[简答题]
Chapter 3, R9. In our rdt protocols, why did we need to introduce sequence numbers?
[参考答案]
接收方使用序列号来判别接收到的数据包是否是按顺序的：是一个新的数据包，还是重复的数据包。

[简答题]
Chapter 3, R10. In our rdt protocols, why did we need to introduce timers?
[参考答案]
用于处理数据包/ACK/NAK 的丢失问题，使用计时器来实现超时重传。如果在数据包的计时器结束时没有收到它的 ACK/NAK，则判定该数据包丢失，重传该数据包。

[简答题]
Chapter 3, R11. Suppose that the roundtrip delay between sender and receiver is constant and known to the sender. Would a timer still be necessary in protocol rdt 3.0 , assuming that packets canbe lost? Explain.
[参考答案]
还需要设置timer。如果有包的丢失，不论是数据包data还是ACK/NAK包的丢失，sender都不会收到receiver的ACK/NAK。这时sender处于等待接收ACK/NAK的死锁状态。因此需要设置timer，在timer超时后，sender启动“重传数据包”的事件，打破死锁。和RTT（round trip time/delay）值是否固定没有关系。如果RTT值固定，则timer就设置成固定的超时时间。否则，超时时间则随RTT值变化。

[简答题]
Chapter 3, P2. Consider Figure 3.5 . What are the source and destination port values in the segments flowing from the server back to the clients’ processes? What are the IP addresses in then network-layer datagrams carrying the transport-layer segments?
[参考答案]
（1）从Server B到host A：
Source port =80, source IP address = B, dest port = 26145, dest IP address = A
（2）从Server B到host A（左侧进程）：
Source port =80, source IP address = B, dest port = 7532, dest IP address = C
（3）从Server B到host A（右侧进程）：
Source port =80, source IP address = B, dest port = 26145, dest IP address = C

[简答题]
Chapter 3, P4.
a. Suppose you have the following 2 bytes: 01011100 and 01100101. What is the 1s complement of the sum of these 2 bytes?
b. Suppose you have the following 2 bytes: 11011010 and 01100101. What is the 1s complement of the sum of these 2 bytes?
c. For the bytes in part (a), give an example where one bit is flipped in each of the 2 bytes and yet the 1s complement doesn’t change.
[参考答案]

a. 相加结果为11000001，补码是00111110。
b. 相加结果为01000000，补码是10111111。
c. 第一个bytes改为01010100，第二个bytes改为01101101。

[简答题]
Chapter 3, P6. Consider our motivation for correcting protocol rdt2.1 . Show that the receiver, shown in Figure 3.60 , when operating with the sender shown in Figure 3.11 , can lead the sender and receiver to enter into a deadlock state, where each is waiting for an event that will never occur.
[参考答案]
假设sender的状态是Wait for call 1 from above，而receiver的状态是Wait for 1 from below。首先sender发送一个sequence number为1的数据包，并且转换到Wait for ACK or NAK 1的状态。假设receiver正确接收到sequence number为1的数据包，并且向sender发送了ACK，状态转换为Wait for 0 from below，即等待sequence number为0的数据包。这时如果出现ACK损坏，当rdt2.1协议的sender得到损坏的ACK，那么它会重新发送sequence number为1的数据包。但是receiver还在等待sequence number为0的数据包，并且在未接收到sequence number为0的数据包时发送NAK。之后sender会一直发送sequence number为1的数据包，而receiver会一直发送NAK，二者的状态都不会改变。

[简答题]
Chapter 3, P15. Consider the cross-country example shown in Figure 3.17 . How big would the window size have to be for the channel utilization to be greater than 98 percent? Suppose that the size of a packet is 1,500 bytes, including both header fields and data.
[参考答案]
已知数据包长度L=1500bytes，链路速率R=10^9bps。
则一个数据包的传输时间t=L/R=1500*8bit/10^9bps=0.012ms
设窗口大小为n，则信道利用率u=n*t/(RTT+t)=(0.012n)/30.012
令u>0.98，得到窗口大小至少应该为2451(取整)。

