



3.2 封装成帧和透明

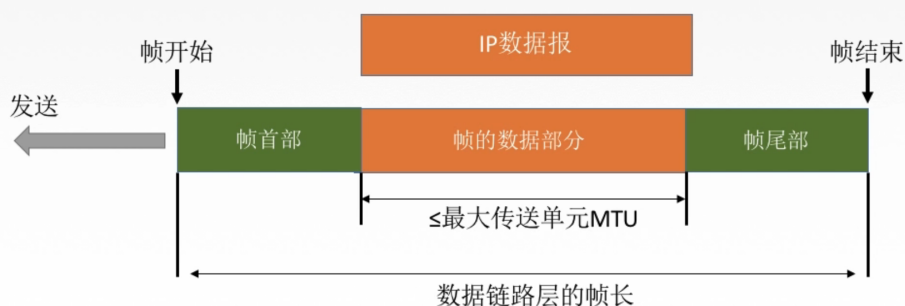
Property

封装成帧就是在一段数据的前后部分添加首部和尾部，这样就构成了一个帧。接收端在收到物理层上交的比特流后，就能根据首部和尾部的标记，从收到的比特流中识别帧的开始和结束。

首部和尾部包含许多的控制信息，他们的重要作用：**帧定界**（确定帧的界限）。

帧同步：接收方应当能从接收到的二进制比特流中区分出帧的起始和终止。

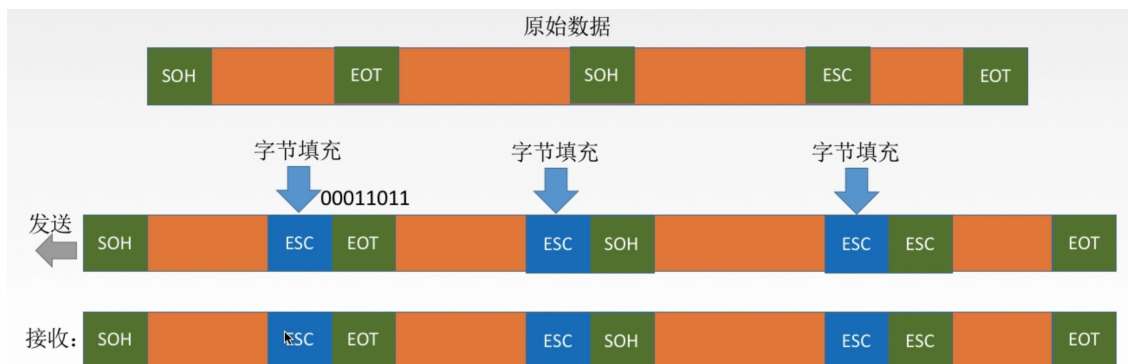
组帧的四种方法：1.字符计数法，2.字符（节）填充法，3.零比特填充法，4.违规编码法。



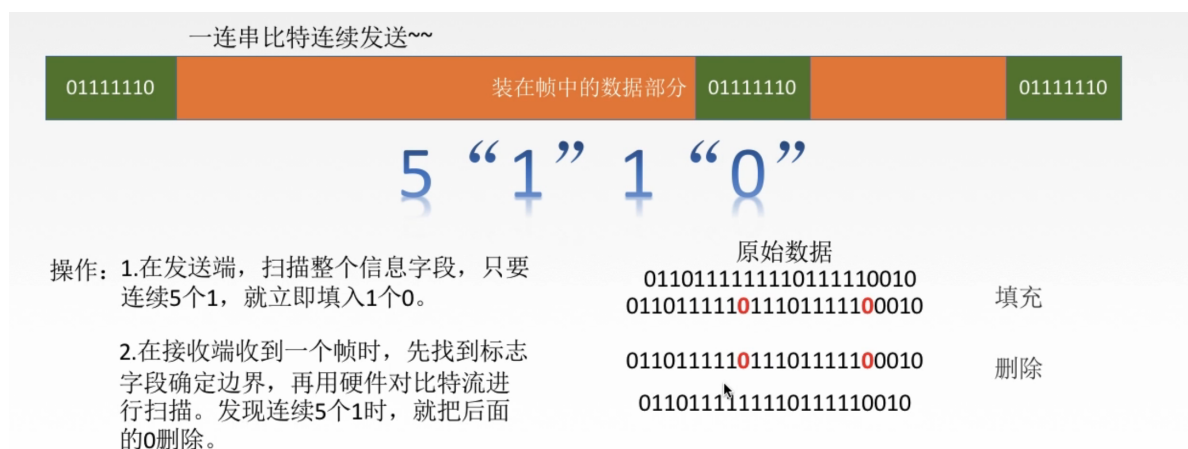
透明传输是指不管所传数据是什么样的比特组合，都应当能够在链路上传送。因此，链路层就“看不见”有什么妨碍数据传输的东西。

当所传数据中的比特组合恰巧与某一个控制信息完全一样时，就必须采取适当的措施，使收方不会将这样的数据误认为是某种控制信息。这样才能保证数据链路层的传输是透明的。

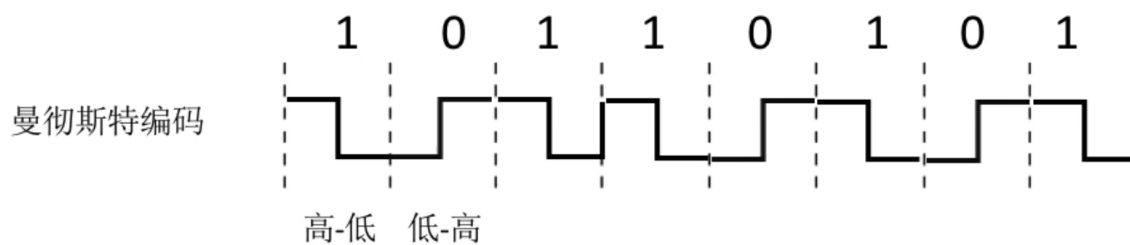
- 字符计数法
 1. 帧首部有一个字符表示帧的长短（真的长短包括这个字符）
 2. 缺点：一错百错
- 字符填充法



- 1.
 2. 首尾都要加一个字符表示start 和 end，在传纯文本的时候没有问题，传其他格式的时候由于数据部分可能会和首位控制字符重合，因此会用转义字符ESC填充
- 0比特填充法



- 违规编码法



可以用“高-高”，“低-低”来定界帧的起始和终止。

- 总结：

由于字节计数法count字段的脆弱，字符填充法的复杂，目前常用0bit填充法和违规编码法