



### Chapter 3

R5 由于安全原因很多机构阻塞UDP流量  
所以对于流式媒体传输也逐渐使用TCP.

R8. 1) 并非通过同一进程, 根据源IP和端口号连接不同的套接字.  
2) 是的. 因为端口80是熟知端口号, 保留给HTTP应用层协议使用. 所以这两个套接字的目的端口都是80, 而当报文来到时, 先通过目的端口号判断, 再进一步依据源IP地址和端口号定位到套接字.

R14. a. (x) 主机B每次接收一个数据报文段都会激活确认机制。

b. (x) rwnd是动态的。

$$rwnd = RcvBuffer - [LastByteRcvd - LastByteRead]$$
  
表明接收方缓存中的空闲空间。

c. (v)  $LastByteSent - LastByteAcked \leq rwnd$   
并且  $rwnd \leq RcvBuffer$ .



d. (X) TCP的序号分配建立在字节流上, 所以相邻报文段中, 报文端按照首字节的编号分配序号。如果  $MSS=1000$ , 那么后继字节流序号是  $m+1000$

\* e. (✓) TCP报文段首部存在  $rwnd$  字段。而TCP实现流量控制服务是通过接收方将  $rwnd$  写入发送的报文段的接收窗口字段中

f. (X)  $TimeoutInterval = EstimatedRTT + 4 \cdot DevRTT$

$$EstimatedRTT = (1 - \alpha) \cdot EstimatedRTT + \alpha \cdot SampleRTT$$

可以看出  $TimeoutInterval$  不由  $SampleRTT$  直接决定。如果  $EstimatedRTT$  的旧值和  $DevRTT$  足够小,  $TimeoutInterval$  可以小于  $SampleRTT$ 。