[简答题]
Chapter 3, P23. Consider the GBN and SR protocols. Suppose the sequence number space is of size k. What is the largest allowable sender window that will avoid the occurrence of problems such as that in Figure 3.27 for each of these protocols?
[参考答案]
为了避免图3.27中的场景，应该避免receiver窗口leading edge(即receiver窗口最右端的序列号)与sender窗口的trailing edge(即sender窗口最左端的序列号)重叠。
假设窗口大小为w，receiver的窗口中，正在等待的数据包是m（即窗口最左端的序列号是m），在这种情况下，它此时的窗口是[m,m+w-1]。在此之前它已经接收了数据包m-1和w-1个数据包并且发送了这w个数据包的ACK。如果发送方还没有接收到这w个数据包的ACK，那么值为[m-w,m-1]的ACK消息可能仍在链路上，一段时间之后会传回sender。此时sender尚未收到这w个数据包的ACK，那么sender的窗口将保持[m-w,m-1]。

因此，sender窗口的最左端序列号为m-w，receiver窗口的最右端序列号为m+w-1。为了使receiver窗口的不与sender窗口发生重叠，序列号范围必须足够大，以容纳2w序列号。也就是说，序列号范围至少是窗口大小的两倍，k>2w。

[简答题]

Chapter 3, P24. Answer true or false to the following questions and briefly justify your answer:

a. With the SR protocol, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.

b. With GBN, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.

c. The alternating-bit protocol is the same as the SR protocol with a sender and receiver window size of 1.

d. The alternating-bit protocol is the same as the GBN protocol with a sender and receiver window size of 1.

[参考答案]

a. 正确。假设 sender 在 t_0 时刻窗口大小为 3，发送的数据包是 1、2、3；在 t1 时刻 receiver 收到数据包 1、2、3 并且发送这三个数据包的 ACK；在 t2 时刻 sender 没有收到 ACK 发生超时，并且重新发送数据包 1、2、3；在 t3 时刻 receiver 收到数据包 1、2、3 并且重新发送这三个数据包的 ACK；在 t4 时刻 sender 收到了 receiver 在 t1 时刻发送的数据包 1、2、3 的 ACK，并且其窗口变为 4、5、6；而在 t5 时刻 sender 收到了 receiver 在 t3 时刻发送的数据包 1、2、3 的 ACK；这个时刻收到的 ACK 已经超出了窗口之外。

b. 正确。可以根据a中的假设来考虑，本质上没有区别。

c. 正确。

d. 正确。SR、GBN和alternating-bit协议在发送窗口和接收窗口为1的情况下功能相同，窗口为1可以保证按照顺序接收数据包，排除出现乱序数据包的情况；此时的ACK只能是窗口内数据包的ACK。

[简答题]

Chapter 3, R14. True or false?

a. Host A is sending Host B a large file over a TCP connection. Assume Host B has no data to send Host A. Host B will not send acknowledgments to Host A because Host B cannot piggyback the acknowledgments on data.

b. The size of the TCP rwnd never changes throughout the duration of the connection.

c. Suppose Host A is sending Host B a large file over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the receive buffer.

d. Suppose Host A is sending a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is m, then the sequence number for the subsequent segment will necessarily be m+1.

e. The TCP segment has a field in its header for rwnd.

f. Suppose that the last SampleRTT in a TCP connection is equal to 1 sec. The current value of TimeoutInterval for the connection will necessarily be ≥ 1 sec.

g. Suppose Host A sends one segment with sequence number 38 and 4 bytes of data over a TCP connection to Host B. In this same segment the acknowledgment number is necessarily 42.

[参考答案]

a. 错误；TCP是一种可靠数据传输协议，协议中有反馈机制；
b. 错误；rwnd是receiver缓冲区中空闲的空间大小，TCP连接中会随着时间改变；
c. 正确；LastByteRcvd−LastByteRead≤RcvBuffer，LastByteRcvd是B中应用程序进程从缓冲区读取的数据流中最后一个字节的数目，LastByteRead是从网络到达的数据流中最后一个字节的数量，该字节被放置在B处的接收缓冲区中，RcvBuffer是主机B的缓冲区大小；
d. 错误； sequence number不一定是m+1，还需要看每一个segment的大小；
e. 正确；TCP的头部有rwnd的信息；
f. 错误；
g. 错误；

[简答题]

Chapter 3, R15. Suppose Host A sends two TCP segments back to back to Host B over a TCP connection. The first segment has sequence number 90; the second has sequence number 110.

a. How much data is in the first segment?
b. Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number?

