第 3 章

数据链路层

数据链路层属于计算机网络的低层，数据链路层有两种截然不同的信道。第一种类型链路层信道是广播信道。这种信道使用一对多的广播通信方式，同一信道上连接的主机很多，需要使用专用的介质访问控制协议来协调传输和避免“碰撞”，局域网中常用这种信道。第二种类型链路层信道是点对点信道。这种信道使用一对一的点对点通信方式。如两台路由器之间的通信链路或一个住宅的拨号调制解调器与一个 ISP 路由器之间的通信链路。协调点对点信道的访问是很容易的，但也存在一些重要问题，如组帧、可靠数据传输、差错检测和流量控制等。

如图 3.1 所示，在两台主机通过互联网通信时，从源主机 H_1 开始，经过一系列路由器（R_1、R_2、R_3），到目的主机 H_2 结束。所经过的网络可以是多种，如电话网、局域网和广域网。该通信路径由一系列通信链路组成。从协议的层次看，主机 H_1 和 H_2 有完整的协议层次，路由器的协议栈只有下面三层。数据进入路由器后先从物理层上到网络层，在网络层的转发表中找到下一条的地址后，再下到物理层转发数据。

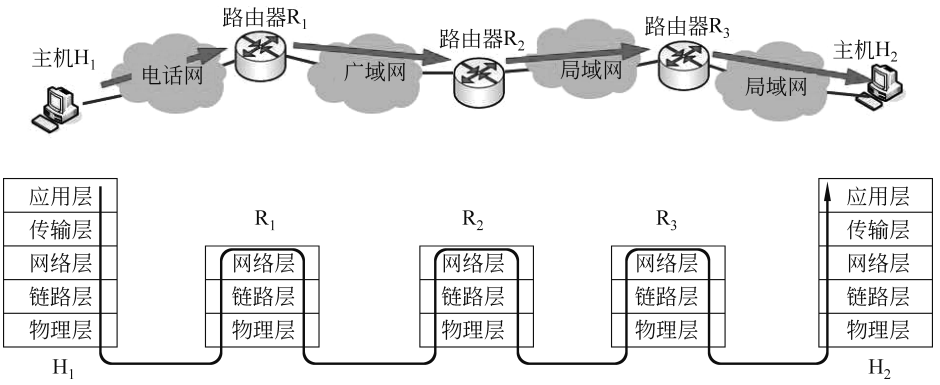


图 3.1　数据链路层

当我们研究数据链路层的问题时，很多情况下可以从各结点协议栈的数据链路层水平方向来着眼，如图 3.2 所示。当主机 H_1 向 H_2 发送数据时，我们可以想象数据是在数据链路层从左向右水平传送，即通过这样的四段链路：H_1 链路层→R_1 链路层、R_1 链路层→R_2 链路层、R_2 链路层→R_3 链路层和 R_3 链路层→H_2 链路层。

协议数据单元如何通过各段链路？在单段链路上，网络层的“分组”如何被封装成数据链路层的“帧”？数据链路层协议能够提供可靠地传输吗？在整个通信路径上不同的链

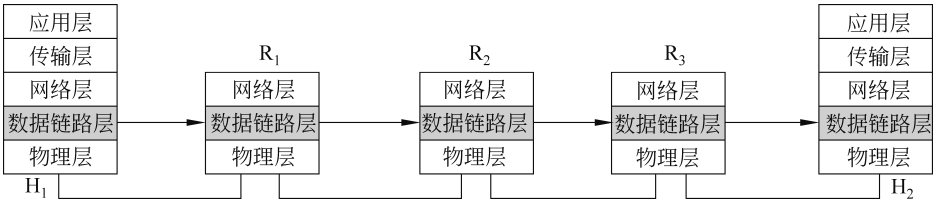


图 3.2　只考虑数据在数据链路层流动

路是采用的相同的链路层协议吗？这类重要问题我们就在这一章来回答。

3.1　数据链路层的基本概念

3.1.1　数据链路和帧

先来学习一些术语。为方便讨论，我们把主机和路由器统称为结点，我们将不关心一个结点是主机还是路由器。我们把沿着通信路径连接相邻结点的通信信道称为链路，有人将其称为物理链路。链路的中间没有任何其他的交换结点。两个主机通信时，通信路径上要经过许多独立的链路。当在一条通信路径上传输数据时，除了要有物理链路外，还必须要有通信协议来控制这些数据的传输。把实现数据传输协议的硬件和软件加到链路上，就构成了数据链路，有人将其称为逻辑链路。网络适配器就是实现这些协议的硬件和软件。一般的适配器包含了数据链路层和物理层这两层的功能。链路层协议交换的数据单元称为帧(frame)。

所有的数据链路层的基本功能都是将数据帧通过单条链路从一个结点移动到相邻结点，如图 3.3 所示，但具体细节依赖于该链路上应用的具体数据链路层协议。链路层协议包括以太网、IEEE 802.11 无线局域网、令牌环和 PPP 等。链路层的一个重要特点是在通信路径的不同链路上可能由不同的链路层协议来处理。例如在第一段链路上可能由 PPP 来处理，在最后一段链路上可能由以太网来处理，在中间的链路上由广域网链路层协议来处理。需要着重注意的是，不同的数据链路层协议的提供功能是不同的。例如一个数据链路层协议可能提供可靠的交付，另一个数据链路层协议可能不提供可靠的交付。

图 3.3　帧在数据链路层一条链路上移动

3.1.2　数据链路层基本问题

数据链路层可能提供的服务包括：组帧、差错控制、流量控制、可靠传输和介质访问控制。

1. 组帧

在网络层分组在链路上传输前，链路层协议用数据链路层的帧将其封装。一个帧由数据字段和首部字段组成，网络层的分组就插在数据字段中。一个帧可能包含尾部字段，我们把首部字段和尾部字段合并起来称为首部字段。接收端在收到物理层上交的比特流后，能根据首部字段的标记，从收到的比特流中识别帧的开始和结束，如图 3.4 所示。帧的结构由数据链路层协议规定。

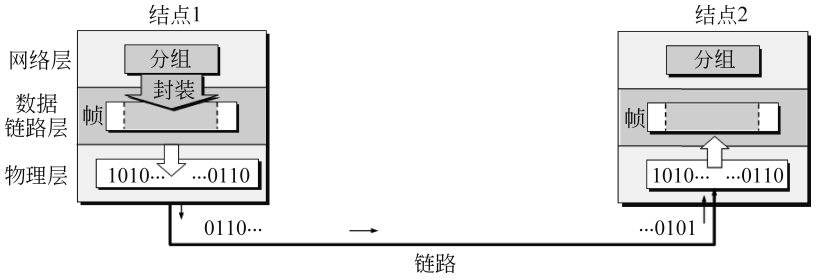


图 3.4　组帧

数据链路层的主要工作是添加一个帧头部和帧尾部，不同的数据链路层协议可能格式不同，但是基本的格式都是相似的，如图 3.5 所示。

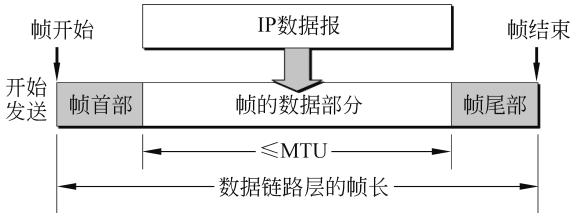


图 3.5　帧格式

这里的数据部分，一般有一个最大程度，我们称为最大传输单元（Maximum Transmission Unit，MTU），是指一种通信协议的某一层上面所能通过的最大数据包大小（以字节为单位），对于时下大多数使用以太网的局域网来说，最大传输单元的值是 1500 字节。这里要说的是，当数据是由可打印的 ASCII 码组成的文件时，可以使用特殊的帧定字符来标明一个帧的开始和结束。比如使用 SOH（Start Of Header）-0x01 和 EOT（End Of Transmission）-0x04 来表示，这样数据链路层就可以识别出帧的开始和结束。

