

3-01 数据链路（即逻辑链路）与链路（即物理链路）有何区别？“电路接通了”与“数据链路接通了”的区别何在？

链路是物理层面的概念，也叫物理链路，是相邻节点间的物理线路，并且该节点之间没有其他任何设备，链路两端称为点对点通信。当两台计算机通信时，中间往往有许多条链路，构成了端到端的通信。

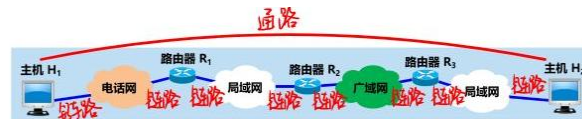


图1 链路

数据链路是逻辑层面的概念，也叫逻辑链路，本质上就是在物理链路的基础上，增加实现规定协议的软硬件。当两台计算机通信时，交换的数据在物理链路上传输，并且通过数据链路提供的软硬件实现相应的协议，如PPP协议和CSMA协议。典型的数据链路有网卡。

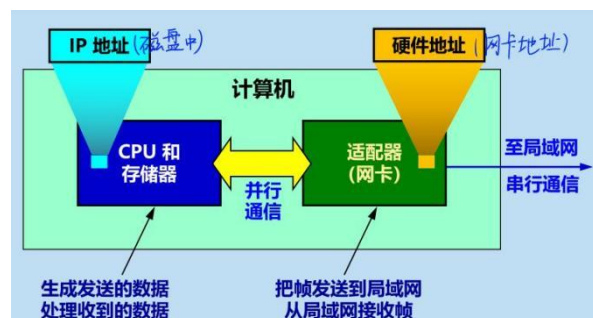


图2 网卡通信

“**电路接通了**”表示链路两端的交换机开机了，物理链路此时已经可以传输比特流。但由于缺少相应协议的约束与控制，并不能保证数据传输的可靠性。而“**数据链路接通了**”则表示建立了数据链路连接，通过其检测、确认、重传等功能确保了可靠的数据传输。

3-04 数据链路层的三个基本问题（封装成帧、透明传输和差错检测）为什么都必须加以解决？

封装成帧是在一段数据的前后分别添加首部和尾部，构成一个帧。首部和尾部中包含了一些控制信号，用来确定帧的界限。由此封装而成的帧就是数据链路层的数据传送单元，可以在数据链路层间进行传输。在实际传输中，数据链路层接收来自物理层比特流，通过控制字符“SOH”和“EOT”读取一整个帧，并将数据部分截取上传给网络层。这是数据链路层传输的基本数据形式，必须加以解决。

透明传输指的是数据部分可以出现任意字符，包括首部和尾部的控制信息字符组合，并且数据链路层在解码的时候不会将其认为是首部和尾部的控制信号。其实现方法就是发送方的数据链路层对待发送帧的数据部分中出现的控制字符“SOH”和“EOT”前插入转义字符“ESC”，接收方的数据链路层对删除接收到的帧的数据部分中出现的转义字符“ESC”。这是使得帧的数据完整传输的规则，必须得到保障，否则帧被误截断，数据丢失，达不到传输的理想效果。

差错检测主要是为了防止比特差错（0误传为1，1误传为0）而进行的准确性检测，目前数据链路层广泛使用的检错技术是循环冗余检验CRC。此外还有一些传输差错如帧丢失、帧重复、帧失序，可以通过帧编号、确认、重传等机制加以避免。差错检测是确保帧正确传输的保障，如果没有差错检测，通信可能出现一定数量的错误。

3-07 要发送的数据为1101011011，采用CRC的生成多项式是 $P(X)=X^4+X+1$ ，试

求应添加在数据后面的余数。

数据在传输过程中最后一个 1 变成了 0, 问接收端能否发现?

若数据在传输过程中最后两个 1 都变成了 0, 问接收端能否发现?

采用 CRC 检验后, 数据链路层的传输是否就变成了可靠的传输?

