Practice Questions:
Congestion Control and Queuing

COS 461: Computer Networks

http://www.cs.princeton.edu/courses/archive/spr13/cos461/

One node on an Ethernet network uses TCP to send data to another node on the same network. If there are no other nodes transmitting on the network, can there be any collisions?

A. Yes
B. No

One node on an Ethernet network uses TCP to send data to another node on the same network. If there are no other nodes transmitting on the network, can there be any collisions?

A. Yes – *Can collide with TCP ACKs*
B. No

Ben Bitdiddle's home network connection can upload at 125,000 bytes/second. His router has a 100,000 byte first in first out buffer for packets awaiting transmission.

If the buffer is completely full, how long will it take for the buffer to clear?

A. 0.4 seconds
B. 0.6 seconds
C. 0.8 seconds
D. 1 second
E. 1.25 seconds

Ben Bitdiddle's home network connection can upload at 125,000 bytes/second. His router has a 100,000 byte first in first out buffer for packets awaiting transmission.

If the buffer is completely full, how long will it take for the buffer to clear?

A. 0.4 seconds
B. 0.6 seconds
C. 0.8 seconds
D. 1 second
E. 1.25 seconds

Ben Bitdiddle's home network connection can upload at 125,000 bytes/second. His router has a 100,000 byte first in first out buffer for packets awaiting transmission.

At time 0, Ben's client starts sending 1,000 byte packets at 150 packets/s. When will the first packet be dropped by the router?

A. 2 seconds
B. 3 seconds
C. 4 seconds
D. Buffer will never discard a packet in this case

Ben Bitdiddle's home network connection can upload at 125,000 bytes/second. His router has a 100,000 byte first in first out buffer for packets awaiting transmission.

At time 0, Ben's client starts sending 1,000 byte packets at 150 packets/s. When will the first packet be dropped by the router?

A. 2 seconds
B. 3 seconds
C. 4 seconds
D. Buffer will never discard a packet in this case

Alyssa P. Hacker and Ben Bitdiddle communicate over a link with capacity of 100 pkts / sec. The latency (RTT) on this link is 100 ms.

If a sliding window protocol with acknowledgement packets is used, and there is a *FIXED* window size of 4 packets, what is the maximum rate of traffic on the link?

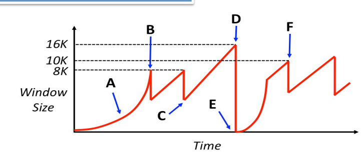
A. 20 pkts / s
B. 40 pkts / s
C. 80 pkts / s
D. 100 pkts /s

Alyssa P. Hacker and Ben Bitdiddle communicate over a link with capacity of 100 pkts / sec. The latency (RTT) on this link is 100 ms.

If a sliding window protocol with acknowledgement packets is used, and there is a *FIXED* window size of 4 packets, what is the maximum rate of traffic on the link?

A. 20 pkts / s
B. 40 pkts / s
C. 80 pkts / s
D. 100 pkts /s

TCP CONGESTION CONTROL

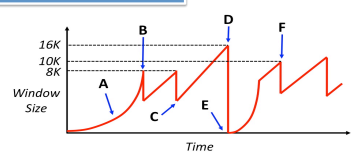


1. Name the event at B which occurs that causes the sender to decrease its window

(a) Triple Duplicate Ack
(b) Slow Start
(c) Packet loss
(d) Time out

10

TCP CONGESTION CONTROL

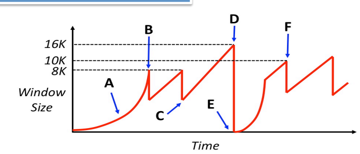


1. Name the event at B which occurs that causes the sender to decrease its window

(a) Triple Duplicate Ack
(b) Slow Start
(c) Packet loss
(d) Time out

11

TCP CONGESTION CONTROL

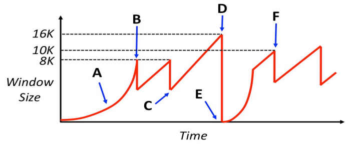


2. Does the event at B necessitate that the network discarded a packet ?

(a) Yes
(b) No
(c) Don't know

12

TCP CONGESTION CONTROL



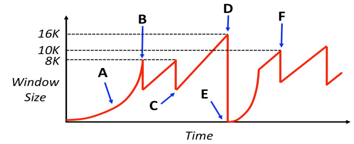
2. Does the event at B necessitate that the network discarded a packet ?

