

物理层

虽然涉及到的知识点众多，但我觉得重点是调制解调器

解调器对收到的数据(data)，将模拟信号转成数字信号。

因为电话线路传输的是模拟信号，所以调制器对于将要发送的数据，将数字信号转成模拟信号。

至于模拟信号和数字信号的话，我的理解是，链路上传输的信号确实是模拟信号，但是这个模拟信号模拟的是数字信号，也就是说把模拟信号当数字信号处理的。

此外还有几种信道复用方式。分别是时分复用，统计时分复用，码分复用，频分复用。时分复用有点像 RR。

集线器：集线器的主要功能是对接收到的信号进行再生整形放大，以扩大网络的传输距离，同时把所有节点集中在以它为中心的节点上。它工作于物理层。也就是说他没法选择路径，他只是放大信号而已。功能很单一。工作很简单。集线器只能连接速率相同的机器。比如都是 10M 的或者都是 100M 的。

拓展：

数字信号在物理层传输时，本质上是数字信号还是模拟信号，为什么？

<https://www.zhihu.com/question/29244850/answer/107289428>

数据链路层

这一层发送的数据单元叫帧，帧其实就是数据链路层把从网络层拿到的数据的首尾各加上一段数据。分别是帧首部(左, SOH)和帧尾部(右, EOT)。

添加首尾的作用之一是区分帧的开始和结束

链路层的协议规定了：

帧的数据部分(上一层拿到的数据)的最大长度(MTU)。

透明传输：字符填充法

可靠传输：差错检测(冗余检验法 CRC) 实际上数据部分被添加了一个冗余码，所以上面写的有问题，并不是在上层拿到的数据两端直接添加首尾，而是先分好组(因为最大长度有限制所以要先分组)，然后在数据末尾添加 n 位冗余码，然后再添加首尾，这时候就叫帧了。

但是具体的冗余检测法我并没有理解。之后有需要再学习。

流量控制：

带宽差异，如果发送快接收慢数据会丢失。所以收方控制发方的速度。

确认和重传机制。接收方正确接收（冗余检验法）帧之后，会有一个反馈信号(ACK)，然后发送方再发下一帧。如果收到的帧是错误的返回一个 NAK，发送方收到 NAK 之后会拿缓存里面的数据重新发送一次。

如果一定时间内接收方没有反馈信号，说明数据丢失了，这时候自动重传。

返回信号丢失也重传，因为发送方也分不出来到底是返回信号丢了还是数据没发送过去。反正[超时计时器](#)到了没收到信号就重传。

为了解决[重复帧](#)的问题，给帧加序号(估计是加在帧首部了)，序号只用 0 和 1 表示(也就是说只占 1 位)

协议：PPP(点对点协议) 任意两个物理设备间适用

上面讲的帧封装，冗余校验，确认重传等在这个协议里面都有详细规定。比如冗余校验的那个占位符，在 PPP 协议里面就有详细规定，规定到底用什么转义字符作为占位符。并且还规定开始符号和结束符号具体都是什么。还有帧的最大长度，并且还隔几分钟就检测一下链路是否通畅(数据链路协议 LCP--建立，配置，测试链路的方法，控制链路上的节点(开关?)，这种控制是物理上的，抱着链路在物理上是连通的。)

16 进制 mark 地址，标识上网设备用的。

数据压缩

协议：P2P(对等计算) 包括 LCP 和 NCP

NCP 是网络控制协议。逻辑上控制节点(路由器)。

ppp 的帧，PPP 是面向字节的(字节为最小单位)，所以所有帧都是整数字节



F: 标志字段，相当于 SOH。

A: 地址字段, 网络层使用的现在是空闲状态。

C: 控制字段

协议: 因为 ppp 可以向上兼容任何协议, 所以这里具体写什么取决于上层吧。

FCS: 检验序列, 也就是冗余码。看来冗余码不是写在 ip 数据报里面的。

F: 标志字段

ppp 为了区分帧首尾和 ip 数据报(透明性), 想出了两种方法。分别是字符填充法和 0 比特填充法。

ppp 的字符填充

由硬件完成, 将 0x7E 转 0x7D 和 0x5E。将 0x7D 转 0x7D 和 0x5D。控制字符前面加 0x7D 并且将控制字符的编码加以改变。

ppp 0 比特填充

发送端发现有 5 个连续 1 时替换成 0。接收端发现五个连续 1 的时候就把 1 后面的 0 删除。

关键: 802.3 标准将数据链路层分为两层, 分别是逻辑链路控制(LLC)和媒体接入控制(MAC)。

但是设备厂商不管 LLC 层, 而是把逻辑控制融入到 MAC 层中。

网卡叫通信适配器

- 1 进行串行并行转换, 就是把运输层的 ip 数据报封装成帧。
- 2 对数据进行缓存(流量控制, 可靠传输)
- 3 在 windows 里面安装驱动程序, 其实就是给 windows 提供控制接口。
- 4 实现以太网协议, 其实就是实现 ppp 协议。