(1) 被除数 $D=1101011011$, 除数 $P=10011$, 利用模 2 除法, 草稿如下:

$$\begin{array}{r} 10011 \overline{) 1101011011} \\ \underline{10011} \\ 00001 \\ \underline{00000} \\ 00010 \\ \underline{00000} \\ 00101 \\ \underline{00000} \\ 01011 \\ \underline{00000} \\ 10110 \\ \underline{10011} \\ 01010 \\ \underline{00000} \\ 10100 \\ \underline{10011} \\ 01110 \\ \underline{00000} \\ 1110 \\ \underline{1110} \\ 0000 \end{array}$$

因此余数 $R1=1110$;

(2) 若数据在传输过程中最后一个 1 变成了 0, 对接收到的数据作模 2 除法:

$$\begin{array}{r} 10011 \overline{) 1101011010} \\ \underline{10011} \\ 00001 \\ \underline{00000} \\ 00010 \\ \underline{00000} \\ 00101 \\ \underline{00000} \\ 01010 \\ \underline{00000} \\ 10101 \\ \underline{10011} \\ 01110 \\ \underline{00000} \\ 11011 \\ \underline{10011} \\ 10000 \\ \underline{10011} \\ 0011 \end{array}$$

此处余数 $R2=0011 \neq 0000$, 接收端能够检测出传输过程有差错;

(3) 若数据在传输过程中最后两个 1 都变成了 0, 对接收到的数据作模 2 除法:

$$\begin{array}{r} 10011 \overline{) 1101011010} \\ \underline{10011} \\ 00001 \\ \underline{00000} \\ 00010 \\ \underline{00000} \\ 00100 \\ \underline{00000} \\ 01000 \\ \underline{00000} \\ 10001 \\ \underline{10011} \\ 01010 \\ \underline{00000} \\ 01011 \\ \underline{00000} \\ 10110 \\ \underline{10011} \\ 0101 \end{array}$$

此处余数 $R3=0101 \neq 0000$, 接收端能够检测出传输过程有差错;

(4) 采用 CRC 检验后, 数据链路层的传输并没有变成可靠的传输。因为 CRC 检验只能

对接收到的数据进行检测，若与发送时数据不同则丢弃。这只能保证“无比特差错”，而不能保障“无传输差错”。前者以**比特**为基准，CRC 只保障了一个帧中比特的准确性；后者以**帧**为基准，有可能出现帧丢失、帧重复、帧失序等问题，CRC 无法改变，需要使用帧标号、确认和重传等机制才能实现可靠传输。

3-09 一个 PPP 帧的数据部分（用十六进制写出）是 7D 5E FE 27 7D 5D 7D 5D 65 7D SE 。试问真正的数据是什么（用十六进制写出）？

PPP 帧通过字节填充实现透明传输，因此真正的数据为 7E FE 27 7D 7D 65 7E

3-10 PPP 协议使用同步传输技术传送比特串 011011111111100。试问经过零比特填充后变成怎样的比特串？若接收端收到的 PPP 帧的数据部分是 000111011111011110110，问删除发送端加入的零比特后变成怎样的比特串？

比特串 011011111111100 经过零比特填充后变成 011011111011111000；

接收端收到的 PPP 帧的数据部分是 000111011111011110110，则删除发送端加入的零比特后比特串为 000111011111111110。

3-13 局域网的主要特点是什么？为什么局域网采用广播通信方式而广域网不采用呢？

局域网最主要的特点是网络为一个单位所有，覆盖范围和站点数目比较有限。其优点主要是具有广播功能、便于系统扩展与演变、可靠性高和可用性强。

正因为局域网地理范围较小，是一个单位所有的，因此采取广播通信比较方便，也可以保证通信的安全性；如果广域网也采用广播通信方式，不仅对信道资源造成极大的浪费，而且一旦非目标地址选择接收广播传来的数据，还会有数据泄露的风险。因此局域网采用广播通信方式而广域网不采用。

3-16 数据率为 10Mbit/s 的以太网在物理媒体上的码元传输速率是多少码元 / 秒？

以太网采用曼彻斯特编码，每个周期内信号还要跳变一次，即一个数据（0 或 1）正中间会出现一次电平跳跃。因此数据率为 10Mbit/s 的以太网在物理媒体上的码元传输速率是 20M 码元 / 秒。