g. (X) TCP通信双方可以随机地选取初始序号。

主机A报文段中的确认号只表明了它期待从主机B收到的下一字节的编号, 和主机A发送的数据的序号无关。

P1

a. Telnet协议代理服务器周知端口号为23, A: 1024 B: 1025  
避开周知端口即可

a.  $A \rightarrow S: \overset{1024}{123} \rightarrow 23$

b.  $B \rightarrow S: \overset{1025}{456} \rightarrow 23$

c.  $S \rightarrow A: 23 \rightarrow \overset{1024}{123}$

d.  $S \rightarrow B: 23 \rightarrow \overset{1025}{456}$

e. 可

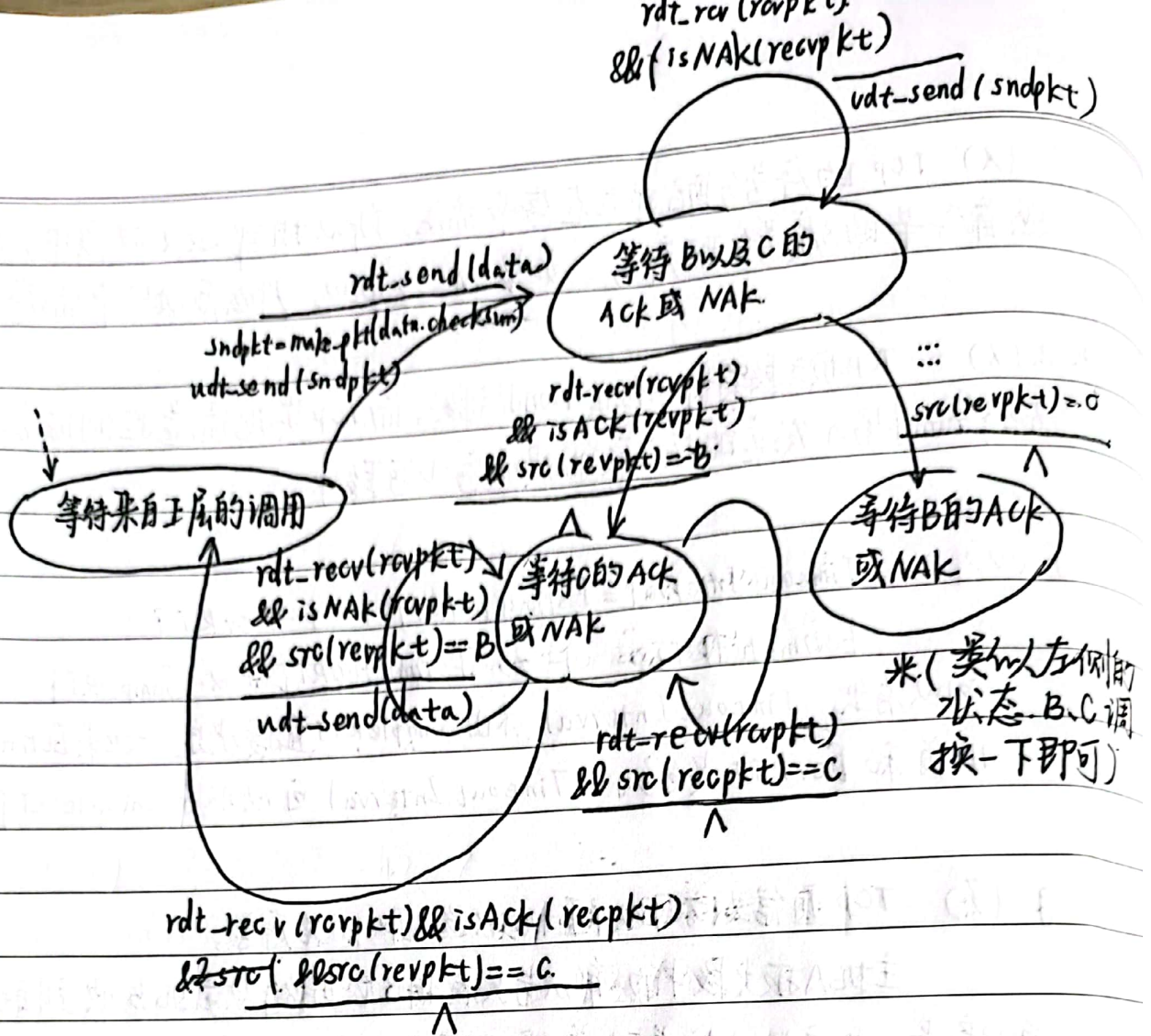
f. A和B是一台主机的两个进程 所以在主机和服务器的两个套接字, 注意A和B不能选用相同源端口号



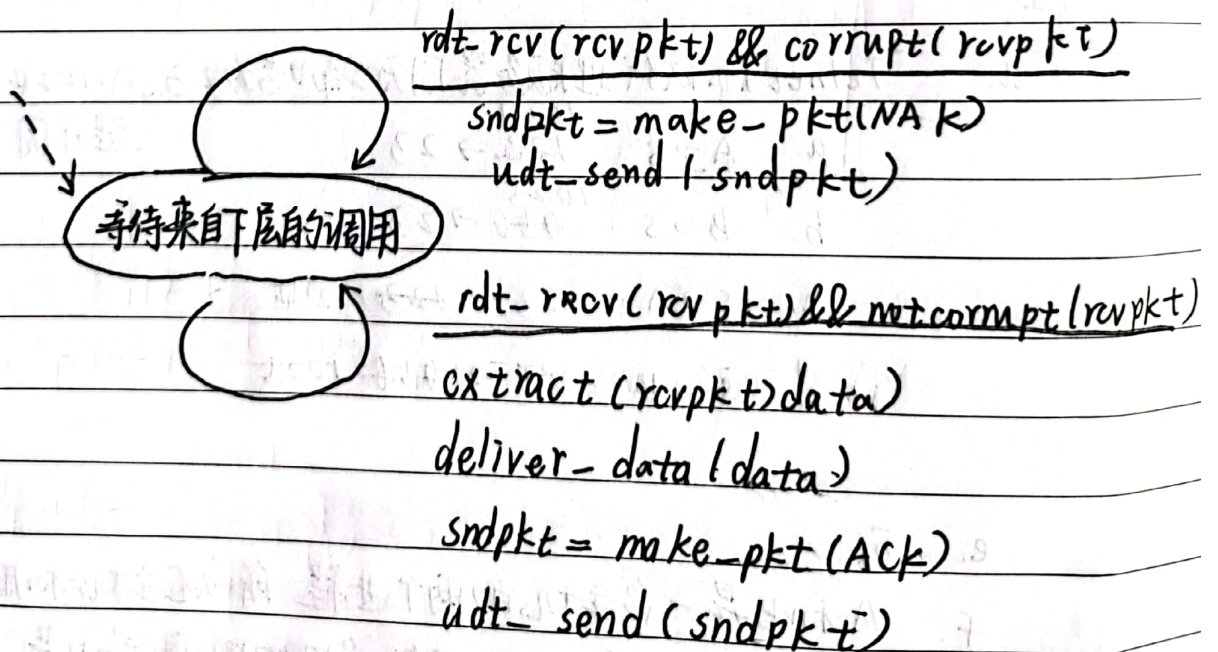




P19.



