Question Completion Status:

Take Test: Assignment 1 - Part A: Problem solving questions

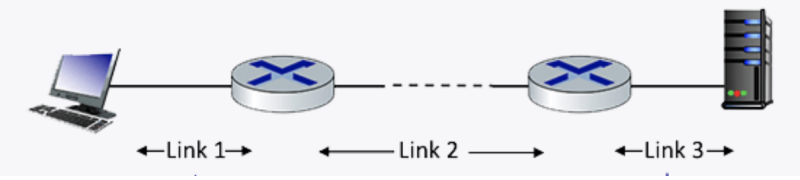
Test Information

Description	Problem solving questions.
Instructions	
Multiple Attempts	This Test allows multiple attempts.
Force Completion	This Test can be saved and resumed later.
	Your answers are saved automatically.

QUESTION 1

2 points Save Answer

End-to-end Delay. Consider the network shown in the figure below, with three links, each with a transmission rate of 1 Mbps, and a propagation delay of 1 msec per link. Assume the length of a packet is 1000 bits.



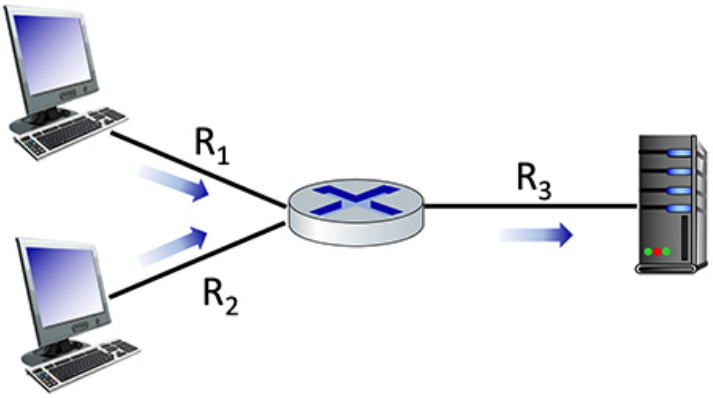
What is the end-end delay of a packet from when it first begins transmission on link 1, until is it received in full by the server at the end of link 3. Assume store-and forward packet transmission.

- ○ 1 msec
- ○ 2 msec
- ○ 3 msec
- ○ 6 msec
- ○ 12 msec

QUESTION 2

2 points Save Answer

Utilisation. Consider the scenario shown below, with a two clients sending to a server. The links attached to clients each have a capacity of $R_1 = R_2$ = 10 Mbps. The link from the router to the server has a capacity of and R_3 = 100 Mbps, which is shared evenly between the two sources when they are each sending at their maximum rate.



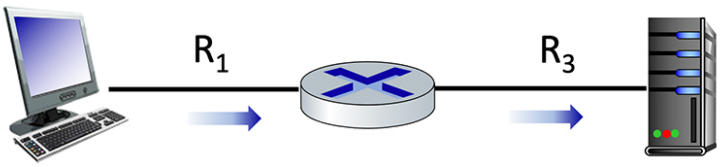
What is the utilization of the R_1 and R_2 links, assuming that the clients are trying to send at their maximum rates?

- .001
- .1
- 1
- 10

QUESTION 3

2 points Save Answer

Maximum end-end throughput. Consider the scenario shown below, with a single source client sending to a server over two links of capacities R_1=10 Mbps and R_3=100 Mbps.



What is the maximum achievable end-end throughput (in Mbps, give an integer value) for the client-to-server pair, assuming that the client is trying to send at its maximum rate?

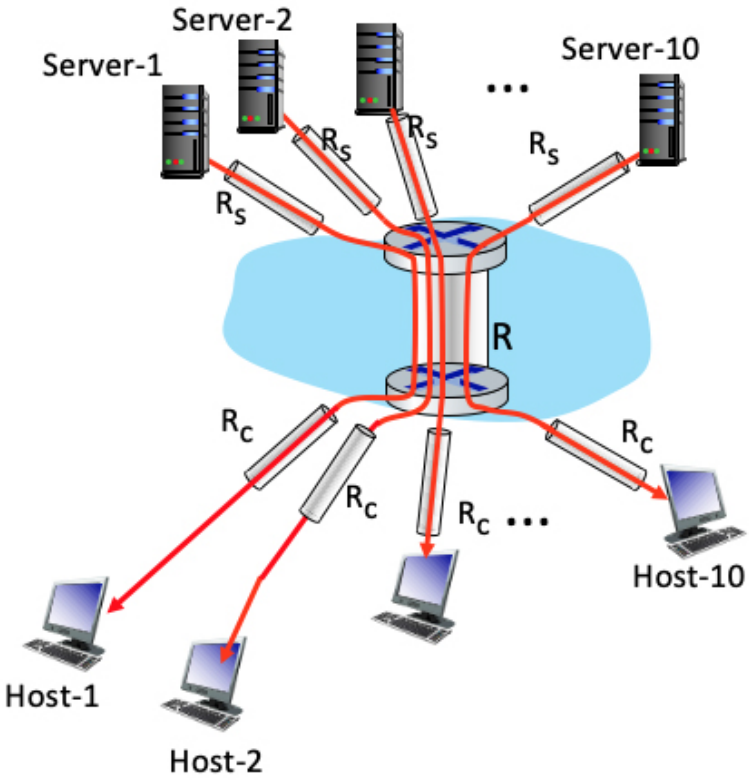
- 1 Mbps
- 10 secs.
- 10 Mbps
- .1 Mbps
- 100 Mbps

