

## Exercise 1

不能保证，在这里我用一个极端的反例来说明：假设发送方发送第一个16比特是11111111 11111111，其余全为0，则检验和为00000000 00000000，而接收方假设接受到的前32比特为11111111 11111110，00000000 00000001，其余也均为零，则加上检验和仍为11111111 11111111，通过了检验，然而明显发送和接受的信息存在不一致，因此检验和不能绝对避免每个比特都不出错

## Exercise 2

协议能正常运行，因为重传并不会产生差错，只不过影响效率。

如果定时器过早超时，那么对于报文1，还未接收到ACK即要重传，接收到ACK 1（下面第x个报文的ACK简称为ACK x）时至少重传了1次，此时接受到ACK 1后传输报文2，首先会收到至少一次ACK 1，其次是收到ACK 2期间定时器至少会超时一次，因此两者加起来至少重传两次；...以此类推，报文n至少接到n-1次ACK n-1，加上定时器至少超时一次，因此至少重传n次（发送n+1次）。n趋于无穷时发送次数也趋于无穷。

## Exercise 3

a. 报文序号可能是  $k-4 \sim k+3$

考虑两个极端情况，1) 假设接收端接受到  $k-4 \sim k-1$ ，则目前在等待k，而发送端目前还没收到大于k-4的报文ACK，因此还停留在  $k-4 \sim k-1$ ，注意不可能有k-5（或更小），因为窗口长度为4，如果有k-5则k-1不可用，不能被发送，更不可能被接收端接受；2) 假设接收端刚接受到k-1，发送k-1的ACK被发送端接受到，发送端窗口可以移到  $k \sim k+3$ ，注意不可能有k+4（或更大），因为窗口长度为4，如果有k+4则说明已经接收到了k的ACK，而此时接收端正等待k的传输，矛盾

b. ACK字段的可能值为  $k-5 \sim k-1$

首先我们得有一个假设，即先发出的ACK应当先被接受（如果未丢失或损坏等），那么当发送方发送k-5超时后重发k-5，收到ACK k-5时发送  $k-4 \sim k-1$ ，接收方收到重发的k-5和  $k-4 \sim k-1$  回复ACK  $k-5 \sim k-1$ ；注意，根据假设，不可能有k-6（或更小），假设有ACK k-6，他肯定先于ACK k-5，而在这种极端情况中，已收到一个ACK k-5，说明ACK k-6或更小的是不存在的。

## Exercise 4

a. 正确，假设某报文因为超时重发，而第一次发送时的ACK被接受后移动窗口，窗口中不存在该序号了，这时接受重发的ACK即满足所述情况

b. 正确，假设发送方窗口长度为1，发送报文时因为超时而重发，收到一个ACK后窗口前移一格，而此时再收到重发的ACK时已经超出窗口

c.d. 正确，GBN和SR都是针对流水线模型的改进，而当  $N=1$  时实际上就退化成了比特交替协议，因此  $N=1$  时GBN、SR、比特交替协议都是等价的

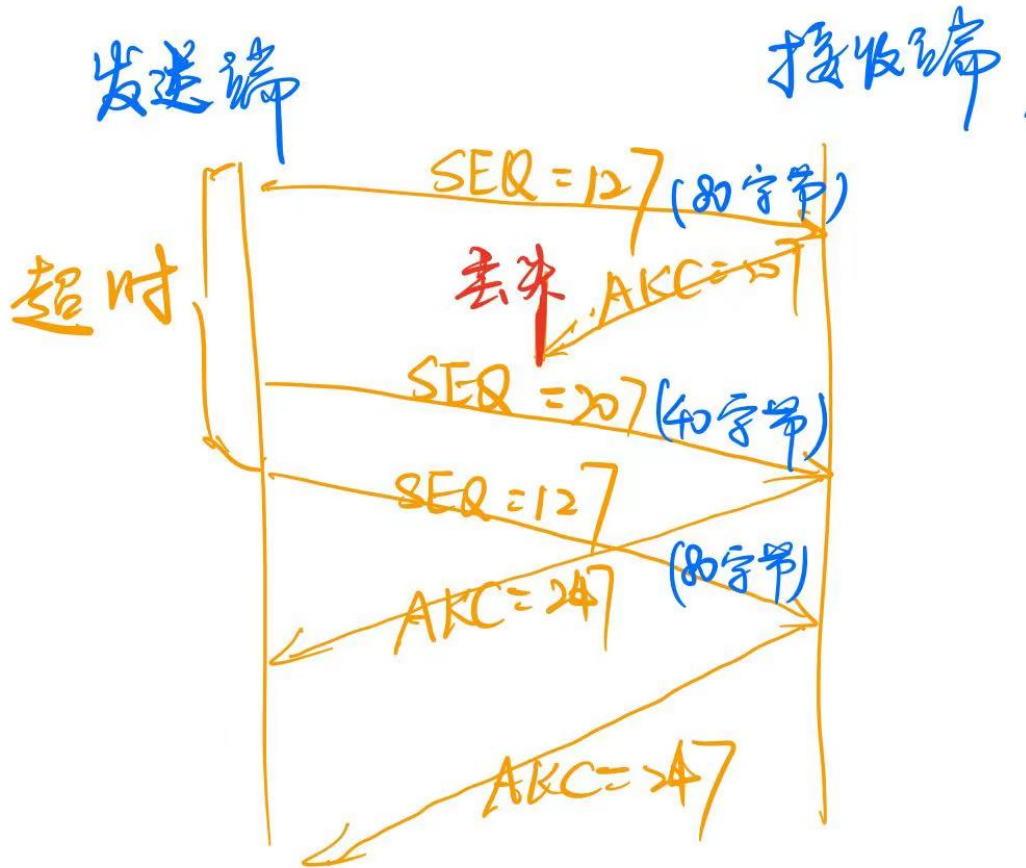
## Exercise 5

a.  $2^{32}$  字节，大约为4.2G文件

b.  $\frac{2^{32} \frac{602}{536} \text{ bit}}{155 \text{ Mbyte/s}} = 237.44 \text{ s}$

## Exercise 6

- a. 序号207 源端口号302 目的地端口号80
- b. 确认号207 源端口号80 目的地端口号302
- c. 确认号247
- d.



## Exercise 7

- a. 6RTT
- b.  $\frac{6+7+8+9+10}{5} MSS = 8MSS$