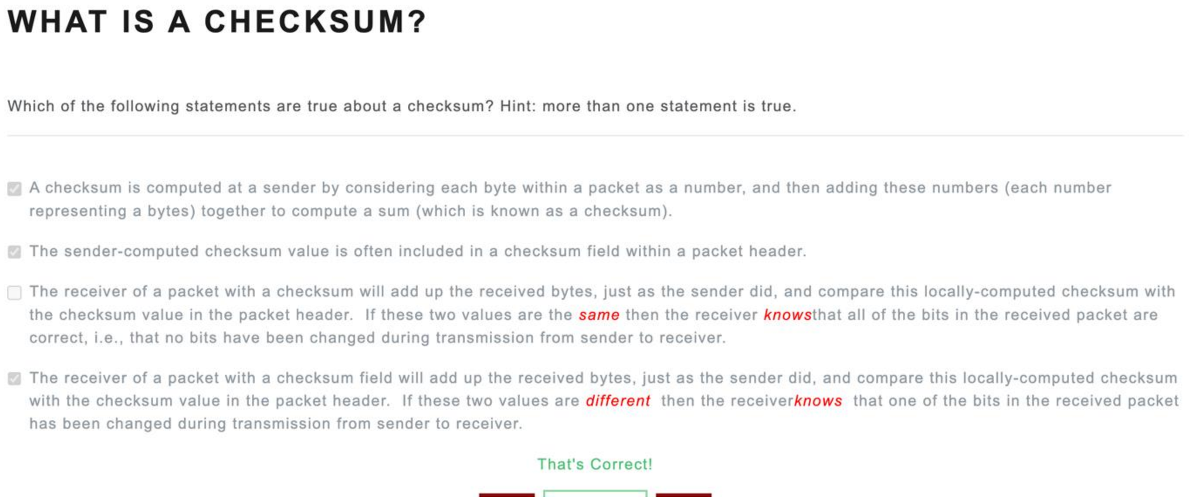
UDP SEGMENT LENGTH FIELD.

Why is the UDP header length field needed?

- ○ (a) and (b) above
- ● Because the payload section can be of variable length, and this lets UDP know where the segment ends.
- ○ Because this field is needed in TCP as well.
- ○ To make the header and even number of bytes

That's Correct!

3.



WHAT IS A CHECKSUM?

Which of the following statements are true about a checksum? Hint: more than one statement is true.

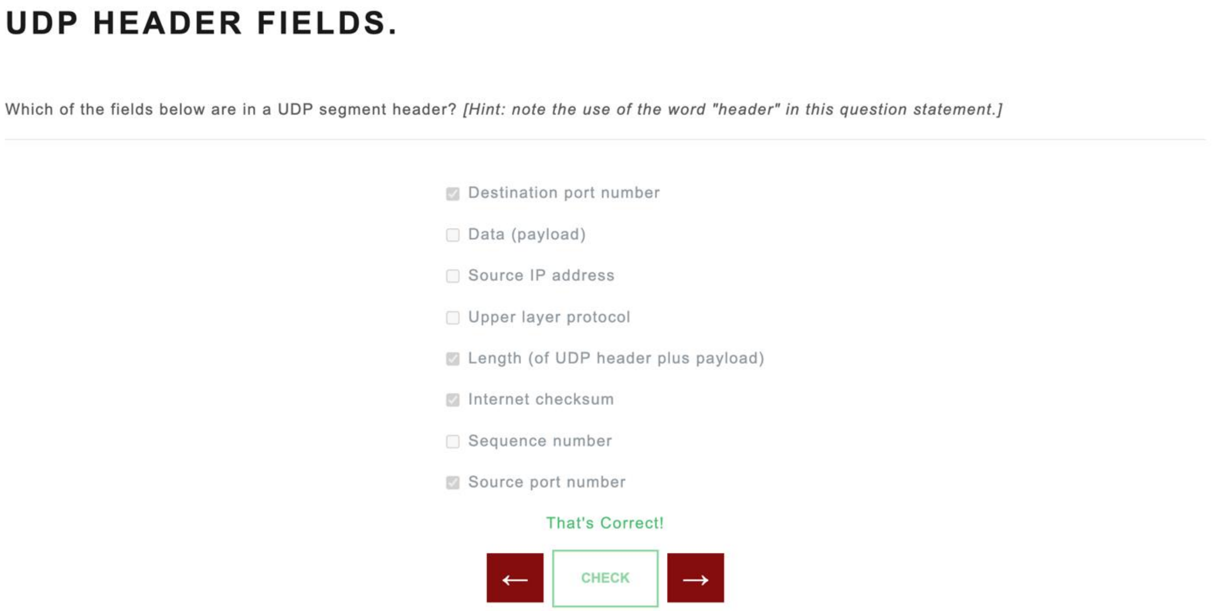
☑ A checksum is computed at a sender by considering each byte within a packet as a number, and then adding these numbers (each number representing a bytes) together to compute a sum (which is known as a checksum).