(a) Yes
(b) No
(c) Don't know

No. It could be due to either reordering or queuing or asymmetric paths.

13

TCP CONGESTION CONTROL

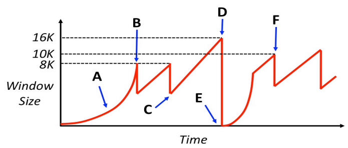


3. Name the event at D which occurs that causes the sender to decrease its window.

(a) Triple Duplicate Ack
(b) Slow Start
(c) Packet loss
(d) Time out

14

TCP CONGESTION CONTROL

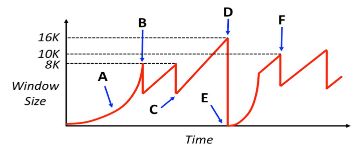


3. Name the event at D which occurs that causes the sender to decrease its window.

(a) Triple Duplicate Ack
(b) Slow Start
(c) Packet loss
(d) Time out

15

TCP CONGESTION CONTROL

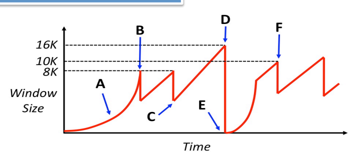


4. Does the event at D necessitate that the network discarded a packet

(a) Yes
(b) No
(c) Don't know

16

TCP CONGESTION CONTROL



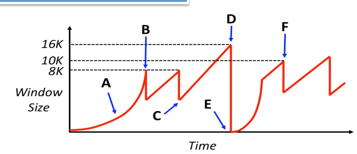
4. Does the event at D necessitate that the network discarded a packet

(a) Yes
(b) No
(c) Don't know

No. Congestion in either direction could cause RTT > RTO (retrans. timeout).

17

TCP CONGESTION CONTROL

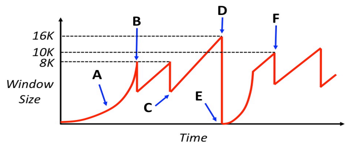


5. For a lightly-loaded network, is the event at D MORE likely or LESS likely to occur when the sender has multiple TCP segments outstanding

(a) MORE
(b) LESS
(c) ALMOST SAME

18

TCP CONGESTION CONTROL

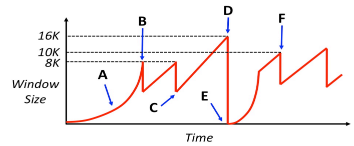


5. For a lightly-loaded network, is the event at D MORE likely or LESS likely to occur when the sender has multiple TCP segments outstanding

(a) MORE
(b) LESS
(c) ALMOST SAME

19

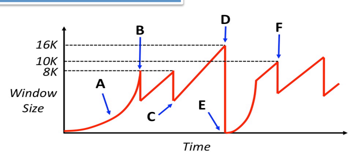
TCP CONGESTION CONTROL



6. Consider the curved slope labeled by point A. Why does the TCP window behave in such a manner, rather than have a linear slope? (Put another way, why would it be bad if region A had a linear slope?)

20

TCP CONGESTION CONTROL

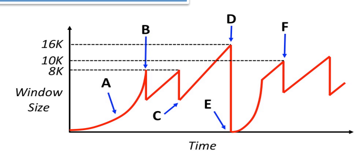


6. Consider the curved slope labeled by point A. Why does the TCP window behave in such a manner, rather than have a linear slope? (Put another way, why would it be bad if region A had a linear slope?)