QUESTION 4

2 points Save Answer

End-to-end delay. Consider the scenario shown below, with 10

transmission capacity of R = 200 Mbps. Each link from a server has to the shared link has a transmission capacity of R_S = 25 Mbps. Each link from the shared middle link to a client has a transmission capacity of R_C = 50 Mbps.

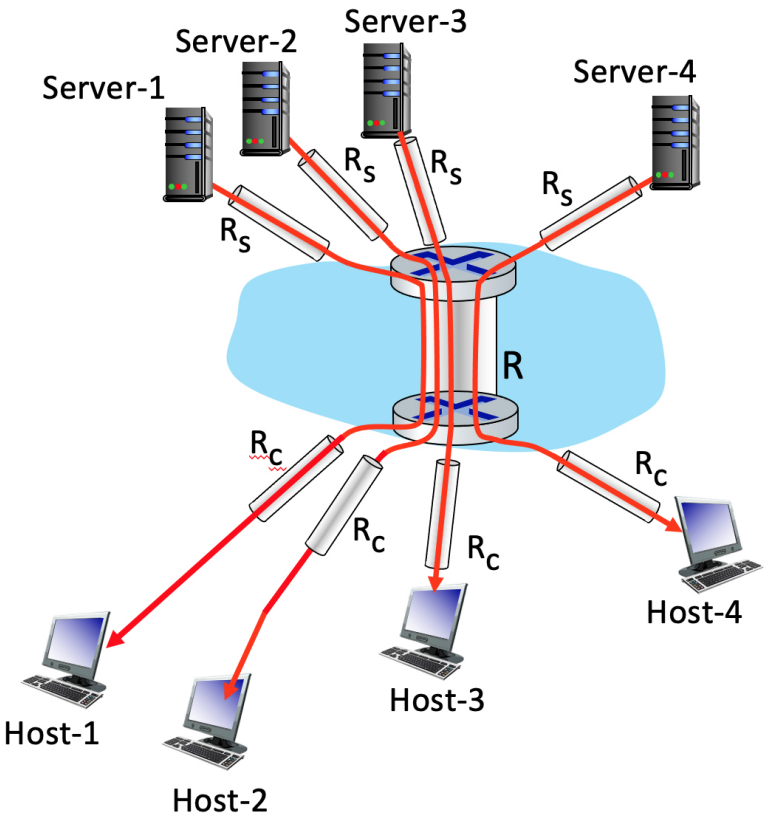


Now consider the three-hop path from Server-10 to host-10, assuming the values of R_S, R_C, and R given above. Assume also that a packet is 1000 bits long, and that each link has a propagation delay of 100 microseconds (1 microsec = 0.000001 sec). You can assume that queueing delay and nodal processing delays are zero. What is the total amount of time from when a server starts sending a packet until a host completely receives that packet?

- ○ 350 microseconds
- ○ 325 microseconds
- ○ 365 microseconds
- ○ 555 microseconds
- ○ 450 microseconds
- ○ 735 microseconds

QUESTION 5 **2 points** Save Answer

Link Utilization. Consider the scenario below where 4 TCP senders are connected to 4 receivers. The servers transmit to the receiving hosts at the fastest rate possible (i.e., at the rate at which the bottleneck link between a server and its destination is operating at 100% utilization, and is fairly shared among the connections passing through that link).



Suppose that R = 1 Gbps and R_c is 300 Mbps and R_s is 200 Mbps. Assuming that the servers are sending at their maximum rate possible, enter the link utilizations for the client links (whose rate is R_c) below.

Enter your answer as a decimal, of the form 1.00 (if the utilization is 1, or 0.xx if the utilization is less than 1, rounded to the closest xx).

The utilization of the client links, whose rate is R_c , is: [A]

QUESTION 6

0 points Save Answer

<<This question has been removed>>

- N/A
- N/A
- N/A
- N/A
- N/A

Reading an Internet RFC. The purpose of this question is to get you to read a (simple) RFC. Read through RFC 768 (User Datagram Protocol), which describes the UDP protocol that we will study in Chapter 3. RFC 768 is only three pages long (the RFC 2616 initial specification for HTTP 1.1 was 176 pages!).

Based on your reading of RFC 768, answer the following question: What is the purpose of the length field in the UDP header?

- ○ It specifies the length in bytes of the UDP header for this UDP segment.
- ○ It specifies the length in bytes of the payload data in the UDP segment.
- ○ It specifies the length in bytes of the UDP segment, including the header the payload data.
- ○ It specifies the maximum amount of time a UDP segment can wait at a host before being transmitted.

