# 数据链路层

## 数据链路层概述：

数据链路层位于物理层之上和网络层之下，主要是为网络层提供有效可靠的帧传输。

链路层的传输单元是数据帧。

## 数据链路层的功能：

### 1、封装成帧

封装成帧就是在一段数据的前后添加首部和尾部，添加的首部和尾部与数据构成一个数据帧，添加首部和尾部的主要功能是进行帧定界。

在用控制字符进行帧定界时，在数据段的首部添加SOH，以SOH作为帧开始，在尾部添加EOT，以EOT作为帧结束。

为了提高帧数据传输的效率，应该使帧的数据部分尽可能大于首部和尾部的长度。但是每一层路层协议都规定了帧的数据部分的长度上限——最大传送单元MTU(Maximum Transfer Unit)。

采用帧传输方式的好处是，在发现有数据传送错误时，只需将有差错的帧再次传送，而不需要将全部数据的比特流进行重传，这就在传送效率上将大大提高。但随后很快又恢复正常，于是重新从头开始发送刚才未发送完的帧。由于使用了帧定界符，在接收端就知道前面收到的数据时个不完整的帧(只有首部SOH，没有传输结束符EOT)，必须丢弃。而后面收到的数据有明显的帧定界符(SOH和EOT)，因此这是一个完整的帧，应当收下。

#### 常用的封装成帧实现透明传输的方法有：

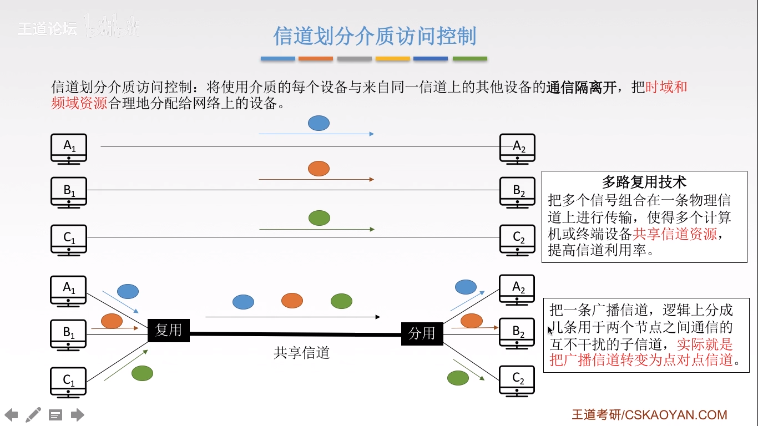
带字节/字符填充的首尾定界符法：让每一帧用一些特殊的字节（标志字节）作为开始和结束。缺点是当传输数据中出现这些特殊的字节时容易造成帧定界混淆。相应的解决方法是在二进制数中偶然出现的标志字节前加入一个转义字节“ESC”，这称为字节/字符填充法。

零比特填充法：我们以01111110作为一帧的开始和结束标志F字段。为避免混淆，当发送端的数据链路层碰到比特流数据中有5个连续1时，就立即在该比特流后填入一个0。接收端接收帧时，先找到F字段确定帧的边界，再对比特流进行扫描。每发现5个连续1时，删除其后的一个“0”，进行还原。零比特填充法使得一帧中两个F字段之间不会出现6个连续1（‘5’ ‘1’ ‘1’ ‘0’）

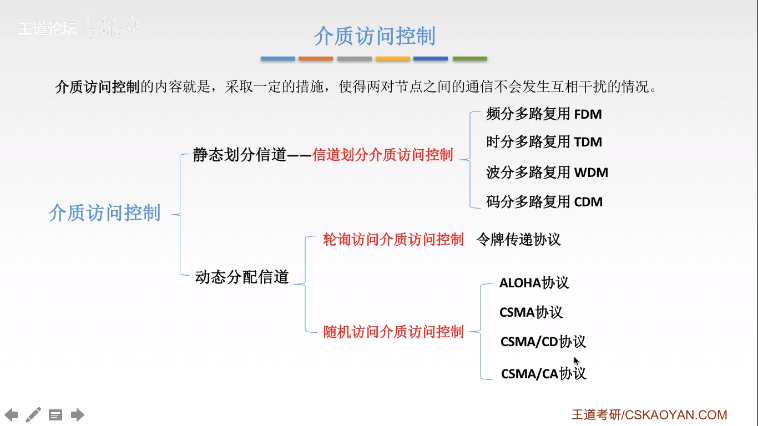
对于帧而言，不同的数据链路层协议对应着不同的帧，因此帧有多种，例如，PPP帧和以太网MAC帧。