This "slow-start" period quickly discovers the maximum acceptable throughput that the path supports – otherwise, AI (additive increase) could take too long (each a full RTT).

21

TCP CONGESTION CONTROL



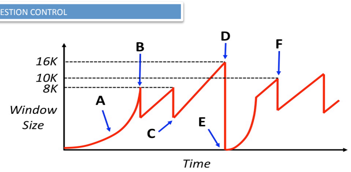
Assume that the network has an MSS of 1000 bytes and the round-trip-time between sender and receiver of 100 milliseconds.

Assume at time 0 the sender attempts to open the connection.

Also assume that the sender can "write" a full window's worth of data instantaneously, so the only latency you need to worry about is the actual propagation delay of the network.

22

TCP CONGESTION CONTROL

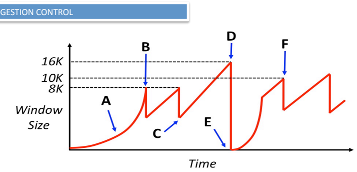


RTT = 100ms, MSS = 1000 bytes
7. How much time has progressed by point B ?

(a) 200ms
(b) 300ms
(c) 400ms
(d) 600ms
(e) 700ms

23

TCP CONGESTION CONTROL

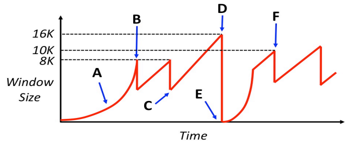


RTT = 100ms, MSS = 1000 bytes
7. How much time has progressed by point B ?

(a) 200ms
(b) 300ms
(c) 400ms
(d) 600ms
(e) 700ms

24

TCP CONGESTION CONTROL

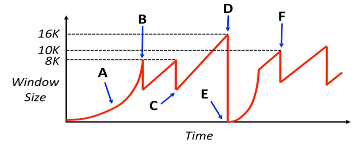


RTT = 100ms, MSS = 1000 bytes
8. How much time has progressed between points C and D?

(a) 800ms
(b) 1000ms
(c) 1200ms
(d) 1400ms

25

TCP CONGESTION CONTROL

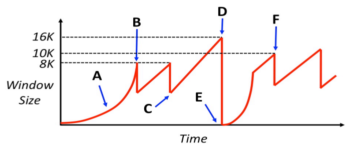


RTT = 100ms, MSS = 1000 bytes
8. How much time has progressed between points C and D?

(a) 800ms
(b) 1000ms
(c) 1200ms
(d) 1400ms

26

TCP CONGESTION CONTROL

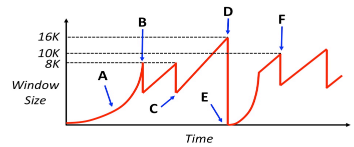


RTT = 100ms, MSS = 1000 bytes
9. How much time has progressed between points E and F?

(a) 400ms
(b) 600ms
(c) 800ms
(d) 900ms

27

TCP CONGESTION CONTROL

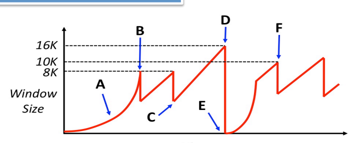


RTT = 100ms, MSS = 1000 bytes
9. How much time has progressed between points E and F?

(a) 400ms
(b) 600ms
(c) 800ms
(d) 900ms

28

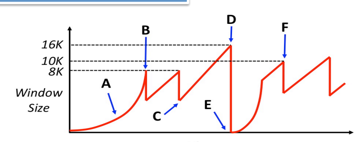
TCP CONGESTION CONTROL



10. If the sender shares its network with other clients whose traffic traverses the same IP routers, give one explanation for why point D is higher than point B?

29

TCP CONGESTION CONTROL



10. If the sender shares its network with other clients whose traffic traverses the same IP routers, give one explanation for why point D is higher than point B?

Changing cross-traffic by other concurrent senders across same routers.

