INTERNET CHECKSUM

COMPUTING AN INTERNET CHECKSUM

Consider the two 16-bit words (shown in binary) below. Recall that to compute the Internet checksum of a set of 16-bit words, we compute the one's complement sum [1] of the two words. That is, we add the two numbers together, making sure that any carry into the 17th bit of this initial sum is added back into the 1's place of the resulting sum); we then take the one's complement of the result. Compute the Internet checksum value for these two 16-bit words:

10000110 10001001 *this binary number is 34441 decimal (base 10)*

10000100 10011000 *this binary number is 33944 decimal (base 10)*

1. QUESTION 1 OF 2

What is the sum of these two 16 bit numbers? Don't put any spaces in your answer

Answer ✓

```
1000 0110 1000 1001 +
1000 0100 1001 1000 =
-------------------
0000 1011 0010 0001 +
                1
-------------------
0000 1011 0010 0010 ✓
```

2. QUESTION 2 OF 2

Using the sum from question 1, what is the checksum? Don't put any spaces in your answer

Answer ✓

1111 0100 1101 1101 ✓

RDT 2.2

RELIABLE DATA TRANSFER: RDT2.2 (SENDER AND RECEIVER ACTIONS)

Consider the rdt2.2 protocol from the text (pages 209-212). The FSMs for the sender and receiver are shown below:

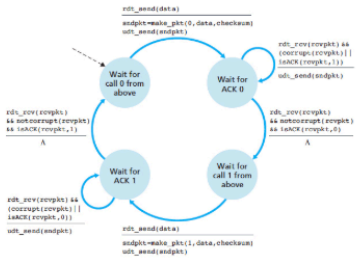


Figure 3.13 ♦ rdt2.2 sender

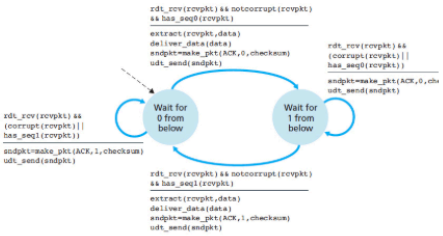
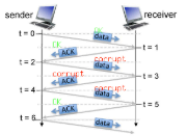


Figure 3.14 ♦ rdt2.2 receiver

Suppose that the channel connecting the sender and receiver can corrupt but not lose or reorder packets. Now consider the figure below, which shows four data packets and three corresponding ACKs being exchanged between an rdt 2.2 sender and receiver. The actual corruption or successful transmission/reception of a packet is indicated by the corrupt and OK labels, respectively, shown above the packets in the figure below.



1. QUESTION 1 OF 13

At time t=0, what is the sender state?

Answer: Wait for ACK 0

2. QUESTION 2 OF 13

At time t=0, what is the receiver state?

Wait for 0 from below

3. QUESTION 3 OF 13

At time t=0, what is the sequence/ack # of the packet?

0

4. QUESTION 4 OF 13

At time t=1, what is the sender state?

Wait for ACK 0

5. QUESTION 5 OF 13

At time t=1, what is the receiver state?

Wait for 1 from below

6. QUESTION 6 OF 13

At time t=1, what is the sequence/ack # of the packet?

0

7. QUESTION 7 OF 13

At time t=2, what is the sender state?

Wait for ACK 1

8. **QUESTION 8 OF 13**

At time t=2, what is the receiver state?

Wait for 1 from below

9. **QUESTION 9 OF 13**

At time t=2, what is the sequence/ack # of the packet?

1

10 **QUESTION 10 OF 13**

At time t=3, what is the sender state?

Wait for ACK 1

11 **QUESTION 11 OF 13**

At time t=3, what is the receiver state?

Wait for 1 from below

12 **QUESTION 12 OF 13**

At time t=3, what is the sequence/ack # of the packet?

0

13 **QUESTION 13 OF 13**

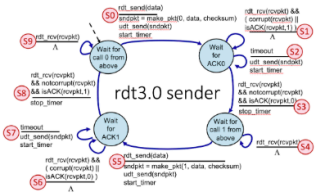
How many times is the payload of the received packet passed up to the higher layer?

