

P3

01010011+01100110=10111001

10111001+01110100=00101110

则反码为 11010001

使用反码，不必依赖系统是大端还是小端，只需将所有数据包包含校验码加起来，计算和为全1即可。

差错检验方法：如果不是全1则说明出现了差错。

1比特的差错一定能检测出，2比特的差错可能检测不出。

P11

前一种情况，可以正常工作，因为上一步状态转换时已经生成了 sndpkt。

后一种情况，sndpkt 还没有生成，如果接收了一个校验错误的报文，那么无法返回一个分组。

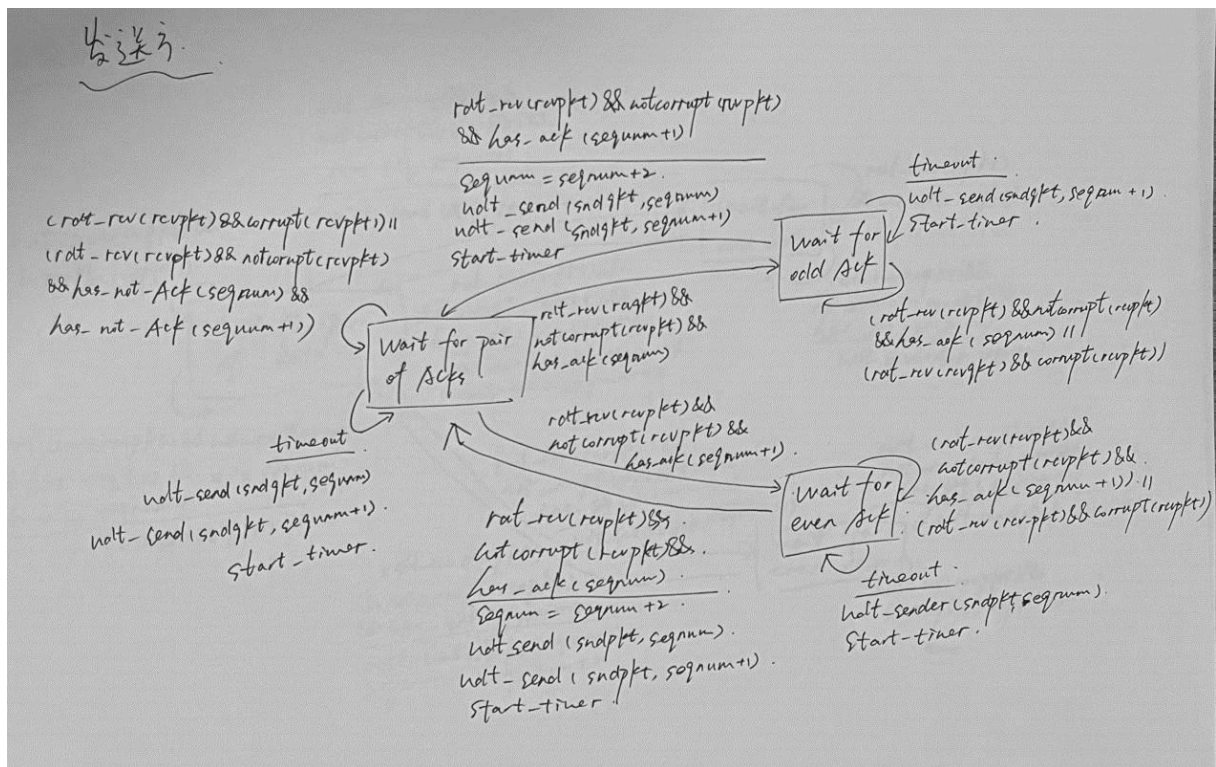
P14

只用 NAK 协议不会比使用 ACK 协议更好。因为接收方判断丢失是依据数据包的上下文，且只是偶尔发送数据，那么只有当丢失的包下一个包被接收时才会发现丢包，这个时间会更长，因此偶尔发送数据，只用 NAK 协议不会比使用 ACK 协议更好。