30

1 In slow start, a sender doubles its window size every RTT if all sent packets were acknowledged

T/F

1 In slow start, a sender doubles its window size every RTT if all sent packets were acknowledged

T/F

1 In slow start, a sender doubles its window size every RTT if all sent packets were acknowledged

T/F

2 In steady state, a sender increases its window size by one packet for each acknowledgement

T/F

1 In slow start, a sender doubles its window size every RTT if all sent packets were acknowledged

T/F

2 In steady state, a sender increases its window size by one packet for each acknowledgement

T/**F –** increases by one MSS for every RTT

1 In slow start, a sender doubles its window size every RTT if all sent packets were acknowledged

T/F

2 In steady state, a sender increases its window size by one packet for each acknowledgement

T/**F –** increases by one MSS for every RTT

3 A sender that underestimates the round-trip time of a connection may unnecessarily induce a TCP timeout

T/F

1 In slow start, a sender doubles its window size every RTT if all sent packets were acknowledged

T/F

2 In steady state, a sender increases its window size by one packet for each acknowledgement

T/**F –** increases by one MSS for every RTT

3 A sender that underestimates the round-trip time of a connection may unnecessarily induce a TCP timeout

T/F

1 In slow start, a sender doubles its window size every RTT if all sent packets were acknowledged

T/F

2 In steady state, a sender increases its window size by one packet for each acknowledgement

T/**F** – increases by one MSS for every RTT

3 A sender that underestimates the round-trip time of a connection may unnecessarily induce a TCP timeout

T/F

4 After detecting packet loss through a timeout, TCP halves its window size as a response to the path congestion

T/F

37

1 In slow start, a sender doubles its window size every RTT if all sent packets were acknowledged

T/F

2 In steady state, a sender increases its window size by one packet for each acknowledgement

T/**F** – increases by one MSS for every RTT

3 A sender that underestimates the round-trip time of a connection may unnecessarily induce a TCP timeout

T/F

4 After detecting packet loss through a timeout, TCP halves its window size as a response to the path congestion

T/**F** – TCP resets its window size to one MSS

38

For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers' queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

•Drop-tail solves the full queue problem T/F

39

For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers' queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

•Drop-tail solves the full queue problem T/**F**

40

For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers' queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

•Drop-tail solves the full queue problem T/**F**

•RED solves the full queue problem T/F

41

For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers' queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

•Drop-tail solves the full queue problem T/**F**

•RED solves the full queue problem **T**/F

42

For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers' queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

•Drop-tail solves the full queue problem T/**F**

•RED solves the full queue problem **T**/F

•RED solves the lockout problem for TCP flows T/F

43

For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers' queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

•Drop-tail solves the full queue problem T/**F**

•RED solves the full queue problem **T**/F

•RED solves the lockout problem for TCP flows **T**/F

44

1 Drop-tail has fewer burst losses than RED

T/F

45

1 Drop-tail has fewer burst losses than RED

T/**F**

46

1 Drop-tail has fewer burst losses than RED

T/**F**

2 Both drop-tail and RED can be used with ECN (explicit congestion notification), so the router can signal congestion to the sender without dropping a packet

T/F

47

1 Drop-tail has fewer burst losses than RED

T/**F**

2 Both drop-tail and RED can be used with ECN (explicit congestion notification), so the router can signal congestion to the sender without dropping a packet

T/**F**

48

1 Drop-tail has fewer burst losses than RED

T/**F**

2 Both drop-tail and RED can be used with ECN (explicit congestion notification), so the router can signal congestion to the sender without dropping a packet

T/**F**

3 RED drops every incoming packet with some probability P > 0

T/F

49

1 Drop-tail has fewer burst losses than RED

T/**F**

2 Both drop-tail and RED can be used with ECN (explicit congestion notification), so the router can signal congestion to the sender without dropping a packet

T/**F**

3 RED drops every incoming packet with some probability P > 0

T/**F**

50