如果我们提供任何数据输入，数据链路层都可以成功传递，那么称之为透明传输，即数据链路层的功能对于网络层和上层是透明的。比如文本字符数据输入，SOH 和 EOT 都可以很好地工作，因为二者没有交集。但是对于二进制数据输入来说，就有可能在数据中出现 0x01 和 0x04，导致帧意外地中断和丢弃。因此，我们需要一种机制来处理这种情况，最经典、最常用的就是字节填充或字符填充的方式。比如在 SOH 和 EOT 的前面分别插入一个转义字符 ESC-0x1B，在接收端的数据链路层在将数据送往网络层之前删除这个插入的转义字符，这就称为字节填充。

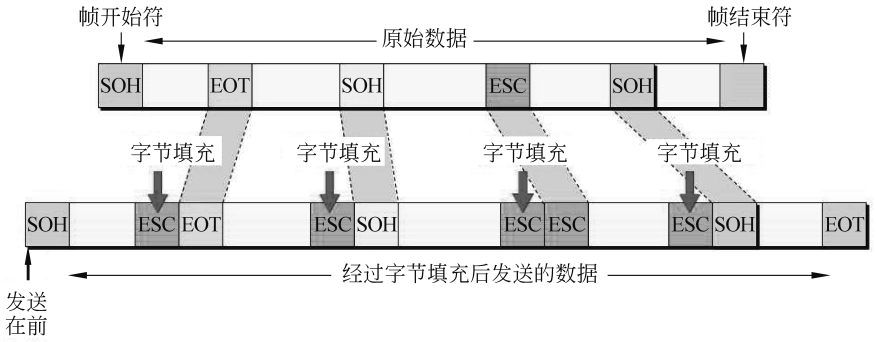


图 3.6　字节填充法实现透明传输

2. 差错控制

设计数据链路层的主要目的就是要将有差错的物理线路改进成无差错的数据链路，因此，差错控制功能是数据链路层中一个非常重要的基本功能，也是确保数据通信正常进行的基本前提。目前，数据链路层所采取的差错控制方法主要包括差错检测技术、差错纠正技术，以及数据帧重传技术等。

数据链路层的许多协议提供检测是否存在差错的机制。这是通过在帧中设置差错检测冗余位，让接收结点对收到的帧进行差错检测来完成的。链路层的差错检测通常很复杂，并且通过硬件来实现。

差错纠正不仅能检测是否帧中出现了差错，而且能够判断帧中的差错出现在哪里并纠正这些错误。一些协议如 ATM 只为分组的首部提供链路层差错纠正。

3. 流量控制

由于链路的每一结点具有有限的帧缓存，接收结点在某个时间段收到帧的速率比其处理的速度快，没有流量控制，接收方的缓存会溢出，帧会丢失。链路层协议提供流量控制机制，当接收方来不及处理发送方发送的数据时，及时控制发送方发送数据的速率，旨在使收发方协调一致。

4. 可靠传输

当数据链路层提供可靠传输服务时，它保证将网络层的分组无差错地通过数据链路层。OSI 的观点是必须把数据链路层做成是可靠传输的。前面讲过，传输层的协议 TCP 也提供可靠的传输服务。和传输层的可靠传输服务类似，链路层的可靠传输是通过确认和重传来获得的。现在通信线路的质量已经大大提高了，通信链路不好引起差错的概率已经大大降低。低差错率的链路，包括光纤、双绞线和同轴电缆，链路层的可靠传输被认为是不必要的开销。为了提高通信效率，许多有线的链路层协议不提供可靠的交付，Internet 广泛使用的数据链路层不提供可靠的服务。数据链路层可靠的传输服务常用于容易产生高差错率的链路，如无线链路。

5. 介质访问控制

介质访问控制协议定义了帧在链路上传输的规则。对于在链路的一端有一个发送方、另一端有一个接收方的点对点链路，介质访问控制协议比较简单，甚至不存在。对于

多个结点共享单个广播链路，就是被称为多址访问的问题，介质访问控制协议用来协调多个结点的帧传输。

数据链路层提供的许多服务和传输层提供的服务非常相似。例如数据链路层和传输层都能提供可靠交付。尽管这两层用于提供可靠交付的机制相似，这两个可靠交付服务却是不同的。传输层协议在端到端的基础上为两个进程之间提供可靠交付；而可靠的链路层协议在一条链路相连的两个结点之间提供可靠的交付服务。同样地，数据链路层和传输层都能提供流量控制和差错检测，传输层协议中的流量控制是在端到端的基础上提供的，而链路层协议是在结点对相邻结点基础上提供的。

6. 链路管理

数据链路层的“链路管理”功能包括数据链路的建立、链路的维持和释放三个主要方面。当网络中的两个结点要进行通信时，数据的发送方必须知道接收方是否已处在准备接收的状态。为此，通信双方必须先要交换一些必要的信息，以建立一条基本的数据链路。在传输数据时，要维持数据链路；而在通信完毕时，要释放数据链路。大多数广域网的通信子网的数据链路层均采用有确认面向连接的服务，源计算机和目标计算机在传输数据前需要先建立一个连接，而且该连接上发送的每一帧也都要被编号，以确保帧传输的内容与顺序的正确性，为此，数据链路层除了可保证每一帧都会被接收方收到之外，还可保证每一帧都只会被按正常顺序接收到一次。

7. MAC 寻址

MAC 寻址是数据链路层中的 MAC 子层的一项主要功能。在以太网中，介质访问控制(Media Access Control，MAC)地址被烧录到每个以太网网卡中作为通信结点数据链路层的唯一标识，而在多点连接的网络通信中，数据链路层必须要保证让每一帧都能准确地送到正确的接收方，且接收方也应当知道发送方到底是哪一个结点，为此，数据链路层必须要能够通过采用 MAC 地址来进行寻址。

3.2 差错控制技术

差错控制指的是在数据通信过程中能发现或纠正差错，把差错限制在尽可能小的允许范围内的技术和方法。

信号在物理信道中传输时，线路本身电器特性造成的随机噪声、信号幅度的衰减、频率和相位的畸变、电器信号在线路上产生反射造成的回音效应、相邻线路间的串扰以及各种外界因素(如大气中的闪电、开关的跳火、外界强电流磁场的变化、电源的波动等)都会造成信号的失真。在数据通信中，将会使接收端收到的二进制数位和发送端实际发送的二进制数位不一致，从而造成由“0”变成“1”或由“1”变成“0”的差错。

通信信道的噪声分为两类：热噪声和冲击噪声。其中，热噪声是信道固有的、持续存在的随机热噪声，它引起的差错是随机差错；冲击噪声是由外界特定的短暂原因所造成的噪声，它引起的差错是突发差错，引起突发差错的位长称为突发长度。在通信过程中产生的传输差错，是由热噪声的随机差错与冲击噪声的突发差错共同构成的。数据通信的差错程度通常是以“误码率”来定义的，它是指二进制比特在数据传输系统中被传错的概率，

它在数值上近似等于 $Pe=Ne/N$。其中，N 为传输的二进制比特总数，Ne 为被传错的比特数。

最常用的差错控制方法是差错控制编码。数据信息位（k 位）在向信道发送之前，先按照某种关系附加上一定的冗余位（n 位），构成一个码字后再发送，这个过程称为差错控制编码过程。编码效率就是 k 除以（$k+n$）的值。接收端收到该码字后，检查信息位和附加的冗余位之间的关系，以检查传输过程中是否有差错发生，这个过程称为检验过程。

