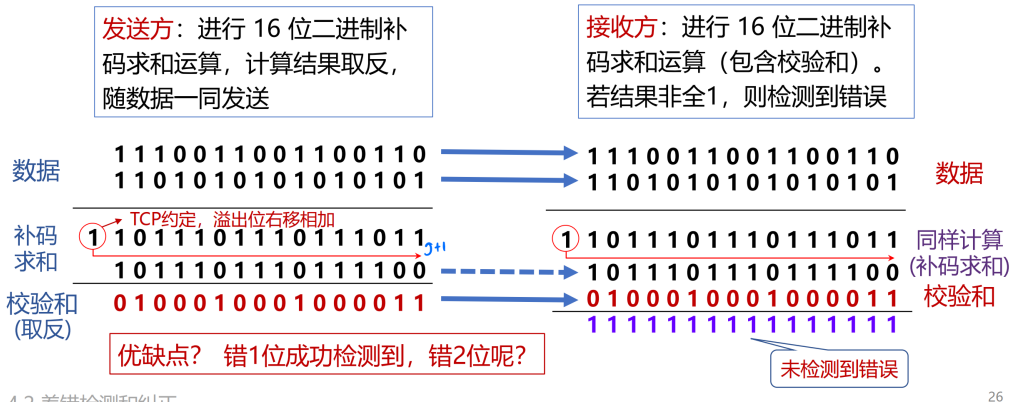
CH4-数据链路层

1.掌握数据链路层将比特流成帧的基本方法

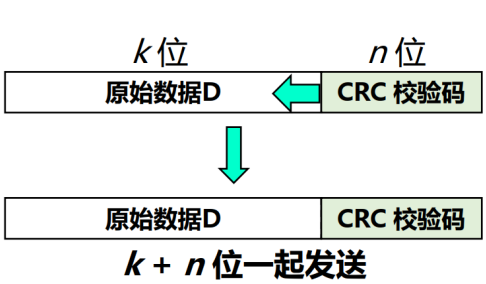
- 帧同步与帧定界
- 成帧方式
  - 字节计数法（Byte count）：破坏了帧的边界，导致一连串帧的错误
  - 带**字节填充**的定界符法：用01111110，即 0x7E 作为FLAG标记首尾，其它涉及FLAG的位置用转移字节ESC+FLAG表示，其他用ESC的地方用ESC+ESC表示。问题：不够灵活
  - 带**比特填充**的定界符法：用01111110作为定界符，对于其它数据中出现的情况，发送方每看到五个1就添加一个0在尾部，接收方每读到五个1，后一个如果是1则是定界符，否则是0就丢弃0得到原来的信息
  - 物理层编码违例
  - - 核心思想：选择的定界符不会在数据部分出现
    - 4B/5B编码方案
      - 4比特数据映射成5比特编码，剩余的一半码字（16个码字）未使用，可以用做帧定界符
      - 例如： 00110组合不包含在4B/5B编码中，可做帧定界符 编码空间扩大，比数间
    - 曼切斯特编码 / 差分曼切斯特编码
      - 正常的信号在周期中间有跳变，持续的高电平（或低电平）为违例码，可以用作定界符
      - 例如：802.5令牌环网
    - 前导码
      - 存在很长的 前导码（preamble），可以用作定界符（并同步时钟）
      - 例如：传统以太网、802.11

2.掌握差错检测和纠正的基本原理和典型的编码方法

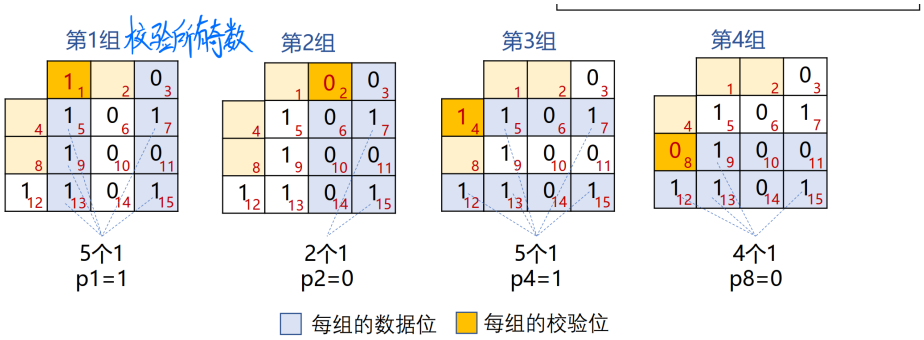
- 码字：一个包含m个数据位和r个校验位的n位单元，描述为(n,m)码，n=m+r
- 码率：码字中不含冗余部分所占的比例，m/n
- 海明距离：两个码字不同对应bit的数目；**n海明，检n-1位错，纠(n-1)/2位错**
- 检错码
  - 奇偶校验：在某位增加一位，如果1的个数为偶数则为0否则为1，可以检验奇数位错误
  - 校验和：可以检验1位错误
    - ➢ 以TCP/IP体系中主要采用的校验方法为例

