

## 1

一种常用的高效的检错码；

发送端在数据流的末尾添加若干个校验位, 使得生成的新数据流能够被预先规定的生成多项式整除, 接收端接收到数据流后使用生成多项式除它, 如果能除尽, 则没错, 否则出错；

要包含  $x+1$  因子；对于任意的  $k$ , 都不能整除  $x^k + 1$

## 2

要将数据流分成离散的帧, 解决帧数据传输的透明性问题 (即怎么分辨两个帧之间的界限)。在帧的首尾都加上 01111110, 并且数据流打包成帧时只要出现 011111 就在其后加 0, 解码时只要遇到连续的 5 个 1 就删除后面跟着的一个 0。

## 3

建立数据链路；码组或帧控制；差错控制；链路流量控制；异常状态的报告和恢复；保证编码透明传输；释放数据链路。

## 4

使用协议, 检错码, 纠错码对传输错误的信息 (数据丢失, 帧丢失, 数据出错, 帧出错) 进行检测甚至纠正。

## 5

否则将错误的的数据上交到网络层, 会导致错误的结果。

## 6

DLE STX A DLE DLE B DLE ETX

## 7

传输数据, 进行数据打包成帧并编码, 以及误差检测和纠正。

## 8

1) 1110 2) 可以 3) 可以 4) 不是, CRC 只能检错, 还得加入确认重传机制才能实现可靠传输

## 9

一个帧的传输时延为  $512 \times 8 / 64000 = 64\text{ms}$   $W_{max} = [2 \times 0.27 \times 64000 / (512 \times 8) + 1] = 9$  窗口为 1 时,  $\eta = 64 / (270 \times 2 + 64) = 10.59$  窗口为 7 时,  $\eta = 7 / (2 \times 0.27 \times 64000 / (512 \times 8) + 1) = 74.17$  窗口为 17 时, 超出限制, 因此只能先发 9 帧, 再发 8 帧,  $\eta = (9 \times 64 + 8 \times 64) / (270 \times 4 + 9 \times 64 + 8 \times 64) = 90.07$  窗口为 117 时, 相当于连续发送 13 个 9 帧,  $\eta = 9 \times 64 / (270 \times 2 + 64) = 95.36$  窗口为 117 时,

## 10

传输时延  $t = 1000 / 50000 = 20\text{ms}$ ,  $W_{max} = [50000 \times 0.27 / 1000] = 13$  1) 每发完一帧, 要等待接收方发送的确认帧, 之后才能发送下一帧, 因此  $\eta = 20 / (20 + 270 \times 2) = 3.572$  一次性发送  $2^3 = 8$ ,  $8$ ,  $= 8$ ,  $= 1$ ,  $\eta = 8 \times 20 / (20 + 270 \times 2) = 28.573$   $2^3 / 2 = 4$ ,  $Ack5(4)$   $4$ ,  $= 4$ ,  $\eta = 4 \times 20 / (20 + 270 \times 2) = 14.29$

因此, 后退 N 帧协议的信道利用率最大, 当然这仅当没有误码的情况下, 如果出现较多误码, 可能选择重传协议会更好。