差错控制编码可分为检错码和纠错码。检错码是能自动发现差错的编码；纠错码是不仅能发现差错而且能自动纠正差错的编码。

差错控制方法分两类，一类是自动请求重发 ARQ，另一类是前向纠错 FEC。在 ARQ 方式中，当接收端发现差错时，就设法通知发送端重发，直到收到正确的码字为止。ARQ 方式只使用检错码。在 FEC 方式中，接收端不但能发现差错，而且能确定二进制码元发生错误的位置，从而加以纠正。FEC 方式必须使用纠错码。

3.2.1 检错

目前在数据链路层广泛使用了循环冗余检验（Cyclic Redundancy Check，CRC）的检错编码。循环冗余检验编码也称为多项式编码，因为能把比特串看作是系数为 0 和 1 的一个多项式，对比特串的操作被解释为多项式算术。

在发送端要发送的 k 比特的数据 M，发送结点要把数据 M 发送给接收结点。发送方和接收方首先要协商一个 $n+1$ 比特生成码 P，称为生成多项式 $P(x)$。要求 $P(x)$ 的最高有效位的比特（最左边）是 1。

对于一个给定的数据 M，发送方要选择 n 位的附加比特 R 即冗余码（冗余码常称为帧检验序列 FCS），附加在 M 后面，使得产生的 $k+n$ 比特的数据一起发送到接收端。在所要发送的数据后面增加 n 位冗余码，虽然增大了数据传输的开销，但却可以进行差错检测，往往是很值得的。

这 n 位的冗余码 R 是如何得出的呢？是用模 2 运算（在加法中不进位，在减法中不借位。这意味着加法和减法是相同的，而且等价于操作数的按位异或（XOR）运算），相当于在 M 后面添加 n 个 0。得到（$k+n$）位的数除以发送方和接收方协商好的 $n+1$ 位除数 P，得出的商是 Q 余数是 R（n 位，比 P 少一位）。

【例 3-1】已知：信息码 M110011，信息多项式 $M(x)=x^5+x^4+x+1$。

生成码 P1101，生成多项式 $P(x)=x^3+x^2+1(n=3)$。

求：冗余码和码字。

【解】 （1）被除数是信息码 M 后添加 $n=3$ 个 0，即 110011000。

（2）除数是 P 即 1101。

（3）用模 2 运算（如图 3.7 所示）。由计算结果知冗余码是 001，码字就是 110011001。

在接收端把收到的数据以帧为单位进行 CRC 检验。用 CRC 进行差错检验的过程很简单：接收方用 P 去除接收到的 $k+n$ 位比特。如果余数为 0，则认为正确而被收下得到信息码；如果余数为非 0，则接收方认为发生错误，就丢弃该帧，请求对方重发。

【例 3-2】 已知：接收码字 1100111001，多项式 $T(x)=x^9+x^8+x^5+x^4+x^3+1$。

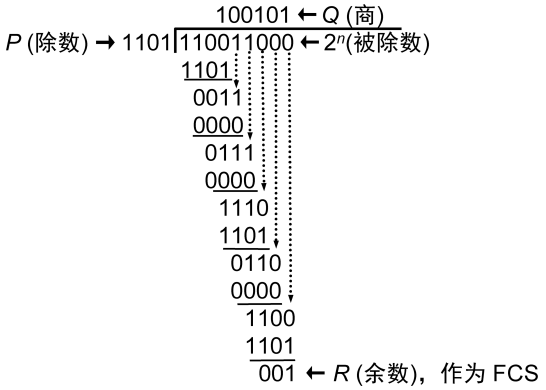


图 3.7　求冗余码的例子

生成码 P11001，生成多项式 $P(x)=x^4+x^3+1(n=4)$。

求：码字的正确性。若正确，则指出冗余码和信息码。

【解】 (1) 用码字除以生成码，余数为 0，如图 3.8 所示，所以码字正确。

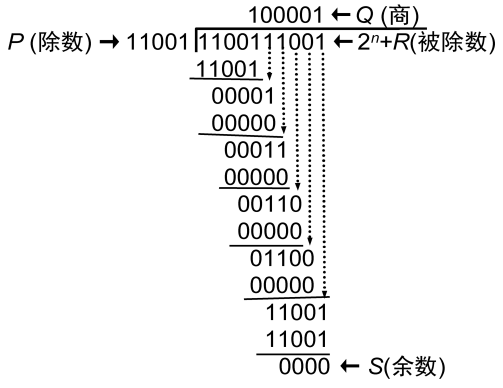


图 3.8　求码字正确性的例子

(2) 因 $n=4$，所以冗余码是：1001，信息码是：110011。

现在广泛使用的生成多项式 $P(x)$有如下几种：

CRC-16$=x^{16}+x^{15}+x^2+1$

CRC16-CCITT$=x^{16}+x^{12}+x^5+1$

CRC-32$=x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

CRC 码不能 100%地发现错误，余数为 0 时可能发生差错。一般生产多项式阶数越高，检错能力越强。凡是接收方数据链路层接受的帧，我们都能以非常接近于 1 的概率认为这些帧在传输过程中没有产生差错。通常都这样近似地认为：凡是接收方数据链路层接受的帧均无差错。

在数据链路层，发送端帧检验序列 FCS 的生成和接收端 CRC 检验都是用硬件完成的，处理很迅速，因此不会延误数据的传输。

CRC 码不可以自动纠错，要做到可靠传输，必须加上确认和重传机制。

采用检错码技术虽然能够检测出帧在传输过程中是否发生了错误，但由于在检测到错误发生之后，需要发送方重传整个帧，故检错码技术仅适用于光纤等具有高可靠性的信道上，而不适合于错误发生很频繁的无线信道上，这是因为在错误发生很频繁的信道上，即便重传也还是很可能出错的，因此难以有效保证帧的正确传送。在无线信道上，最好的办法是使用纠错码(Error Correcting Code)技术，通过在每一个发送的数据帧中包含足够的冗余信息(校验位)，让接收方在收到该数据帧后不但可以检测出其中是否发生了错误，而且一旦发现出错，还可以还原出原始的帧内容，不需要依靠重传来解决问题。

3.2.2　纠错

采用纠错码进行差错控制时，接收端不仅能发现差错，而且知道出错码元的位置，从而自动进行纠正。这种方式称为前向纠错(Forward Error Correction，FEC)。海明码就是一种纠错码。

发送方进行海明码编码，所需步骤如下：

(1) 确定最小的校验位数 k。

(2) 原有信息和 k 个校验位一起编成长为 $m+k$ 位的新码字——海明码。选择 k 校验位(0 或 1) 以满足必要的奇偶条件。

接收方对收到的码字进行译码，所需步骤如下：

(1) 接收端对所接收的信息作所需的 k 个奇偶检查。

(2) 如果所有的奇偶检查结果均为正确的，则认为信息无错误。如果发现有一个或多个错了，则错误的位由这些检查的结果来唯一地确定。

1. 校验位的位数

推求海明码时的一项基本考虑是确定所需最少的校验位数 k。考虑长度为 m 位的信息，若附加了 k 个校验位，则所发送的总长度为 $m+k$。在接收器中要进行 k 个奇偶检查，每个检查结果或是真或是伪。这个奇偶检查的结果可以表示成一个 k 位的二进字，它可以确定最多 2^k 种不同状态。这些状态中必有一个其所有奇偶测试都是真的，它便是判定信息正确的条件。于是剩下的(2^k-1) 种状态，可以用来判定误码的位置。于是导出下一关系：

$$2^k-1 \geqslant m+k \tag{3-1}$$

2. 码字格式

从理论上讲，校验位可放在任何位置，但习惯上校验位被安排在 1、2、4、8、…的位置上。

表 3.1 列出了 $m=4$，$k=3$ 时，信息位和校验位的分布情况。

表 3.1　海明码中校验位和信息位的定位

码字位置	B_1	B_2	B_3	B_4	B_5	B_6	B_7
校验位	x	x		x			
信息位			x		x	x	x
复合码字	P_1	P_2	D_1	P_3	D_2	D_3	D_4