[参考答案]

a. 通过sequence number，可以知道第一个segment是20 bytes；
b. ACK的序号是90。

[简答题]

Chapter 3, P28. Host A and B are directly connected with a 100 Mbps link. There is one TCP connection between the two hosts, and Host A is sending to Host B an enormous file over this connection. Host A can send its application data into its TCP socket at a rate as high as 120 Mbps but Host B can read out of its TCP receive buffer at a maximum rate of 50 Mbps. Describe the effect of TCP flow control.

[参考答案]

链路容量只有100 Mbps，那么主机A的发送速率最高可达100 Mbps。但是主机B读取数据的速率最高为50 Mbps，TCP的流量控制会把主机A的发送速率限制在50Mbps。

[简答题]

Chapter 3, P31. Suppose that the five measured SampleRTT values (see Section 3.5.3) are 106 ms, 120ms, 140 ms, 90 ms, and 115 ms. Compute the EstimatedRTT after each of these SampleRTT values is obtained, using a value of $\alpha = 0.125$ and assuming that the value of EstimatedRTT was 100 ms just before the first of these five samples were obtained. Compute also the DevRTT after each sample is obtained, assuming a value of

$\beta = 0.25$ and assuming the value of DevRTT was 5 ms just before the first of these five samples was obtained. Last, compute the TCP TimeoutInterval after each of these samples is obtained.

[参考答案]

根据下面的公式迭代计算每一轮的 DevRTT、EstimatedRTT 与 TimeoutInterval 值：

EstimatedRTT=(1-α)*EstimatedRTT+α*SampleRTT

DevRTT=(1-β)*DevRTT+β*|SampleRTT-EstimatedRTT|

TimeoutInterval= EstimatedRTT+4*DevRTT

得到样本 106ms 后:
EstimatedRTT = 0.875 * 100 + 0.125 * 106 = 100.75 ms
DevRTT = 0.75*5 + 0.25 * | 106 – 100.75 | = 5.06ms
TimeoutInterval = 100.75+4*5.06 = 120.99 ms

得到样本 120ms 后:
EstimatedRTT = 0.875 * 100.75 + 0.125 * 120 = 103.16 ms
DevRTT = 0.75*5.06 + 0.25 * | 120 – 103.15 | = 8ms
TimeoutInterval = 103.15+4*8 = 135.15 ms

得到样本 140ms 后:
EstimatedRTT = 0.875 * 103.15 + 0.125 * 140 = 107.76 ms
DevRTT = 0.75*8 + 0.25 * | 140 – 107.76| = 14.06 ms
TimeoutInterval = 107.76+4*14.06 = 164 ms

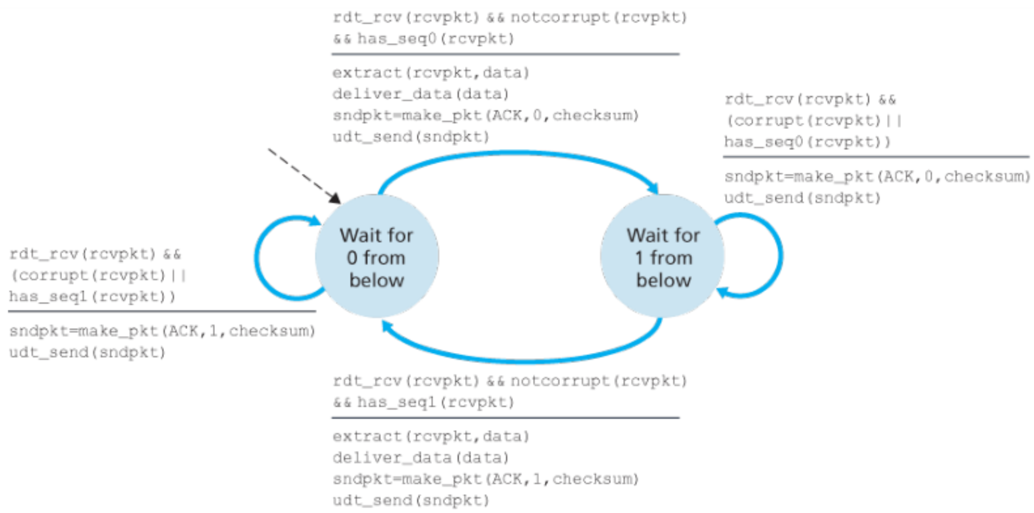
得到样本 90ms 后:
EstimatedRTT = 0.875 * 107.76 + 0.125 * 90 = 105.54 ms
DevRTT = 0.75*14.06 + 0.25 * | 90 – 105.54 | = 14.42 ms
TimeoutInterval = 105.54+4*14.42 =163.22 ms