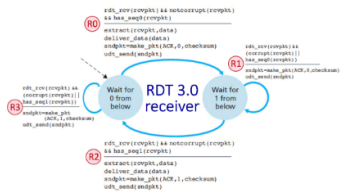
1

RDT 3.0

RELIABLE DATA TRANSFER: RDT 3.0

Consider the RDT 3.0 protocol, for reliably communicating data from a sender to receiver over a channel that can lose or corrupt packets in either direction, and when the maximum delay from sender to receiver and back is not known. The FSMs for the sender and receiver are shown below, with their transitions labeled as SX and RY, respectively.





Now let's consider the sequence of sender and receiver transitions that would happen when one or more of the following complications occur: a packet (data or ACK) is lost, a timer times out (prematurely or not), or a message is corrupted. One or more of these events has occurred to produce the sequence of transitions below. In the sequence below, one transition has been omitted and replaced with a "*".

Transition Sequence: S0, *, S1, S2, R1, S3, S4, S5, R2, S8

1

QUESTION 1 OF 1

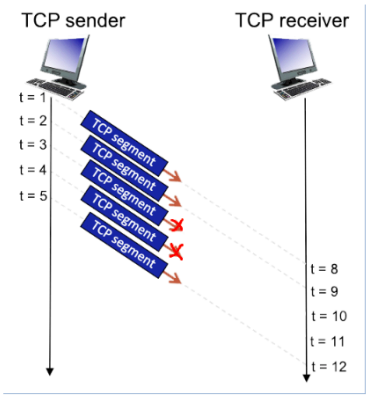
What is the missing transition? To indicate the missing transition, enter S or R, followed by an index.

R0

TCP SEQ. & ACK

TCP SEQUENCE AND ACK NUMBERS WITH SEGMENT LOSS

Consider the figure below in which a TCP sender and receiver communicate over a connection in which the sender->receiver segments may be lost. The TCP sender sends an initial window of 5 segments. Suppose the initial value of the sender->receiver sequence number is 326 and the first 5 segments *each* contain 433 bytes. The delay between the sender and receiver is 7 time units, and so the first segment arrives at the receiver at t=8. As shown in the figure below, 2 of the 5 segment(s) are lost between the segment and receiver.



1. QUESTION 1 OF 2

Give the sequence numbers associated with each of the 5 segments sent by the sender. Format your answer as: a,b,c,...

Answer ✓

326,

759,

1192,

1625,

2058

2. QUESTION 2 OF 2

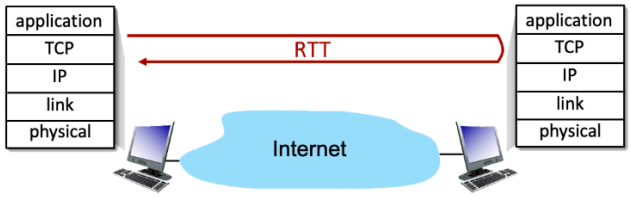
Give the ACK numbers the receiver sends in response to each of the segments. If a segment never arrives use 'x' to denote it, and format your answer as: a,b,c,...

759,1192,x,x,1192

TCP: RTT & TIMEOUT

COMPUTING TCP'S RTT AND TIMEOUT VALUES

Suppose that TCP's current estimated values for the round trip time (*estimatedRTT*) and deviation in the RTT (*DevRTT*) are 390 msec and 32 msec, respectively (see Section 3.5.3 for a discussion of these variables). Suppose that the next three measured values of the RTT are 200 msec, 390 msec, and 390 msec respectively.



Compute TCP's new value of *DevRTT, estimatedRTT,* and the TCP timeout value after each of these three measured RTT values is obtained. Use the values of $\alpha = 0.125$, and $\beta = 0.25$. Round your answers to two decimal places after leading zeros.

Timeout = E.RTT + C · DevRTT

E.RTT = (1-α) · ERT + α · RTT

DevRTT = (1-β) · DevRTT +

β · |ERT - RTT|

1. QUESTION 1 OF 9

What is the estimatedRTT after the first RTT?

Answer ✓

$ERTT = 0.125 \cdot 200ms + (1-0.125) \cdot 390ms$

$= 366.25$

2. QUESTION 2 OF 9

What is the RTT Deviation for the the first RTT?

Answer ✓

- Prendere le variabili del RTT precedente anche di ERTT

$DevRTT = (1-0.25) \cdot 32 + 0.25 \cdot |390-200| = 71.5\ ms$

3. QUESTION 3 OF 9

What is the TCP timeout for the first RTT?