3. 各校验位的确定

k 个校验位是通过对 $m+k$ 位复合码字进行奇偶校验而确定的。

其中：P_1 位负责校验海明码的第 1、3、5、7、…（P_1、D_1、D_2、D_4、…）位（包括 P_1 自己，检验 1 位，跳过 1 位）。

P_2 负责校验海明码的第 2、3、6、7、…（P_2、D_1、D_3、D_4、…）位（包括 P_2 自己，检验 2 位，跳过 2 位）。

P_3 负责校验海明码的第 4、5、6、7、…（P_3、D_2、D_3、D_4、…）位（包括 P_3 自己，检验 4 位，跳过 4 位）。

对 $m=4$，$k=3$，偶校验的例子，只要进行三次偶性测试。这些测试（以 A、B、C 表示）在表 3.2 所示各位的位置上进行。

表 3.2　奇偶校验位置

奇偶条件	码字位置						
	B_1	B_2	B_3	B_4	B_5	B_6	B_7
A	x	x		x			
B			x		x	x	x
C	P_1	P_2	D_1	P_3	D_2	D_3	D_4

因此可得到三个校验方程及确定校验位的三个公式：

$$A=B_1\oplus B_3\oplus B_5\oplus B_7=0 \text{ 得 } P_1=D_1\oplus D_2\oplus D_4 \tag{3-2}$$

$$B=B_2\oplus B_3\oplus B_6\oplus B_7=0 \text{ 得 } P_2=D_1\oplus D_3\oplus D_4 \tag{3-3}$$

$$C=B_4\oplus B_5\oplus B_6\oplus B_7=0 \text{ 得 } P_3=D_2\oplus D_3\oplus D_4 \tag{3-4}$$

例如四位信息码为 1001，利用这三个公式可求得三个校验位 P_1、P_2、P_3 值和海明码。

$$P_1\ P_2\ 1\ P_3\ 0\ 0\ 1$$

三个校验位：$P_1=0$；$P_2=0$；$P_3=1$。

海明码：0011001。

上面是发送方的处理。

4. 接收方译码

在接收方，也可根据这三个校验方程对接收到的信息进行同样的奇偶测试：

$$A=B_1\oplus B_3\oplus B_5\oplus B_7=0 \tag{3-5}$$

$$B=B_2\oplus B_3\oplus B_6\oplus B_7=0 \tag{3-6}$$

$$C=B_4\oplus B_5\oplus B_6\oplus B_7=0 \tag{3-7}$$

若三个校验方程都成立即方程式右边都等于 0，则说明没有错。若不成立即方程式右边不等于 0，说明有错。从三个方程式右边的值，可以判断哪一位出错。例如，如果第 3 位数字反了，则 $C=0$（此方程没有 B_3），$A=B=1$（这两个方程有 B_3）。可构成二进制数 CBA，以 A 为最低有效位，则错误位置就可简单地用二进制数 $CBA=011$ 指出。

同样，若三个方程式右边的值为 001，说明第一位出错。若三个方程式右边的值为

100，说明第四位出错。

例如，接收码字为0110111经测试$A=1$；$B=0$；$C=1$。说明第五位有错，则只须将第五位变反，就可还原成正确的数码0110011。

海明码能够检测出两位同时出错，或者能检测出一位出错并能自动给出错位的正确值。

3.3　流量控制技术

在本节，我们介绍一般意义上的可靠数据传输问题。可靠数据传输可以在数据链路层实现，也可在传输层和应用层实现。一般性的问题对网络来说更为重要。

OSI的观点是必须把数据链路层做成是可靠传输的。因此在CRC检错的基础上，增加了流量控制、确认和重传机制。采取适当的措施限制发送速率，避免由于接收方来不及接收而造成数据丢失。接收方收到正确的帧就要向发送方发送确认。发送端在一定期限内没有收到对方的确认，就认为出现了差错，因而就进行重传，直到收到对方的确认为止。但现在通信线路的质量已经大大提高了，由于通信链路不好引起差错的概率已经大大降低。为了提高通信效率，Internet广泛使用的数据链路层不提供可靠的服务。数据链路层可靠的传输服务常用于容易产生高差错率的链路，如无线链路。

3.3.1　流量控制与滑动窗口

当发送数据的速率高于接收数据的速率时，必须采取适当的措施限制发送速率，否则会由于来不及接收而造成数据丢失。流量控制就是让发送方的发送速率不要太快，要让接收方来得及接收。在本节，我们介绍如何用滑动窗口机制来达到流量控制的目的。

尽管流量控制可以用多种方式实现，但通常的方式是使用两个缓冲区：一个位于发送方，另一个位于接收方。缓冲区是一组内存单元，它可以在发送端和接收端存储分组。这样的缓冲区又称为窗口，在发送端和接收端分别设置发送窗口和接收窗口。

通常，数据链路层采用基于确认的流量控制机制来进行流量控制。在该方法中，一般是通过定义一些良好的规则，这些规则规定了发送方什么时候可以发送下一帧。通常，由接收方给发送方回送信息，告诉发送方被允许发送多少数据，而在没有得到接收方许可之前，禁止发送方向接收方发送数据帧。基于确认的流量控制机制包括停止—等待技术、基于回退N帧(Go Back N)技术的机制与基于选择性重传(Selective Repeat)技术的机制，这些机制统称为滑动窗口机制(Sliding Window Mechanism)。此外，除了利用滑动窗口机制来进行流量控制之外，为了进一步提高数据传输效率，数据链路层通常还采用捎带确认、发送窗口与接收窗口等设计思想与技术。

(1) 捎带确认：在收发双方在进行通信时，为了提高信道的利用率，将利用捎带确认的方法进行数据帧的确认。其原理如下。当一方收到一帧A，如果其网络层有一个新的分组到来要发送给对方，则通过在其数据帧的头部设置的ack域中捎带对A的确认，然后再将该数据帧发送给对方(而不单独发送确认帧给对方，从而节约了网络带宽)。否则，若在一定的时间周期内其网络层都没有新的分组到来，则发送一个单独的确认帧给对方。

(2) 发送窗口：是指发送方维持的一组序列号，分别对应于发送方允许它发送的帧，或它已发送但仍未被确认的帧。由于当前发送窗口内的帧最终可能在传输过程中丢失或被破坏，因此，发送方需要用缓存区保存好这些帧以备重传。若发送窗口的大小为 n，则发送方需要 n 个缓存区来存放未被确认的帧。若发送窗口达到最大尺寸，则发送方的链路层需要强制关闭其网络层，直到有一个缓存区空闲出来为止。

(3) 接收窗口：是指接收方维持的一组序列号，分别对应于一组接收方允许接收的帧。任何落在接收窗口外面的帧都将被接收方无条件丢弃。当一个新到的帧的序列号等于接收窗口的下界时，接收方会把该新到帧以及接收窗口(即接收方缓存区)中原来保存的其后续各帧依次传递给网络层，并生成一个确认帧给发送方，然后将接收窗口前移。