### 2、介质访问控制

介质访问控制是指采取一定的措施，是的两对节点之间的通信不发生冲突



介质访问控制分为两类：静态划分信道和动态划分信道



静态信道划分是在进行数据传输之前将传输信道分开

动态信道划分是在进行数据传输时进行信道划分，动态信道划分又分为：轮询访问介质访问控制和随机访问介质访问控制

轮询访问介质访问控制：

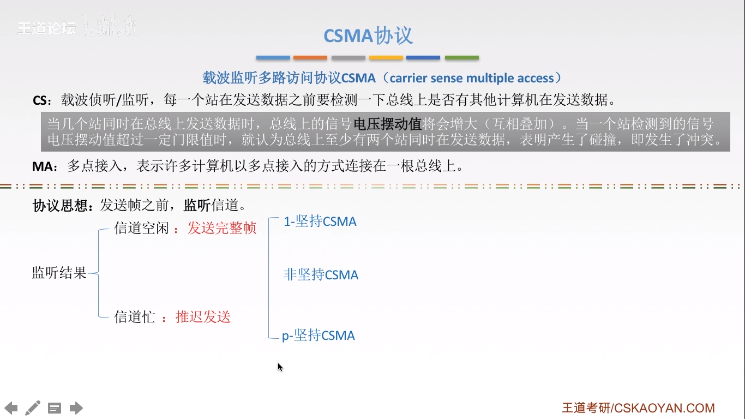






随机访问介质访问控制：

ALOHA协议：想发就发，隔一段时间再重新发送



CSMA协议：



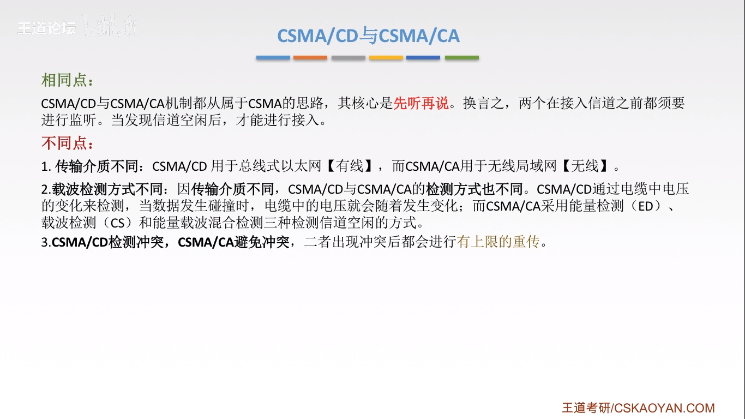