Answer

$C = 4. \Rightarrow 366.25 + 4 \cdot 71.5 = 652.25\ ms$

4. QUESTION 4 OF 9

What is the estimatedRTT after the second RTT?

Answer ✓

$RTT_2 = 390ms$

$\alpha = 0.125$ DevRTT: 71.5ms

$ERTT_0 = 366.25ms$

$E.RTT_1 = (1-\alpha) \cdot ERTT + \alpha \cdot SRTT$

$= (1-0.125) \cdot 366.25ms + 0.125 \cdot 390ms$

$= 369.22$

5. QUESTION 5 OF 9

What is the RTT Deviation for the the second RTT?

Answer ✓

$DevRTT_1 = (1-\beta) \cdot DevRTT_0 + \beta \cdot |ERTT_0 - S.RTT_1|$

$= (1-0.25) \cdot 71.5\,ms + 0.25 \cdot |366.25\,ms - 390ms|$

$= 0.75 \cdot 71.5ms + 0.25 \cdot 23.75ms$

$= 59.56$

6 QUESTION 6 OF 9

What is the TCP timeout for the second RTT?

Answer ✓

$Timeout_1 = ERTT_1 + C \cdot DevRTT_1 \quad con\ C=4$

$= 369.21875 + 4 \cdot 59.5625\,ms$

$\overset{N}{=} 607.47ms$

7. QUESTION 7 OF 9

What is the estimatedRTT after the third RTT?

Answer ✓

$ERTT_1 = 369.22\,ms$

$DevRTT_1 = 59.56\,ms$

$RTT_2 = 390ms$

$ERTT_2 = (1-\alpha)\, ERTT_1 + \alpha \cdot RTT_2$

$= (1-0.125) \cdot 369.22\,ms + 0.125 \cdot 390ms$

$= 0.875 \cdot 369.22\,ms + 0.125 \cdot 390ms$

$= 371.8175 \overset{2}{\approx} 371.82$

8. QUESTION 8 OF 9

What is the RTT Deviation for the the third RTT?

Answer ✓

$DevRTT_2 = (1-\beta) \cdot DevRTT_1 + \beta \cdot |ERTT_1 - RTT_2|$

$= (1-0.25) \cdot 59.56ms + 0.25 \cdot |369.22 - 390ms|$

$= 0.75 \cdot 59.56ms + 0.25 \cdot 20.78\,ms$

$= 49.865 \overset{2}{\approx} 49.87$

9. QUESTION 9 OF 9

What is the TCP timeout for the third RTT?

Answer ✓

$Timeout_2 = ERTT_2 + C \cdot DevRTT_2 \quad C=4$

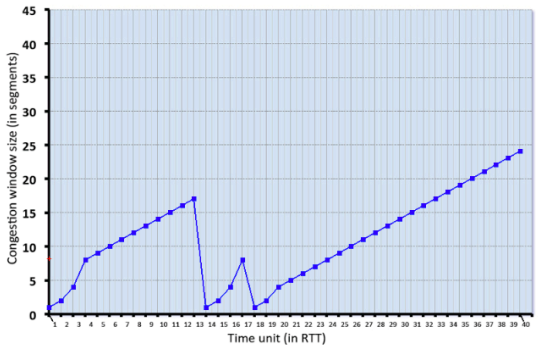
$= 371.8175 + 4 \cdot 49.865$

= 571.2775 ≈ 571.28 ✓

TCP : SLOW START, AVOIDANCE & FAST RETRANSMIT

TCP IN ACTION: SLOW START, CONGESTION AVOIDANCE, AND FAST RETRANSMIT

Consider the figure below, which plots the evolution of TCP's congestion window at the beginning of each time unit (where the unit of time is equal to the RTT); see Figure 3.53 in the text. In the abstract model for this problem, TCP sends a "flight" of packets of size *cwnd* at the beginning of each time unit. The result of sending that flight of packets is that either *(i)* all packets are ACKed at the end of the time unit, *(ii)* there is a timeout for the first packet, or *(iii)* there is a triple duplicate ACK for the first packet. In this problem, you are asked to reconstruct the sequence of events (ACKs, losses) that resulted in the evolution of TCP's *cwnd* shown below.



Consider the evolution of TCP's congestion window in the example above and answer the following questions. The initial value of *cwnd* is 1 and the initial value of *ssthresh* (shown as a red +) is 8.