发送窗口和接收窗口统称为滑动窗口。

1. 发送窗口的规则

发送窗口用来对发送端进行流量控制。发送窗口是发送方用来保存允许发送和已发送但尚未经确认的数据分组。发送窗口的大小 W_T 代表在还没有收到对方确认信息的情况下发送端最多可以发送多少个分组。图 3.9 表示了发送窗口的规则。

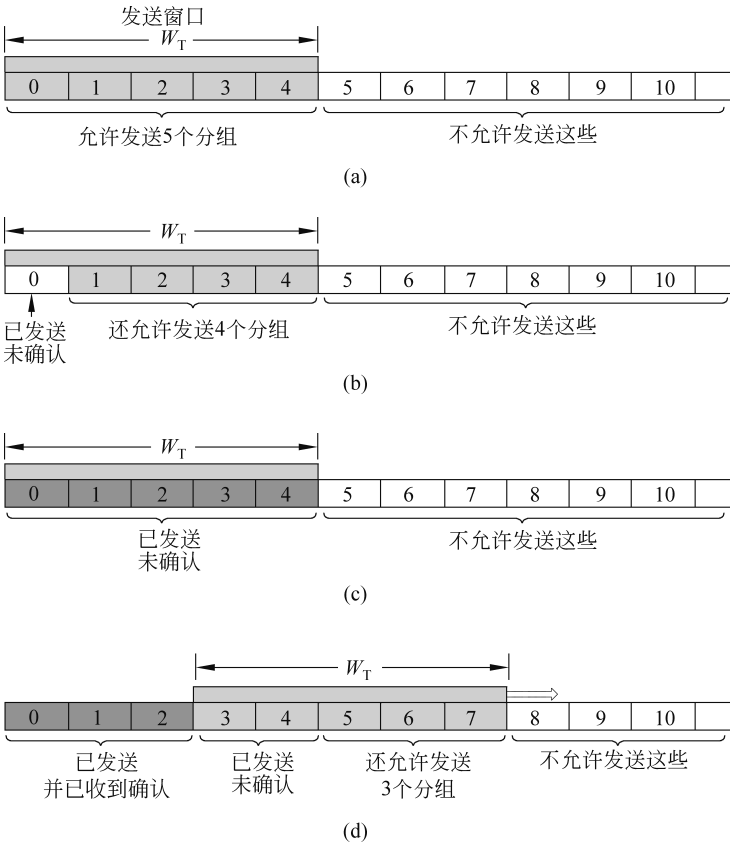


图 3.9　发送窗口的规则

(1) 发送窗口内的分组是允许发送的分组(如图 3.9(a)，发送窗口大小为 5)。

(2) 每发送完一个分组，允许发送的分组数减 1，但发送窗口的位置、大小不变(如图 3.9(b))。

(3) 若所允许发送的分组都发送完了，但还没有收到任何确认，发送方就不能再发送，进入等待状态(如图 3.9(c))。

(4) 发送方收到对方对一个分组的确认信息后，将发送窗口向前滑动一个分组的位置(如图 3.9(d)，依次收到 3 个确认分组)。

(5) 发送方设置一个超时计时器，当超时计时器满且未收到应答，则重发分组。

2. 接收窗口的规则

接收窗口是为了控制可以接收的数据分组的范围。接收窗口是接收方用来保存已正确接收但尚未交给上层的分组。接收窗口的规则如下：

(1) 只有当收到的分组序号落入接收窗口内才允许收下，否则丢弃它。

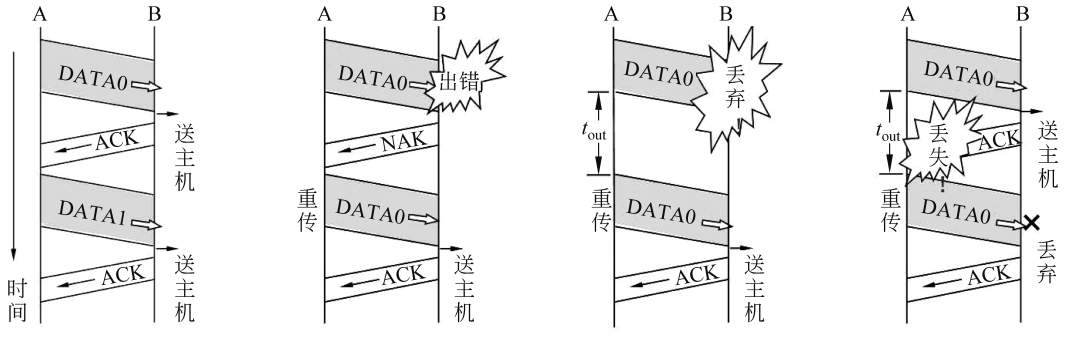
(2) 当接收方接收一个序号正确的分组，接收窗口向前滑动，并向发送端发送对该分组的确认。

只有在接收窗口向前滑动时(与此同时也发送了确认)，发送窗口才有可能向前滑动。收发两端的窗口按照以上规律不断地向前滑动，因此这种协议又称为滑动窗口协议。使用滑动窗口机制，由接收方控制发送方的数据流，实现了流量控制。同时采用有效的确认重传机制，向高层提供可靠传输的服务。

下面的停止—等待协议、回退 N 帧协议和选择重传协议三个协议都实现了流量控制，是保证数据可靠传输常采用的协议。

3.3.2　停止—等待协议

停止—等待协议(Stop-and-Wait-Protocol)，发送方和接收方都使用大小为 1 的滑动窗口。停止—等待协议(也称为停－等协议)的规则是：发送方每发送一个分组后就要停下来等待接收方的确认返回，仅当接收方正确接收，并返回确认分组 ACK，发送方接收到确认分组后，才可以发送下一分组，如图 3.10(a)所示。



(a) 正常情况　(b) 分组出错　(c) 分组丢失　(d) 分组丢失

图 3.10　停止—等待协议

接收方收到一个数据分组，通过差错检测发现数据是错误的，接收方向发送方发送一

个否认分组 NAK，发送方收到否认分组后重传出错分组，如图 3.10(b)。

由于链路干扰或其他原因，在发送方发送的数据分组或接收方发送的确认分组 ACK 丢失的情况下，发送方没有收到确认分组，到了超时计时器所设置的重传时间，发送方会重传该分组，如图 3.10(c)和图 3.10(d)。

3.3.3 回退 N 帧协议

为了提高传输效率，当发送端等待确认时，可以传输多个分组。换言之，当发送端等待确认时，我们需要让不止一个分组处于未完成状态，以此确保信道忙碌。回退 N 帧协议(Go-back-N，GBN)也称为连续自动重传请求(Automatic Repeat Request，ARQ)协议，可以看成是发送窗口大于 1，接收窗口等于 1 的滑动窗口协议。该协议的规则如下：

(1) 在发送完一个数据帧后，不是停下来等待确认帧，而是可以连续再发送若干个数据帧。由于减少了等待时间，整个通信的吞吐量就提高了。

(2) 如果这时收到了接收端发来的确认帧，那么还可以接着发送数据帧。

(3) 如果发送方发送了前五个帧，而中间的第三个帧丢失了。这时接收方只能对前两个帧发出确认。发送方无法知道后面三个帧的下落，而只好把后面的三个帧都再重传一次。这就称为 Go-back-N(回退 N)，表示需要再退回来重传已发送过的 N 个帧，如图 3.11 所示。