☑ The sender-computed checksum value is often included in a checksum field within a packet header.

□ The receiver of a packet with a checksum will add up the received bytes, just as the sender did, and compare this locally-computed checksum with the checksum value in the packet header. If these two values are the ***same*** then the receiver ***knows***that all of the bits in the received packet are correct, i.e., that no bits have been changed during transmission from sender to receiver.

☑ The receiver of a packet with a checksum field will add up the received bytes, just as the sender did, and compare this locally-computed checksum with the checksum value in the packet header. If these two values are ***different*** then the receiver***knows*** that one of the bits in the received packet has been changed during transmission from sender to receiver.

That's Correct!



UDP HEADER FIELDS.

Which of the fields below are in a UDP segment header? *[Hint: note the use of the word "header" in this question statement.]*

☑ Destination port number

□ Data (payload)

□ Source IP address

□ Upper layer protocol

☑ Length (of UDP header plus payload)

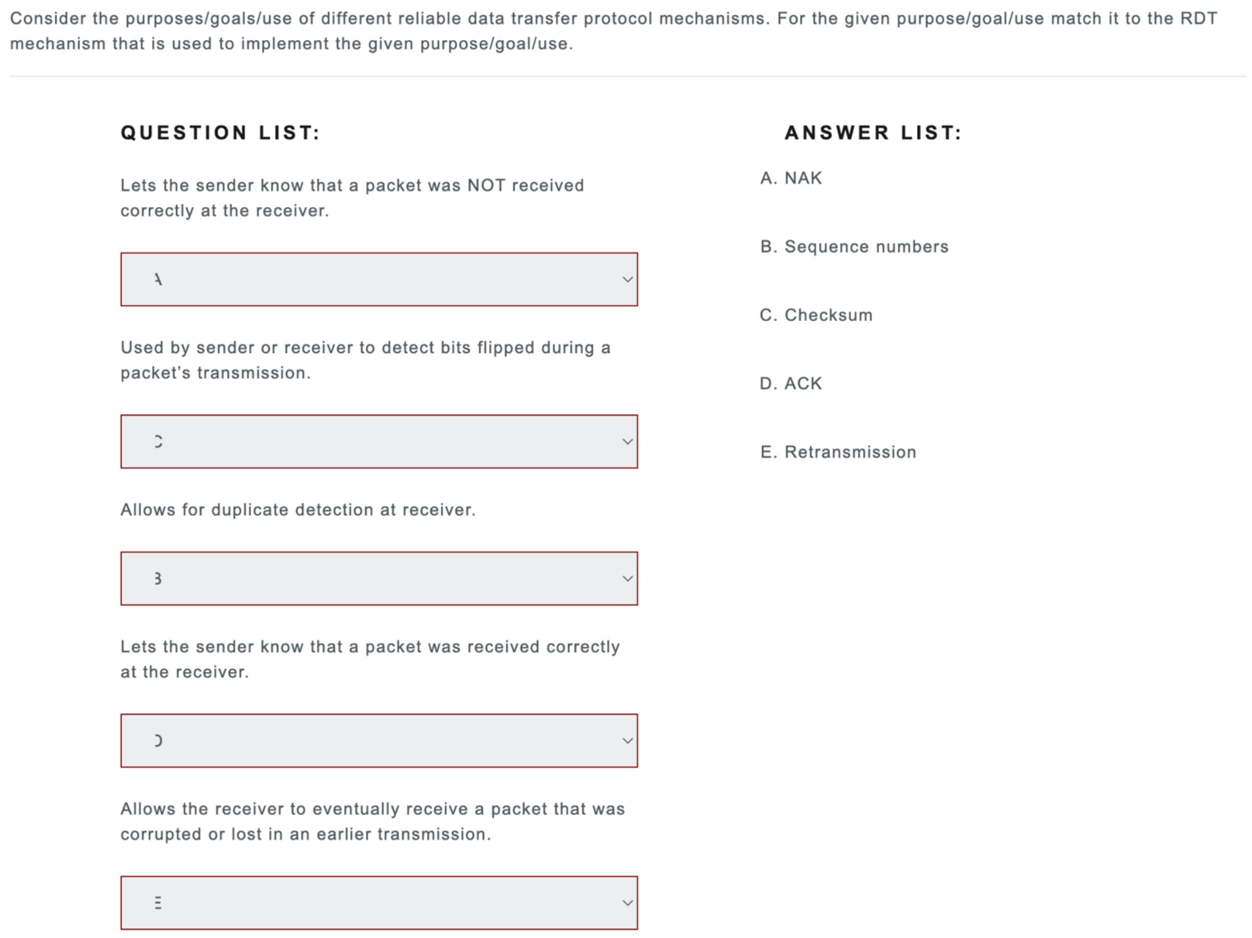
☑ Internet checksum

□ Sequence number

☑ Source port number

That's Correct!

4.



Consider the purposes/goals/use of different reliable data transfer protocol mechanisms. For the given purpose/goal/use match it to the RDT mechanism that is used to implement the given purpose/goal/use.

QUESTION LIST:

Lets the sender know that a packet was NOT received correctly at the receiver.

A

Used by sender or receiver to detect bits flipped during a packet's transmission.

C

Allows for duplicate detection at receiver.

B

Lets the sender know that a packet was received correctly at the receiver.

D

Allows the receiver to eventually receive a packet that was corrupted or lost in an earlier transmission.

E

ANSWER LIST:

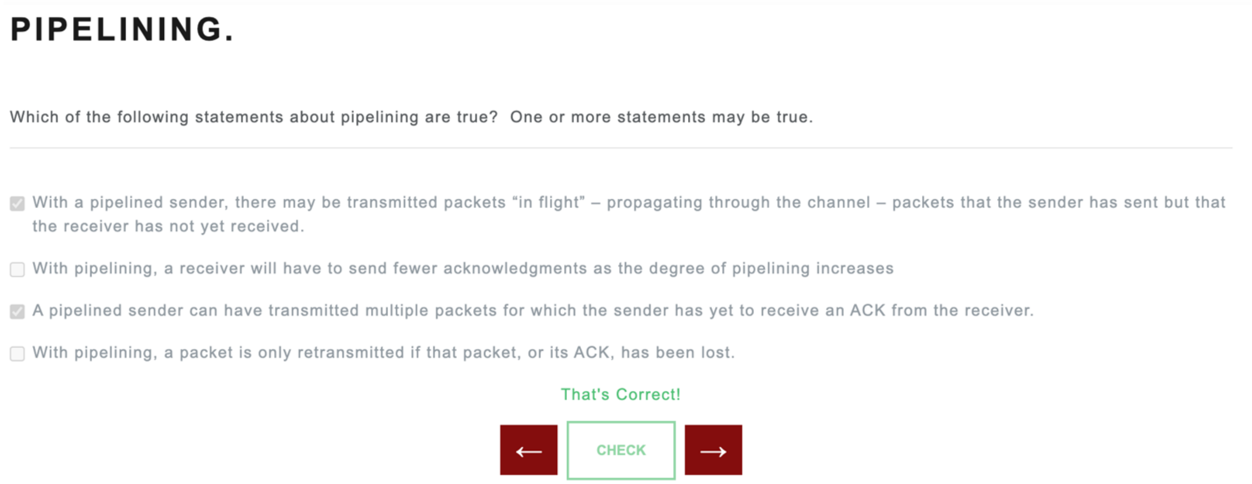
A. NAK

B. Sequence numbers

C. Checksum

D. ACK

E. Retransmission



PIPELINING.