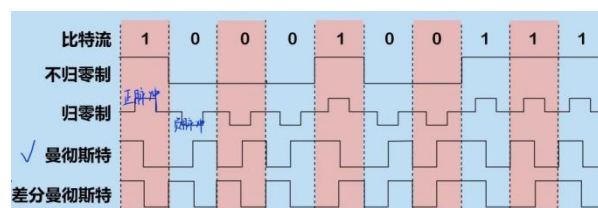


图 3 曼彻斯特编码

3-20 假定 1km 长的 CSMA/CD 网络的数据率为 1Gbit/s。设信号在网络上的传播速率为 200000 km/s。求能够使用此协议的最短帧长。

信号传播速率 $v=200000 \text{ km/s}$ ，电缆长度 $l=1\text{km}$ ，所以传播时延 $\tau=l/v=5\mu\text{s}$ ；

因此端到端的传播时延 $t=2\tau=10\mu\text{s}$ ；

因此此协议的最短帧长=端到端的传播时延*发送速率= $10\mu\text{s} \times 1\text{Gbit/s}=10\text{kbit}$

3-24 假定站点 A 和 B 在同一个 10 Mbit/s 以太网网段上。这两个站点之间的传

播时延为 225 比特时间。现假定 A 开始发送一帧，并且在发送结束之前 B 也发送一帧。如果 A 发送的是以太网所容许的最短的帧，那么 A 在检测到和 B 发生碰撞之前能否把自己的数据发送完毕？换言之，如果 A 在发送完毕之前并没有检测到碰撞，那么能否肯定 A 所发送的帧不会和 B 发送的帧发生碰撞？（提示：在计算时应当考虑到每一个以太网帧在发送到信道上时，在 MAC 帧前面还要增加若干字节的前同步码和帧定界符）

根据协议规定，以太网所容许的最短的帧为 512bit，封装成 MAC 帧时需要在前面插入 7byte 的前同步码和 1byte 的帧开始界定符。因此 A 发送的数据长度为 $512+8*8=576\text{bit}$ ，发送完需要 576 比特时间；

假设 A 第 0s 开始发送数据，B 第 t 秒发送数据。而 A 和 B 两个站点的传播时延为 225 比特时间，根据 CSMA/CD 规则，B 只会在没有侦听到其他站的载波信号时发送数据，因此只会在 $t < 225$ 比特时间内发送数据。哪怕 B 是在 $225-\delta$ 时发出信号，传递到 A 时 $t=450-\delta$ ，此时 A 并没有发送完。因此如果 A 在发送完毕之前并没有检测到碰撞，那么可以肯定 A 所发送的帧不会和 B 发送的帧发生碰撞。

3-25 在上题中的站点 A 和 B 在 $t=0$ 时同时发送了数据帧。当 $t=225$ 比特时间，同时检测到发生了碰撞，并且在 $t=225+48=273$ 比特时间完成了干扰信号（**强化碰撞**）的传输。A 和 B 在 CSMA/CD 算法中选择不同的 r 值退避。假定 A 和 B 选择的随机数分别是 $r_A=0$ 和 $r_B=1$ 。试问 A 和 B 各在什么时间开始重传其数据帧？A 重传的数据帧在什么时间到达 B？A 重传的数据会不会和 B 重传的数据再次发送碰撞？B 会不会在预定的重传时间停止发送数据？

基本退避时间 $= 2\tau = 450$ 比特时间，重传时延 $= r \times \text{基本退避时间}$ 。因为 $r_A=0$ ， $r_B=1$ ，所以 A 立即检测信道，B 推迟 **450 512** 比特时间再检测信道。

在 $t=225$ 比特时间时 A 和 B 分别检测到其他站点的载波，发现冲突，于是发送 48bit 的干扰信号，在 $t=225+48+225=498$ 时分别接收到最后 1bit 的干扰信号。