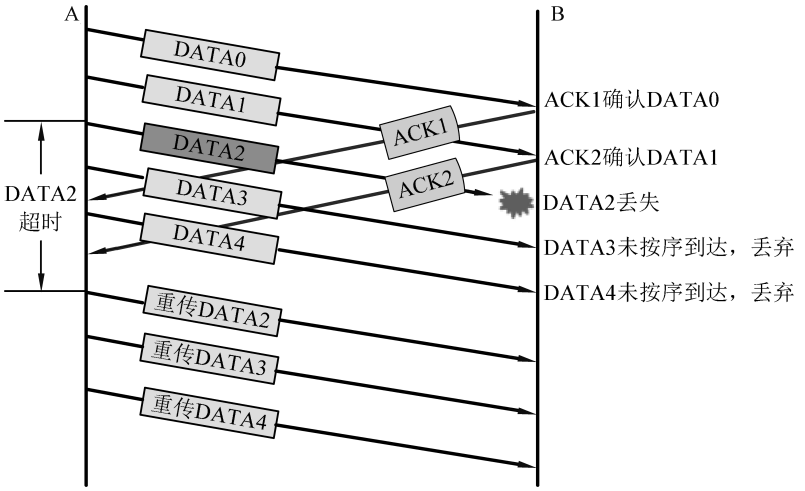


图 3.11 后退 N 帧协议

接收端只按序号顺序接收数据帧。虽然在有差错的 2 号帧之后接着又收到了正确的三个数据帧，但接收端都必须将这些帧丢弃，因为在这些帧前面有一个 2 号帧还没有收到。虽然丢弃了这些不按序的无差错帧，但应重复发送已发送过的最后一个确认帧(防止确认帧丢失)。在图 3.11 中 ACK1 表示确认 0 号分组 DATA0，并期望下次收到 1 号分组；ACK2 表示确认 1 号分组 DATA1，并期望下次收到 2 号分组，依此类推。

结点 A 在每发送完一个数据帧时都要设置该分组的超时计时器。如果在所设置的超时时间内收到确认帧，就立即将超时计时器清零。但若在所设置的超时时间到了而未

收到确认分组，就要重传相应的数据帧（仍需重新设置超时计时器）。

在等不到2号分组的确认而重传2号数据帧时，虽然结点A已经发完了4号分组，但仍必须向回走，将2号帧及其以后的各帧全部进行重传。这就是回退N帧协议，意思是当出现差错必须重传时，要向回走N个帧，然后再开始重传。

在回退N帧协议中，接收窗口的大小$W_R=1$，如图3.12所示。只有当收到的帧的序号与接收窗口一致时才能接收该帧（如图3.11中的DATA0、DATA1），否则就丢弃它（如图3.11中的DATA3、DATA4、DATA5）。

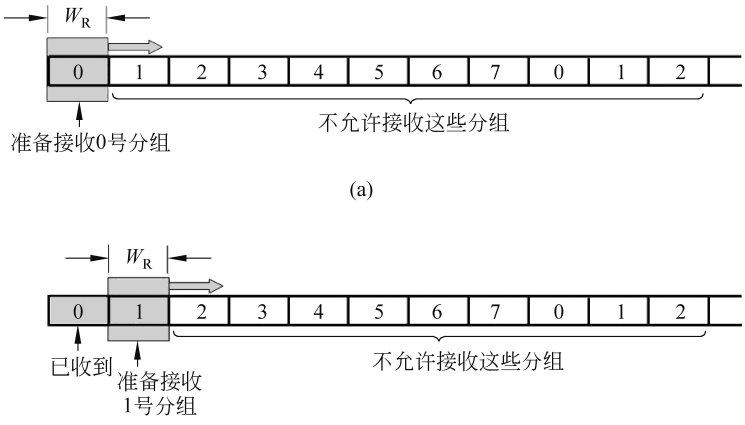


图3.12　接收窗口为1的情况

3.3.4　选择重传协议

回退N帧协议简化了接收方的进程。接收方只记录一个变量，没有必要缓冲时序分组；它们被简单地丢弃。然而，如果下层网络层丢失很多分组，那么这个协议就是低效的。每当一个分组丢失或被破坏，发送方要重新发送所有未完成分组，即使有些时序分组已经被完全完整地接收了。如果网络层由于网络拥塞丢失了很多分组，那么重发所有这些未完成分组将会使得拥塞更严重，最终更多的分组丢失。

选择重传协议（Selective-Repeat Protocol）已经被设计出来，正如其名，支持选择性重发分组，即那些确实丢失的分组。在选择重传协议中，发送窗口大于1，接收窗口大于1。选择重传协议规则是加大接收窗口，先收下发送序号不连续但仍处在接收窗口中的那些数据分组。等到所缺序号的数据分组收到后再一并送交主机。

选择重传协议可避免重复传送那些本来已经正确到达接收端的数据分组，但我们付出的代价是在接收端要设置具有相当容量的缓存空间。

3.4　点到点信道的数据链路层

广域网一般最多包括OSI参考模型的下三层，网络层提供的服务有虚电路和数据报服务，数据链路层协议有PPP、HDLC和帧中继等，PPP协议占有绝对优势。

现在我们讨论点对点链路的数据链路层协议，即点对点协议（Point-to-Point Protocol，PPP）。这种链路提供全双工操作，并按照顺序传递数据包。设计目的主要是用来通过拨号或专线方式建立点对点连接发送数据，使其成为各种主机、网桥和路由器之间简单连接的一种共通的解决方案。使用拨号电话线接入 Internet 采用 PPP 协议，如图 3.13 所示。

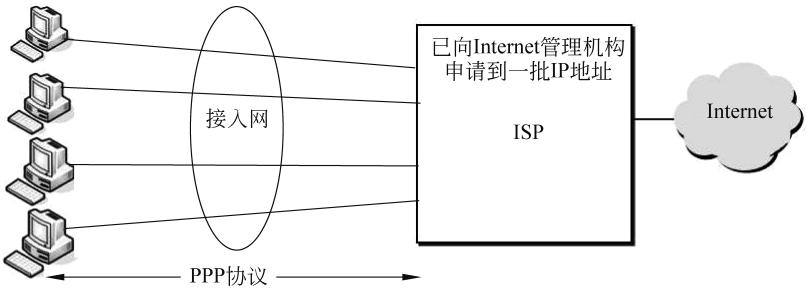


图 3.13　用户采用 PPP 协议接入 Internet

PPP 协议是 Internet 工程任务组（Internet Engineering Task Force，IETF）推出的一种适用于点到点连接的数据链路控制协议，是一种正式的 Internet 数据链路层协议标准，该协议在 RFC 1661、RFC 1662 和 RFC 1663 中进行了描述。PPP 协议有三个组成部分：组帧即一个将 IP 数据报封装到串行链路的方法；一个用来建立、维护和拆除数据链路连接的链路控制协议（Link Control Protocol，LCP）；一套网络控制协议（Network Control Protocol，NCP）族，PPP 允许多个网络协议共用一个链路，网络控制协议（NCP）负责连接 PPP（第二层）和网络协议（第三层）。对于所使用的每个网络层协议，PPP 都分别使用独立的 NCP 来连接。例如，IP 使用 IP 控制协议（IPCP），IPX 使用 Novell IPX 控制协议（IPXCP）。

3.4.1　功能

（1）PPP 具有动态分配 IP 地址的能力，允许在连接时刻协商 IP 地址。

（2）PPP 支持多种网络协议，比如 TCP/IP、NetBEUI、NWLINK 等。

（3）PPP 协议只检错不纠错。

（4）PPP 具有身份验证功能。

（5）PPP 可以用于多种类型的物理介质上，包括串口线、电话线、移动电话和光纤（例如 SDH），PPP 也用于 Internet 接入。

3.4.2　PPP 帧填充方式

1. 帧格式各字段的含义

PPP 协议的帧格式如图 3.14 所示。

标志字段 F 为 0x7E（符号“0x”表示后面的字符是用十六进制表示。十六进制的 7E 的二进制表示是 01111110），每个 PPP 帧都是以 01111110 的 1 字节标志字段来作为开始和结束。

