**Assignment #004**

1. Consider the following behavior of a TCP connection (using the congestion control algorithm we learned in class).

At time 0, a TCP sender initiates a connection. As soon as the connection is established, the TCP sender will begin sending data. The MSS is 1KB and RTT is 100 ms.

1) Assuming the connection does not lose any data or experience any timeouts, at what time will the sender’s congestion window be 16KB? (Assuming *threshold* is 32MSS)

**Answer:**

**首先经过一个RTT建立TCP。**

**threshold =32MSS=32KB > 16KB。**

**根据the congestion control algorithm，在congestion window达到16KB这个过程都是slow start过程。**

**所以在时间500ms时，congestion window达到16KB。**

Right after the sender’s congestion window has reached a size of 16KB, a timeout occurs. After the timeout is detected, the sender continues sending more data over the established connection.

2) Assuming no additional packets loss or timeouts, how long (since the observed timeout) will it take for the congestion window to build to size 14KB?

**Answer：**

**在发生timeout那一刻之前，congestion window=16KB，threshold=32KB。**

**根据the congestion control algorithm，调整为congestion window=1MSS=1KB，threshold=8KB。**

**之后进入慢启动，3个RTT即300ms后，congestion window=8KB。**

**之后进入线性增加，6个RTT即600ms后，congestion window=14KB。**

**所以需要300ms+600ms=900ms，congestion window达到14KB。**

3) While its congestion window is at 14KB, the sender receives triple duplicate acknowledgements for the same sequence number. How long after receiving the third duplicate acknowledgement will it take for the sender’s congestion window to be at least 9KB again?

**Answer：**

**收到3个冗余ACK时，congestion window=14KB，threshold=8KB。**

**根据the congestion control algorithm，应该进入到拥塞避免状态，**

**调整为congestion window=threshold=7KB。**

**之后线性增加，2个RTT即200ms后，congestion window=9KB。**

**所以需要200ms，congestion window达到9KB。**

1. Consider a scenario with two hosts, Alice and Bob. A web server running on Alice is trying to send data to a browser on Bob. For each TCP connection, Alice’s TCP stack maintains a send buffer of 512 bytes and Bob’s TCP stack maintains a receive buffer of 1024 bytes. For simplicity, assume TCP sequence numbers began at 0 in this problem.

1) Bob’s stack received up to byte 560 in order from Alice, although its browser has only read up to the first 60 bytes. What will be the ***ACK#***and ***rcvr window size***in the TCP headers that Bob next sends to Alice?

**Answer：**

**ACK#=561**

***rcvr window size* = 1024-(560-60) = 524bytes。**

2) Later in the same connection, Alice’s congestion window is set to 1 MSS = 536 bytes and the advertised flow-control window from Bob is 560 bytes. The last ***ACK#*** that Alice received from Bob is byte 700, and the last byte that Alice sends to Bob is byte 900.

A) What is the smallest byte number that Bob will not accept?

**Answer：**

**Bob还可以接收560-(900-700)=360bytes**

**则Bob还能接收到最后一个的byte的number为1260，即number为1261及以后的byte都不能被Bob接收到。**

**所以the smallest byte number that Bob will not accept is 1261。**

B) Assuming that Alice doesn’t receive any more ACKs and her window does not change, what is the greatest byte number that Alice can send?

**Answer：**

**Alice收不到任何ACK，说明701-900的包丢失，所以要重新传这200个bytes，又因为一次最多能传min（536bytes，560bytes）=536bytes。**

**加上之前传的700bytes，the greatest byte number that Alice can send is 1236。**

C) Again assuming that Alice doesn’t receive any additional ACKs, how many more bytes can the web server running on Alice write to its network socket before blocking?

**Answer：**

**因为还有200bytes需要重新传，所以这200bytes仍在Alice的缓存中，那么Alice的缓存中还有312bytes可以供Alice发送其他东西。**

**学号：1210565**

**2015年5月15日星期五**