QUESTION 8

2 points Save Answer

Simple HTTP GET request response time. Suppose an HTTP client makes a request to the gaia.cs.umass.edu web server. The client has never before requested a given base object, nor has it communicated recently with the gaia.cs.umass.edu server. You can assume, however, that the client host knows the IP address of gaia.cs.umass.edu.

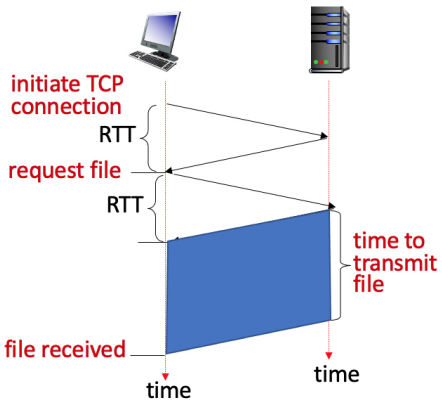
How many round trip times (RTTs) are needed from when the client first makes the request to when the base page is completely downloaded, assuming the time needed by the server to transmit the base file into the server's link is equal to 1/2 RTT and that the time needed to transmit the HTTP GET into the client's link is zero? (You should take into account any TCP setup time required before the HTTP GET is actually sent by the client, the time needed for the server to transmit the requested object, and any propagation delays not accounted for in these amounts of time.)

- ○ 0 RTT
- ○ 0.5 RTT
- ○ 1.5 RTT
- ○ 2.5 RTT
- ○ 3.5 RTT

QUESTION 9

2 points Save Answer

Download delays for 100 objects (HTTP 1.0). Consider an HTTP 1.0 client and server. The RTT delay between the client and server is 2 seconds. Suppose the time a server needs to transmit an object into its outgoing link is 3 seconds, as shown below for the first of these 100 requests.



You can assume that any other HTTP message not containing an object sent by the client and server has a negligible (zero) transmission time. Suppose the client makes 100 requests, one after the other, waiting for a reply to a request before sending the next request.

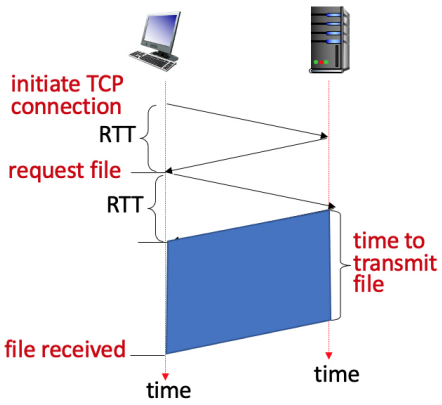
Using HTTP 1.0, how much time elapses between the client transmitting the first request, and the receipt of the last requested object?

- ○ 502 secs
- ○ 300 secs
- ○ 700 secs
- ○ 203 secs
- ○ 500 secs

QUESTION 10

2 points Save Answer

Download delays for 100 objects (HTTP 1.1). Consider an HTTP 1.1 client and server. The RTT delay between the client and server is 2 seconds. Suppose the time a server needs to transmit an object into its outgoing link is 3 seconds, as shown below for the *first* of these 100 requests.



You can assume that any other HTTP message not containing an object sent by the client and server has a negligible (zero) transmission time. Suppose the client makes 100 requests, one after the other, waiting for a reply to a request before sending the next request.

Using HTTP 1.1, how much time elapses between the client

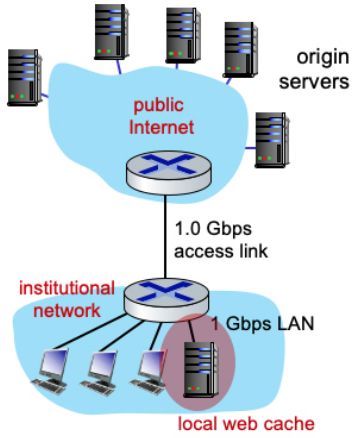
- ○ 203 secs
- ○ 700 secs
- ○ 500 secs
- ○ 502 secs
- ○ 300 secs

QUESTION 11

2 points Save Answer

Download delays for 100 objects (HTTP 1.1 with local web caching). Consider an HTTP 1.1 client and server. The RTT delay between the client and server is 2 seconds. Suppose the time a server needs to transmit an object into its outgoing link is 3 seconds.

There is also a local web cache, as shown in the figure below, with negligible (zero) propagation delay and object transmission time. The client makes 100 requests one after the other, waiting for a reply before sending the next request. All requests first go to the cache (which also has a 2.0 sec. RTT delay to the server and a 0.1sec. RTT to the client).



How much time elapses between the client transmitting the first request, and the receipt of the last requested object, *assuming no use of the IF-MODIFIED-SINCE header line anywhere*, *and assuming that 50% of the objects requested are "hits"* (found) in the local cache?