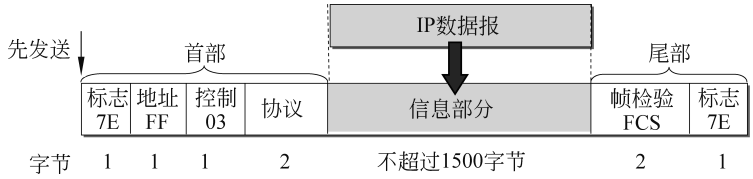


图 3.14　PPP 的帧格式

地址字段 A 只置为 0xFF，地址字段实际上并不起作用。

控制字段 C 通常置为 0x03，表明这是一个无序号帧，即意味着：在默认方式下，PPP 并没有采用序号和确认来实现可靠的传输。

PPP 是面向字节的，所有的 PPP 帧的长度都是整数字节。

PPP 有一个 2 字节的协议字段：当协议字段为 0x0021 时，PPP 帧的信息字段就是 IP 数据报。若为 0xC021，则信息字段是 PPP 链路控制数据。若为 0x8021，则表示这是网络控制数据。

信息字段的长度是可变的，不超过 1500 字节。

2. 比特填充

当 PPP 用在同步传输链路时，协议规定通常采用硬件来完成比特填充，采用零比特填充实现透明传输。

采用零比特填充法使信息字段不会出现 6 个连续 1。在发送端，当一串比特流数据中有 5 个连续 1 时，就立即填入一个 0。在接收帧时，每当发现 5 个连续 1 时，就将其后的一个 0 删除，以还原成原来的比特流。

例如，比特流“011111011111110…”的输出位流模式为“01111100111110110…”，当接收方收到“01111100111110110…”时，则自动删除连续 5 个输入位“1”后的位“0”，即将输入流还原成“011111011111110…”。

3. 字符填充

当 PPP 用在异步传输时，就使用一种特殊的字符填充法。字符填充法是将信息字段中出现的每一个 0x7E 字节转变成为 2 字节序列(0x7D;0x5E)。若信息字段中出现一个 0x7D 的字节；则将其转变成为 2 字节序列(0x7D;0x5D)。若信息字段中出现 ASCII 码的控制字符(即数值小于 0x20 的字符)，则在该字符前面要加入一个 0x7D 字节，同时将该字符的编码加以改变。

PPP 协议之所以不使用序号和确认机制是出于以下的考虑：在数据链路层出现差错的概率不大时，使用比较简单的 PPP 协议较为合理；在 Internet 环境下，PPP 的信息字段放入的数据是 IP 数据报。数据链路层的可靠传输并不能够保证网络层的传输也是可靠的；帧检验序列 FCS 字段可保证无差错接受。

3.4.3　身份认证模式

有两种身份认证模式：一种是 PAP，一种是 CHAP。相对来说 PAP 的认证方式安全性没有 CHAP 高。PAP 在传输 password 是明文的，而 CHAP 在传输过程中不传输密

码，取代密码的是 hash(哈希值)。PAP 认证是通过两次握手实现的，而 CHAP 则是通过三次握手实现的。PAP 认证是被叫提出连接请求，主叫响应。而 CHAP 则是主叫发出请求，被叫回复一个数据包，这个包里面有主叫发送的随机的哈希值，主叫在数据库中确认无误后发送一个连接成功的数据包连接。

3.4.4 PPP 的工作过程

PPP 协议的工作过程如下。

(1) 当用户拨号接入 ISP 后，就建立了一条从用户 PC 到 ISP 的物理连接。

(2) 这时用户 PC 向 ISP 发送一系列的 LCP 分组(封装成多个 PPP 帧)，以便建立 LCP 连接。这些分组及其响应选择了将要使用的一些 PPP 参数。

(3) 协商结束后就进入鉴别状态。若通信的双方鉴别身份成功，则进入网络层协议状态。

(4) 接着还要进行网络层配置，NCP 给新接入的用户 PC 分配一个临时的 IP 地址。这样，用户 PC 就成为 Internet 上的一个有 IP 地址的主机了。

(5) 当用户通信完毕时，NCP 释放网络层连接，收回原来分配出去的 IP 地址。接着，LCP 释放数据链路层连接。最后释放的是物理层的连接。

注：PPP 链路的起始和终止状态永远是“链路静止”(Link Dead)状态，这时在 PC 和 ISP 的路由器之间并不存在物理层的连接。

图 3.15 的右边方框是对 PPP 协议的几个状态的说明。从设备之间无链路开始，到先建立物理链路，再建立链路控制协议 LCP 链路。经过鉴别后再建立网络控制协议 NCP 链路，然后才能交换数据。由此可见，PPP 协议已不是纯粹的数据链路层的协议，它还包含了物理层和网络层的内容。

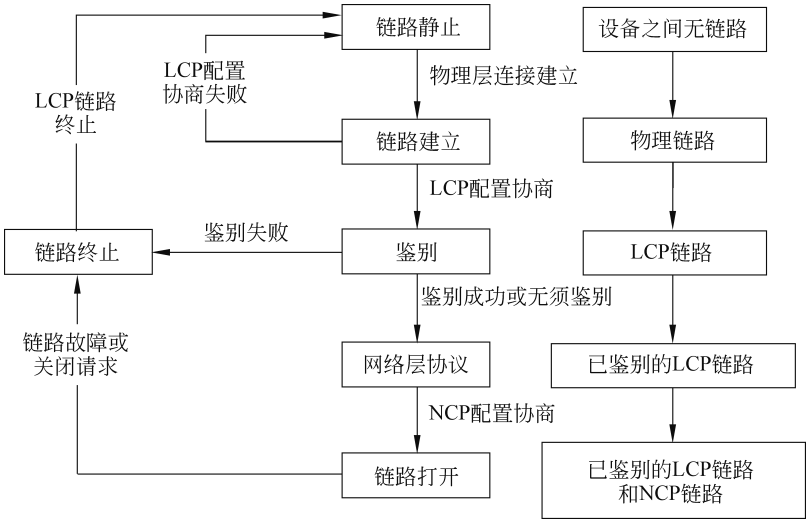


图 3.15 PPP 协议的工作流程和工作状态

3.5　广播信道的数据链路层

广播信道使用一对多的广播通信方式，同一信道上连接的主机很多。以太网起源于20世纪70年代，是目前使用最为广泛的局域网。

3.5.1　局域网概述

目前，全球范围内局域网(Local Area Network，LAN)的数量远超广域网。局域网已经从低速向高速，从共享式向交换式，从半双工式向全双工式发展与进步。通常把局域网定义为：在较小的地理范围内，局域网主要利用通信线路将办公室、企业、校园、小区等较小区域内的计算机、网络通信设备等连接在一起，配以接口和高层软件，进行高速数据传输和软硬件资源共享的系统。

1. 局域网的特点

局域网的特点主要有以下几点：

(1) 通常为一个单位拥有，地理范围有限，站点数量有限。

(2) 所有的站点共享较高的总带宽。

(3) 较高的时延和较低的误码率。

(4) 支持几种传输介质，包括双绞线、同轴电缆、光纤和无线介质等。

(5) 拓扑结构简单，主要有总线型、环型、星型结构等。

2. 局域网的体系结构

由于局域网大多采用共享信道，当通信局限于一个局域网内部时，任意两个结点之间都有唯一的链路，即网络层的功能可由链路层来完成，所以局域网中不单独设立网络层。IEEE 802提出的局域网参考模型(LAN/RM)如图3.16所示。