得到样本 115ms 后:
EstimatedRTT = 0.875 * 105.54 + 0.125 * 115 = 106.71 ms
DevRTT = 0.75*14.42 + 0.25 * | 115 – 106.71| = 12.88 ms
TimeoutInterval = 106.71+4*12.88 =158.23 ms

[简答题]

Chapter 3, P8. Draw the FSM for the receiver side of protocol rdt3.0.

[参考答案]

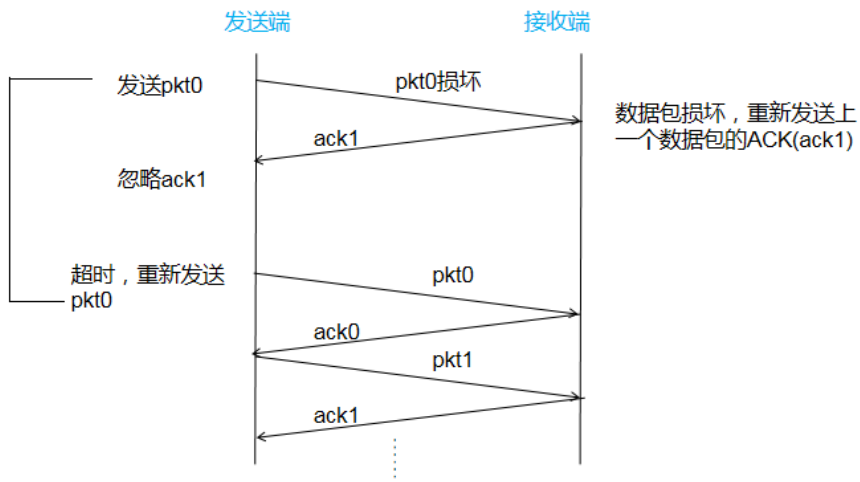


[简答题]

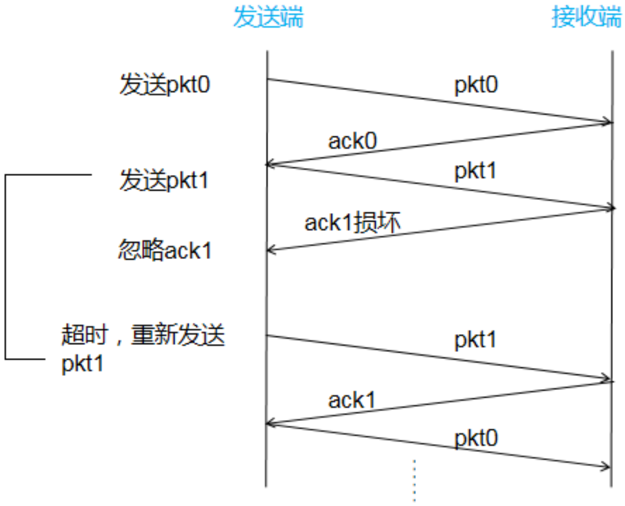
Chapter 3, P9. Give a trace of the operation of protocol rdt3.0 when data packets and acknowledgment packets are garbled. Your trace should be similar to that used in Figure 3.16 .