- ○ 208 secs
- ○ 260 secs
- ○ 110 secs
- ○ 300 secs
- ○ 160 secs

QUESTION 12

2 points Save Answer

HTTP Performance: access link utilization. Consider again the scenario in the question directly above, where there is again a web cache in the lower network. Suppose that:

- The link capacties R_C and R_S are so large that they are never a bottleneck and can be ignored.
- Each object requested by a client is 1 Mbits (1×10^6 bits) in size
- Each of four clients is individually making HTTP GET requests at a rate of 10 requests per second.

Assuming that an HTTP request is *never* satified in the cache, what is the utilization of the link of capacity R_I?

- ○ .40
- ○ 1.0
- ○ .5
- ○ .10
- ○ .20

QUESTION 13 **2 points** Save Answer

UDP client-side socket actions. Match the genera*l client-side* action stated with the specific UDP socket-related action that implements it.

- \- Create a socket.
- \- When sending to the server, this is how a specific server is identified.
- \- Send to server, using this socket.

A. Use the call `socket(AF_INET, SOCK_DGRAM)`

B. The client explicitly includes the destination IP address and port #, when sending

C. Send using the socket created using `socket(AF_INET, SOCK_DGRAM)`

D. Use the call `socket(AF_INET, SOCK_STREAM)`

E. As the result of an `accept()`, a new socket is created, which binds the client and server together via this new socket without the need to to explicitly specify the destination IP address and port # when sending

F. Send using a socket not explicitly created via a call to `socket()`

G. Send using the socket created using `socket(AF_INET, SOCK_STREAM)`

QUESTION 14 **2 points** Save Answer

- Create a socket.
- When sending to a client, this is how a specific client is identified.
- Send to client, using this socket.
- When sending to the client, this is how the server knows the client IP address and port number.

A. Use the call `socket(AF_INET, SOCK_DGRAM)`

B. The client explicitly includes the destination IP address and port #, when sending

C. Send using the socket created using `socket(AF_INET, SOCK_DGRAM)`

D. Use the call `socket(AF_INET, SOCK_STREAM)`

E. As the result of an `accept()`, a new socket is created, which binds the client and server together via this new socket without the need to to explicitly specify the destination IP address and port # when sending

F. Send using a socket not explicitly created via a call to `socket()`

G. Send using the socket created using `socket(AF_INET, SOCK_STREAM)`

H. The server determines the client IP address and port # from an earlier datagram sent by this client.

QUESTION 15

2 points Save Answer

TCP client-side socket actions. Match the genera*l client-side* action stated with the specific TCP socket-related action that implements it.

- Create a socket.
- When sending to a server, this is how a specific server is identified.
- Send to server, using this socket.

A. Use the call `socket(AF_INET, SOCK_STREAM)`

B. The client uses `connect()` to explicitly bind its socket to specific server, and so the server IP address and port number need not be explicitly stated in a send operation.

C. Send using the socket created using `socket(AF_INET, SOCK_STREAM)`

D. Use the call `socket(AF_INET, SOCK_DGRAM)`

E. As the result of an `accept()`, a

without the need to to explicitly specify the destination IP address and port # when sending

F. Send using a socket not explicitly created via a call to `socket()`

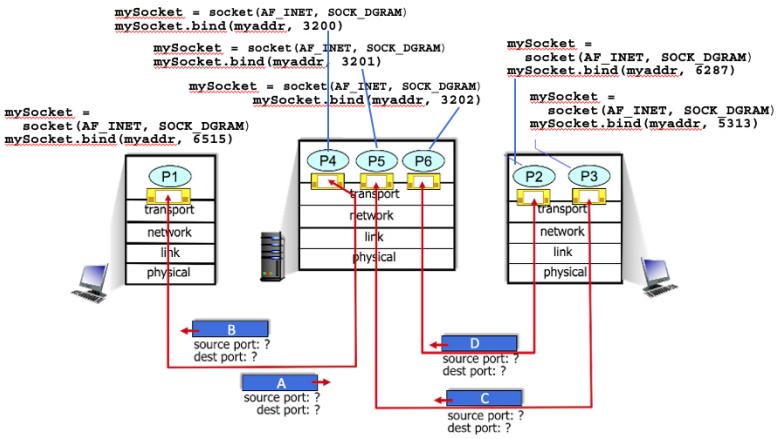
G. Send using the socket created using `socket(AF_INET, SOCK_DGRAM)`

H. The client must explicitly include the server's IP address, port #, when sending

QUESTION 16

2 points Save Answer

UDP multiplexing and demultiplexing. Consider the figure below, with 6 sockets shown across the network, and the corresponding Python code at each host. There are four UDP segments in flight. Match the source and destination port numbers for each segment with a value below.



Note: you can generate/solve/practice many similar instances of this question here.

- Segment A source port #
- Segment B source port #
- Segment C source port #
- Segment C destination port #
- Segment D source port #
- Segment D destination port #

A. 6515
B. 3201
C. 3202
D. 5313
E. 3200
F. 6287

QUESTION 17

2 points Save Answer

Internet Checksum. Consider the two sixteen bit numbers:

```
10110100 01000110
01001000 01101111
```

Compute the Internet Checksum of these two values

Enter the 2 bytes each as an 8-bit number with only 0's and 1's, and make a single blank space between the two 8-bit numbers (e.g., 01010101 00101000).