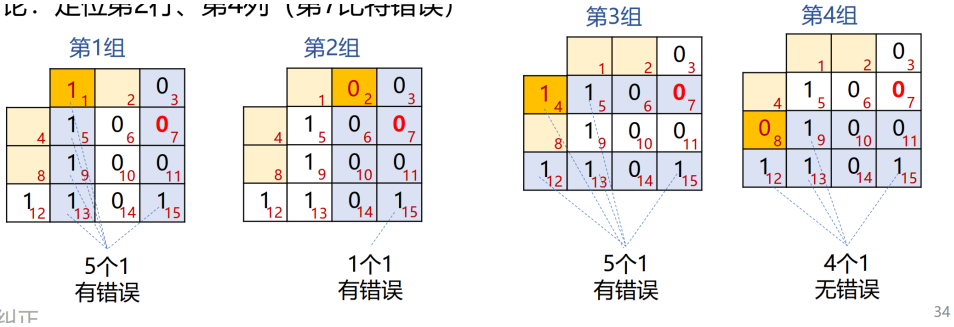


  - 循环冗余校验CRC
    - 双方实现商定一个n+1位的二进制G

- CRC内填充原始数据$D \times 2^n$对G的余数（异或除）
- 

- 接收方通过判断数据对G模是否为0来判断是否有错，可以检测少于N+1位的错误

- 纠错码
  - 海明码
    - 校验位： 2的幂次方位（记为p1, p2, p4, p8） ，缺省值为偶检验
    - 发送方计算校验值
    - 

    - 接收方再计算一遍，和发送方传过来的值做对比，定位错误
    - 

3.掌握无错信道和有错信道上停等协议的设计和实现方法

- 物理层进程和某些数据链路层进程，运行在**专用硬件**上
- 数据链路层进程的其他部分和网络层进程，作为操作系统的一部分（协议栈）， **运行在CPU上**
- 乌托邦式单工协议P1
  - 单工；完美信道（不需要纠错）；始终就绪；瞬间完成（不需要流量控制）无确认无连接
- 无错信道上的停等式协议P2
  - [ADD] 流量控制
  - [SOLVE] 停-等：等待确认，ACK帧（哑帧）
  - 停-等式协议（stop-and-wait）
- 有错信道上的单工停等式协议P3
  - [ADD] 差错控制
  - [SLOVE]

        - 数据帧丢失--计时器：如果发送的帧没有得到确认；
        - ACK帧丢失--序号SEQ：识别重复的帧
        - 信道利用率 (line utilization)=F/(F+R · RTT)
    - 问题：长肥网络（时延带宽积很大）下信道利用率过低

4.理解滑动窗口协议的基本原理

- 流水线需求
    - 发送方：要暂存哪些帧以便可能的重传
    - 接收方：如何能向网络层按序提交数据
    - 双方：允许发送方发多少帧以不淹没接收方
- 发送窗口
    - 发送端始终保持一个已发送但尚未确认的帧的序号表
    - 发送窗口大小 = **上界 - 下界**， 大小可变
- 一比特滑动窗口协议P4
    - 哑帧确认->捎带确认（piggybacking） ：将确认帧与反向数据帧合并，可以暂时延迟待发确认

5.掌握回退N和选择重传两种典型滑动窗口协议的工作机制

- 回退N协议P5
    - 基本原理：当发送方发送了N个帧后，若发现该N帧的前一个帧在**计时器超时**后仍未返回其确认信息，则该帧被判为出错或丢失，此时发送方就**重新发送出错帧及其后的N帧**
    - **接收方窗口大小为1，发送方窗口大小小于$2^n - 1$即可**
    - 累计确认：仅对最后一个分组发送确认
    - 缺点：正确帧如果乱序也要丢弃
- 选择重传协议P6
    - 接收窗口大于1
    - [ADD] 仅重传错误帧
    - [SOLVE] 接收窗口扩大，因此需要占用缓存
    - 如果有丢失，由于要等待超时，可能不如GBN!
    - 不能使用累计确认，因为如果出现选择重传无法定义是否返回确认
    - 与P5 GBN不同， P6是**给每个帧设置定时器**，发送端**只重传出错帧/超时**
    - **接收端缓存**所收到的**乱序帧**，当前面帧到达后一起按序提交上层
    - 重传优化：否认确认帧NAK，在计时器超时前就告知重发
    - 发送方和接收方**窗口尺寸都不能超过总数的一半**： 1<WR≤2n-1
- 二者比较
    - 累计确认在ack丢失时效率高，选择重传在ack丢失时效率低 ： 因为对累计确认来说后面的确认可以起到覆盖前一个确认的作用