[参考答案]

（1）数据包破坏



（2）ACK破坏



[简答题]

Chapter 3, P32. Consider the TCP procedure for estimating RTT. Suppose that $\alpha = 0.1$. Let $SampleRTT_1$ be the most recent sample RTT, let $SampleRTT_2$ be the next most recent sample RTT, and so on.

a. For a given TCP connection, suppose four acknowledgments have been returned with corresponding sample RTTs: $SampleRTT_4$, $SampleRTT_3$, $SampleRTT_2$, and $SampleRTT_1$. Express EstimatedRTT in terms of the four sample RTTs.

b. Generalize your formula for n sample RTTs.

c. For the formula in part (b) let n approach infinity. Comment on why this averaging procedure is called an exponential moving average.

[参考答案]

a.

$$\begin{aligned} Estimate^{(4)} &= xSampleRTT_1 \\ &+ (1-x)\big[xSampleRTT_2 \\ &+ (1-x)[xSampleRTT_3 + (1-x)SampleRTT_4]\big] \\ &= xSampleRTT_1 + (1-x)xSampleRTT_2 + (1-x)^2 xSampleRTT_3 + (1 \\ &- x)SampleRTT_4 \end{aligned}$$

b.

$$Estimate^{(n)} = x\sum_{j=1}^{n-1}(1-x)^{j-1}SampleRTT_j + (1-x)^{n-1}SampleRTT_n$$

c.

$$Estimate^{(\infty)} = \frac{x}{1-x}\sum_{j=1}^{\infty}(1-x)^{j-1}SampleRTT_j + \frac{1}{9}\sum_{j=1}^{\infty}9^{j-1}SampleRTT_j$$

每一次样本的权重呈指数衰减。

[简答题]

Chapter 3, P37. Compare GBN, SR, and TCP (no delayed ACK). Assume that the timeout values for all three protocols are sufficiently long such that 5 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A) respectively. Suppose Host

A sends 5 data segments to Host B, and the 2nd segment (sent from A) is lost. In the end, all 5 data segments have been correctly received by Host B.
a. How many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers? Answer this question for all three protocols.
b. If the timeout values for all three protocol are much longer than 5 RTT, then which protocol successfully delivers all five data segments in shortest time interval?
[参考答案]
a. GBN：
A一共发送了9个segment，开始发送序号为1、2、3、4、5的segment，在第2个segment丢失之后，发送序号为2、3、4、5的segment。
B发送了8个ACK，分别是四个序号为1的ACK和序号为2、3、4、5的ACK。
SR：
A一共发送了6个segment，开始发送序号为1、2、3、4、5的segment，在第2个segment丢失之后，发送序号为2的segment。
B发送了5个ACK，分别发送的是1、3、4、5的ACK，之后发送的是序号为2的ACK。
TCP：
A一共发送了6个segment，开始发送序号为1、2、3、4、5的segment，在第2个segment丢失之后，发送序号为2的segment。
B发送了5个ACK，分别发送的是四个序号为2的ACK，之后发送的是序号为6的ACK。
b. TCP，这是因为TCP使用快速重传，而不需要等待超时。

[简答题]
Chapter 3, P40. Consider Figure 3.61 . Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. In all cases, you should provide a short discussion justifying your answer.
a. Identify the intervals of time when TCP slow start is operating.
b. Identify the intervals of time when TCP congestion avoidance is operating.
c. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
d. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
e. What is the initial value of ssthresh at the first transmission round?
f. What is the value of ssthresh at the 18th transmission round?
g. What is the value of ssthresh at the 24th transmission round?
h. During what transmission round is the 70th segment sent?
i. Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of ssthresh?
j. Suppose TCP Tahoe is used (instead of TCP Reno), and assume that triple duplicate ACKs are received at the 16th round. What are the ssthresh and the congestion window size at the 19th round?
k. Again suppose TCP Tahoe is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive?
[参考答案]
a. [1,6]，[23,26]；
b. [6,16]，[17,22]；