1. 坚持：一直监听

非坚持：随机监听

p-坚持：按照一定的概率进行发送

ALOHA：不监听就说

CSMA、CSMA/CD、CSMA/CA：先听再说

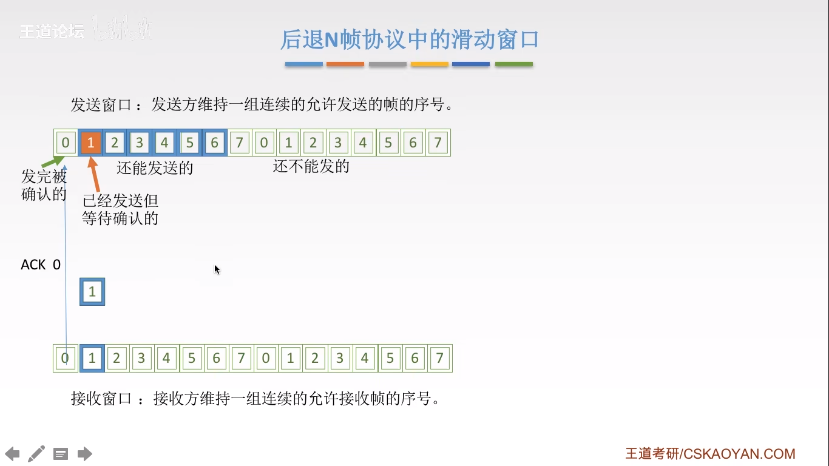


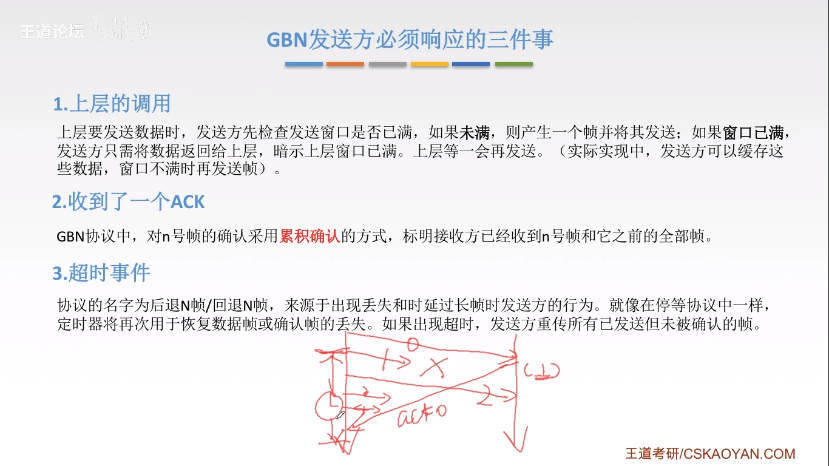
### 3、差错控制

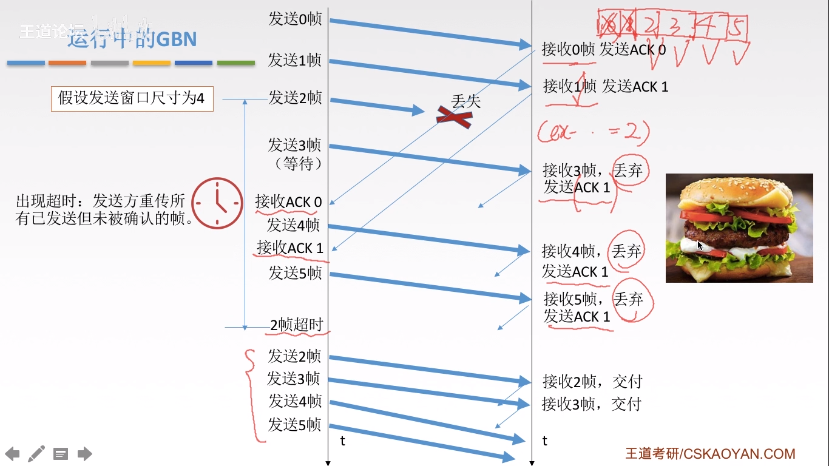
### 4、流量控制

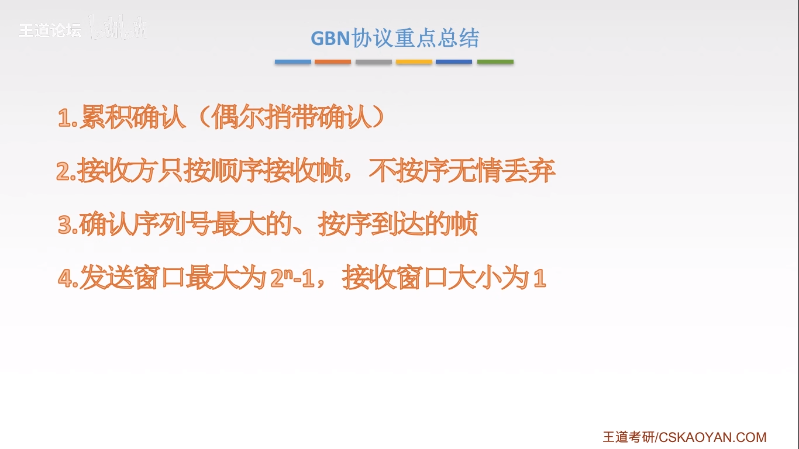
停止等待协议：信道利用率很低，发送方发送数据，接收方接受到数据回复确认信号，发送方再继续发送数据