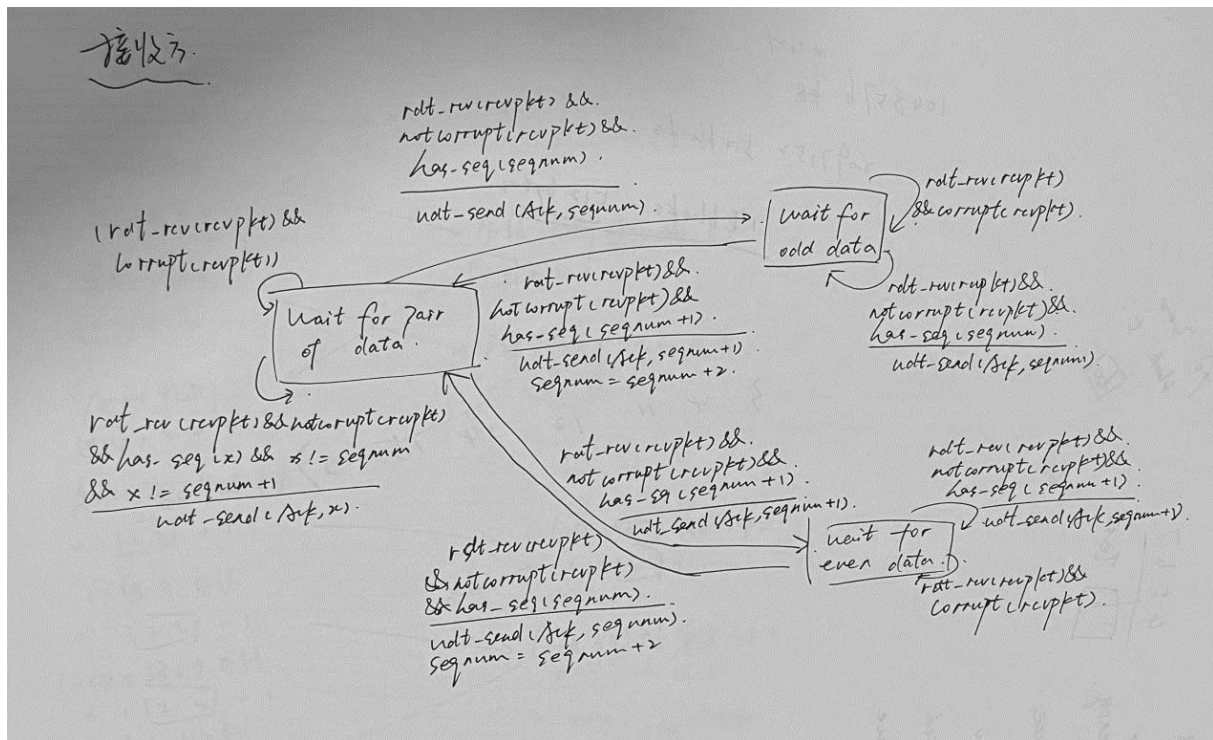
如果发送大量数据并且端到端很少丢包，用 NAK 协议会比使用 ACK 协议更好，因为这样能够减小数据流量。