以太网:

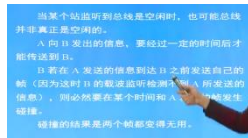
站的概念。

数据链路层在以太网的情况下, 没有重传, 也就不需要对帧编号。

对数据的检测交给上一层来管理(网络层? 运输层? 应用层?), 如果上一层需要重传那就给发送方发送信号。在这种重传机制下, 其实数据链路层就是个傻子, 它只做数据发送和接收, 即使数据重新发送了它也不知道, 因为它不检测数据。

多点接入: 多个计算机接一根总线上。

载波监听: 发送数据前用电子技术检测总线上有没有其他计算机在发送的信号。有就不发,



避免碰撞。

碰撞：碰撞发生时总线上的信号会失真。无法从信号中恢复出有效信息。碰撞之后会重传最多重传 16 次(二进制指数类型退避算法)

强化碰撞：发生碰撞后，发送干扰信号，让总线上所有计算机都知道发生了碰撞。

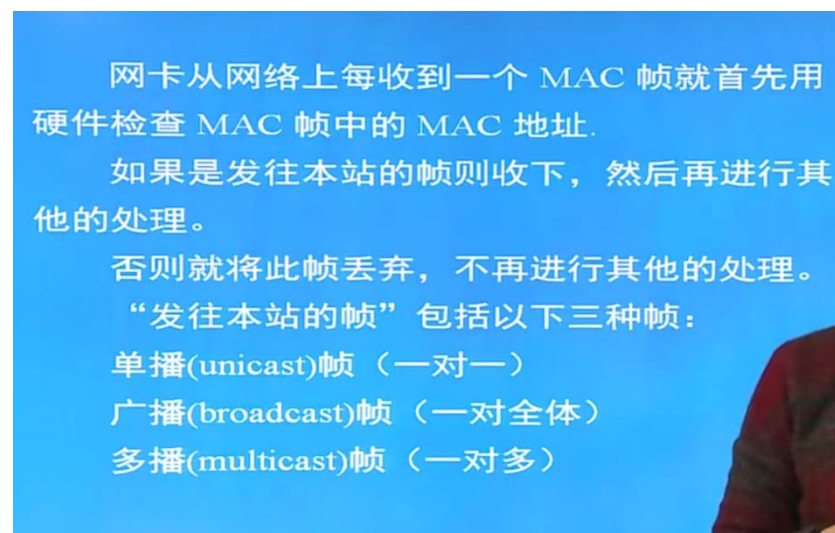
以太网规定凡是小于 64 字节的数据是由于发生碰撞而异常终止的无效帧

利用集线器扩展以太网

信道利用率的计算（帧越长，时延带宽积越低，a 的值就越小。这时候信道利用率也就越高）

mac 地址(传输过程中的物理设备的名字管理。)这里所说的地址其实是名字或标识符。802 标准规定 mac 地址是 48bit。为了方便，采用 12 个 16 进制表示 mac 地址。前三个 16 进制是厂家的标识符。后三个是拓展。最后 6 个是用来确定具体机器的？（“前 6 位用来标记特定厂商，后 6 位用来标记具体产品。”百度上有人这么说的）

路由有两个 mac 地址，其他设备是一个 mac 地址。一个网卡对应一个 mac 地址，路由有两个网卡。



网卡：收到 mac 帧之后，先看看是不是发给自己这个站的。如果不是就丢弃。如果是发给本站的(比如发给连到自己网络上的其他主机或者其他路由的)就收下。然后再进行其他处理。

发往本站的帧有三种：

单播帧：一对一（我觉得应用 SDN 之后就会变成单播帧）

广播帧：一对全体

多播帧：一对多

mac 帧和 PPP 帧的区别

这里 mac 帧存在于以太网，而 ppp 帧是广域网。广域网是用 ip 地址寻址的(ip 地址是网络层的寻址机制)，但是局域网内部有自己的寻址机制，局域网内部使用 mac 地址来做寻址(链路层)。