Which of the following statements about pipelining are true? One or more statements may be true.

☑ With a pipelined sender, there may be transmitted packets "in flight" – propagating through the channel – packets that the sender has sent but that the receiver has not yet received.

□ With pipelining, a receiver will have to send fewer acknowledgments as the degree of pipelining increases

☑ A pipelined sender can have transmitted multiple packets for which the sender has yet to receive an ACK from the receiver.

□ With pipelining, a packet is only retransmitted if that packet, or its ACK, has been lost.

That's Correct!

PACKET BUFFERING IN GO-BACK-N.

What are some reasons for discarding received-but- out-of-sequence packets at the receiver in GBN? Indicate one or more of the following statements that are correct.

- □ If some packets are in error, then its likely that other packets are in error as well.
- □ Discarding an out of sequence packet will really force the sender to retransmit.
- ☑ The implementation at the receiver is simpler.
- ☑ The sender will resend that packet in any case.

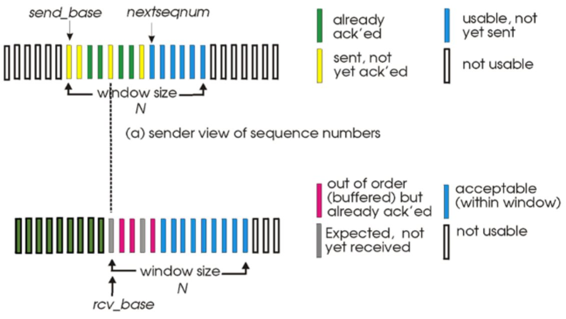
That's Correct!

← CHECK →

11/14

In SR, why does the receiver have to acknowledge packets with sequence numbers that are less than (and to the left of) those in its window, which starts at *rcv_base*.

Selective repeat: sender, receiver windows



(a) sender view of sequence numbers

(b) receiver view of sequence numbers

- □ Actually, this ACK retransmission can be ignored and the protocol will still function correctly, but its performance won't be as good.
- □ Because, at the time of the data packet arrival at the receiver, the sender has definitely still not received an ACK for that packet.
- ☑ Because the sender may not have received an ACK for that packet yet.

That's Correct!

What are some reasons for *not* discarding received-but- out-of-sequence packets at the receiver in GBN? Indicate one or more of the following statements that are correct.

☑ Even though that packet will be retransmitted, its next retransmission could be corrupted, so don't discard a perfectly well-received packet, silly!

□ Complex protocols are always better.

□ By not discarding, the receiver can implicitly let the sender know that it (the sender) does not necessarily have to retransmit that packet.

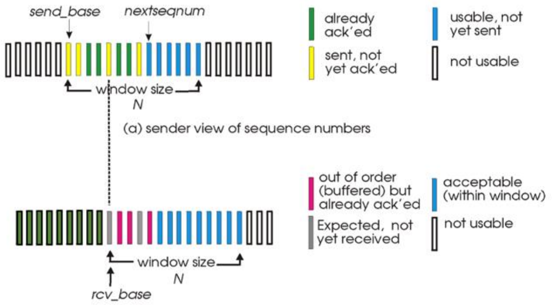
That's Correct!

← CHECK →

12/14

In the SR receiver window (see diagram below, taken from PPT slides and video), why haven't the red packets been delivered yet? Check the one or more reasons below that apply.

Selective repeat: sender, receiver windows



(a) sender view of sequence numbers

(b) receiver view of sequence numbers

□ Red packets have a lower delivery priority up to the application.

☑ There is a packet with a lower sequence number than any of the red packets that has yet to be received, so in-order delivery of data in the red packets up to the application layer is not possible.

□ There is a packet with a higher sequence number than any of the red packets that has yet to be received, so in-order delivery of data in the red packets to the application layer is not yet possible.

That's Correct!

← CHECK →