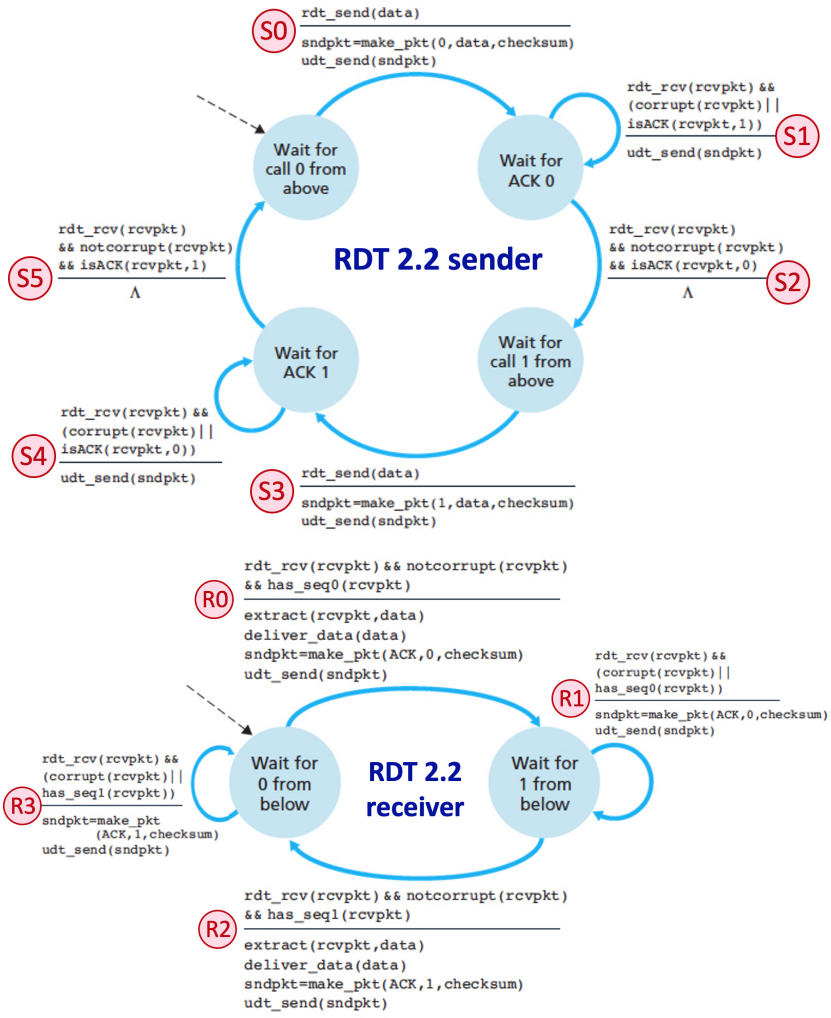
[Note: you can generate/solve/practice many similar instances of this question here.]

QUESTION 18

4 points Save Answer

The RDT 2.2 protocol. Consider the rdt 2.2 sender and receiver below, with FSM transitions labeled in red.

Which of the following sequences of transitions could possibly occur as a result of an initial rdt_send() call at the sender (with no messages initially in the channel) , and possible later message corruption and subsequent error recovery?



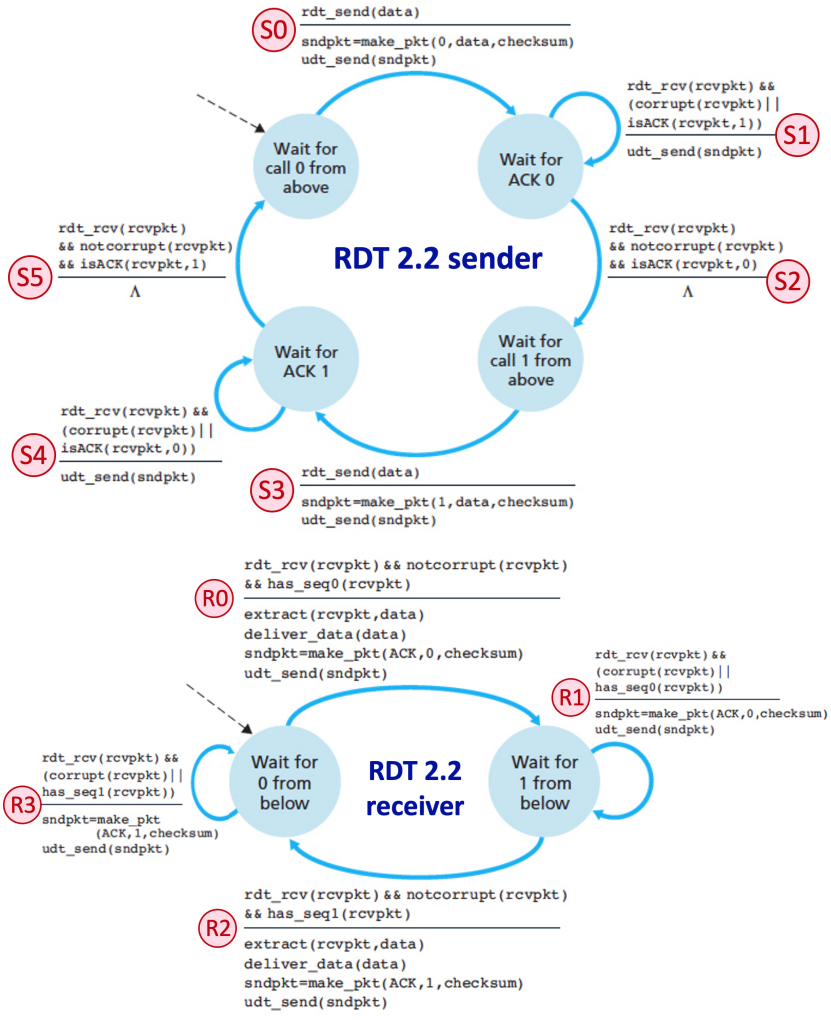
- [ ] S0, R0, S2, S3, R2, S5
- [ ] S0, R0, S2, R2
- [ ] S0, R0, S1, R2
- [ ] S0, R0, S1, R1, S1
- [ ] S0, R0, S1, R1
- [ ] S0, R0, S2, R1