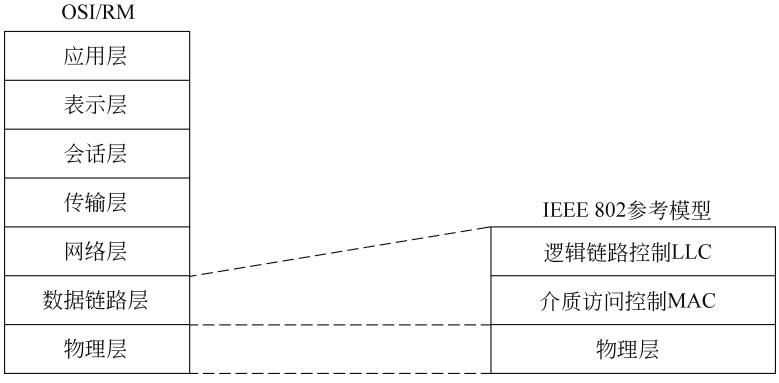


图3.16　IEEE 802参考模型与OSI参考模型的对应关系

与OSI参考模型相比，局域网的参考模型就只相当于OSI的最低两层。为了使数据链路层能更好地适应多种局域网标准，IEEE 802委员会就将局域网的数据链路层拆成逻辑链路控制LLC(Logical Link Control)子层和媒体接入控制MAC(Medium Access

Control)子层两个子层,与接入到传输媒体有关的内容都放在 MAC 子层,而 LLC 子层则与传输媒体无关,不管采用何种协议的局域网对 LLC 子层来说都是透明的。由于 TCP/IP 体系经常使用的局域网是 DIX Ethernet V2 而不是 IEEE 802.3 标准中的几种局域网,因此现在 IEEE 802 委员会制定的逻辑链路控制子层 LLC(即 IEEE 802.2 标准)的作用已经不大了。很多厂商生产的适配器上就仅装有 MAC 协议而没有 LLC 协议。

3.5.2 以太网

以太网(Ethernet)指的是由 Xerox 公司创建并由 Xerox、Intel 和 DEC 公司联合开发的基带局域网规范。1982 年 12 月,IEEE 公布了与以太网规范兼容的 IEEE 802.3 标准,它们的出现标志着以太网技术标准的起步,为符合国际标准、具有高度互通性的以太网产品的面世奠定了基础。

通常我们所说的以太网主要是指以下三种不同的局域网技术。

(1) 10 Mb/s 以太网,又称为标准以太网、传统以太网。采用同轴电缆作为网络媒体,传输速率达到 10 Mb/s。

(2) 100 Mb/s 以太网,又称为快速以太网。采用双绞线作为网络媒体,传输速率达到 100 Mb/s。

(3) 1000 Mb/s 以太网,又称为千兆以太网。采用光缆或双绞线作为网络媒体,传输速率达到 1000 Mb/s。

以太网跨越数据链路层和物理层,有许多不同的以太网标准,它们有不同的速率(2Mb/s、10Mb/s、100Mb/s、1Gb/s、10Gb/s)、不同的物理层媒体,但它们有共同的 MAC 协议和帧格式。

由于以太网结构简单、组网容易、建网成本低、扩充方便,一出现就受到业界的普遍欢迎而迅速发展起来,在很大程度上逐步取代了其他局域网标准。如当时比较流行的令牌环、FDDI 和 ARCNET 都逐渐被以太网淘汰。目前以太网成为局域网技术的主流技术。

1. 以太网工作原理

以太网是一种以总线方式连接、广播式传输的网络,所有站点通过共享总线实现数据传输,一个站发出的数据帧,所有的站都能收到,这种工作方式带来了冲突问题,需要采用相应的介质访问控制方式解决。以太网采用带有冲突检测的载波侦听多路访问(CSMA/CD)协议实现介质访问控制,在 IEEE 802 标准中,以太网标准为 IEEE 802.3 标准。以太网结构示意图如图 3.17 所示。

10Base-5 以太网与 10Base-2 以太网缺点很多,因此在 1990 年,IEEE 发布了新的以太网标准,这就是 10Base-T 以太网。10Base-T 以太网有两项革命性的改进,一是用双绞线代替同轴电缆,二是用星型拓扑结构代替总线型拓扑结构,这在以太网的发展史上有里程碑性的意义。经此改进,以太网安装简单、设备价格低廉、故障易于定位,为以太网战胜其他局域网奠定了牢固的基础。“10”与“Base”的含义与“10Base-5”以太网相同,“T”表示传输介质使用双绞线。

10Base-T 以太网的传输媒介质使用 3 类非屏蔽双绞线,更好的双绞线当然也可以。与同轴电缆与 T 型头相比,双绞线与 RJ-45 头的价格低廉,连接简单;同轴电缆中只有一

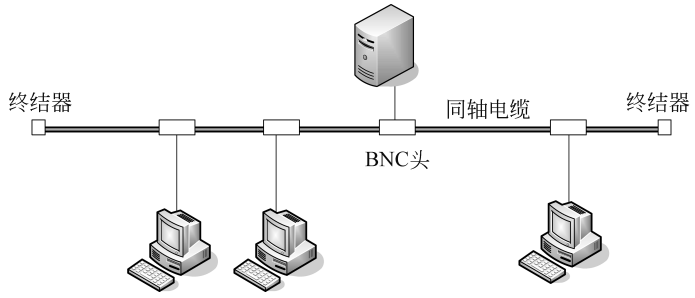


图 3.17　以太网结构示意图

个电回路，不能进行全双工通信，而双绞线里面有 8 根电线，两根构成发送电回路，另两根构成接收电回路，可以全双工通信。10Base-T 以太网中每根双绞线最长为 100m，再长时信号衰减严重，可能无法正确接收，如图 3.18 所示。

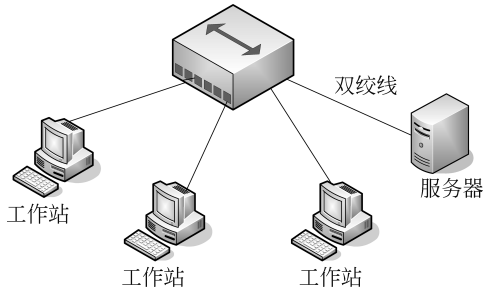


图 3.18　10Base-T

以太网通常使用专门的网络接口卡或通过系统主电路板上的电路实现。以太网使用收发器与网络媒体进行连接。收发器可以完成多种物理层功能，其中包括对网络碰撞进行检测。收发器可以作为独立的设备通过电缆与终端站连接，也可以直接被集成到终端站的网卡中。

以太网采用广播机制，所有与网络连接的工作站都可以看到网络上传递的数据。通过查看包含在帧中的目标地址，确定是否进行接收或放弃。如果证明数据确实是发给自己的，工作站将会接收数据并传递给高层协议进行处理。

作为一种基于竞争机制的网络环境，以太网允许任何一台网络设备在网络空闲时发送信息。因为没有任何集中式的管理措施，所以很有可能出现多台工作站同时检测到网络处于空闲状态，进而同时向网络发送数据的情况。这时，发出的信息会相互碰撞而导致损坏。工作站必须等待一段时间之后，重新发送数据。补偿算法用来决定发生碰撞后，工作站应当在何时重新发送数据帧。

2. 以太网标准

1982 年 2 月，IEEE 推出了 IEEE 802.3 规范，这是最早的 10Mb/s 以太网的标准。

1995 年 3 月，IEEE 通过了 IEEE 802.3u 规范，这是一个关于以 100Mb/s 的速率运行的快速以太网的规范。