然后 A 站点立即检测信道，连续 96 比特时间没有监听到其他站点的载波后便开始第一次重传，此时 $t=498+96=594$ 比特时间。重传的数据在 $t=594+225=819$ 比特时间时到达 B 端口，最后 1bit 在 $819+576=1395$ 比特时间到达。

B 在 $t=498$ **273** 时执行退避算法的推迟（不必等到强化信号全都发完且到达对面），在 $t=273+512=785$ 比特时间时开始检测信道，持续 96 比特时间后 $t=785+96=881$ ，但在这之前 A 重传的数据已经到达 B，因此 B 还是不能发送数据。

综上，预计 A 在 $t=594$ 比特时间重传数据帧，B 在 $t=881$ 比特时间重传数据帧；A 重传的数据帧在 $t=819$ 比特时间到达 B，且不会与 B 发生碰撞（因为 B 还没发送）；B 会在预定的重传时间停止发送数据。

3-30 在图 3-30 中，某学院的以太网**交换机**有三个接口分别和学院三个系的以太网相连，另外三个接口分别和电子邮件服务器、万维网服务器以及一个连接互联网的路由器相连。图中的 A、B 和 C 都是 **100 Mbit/s 以太网交换机**。假定所有的链路的速率都是 100 Mbit/s，并且图中的 9 台主机中的任何一个都可以和任何一个服务器或主机通信。试计算这 9 台主机和两个服务器产生的总的吞吐量的最大值。为什么？

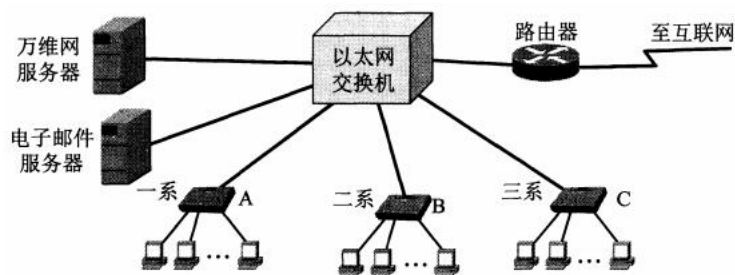


图 3-30 习题 3-30 的图

因为 A, B 和 C 都是 100Mbit/s 交换机, 所以 A, B 和 C 各自总吞吐量为 300Mbit/s, 9 台主机和两个服务器产生的总的吞吐量为 $3 \times 300 + 100 + 100 = 1100 \text{ Mbit/s}$ 。

(若 A, B 和 C 都是 100Mbit/s 集线器就是另一种情况, 答案是 500Mbit/s)

3-33 在图 3-31 中, 以太网交换机有 6 个接口, 分别接到 5 台主机和一个路由器。

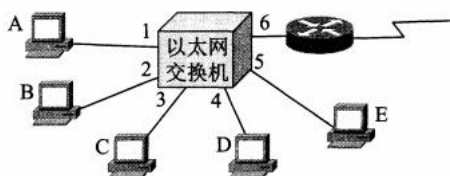


图 3-31 习题 3-33 的图

在下面表中的“动作”一栏中, 表示先后发送了 4 个帧。假定在开始时, 以太网交换机的交换表是空的。试把该表中其他的栏目都填写完。

| 动作 | 交换表的状态 | 向哪些接口转发帧 | 说明 |
|----------|---------------------|---------------------------|---------------------------------|
| A 发送帧给 D | 记录接口 1 对应 MAC 地址为 A | 2-B, 3-C, 4-D, 5-E, 6-路由器 | 发送前交换表为空, 因此只能广播发送 |
| D 发送帧给 A | 记录接口 4 对应 MAC 地址为 D | 1-A | 交换表中已经记录了 A 对应的接口为 1 |
| E 发送帧给 A | 记录接口 5 对应 MAC 地址为 E | 1-A | 交换表中已经记录了 A 对应的接口为 1 |
| A 发送帧给 E | 不添加地址 | 5-E | 交换表中已经记录了 E 对应的接口为 5, 所以不用添加新纪录 |