QUESTION 19

2 points Save Answer

The RDT 2.2 protocol. Consider the rdt 2.2 sender and receiver below, with FSM transitions labeled in red. Complete the sequence of transitions that the sender and receiver FSMs would make, in an global order, to deliver two messages from sender to receiver, assuming no errors occur (including the ACK received at the sender for the second packet).

The transition sequence is: S0, R0, <t1>, <t2>, <t3>, <t4>. Match unspecified transitions <t1>, <t2>, <t3> and <t4> with a labeled transition from the figures below.



- Transition <t1> is: A. R2

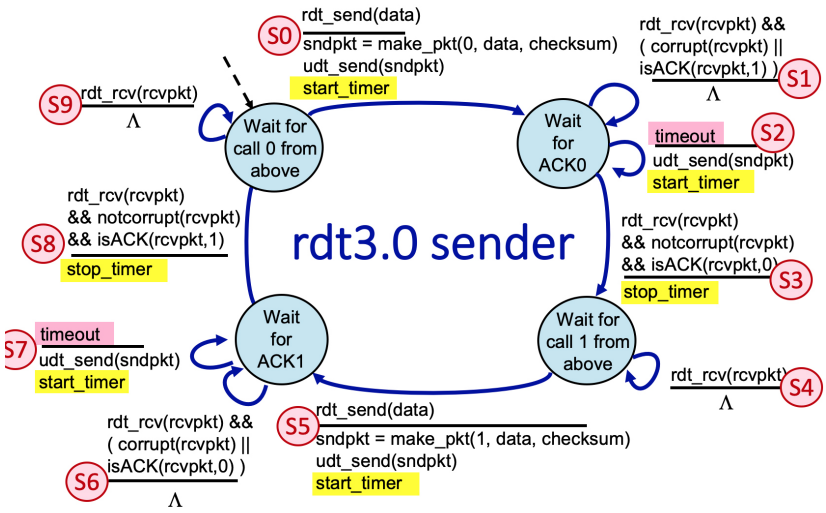
- Transition <t3> is: C. S3
- Transition <t4> is: D. S2

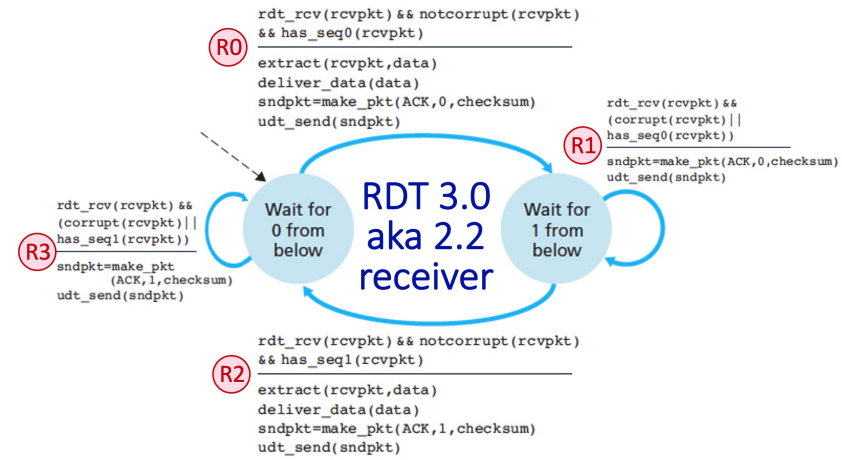
QUESTION 20

2 points Save Answer

The RDT 3.0 protocol. Consider the rdt3.0 sender and receiver shown below, with FSM transitions labeled in red. Complete the sequence of transitions that the sender and receiver FSMs would make, in an global order, to deliver two messages from sender to receiver, assuming no errors occur (including the ACK received at the sender for the second packet).