1. QUESTION 1 OF 6

Give the times at which TCP is in slow start. Format your answer like: 1,3,5,9 (If none submit blank)

1,2,3,14,15,16,17,18,19 ✓

2. QUESTION 2 OF 6

Give the times at which TCP is in congestion avoidance. Format your answer like: 1,3,5,9 (If none submit blank)

4,5,6,7,8,9,10,11,12,13,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40x ✓

3 QUESTION 3 OF 6

Give the times at which TCP is in fast recovery. Format your answer like: 1,3,5,9 (If none submit blank)

Answer ✗ ✓

4. QUESTION 4 OF 6

Give the times at which packets are lost via timeout. Format your answer like: 1,3,5,9 (If none submit blank)

13,17 ✓

Guarda le parti dove cwd scende a 1

(il momento prima del drop)

5. QUESTION 5 OF 6

Give the times at which packets are lost via *triple ACK*. Format your answer like: 1,3,5,9 (If none submit blank)

Answer ✗ ✓

Guarda dove c'è un drop di $\frac{cwd}{2}$

6. QUESTION 6 OF 6

Give the times at which the value of *ssthresh* changes (if it changes between t=3 and t=4, use t=4 in your answer)

14,18 ✓

Guarda dove ci sono perdite in generale

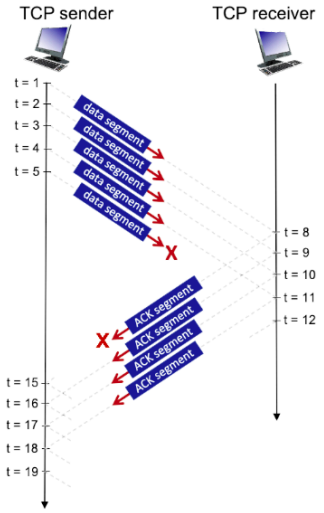
cwd: 1

ssthresh: 8 → 2 (3) → 6 (15) → 5 (21) → 6.5 (30)

TCP: RETRANSMIT

TCP RETRANSMISSIONS (RELIABLE DATA TRANSMISSION WITH ACK LOSS)

Consider the figure below in which a TCP sender and receiver communicate over a connection in which the segments can be lost. The TCP sender wants to send a total of 10 segments to the receiver and sends an initial window of 5 segments at t = 1, 2, 3, 4, and 5, respectively. Suppose the initial value of the sequence number is 59 and every segment sent to the receiver each contains 349 bytes. The delay between the sender and receiver is 7 time units, and so the first segment arrives at the receiver at t = 8, and an ACK for this segment arrives at t = 15. As shown in the figure, 1 of the 5 segments is lost between the sender and the receiver, but *one* of the ACKs is lost. Assume there are no timeouts and any out of order segments received are thrown out.



$seq_0 = 59$

$seq_5 = 349 + 59 = 408$

1. QUESTION 1 OF 15

What is the sequence number of the segment sent at t=1?

59 ✓

2. QUESTION 2 OF 15

What is the sequence number of the segment sent at t=2?

Answer 59+349 = 408 ✓

3. QUESTION 3 OF 15

What is the sequence number of the segment sent at t=3?

59+349*2=757 ✓

4. QUESTION 4 OF 15

What is the sequence number of the segment sent at t=4?

59+349*3=1106 ✓

5 QUESTION 5 OF 15

What is the sequence number of the segment sent at t=5?

59+349*4=1455 ✓

6 QUESTION 6 OF 15

What is the value of the ACK sent at t=8? (If segment lost, write 'x')

408 ✓

7 QUESTION 7 OF 15

What is the value of the ACK sent at t=9? (If segment lost, write 'x')

757 ✓

8 QUESTION 8 OF 15

What is the value of the ACK sent at t=10? (If segment lost, write 'x')

1106 ✓

9. **QUESTION 9 OF 15**

What is the value of the ACK sent at t=11? (If segment lost, write 'x')

1455 ✓

10 **QUESTION 10 OF 15**

What is the value of the ACK sent at t=12? (If segment lost, write 'x')

x ✓

11 **QUESTION 11 OF 15**

What is the sequence number of the segment sent at t = 15? (If ACK never arrives, write 'x')

x ✓

12 **QUESTION 12 OF 15**

What is the sequence number of the segment sent at t = 16? (If ACK never arrives, write 'x')

1455+349=1804 ✓

13 **QUESTION 13 OF 15**

What is the sequence number of the segment sent at t = 17? (If ACK never arrives, write 'x')