c. 三个重复（duplicate）ACK。因为收到三个重复 ACK 后拥塞窗口的大小降至当前窗口的一半再加 3；从图上看第 16 轮拥塞窗口为 42，第 17 轮第拥塞窗口为 24（=42/2+3）；
d. 超时。因为拥塞窗口大小被设置为 1；
e. 32．因为在第 16 轮，拥塞窗口的大小从指数增长转变为线性增长，即慢启动阶段（slow start）结束，进入拥塞避免(congestion avoidance)阶段。转变的条件是拥塞窗口大于拥塞阈值，16 轮时拥塞窗口为 32，所以拥塞阈值也是 32。
f. 21。第 16 轮时检测到 3 个重复 ACK，拥塞阈值下降为当前拥塞窗口的一半，拥塞窗口下降为当前拥塞窗口的一半再加 3，进入拥塞避免阶段。因此，第 17 轮开始，拥塞阈值为 21=（42/2）.在第 16-18 轮间没有发生任何能够让阈值改变的事件，因此拥塞阈值 ssthresh 保持为 21；
g. 14。当发生超时后，拥塞阈值被设置为当前拥塞窗口的一半，拥塞窗口置为 1，进入慢启动阶段。第 22 轮发生了超时，当前拥塞窗口大小为 29，因此拥塞阈值被置为 14（=29/2），拥塞窗口置为 1，进入了慢启动阶段。第 24 轮依然在慢启动阶段，因此拥塞阈值仍为 14；
h. 第 7 轮。在第一轮发送中，发送分组 1；在第二轮发送分组 2-3；在第三轮发送分组 4-7；在第四轮发送分组 8-15；在第五轮发送分组 16-31；在第六轮发送分组 32-63；数据包 64-96 在第 7 轮传输中发送。因此，分组 70 在第 7 轮中被发送；
i. 拥塞阈值为 4，拥塞窗口为 7。当收到三个重复 ACK 时，拥塞阈值被设置为当前拥塞窗口值的一半，并且拥塞窗口被设置为当前拥塞窗口到一半再加 3。第 26 轮时，拥塞窗口为 8，因此，阈值和窗口的新值将分别为 4 和 7，进入拥塞避免阶段。
j. 阈值为 21，拥塞窗口大小为 4。TCP Tahoe 不区分丢包的原因。不管是收到三个重复 ACK，还是超时，都把拥塞阈值设置为当前拥塞窗口的一半，拥塞窗口置为 1，进入慢启动阶段。因此在第 16 轮收到 3 个重复 ACK 后，在第17 轮拥塞阈值会降为 21（=42/2），拥塞窗口降为 1，进入慢启动阶段。在第 19 轮时，拥塞窗口增长到 4.
k. 52；第 17 轮到第 22 轮，共 1+2+4+8+16+21=52 个包。从第 17 轮到第 21 轮，TCP Tahoe 处于慢启动阶段，拥塞窗口从 1 开始指数增加。在第 22 轮，拥塞窗口增加到 21 时，和阈值相等，不再指数增加，转为线性增加，进入拥塞避免阶段。此时发生超时，拥塞窗口在下一轮（第 23 轮）降到 1。因此在第 22 轮时拥塞窗口为 21。

[简答题]

Chapter 3, P48. Consider that only a single TCP (Reno) connection uses one 10Mbps link which does not buffer any data. Suppose that this link is the only congested link between the sending and receiving hosts. Assume that the TCP sender has a huge file to send to the receiver, and the receiver's receive buffer is much larger than the congestion window. We also make the following assumptions: each TCP segment size is 1,500 bytes; the two-way propagation delay of this connection is 150 msec; and this TCP connection is always in congestion avoidance phase, that is, ignore slow start.

a. What is the maximum window size (in segments) that this TCP connection can achieve?
b. What is the average window size (in segments) and average throughput (in bps) of this TCP connection?
c. How long would it take for this TCP connection to reach its maximum window again after recovering from a packet loss?

[参考答案]

a.　设最大窗口中包含N个segment，根据公式：W*MSS/RTT = 10Mbps,，可得

$$\frac{N * 1500 * 8bits}{150 * 10^{-3}s} = 10 * 10^{6}bps$$

由上式可得，N=125。

b.　拥塞窗口变化的范围是N/2到N，所以平均窗口大小为0.75N=93.75,，取值为94。平均throughput为$94 * 1500 * \frac{8}{0.15} = 7.52$Mbps

c.　窗口从N/2到N，N=125. 125/2=62。 所以用到的时间为(125 - 62) *0.15 = 9.45。