The transition sequence is: S0, R0, *t1, t2, t3, t4*. Match unspecified transitions *t1, t2, t3* and *t4* with a labeled transition from the figures below.





- Transition *t1* is: A. S5
- Transition *t2* is: B. S3
- Transition *t3* is: C. S8
- Transition *t4* is: D. R2

QUESTION 21

2 points Save Answer

TCP RTT estimation and timeout value. Suppose that TCP's current estimated values for the round trip time (*estimatedRTT*) and deviation in the RTT (*DevRTT*) are 300 msec and 13 msec, respectively. Suppose that the next two measured RTTs are 330 msec and 240 msec respectively. We want to calculate TCP's RTT estimate, and the value of TCP's timeout interval. Note that given a new measured RTT, you should first compute *devRTT*, then *estimatedRTT* (the textbook incorrectly reverses those computations), and then (lastly) the timeout interval. Use the values of $\alpha = 0.125$, $\beta = 0.25$.

Following the newly measured RTT of 330 msec, what is the new value for *devRTT* in msec? [Note: round your answer to the nearest msec - enter an integer value, do not include any decimal places/point or any leading zeros, but keep your full resolution value of *devRTT* handy and use it in any later calculations you perform using *devRTT*.

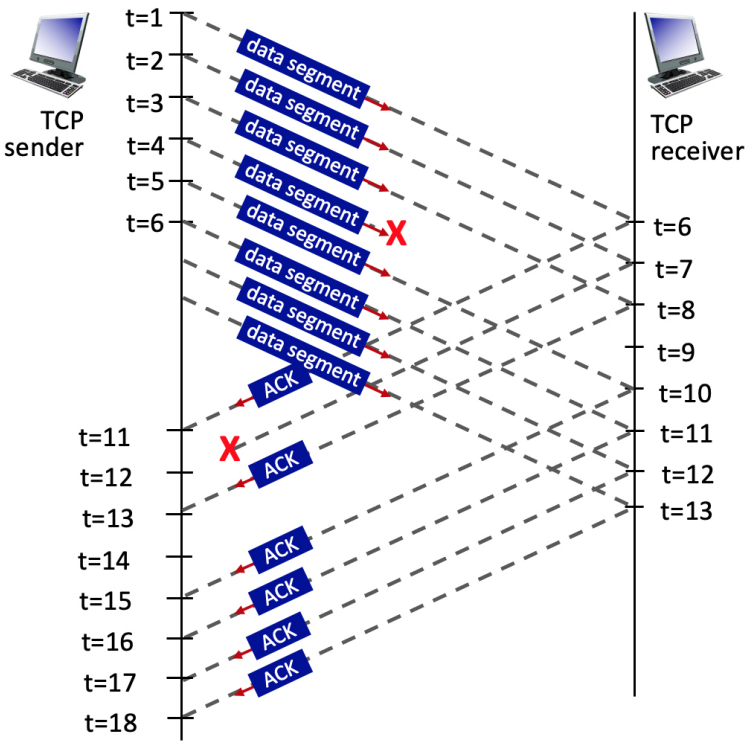
[Note: you can generate/solve/practice many similar instances of this question here.]

QUESTION 22

2 points Save Answer

TCP sequence and ACK numbers. Consider the figure below, where a TCP sender sends 8 TCP segments at *t = 1, 2, 3, 4, 5, 6, 7, 8.* Suppose the initial value of the sequence number is 0 and every segment sent to the receiver each contains 100 bytes. The delay between the sender and receiver is 5 time units, and so the first segment arrives at the receiver at t = 6. The ACKs sent by the receiver at *t = 6, 7, 8, 10, 11, 12* are shown. The TCP segments (if any) sent by the sender at t = 11, 13, 15, 16, 17, 18 are *not* shown.

The segment sent at t=4 is lost, as is the ACK segment sent at t=7.



