

R5. 当今的因特网防火墙大都阻塞UDP流量, 故采用TCP协议

R8. 套接字不相同. 对于每个连接, Web服务器均会分别创建一套套接字, 而TCP协议下套接字有源IP地址部分, 故主机不同的A、B套接字一定不同. 但同一IP下目的端口号一样  
由上面的讨论可知, 若具有端口80

R14. A. 错, B. 错, 单独发送确认用报文  
C. 对

b. 错  $rwnd = rwnd - [lastByteRecv - lastByteRead]$  故为动态的

d. 错, 若该轮报文为数据, 后续报文应为该段报文最后十数据字段的最后字节

e. 对

f. 错  $TimeoutInterval = EstimateRTT + 4 \cdot DevRTT$

$= (1 - \alpha) EstimateRTT + \alpha \cdot SampleRTT + 4(1 - \alpha) \cdot DevRTT + 1 SampleRTT$   
Estimate

g. 错, 确认号与报文序号无关.

从入发现 Time out Interval 不大于 15

P1. 设 A 的默认端口号为 A, B 的为 B, Telnet 默认 23

a. source port: A dest port: 23

b. source port: B dest port: 23

c. source port: 23 dest port: A

d. source port: 23 dest port: B

e. 可能相同. 如果 A 的默认端口号与 B 的相同的话 f. 则发向 A、B 的报文均发向 3 个不同的端口

P3.  $1010011 + 0110110 + 0110110 = 10010111 = 00101101 = 0010110$  取反为 1101001

在本地先取反, 验证时再用两次取反, 提高效率

根据数据后, 接收方将所有字节相加, 与发送方验证码作和. 若结果不为 0, 则一定有问题, 若为 0 可能没有问题  
1. 出错一定会被检测出, 2. 出错可能无法检测: 有两数据同时出现改变, 求和仍不变

P5. 不足, 见 P3 最后问题的回答

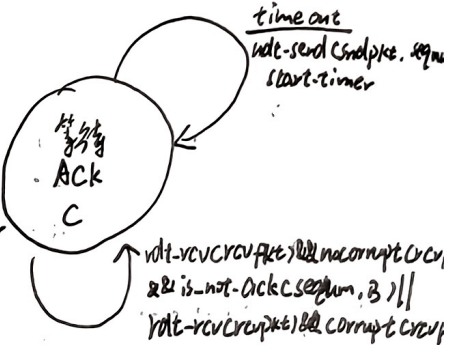
P19.  $rdt\_recv(crcv\_pkt) \&\& corrupt(crcv\_pkt)$

$rdt\_recv(crcv\_pkt) \&\& no\_corrupt(crcv\_pkt) \&\& is\_not\_ack(seqnum, A)$

$rdt\_recv(crcv\_pkt) \&\& no\_corrupt(crcv\_pkt) \&\& is\_not\_ack(seqnum, B)$

$rdt\_recv(crcv\_pkt) \&\& no\_corrupt(crcv\_pkt) \&\& is\_not\_ack(seqnum, C)$

$seqnum = seqnum + 1$   $udt\_send(csendpkt, seqnum, start\_timer)$



C 下粘上粘相同, B、C 字节调换即可

B 接收方:

$rdt\_recv(crcv\_pkt) \&\& corrupt(crcv\_pkt)$



$rdt\_recv(crcv\_pkt) \&\& no\_corrupt(crcv\_pkt) \&\& has\_seq(seqnum)$

$udt\_send(ACK, seqnum, B)$   
 $seqnum = seqnum + 1$

$rdt\_recv(crcv\_pkt) \&\& no\_corrupt(crcv\_pkt) \&\& has\_seq(x) \&\& x = seqnum$

$udt\_send(ACK, x, B)$



