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 slowstart 过程。

$$\mathbf{100}\boldsymbol{ms}+\left(\log_2\frac{\mathbf{16}\boldsymbol{KB}}{\mathbf{1}\boldsymbol{KB}}\right)\times\mathbf{100}\boldsymbol{ms}=\mathbf{500}\boldsymbol{ms}$$

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。

2. 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