### 发送方A的FSM



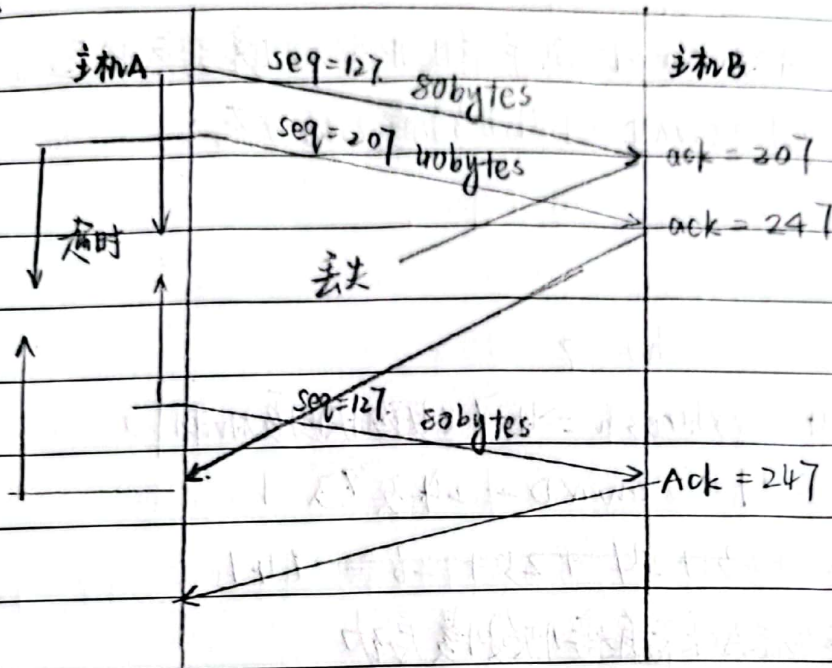
### 接收方B/C的FSM





P27.

- 序号: 207 源端口号: 307. 目的端口号 80
- 确认号: 207 源端口号: 80. 目的端口号 307
- 确认号: 127
- 



P32. a). Estimated RRTT<sub>70</sub> = SampleRT<sub>1</sub>

$$\text{Estimate RRTT}_{71} = \text{SampleRT}_1$$

$$\text{Estimate RRTT}_{72} = 0.9 \times \text{SampleRT}_1 + 0.1 \times \text{SampleRT}_2$$

$$\text{Estimate RRTT}_{73} = 0.81 \times \text{SampleRT}_1 + 0.09 \times \text{SampleRT}_2 + 0.1 \times \text{SampleRT}_3$$

$$\text{Estimate RRTT}_{74} = 0.729 \times \text{SampleRT}_1 + 0.081 \times \text{SampleRT}_2 + 0.09 \times \text{SampleRT}_3 + 0.1 \times \text{SampleRT}_4 \text{ (符号1,2,3,4写反了)}$$

$$\text{Estimate RRTT}_4 = 0.729 \times \text{SampleRT}_4 + 0.081 \times \text{SampleRT}_3 + 0.09 \times \text{SampleRT}_2 + 0.1 \times \text{SampleRT}_1$$

$$b) \text{Estimate RRTT}_n = (1-\alpha)^n \times \text{SampleRT}_n + \sum_{i=1}^{n-1} \alpha(1-\alpha)^{i-1} \times \text{SampleRT}_i$$

c).  $n$  趋于无穷  $(1-\alpha)^n \times \text{SampleRT}_n \rightarrow 0$ . 而剩余的项的权值(系数)随着  $i$  增大而指数衰减。







P40 a) 慢启动: 1~6, 23~26

b) 拥塞避免: 6~16, 17~22

c) 冗余ACK。因为  $cwnd$  减半并且进入拥塞避免状态。

d) timeouts。因为  $cwnd=1$  且进入慢启动状态。

e) 32

f) 24/2?