59+349*6=2153 ✓

14 **QUESTION 14 OF 15**

What is the sequence number of the segment sent at t = 18? (If ACK never arrives, write 'x')

59+349*7=2502 ✓

15 **QUESTION 15 OF 15**

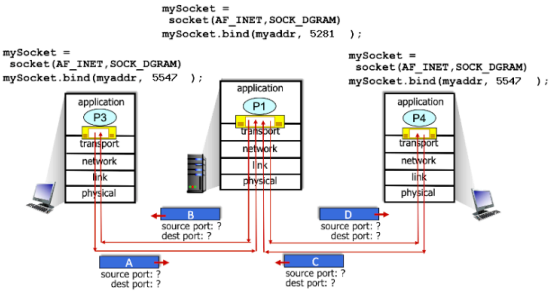
What is the sequence number of the segment sent at t = 19? (If ACK never arrives, write 'x')

59+349*8=2851 ✓

UDP MULTIPLEXING & DEMULTIPLEXING

UDP MULTIPLEXING AND DEMULTIPLEXING

In the scenario below, the left and right clients communicate with a server using UDP sockets. The same socket at the server is used to communicate with both clients. The Python code used to create the sockets is shown in the figure. Consider the four transport-layer packets – A, B, C and D – shown in the figure below.



1 QUESTION 1 OF 8

What is the source port # for packet D?

5281 ✓

2 QUESTION 2 OF 8

What is the destination port # for packet D?

5547 ✓

3 QUESTION 3 OF 8

What is the source port # for packet A?

5547 ✓

4. QUESTION 4 OF 8

What is the destination port # for packet A?

5281 ✓

5. QUESTION 5 OF 8

What is the source port # for packet B?

5281 ✓

6 QUESTION 6 OF 8

What is the destination port # for packet B?

5547 ✓

7 QUESTION 7 OF 8

What is the source port # for packet C?

5547 ✓

8 QUESTION 8 OF 8

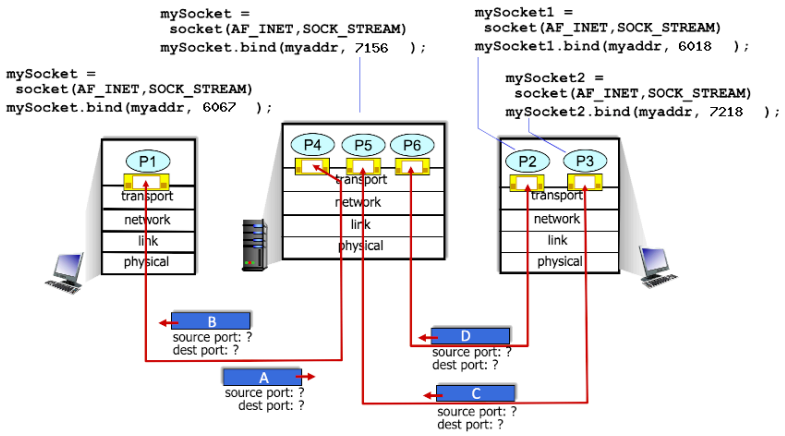
What is the destination port # for packet C?

5281 ✓

TCP: mux & demux

TCP MULTIPLEXING AND DEMULTIPLEXING

In the scenario below, the left and right TCP clients communicate with a TCP server using TCP sockets. The Python code used to create a single welcoming socket in the server is shown in the figure (the welcoming socket itself is not shown graphically); code is also shown for the client sockets as well. The three sockets shown in server were created as a result of the server accepting connection requests on this welcoming socket from the two clients (one connection from the client on the left, and two connections from the client on the right).



1 QUESTION 1 OF 8

What is the source port # for packet D?

6018 ✓

2 QUESTION 2 OF 8

What is the destination port # for packet D?

7156 ✓

3. QUESTION 3 OF 8

What is the source port # for packet B?

7156 ✓

4. QUESTION 4 OF 8

What is the destination port # for packet B?

6067 ✓

5. QUESTION 5 OF 8

What is the source port # for packet C?

7218 ✓

6. QUESTION 6 OF 8

What is the destination port # for packet C?

7156 ✓

7 QUESTION 7 OF 8

What is the source port # for packet A?

6067 ✓

8 QUESTION 8 OF 8

What is the destination port # for packet A?

7156 ✓