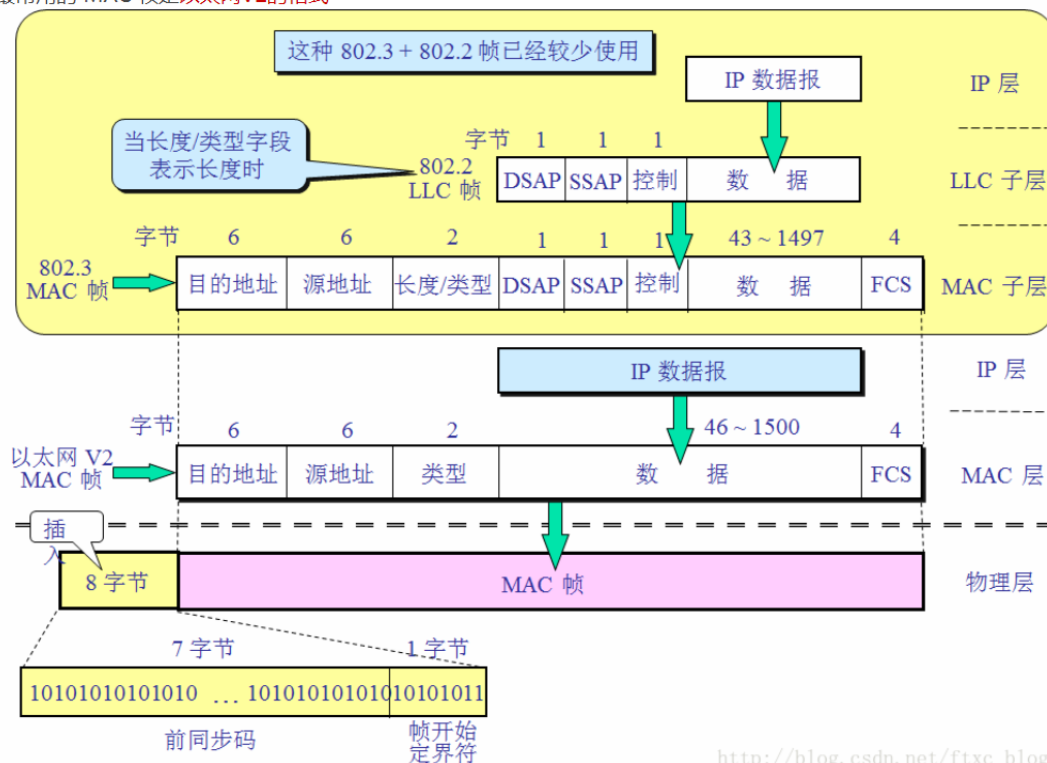
ppp 属于广域网范畴，MAC 是局域网范畴，按实际情况和环境就选用不同的协议，ppp 支持的网络结构只能是点对点，mac 支持多点对多点。

以太网中用 mac，远程的话就用 ppp（如 ADSL 拨号，就是基于 ppp 的）。

以太网封装和 PPP 封装都是对 IP 层的封装

这里 mac 帧其实是在 ppp 帧的基础上再进行添加首尾部分，从而组成 mac 帧。

最常用的 MAC 帧是以太网V2的格式



类型字段（2 个字节）：

用来标志上一层使用的是什麼协议，以便把收到的 MAC 帧的数据上交给上一层的这个协议。其实就是为了向上兼容。(不过我猜测，它的作用只是让下一个接收这个帧的机器以此来判断我这个帧是遵循哪个协议规范的帧格式。因为格式不同对于数据解释的规则是不一样的。（就好像编码，不过这好像就是编码？））

数据字段（46-1500）：

正式名称是 MAC 客户数据字段最小长度 64 字节-18 字节的首部和尾部 = 数据字段的最小长度。最小长度就是 46 字节，因为规定传输的最小字节是 64，首尾加起来有 18 了，所以数据字段最小是 46。有效长度是 64~1518。超过了也不行。

FCS 字段（4 字节）：

当数据字段的长度小于 46 字节时，应在数据字段的后面加入整数字节的填充字段，以保证以太网的 MAC 帧长不小于 64 字节(视频里面说这是冗余校验序列，我觉得这个比较靠谱，因为你填充不可能只有 4 字节啊，46 加 4 也不等于 64 啊)