200

100

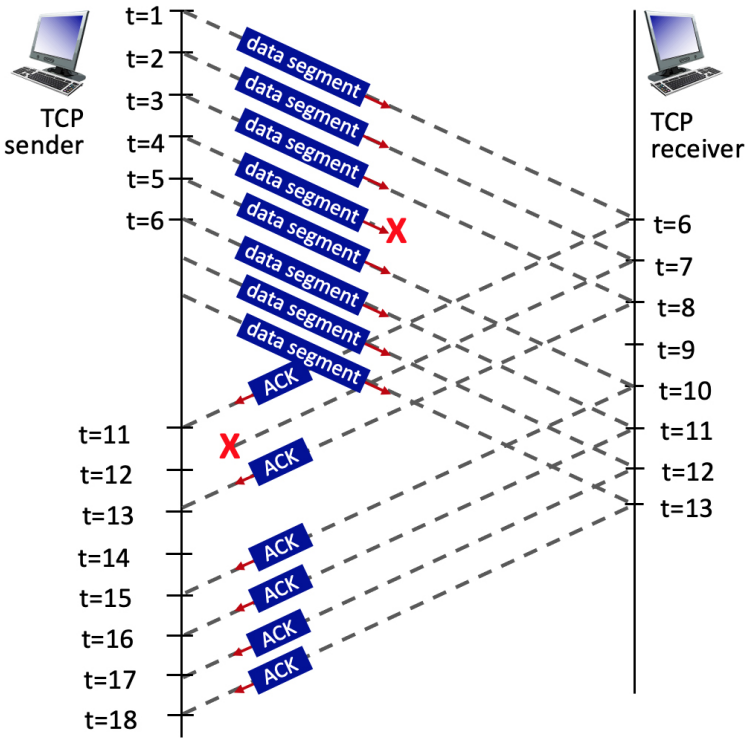
2

QUESTION 23

2 points Save Answer

TCP sequence and ACK numbers. Consider the figure below, where a TCP sender sends 8 TCP segments at *t = 1, 2, 3, 4, 5, 6, 7, 8.* Suppose the initial value of the sequence number is 0 and every segment sent to the receiver each contains 100 bytes. The delay between the sender and receiver is 5 time units, and so the first segment arrives at the receiver at t = 6. The ACKs sent by the receiver at *t = 6, 7, 8, 10, 11, 12* are shown. The TCP segments (if any) sent by the sender at t = 11, 13, 15, 16, 17, 18 are *not* shown.

The segment sent at t=4 is lost, as is the ACK segment sent at t=7.



What is the ACK value carried in the receiver-to-sender ACK sent at t = 10?

100

3

None of these other answers.

300

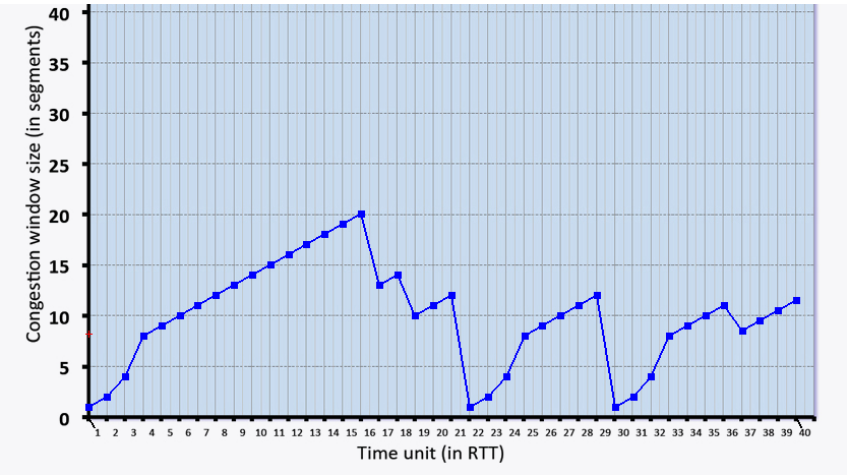
400

200

Phases of TCP congestion control. 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.

During which of the following intervals of time is TCP performing slow start?

[Note: you can generate/solve/practice many similar instances of this question here.]



- [ ] [1,3]
- [ ] [4,15]
- [ ] 16
- [ ] 17
- [ ] 18
- [ ] [19,20]
- [ ] 21
- [ ] [22,24]

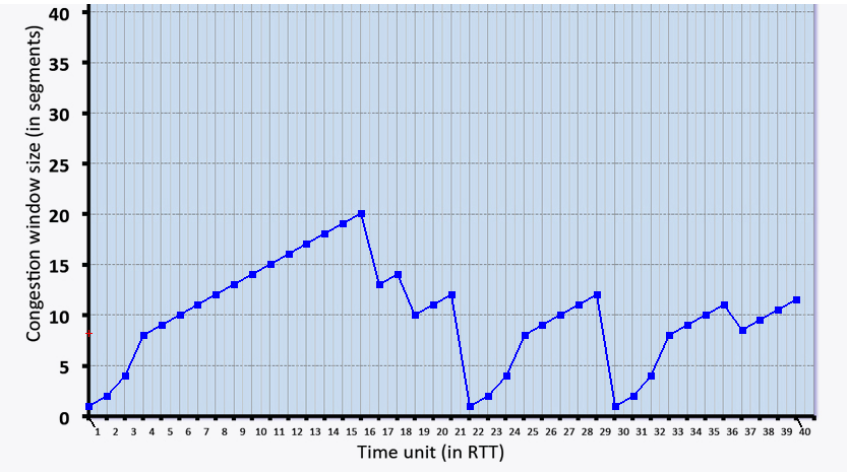
QUESTION 25

2 points Save Answer

Phases of TCP congestion control. 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.

During which of the following intervals of time is TCP performing congestion avoidance?

[Note: you can generate/solve/practice many similar instances of this question here.]



- [ ] [1,3]
- [ ] [4,15]
- [ ] 16
- [ ] 17
- [ ] 18
- [ ] [19,20]
- [ ] 21
- [ ] [22,24]