GBN协议（后退N帧协议）：连续发送数据，如果某一帧数据丢失，那么从这一帧开始的数据需要全部重新发送

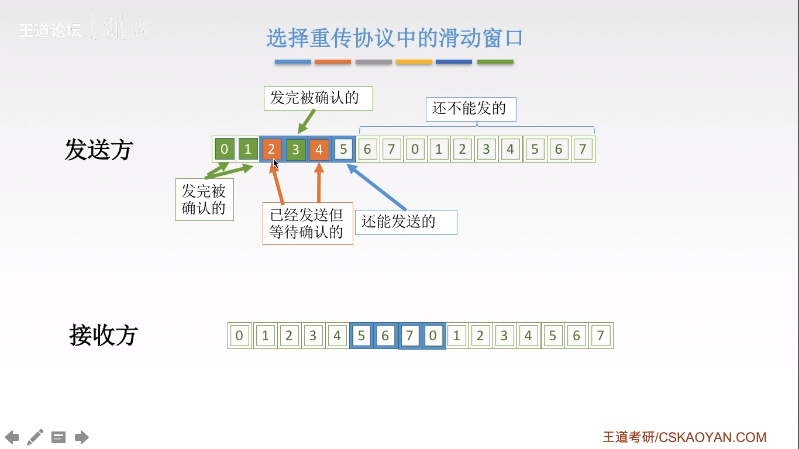


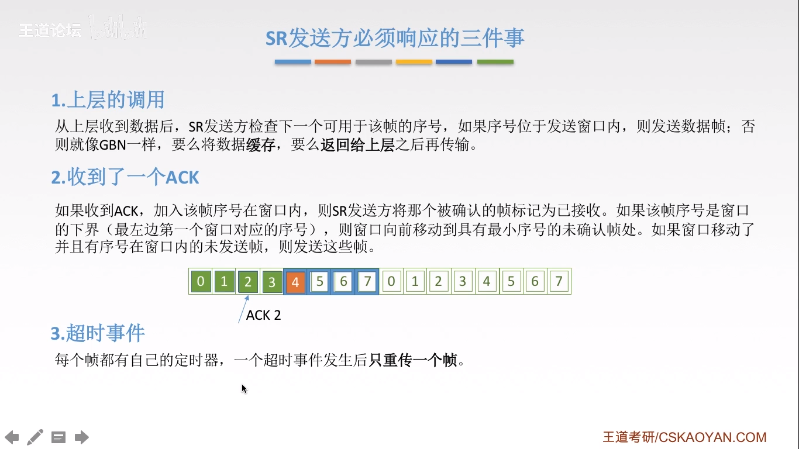


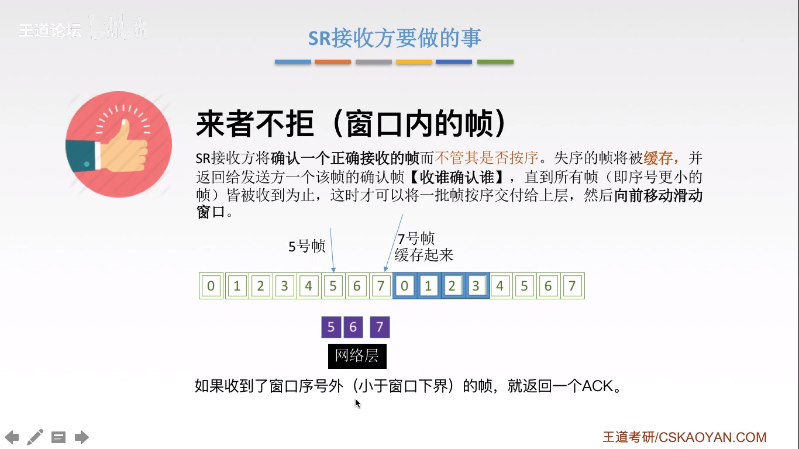


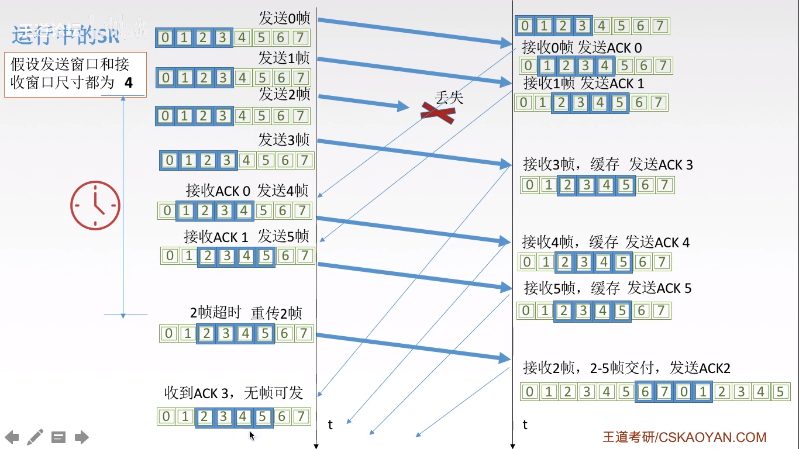