在帧的前面插入的 8 字节中的第一个字段共 7 个字节，是前同步码，用来迅速实现 MAC 帧的比特同步。第二个字段是帧开始定界符，表示后面的信息就是 MAC 帧。一定要注意，8 个字节不属于 mac 帧。

如果 mac 帧无效(不是整数字节、帧长不符合要求、FCS 检测)，那就就把帧直接丢弃。以太网不需要接收端发送确认帧。也就是不进行确认。(那以太网没有流控吗？还是说重传就已经算流控了)

帧间最小间隔是 9.6us，也就是争用期的时间。也是发送 96bit 数据的时间，一方面是用来确认对方是否正确接收数据(没有返回帧，所以到底怎么实现流控的呢？难道只要不发生碰撞就一定正确传输了？明显不是啊，那如果帧无效发送方怎么知道要不要重传呢？)，另一方面是给接收端一个处理帧的时间。

集线器：半双工(碰撞终止发送)

物理层 mac 帧。多个集线器搭建网络的话，碰撞域就变大了，一个站发送信息，所有集线器上面的站都要受到广播。其实多个集线器还是一根总线。

网桥：半双工(碰撞终止发送)

数据链路层 mac 帧，拿到 mac 帧之后不是广播，而是先解析目的地址，然后根据目的

mac 地址一对一的发送 mac 帧。

维护一个表

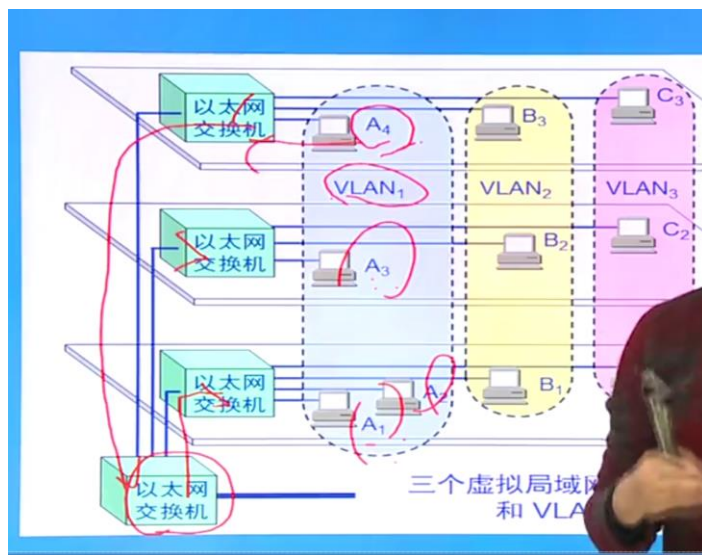
| | |
|--------------------------|----------------------------|
| 站地址(所有连接到总线上的机器的 mac 地址) | 端口(网桥一共俩端口，所以端口号不是 1 就是 2) |
| | |

网桥替代集线器。一个集线器相当于一根总线，但是网桥拉出来的线都是总线。

交换机：全双工(无碰撞，每个站一条总线)

VLAN (虚拟局域网) 因为每台 PC 独占一条总线，所以用逻辑上构建 VLAN。

详细解释的话，其实就是在 mac 帧原地址和类型之间再插入一个 4 字节的片段。叫做 vlan 标记(tag)，交换机通过这个 tag 可以知道这条信息属于哪个站。值得注意的是，即使这里插入了 4 字节的片段，也不和原来的 46~1500 冲突。最小字节数还是 46~1500。



载波延伸(只是半双工 1000M 网络才用到，扩充 mac 帧，最小 512 字节(原来 64))

分组突发(只延伸第一个帧，后面帧与前面的是 0.96us 就算是一起的数据。不需要单独进行

载波延伸)

网络层

虚电路服务：X.25 协议

流控是由网络负责的(路由器负责？)

虚电路号存在 X.25 分组的首部

数据报服务：IP 协议(TCP/IP 协议体系之一)

流控是由发送方和接收方负责的。(也就是之前学的确认重传机制，做这件事情的不是路由而是两头的 pc(的网卡)。？)

IP 协议配套使用的有三个协议，分别是

地址解析协议(ARP)、逆向地址解析协议(RARP)：为了向下兼容

网际控制报文协议(ICMP)、网际组管理协议(IGMP)：为了向上兼容