P18

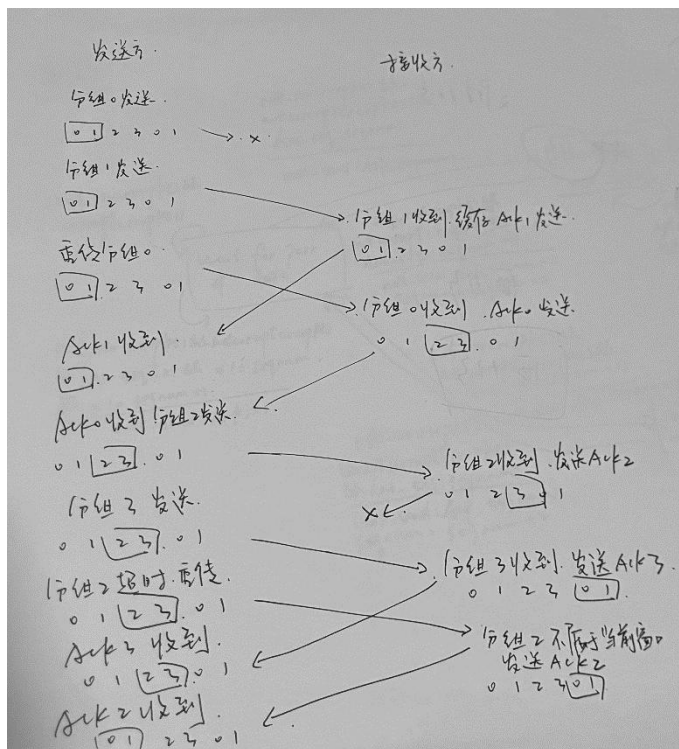
发送方



接收方

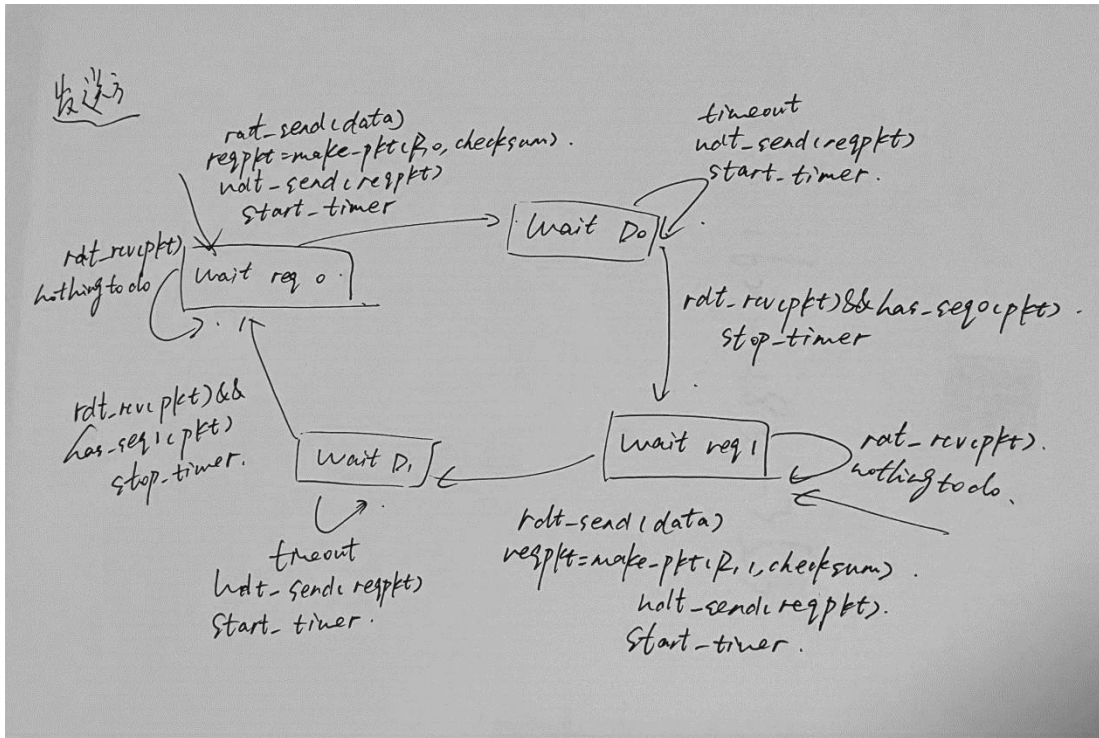


问题处理

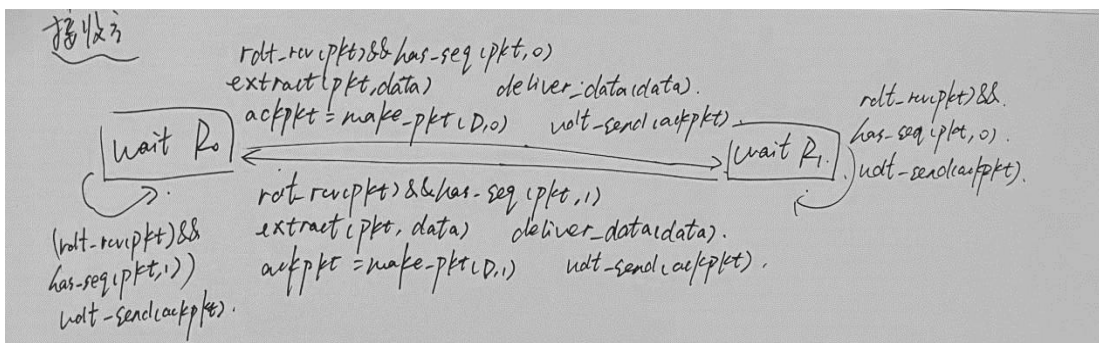


P21

发送方



接收方



P22

- 若 $k-4, k-3, k-2, k-1$ 的 ACK 都未到达发送端，此时发送方窗口序号是 $k-4, k-3, k-2, k-1$ 。
若 ACK 全部传回，发送方更新 base，此时发送方窗口序号是 $k, k+1, k+2, k+3$ 。
- 要接收 k ，那么比 $k-1$ 小的 ACK 已经发送出去，如果发送方发送 $k-1$ ，那么至少接收到了 $k-5$ 的 ACK。故 ACK 字段的所有可能值是 $k-4, k-3, k-2, k-1$ 。

P23

GBN 协议：允许的发送方窗口大小最大为 k 。

SR 协议：允许的发送方窗口大小最大为 $k/2$ 。

P26

a. L 的最大值是 2^{32} byte。

b. 设 N 为报文数 $N = \left\lceil \frac{2^{32}}{536} \right\rceil = 801299$

$$T = \frac{(66 * N + 2^{32})}{155} = 249 \text{秒}$$

P36

如果到达顺序错误，例如分组 3 提前于分组 2 到达，发送一个冗余 ACK，立即快速重传，重传的是分组 2，但是分组 2 并没有丢失，这没有必要。

P37

a.

GBN:

A: 序号为: 1 2 3 4 5 2 3 4 5 共 9 个报文段。

B: 序号为: 1 1 1 1 2 3 4 5 共 5 个 ACK。

SR:

A: 序号为: 1 2 3 4 5 2 共 6 个报文段。

B: 序号为: 1 3 4 5 2 共 5 个 ACK。

TCP:

A: 序号为: 1 2 3 4 5 2 共 6 个报文段。

B: 序号为: 2 2 2 2 6 共 5 个 ACK。

b. TCP 因为 GBN 和 SR 需要等待超时，而 TCP 使用有快速重传，不必等待 5RTT。