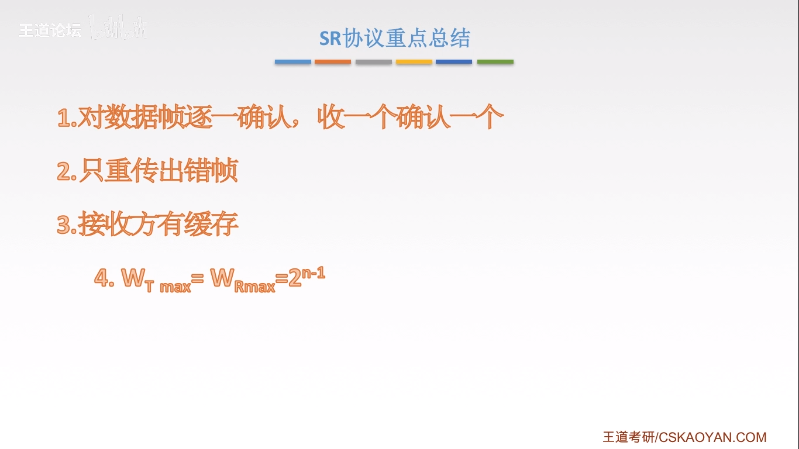
SR协议（选择重传协议）：只重新发送丢失的数据帧







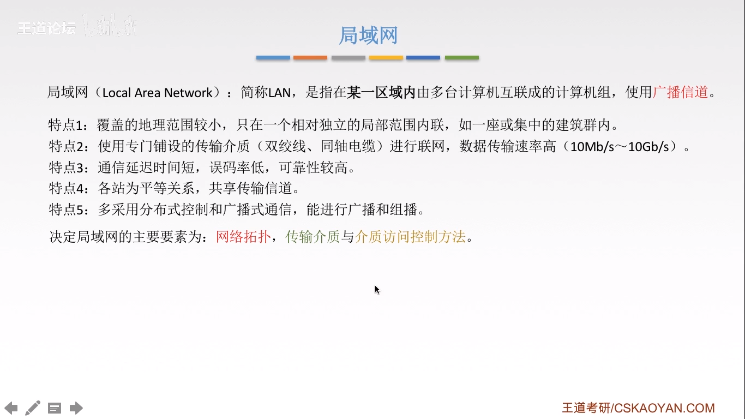




传输数据的两种链路



局域网：



链路层分为两层：LLC和MAC，LLC为网络层提供服务，MAC对接物理层



广域网：