g) 14

h) 7

i)  $cwnd=4$ ,  $ssthresh=7$  (因为快恢复机制)

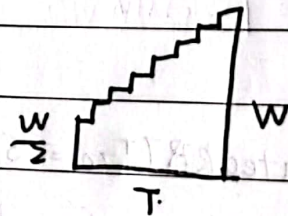
j)  $ssthresh=21$ ,  $cwnd=24$  及 73

k) ~~21+22+23+24+25+26 = 141~~

Tahoe 版本冗余ACK后直接恢复慢启动

$$1+2+4+8+16+21=52$$

P45. a. 发生一次丢包中发送了



$$\frac{W}{2} + \frac{W}{2} + 1 + \frac{W}{2} + 2 + \frac{W}{2} + \frac{W}{2}$$

$$= \frac{W}{2} \cdot \left( \frac{W}{2} + 1 \right) + \sum_{i=1}^{\frac{W}{2}} i$$

$$= \frac{W^2}{4} + \frac{W}{2} + \frac{\frac{W}{2}(\frac{W}{2} + 1)}{2} = \frac{W^2}{4} + \frac{W^2}{8} + \frac{3}{4}W = \frac{3}{8}W^2 + \frac{3}{4}W$$

$$\text{那么 } L = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

b. 传输过程是从  $W/(2 \times RTT)$  到  $W/RTT$  过程的反复  
在一个周期中

$$\text{平均速率} = \frac{\frac{3}{4}W \cdot \text{MSS}}{RTT} \approx \frac{\frac{3}{4} \cdot \frac{3}{8} \cdot \text{MSS}}{\frac{1}{\sqrt{32}} \cdot RTT} = \frac{1.22 \cdot \text{MSS}}{RTT \cdot \sqrt{2}}$$





Prob. a)  $10 \times 10^6 \div 8 \times 0.15 / 1500 = 125$

b)  $0.75 \times 125 = 93.75$

$93.75 \times 1500 \times 8 / 0.15 = 7.5 \text{ Mbps}$

c)  $0.15 \times (125 - 62) = 9.45 \text{ s}$





P50 初始时:  $CWND_G = 10$   $CWND_{G_2} = 10$

总速率为  $10 \div 0.05 + 10 \div 0.1 = 300 \text{ 个/s} > 30$

50ms 时  $G$  更改  $CWND_G = 5$ .  $G_2$  未结束 RTT 调整窗口

100ms 时. 总速率为  $5 \div 0.05 + 10 \div 0.1 = 200 > 30 (50\text{ms})$

$G$  更改  $CWND_G = 2$ .

$2 \div 0.05 + 5 \div 0.1 = 90 > 30$

150ms 时  $G$  更改  $CWND_G = 1$ .

$1 \div 0.05 + 5 \div 0.1 = 25 < 30$

200ms.  $G$  减无可减  $CWND_G = 1$

$G_2$ .  $CWND_{G_2} = 2$

$1 \div 0.05 + 2 \div 0.1 = 40 > 30$

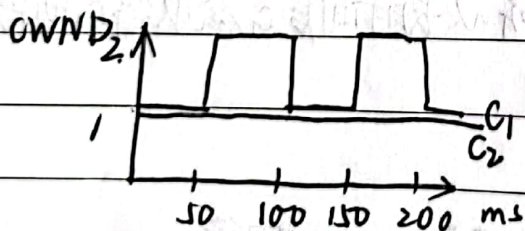
300ms:  $G$ :  $CWND = 1$   $G_2$ :  $CWND = 1$

$1 \div 0.05 + 1 \div 0.1 = 30$

之后的 50ms 时 50ms 后  $G$  加倍, 再过 50ms  $G$  减半  
 $G_2$  不变

所以到 1000ms 时  $CWND_G = 1$   $CWND_{G_2} = 1$

b) 不是。



所以由于 RTT 不同,  $G$  享用了更多的带宽  
例如在 100ms 中.

$G$  发送了:  $1 \div 0.05 + 2 \div 0.05 = 60 \text{ MSS}$

$G_2$  发送了:  $1 \div 0.05 + 1 \div 0.1 = 20 \text{ MSS}$

