第 3 章　数据链路层

数据链路层 DLL（Data Link Layer）以物理层为基础，向网络层提供可靠的服务。它在物理层之上通过数据链路层协议，加上必要的规程，控制节点间数据传输过程，在一条不太可靠的通信链路上，实现有结构的数据块（帧）的可靠传输。数据链路层对等实体之间的数据传输通道称为数据链路（Data Link），数据链路和链路是有区别的，链路是一条物理线路，而数据链路是一个逻辑概念，它包括物理线路和必要的控制规程。

3.1　数据链路层功能及成帧

3.1.1　数据链路层功能

数据链路层实现实体间数据的可靠传送。数据链路层的作用是利用物理层提供的位串传输功能，将物理层传输原始比特流时可能出错的物理连接，改造成为逻辑上无差错的数据链路，在相邻节点间实现透明的高可靠性传输，同时为网络层提供有效的服务。

透明是指该层传输数据时，对数据的内容、格式及编码没有限制，也没有必要解释信息结构的意义，不论传输的数据是什么比特组合，都能原样传输到目的节点，其处理过程上层是不可见的（对上层提供统一的界面），下面介绍的帧同步就是为了解决数据传输过程中代码透明性问题。

数据链路层主要功能是：成帧、差错控制、流量控制和链路管理等。成帧是将数据组合成数据块（数据链路层中将这种数据块称为帧，帧是数据链路层的传送单位）；差错控制是指控制帧在物理信道上的传输，包括如何处理传输差错；流量控制是指调节发送速率使之与接收方相匹配；链路管理是在两个网络实体之间提供数据链路通路的建立、维持和释放管理。

注意：物理层的传输单位是二进制位（bit），而数据链路层的传输单位是帧（Frame），是在 LAN 内节点间的传输。如果传输时经过中间节点，那就是广域网，存在路由选择问题，需要用到网络层。

物理层和数据链路层解决了局域网中的大部分问题，所以，组建局域网只用到通信子网内的低两层。

3.1.2　成帧和帧同步

1．成帧

数据链路层将物理层传送过来的比特流按照一定的格式分割成若干个帧，成帧的目的在于：

① 一旦数据在传输时出错，只需重传或纠正有错的帧，而不必重发全部数据，从而提高效率；

② 如果报文不分割成许多个短帧，可能会多次重传，多次出错，效率较低。报文分成若干短帧后，较小的帧出错的概率也小；

③ 检查一个较短帧的错误要比检查一个大的报文传输错误要容易，算法也较简单。

成帧时通常还为每个帧增加校验和（Checksum），接收方通过检查每帧的校验和，检查传输中是否出现差错，以决定是否让发送方重传。

2．帧同步

帧同步是指为了能让接收方收到的比特流中明确区分出一帧，发送方必须要建立和区分帧的边界（起始和终止），方法是在帧的起始和终止位置增加一些特殊的位组合。

作为帧边界的位组合在数据部分也有可能出现。如中文一般用双引号表示一句话，当这句话中又引用其他人的话时，一般用单引号引起来。例如：

张三说：“今天我听到李四说‘……’，……”

这个双引号可以理解为帧的边界（一个特殊的字符），中文要求双引号里面的内容不能再包含或出现这个双引号了（如果‘……’也用双引号就无法区分出起始和终止了）。

所以数据链路层中的帧为避免混淆，就采取专门的措施区分这个位组合是数据还是帧边界，这就是代码透明性问题。

常用的帧同步方法有下面 4 种。

1）字节计数法

字节计数法用一个特殊字符表示一帧的起始，并用一个专门的字段来标明帧内的字节数。接收方通过这个特殊字符从比特流中识别出帧的起始，并从专门的字段中获知该帧中随后跟随的信息长度，从而可确定出帧的终止位置。由于通过字节计数方法可以确定帧的终止边界，不会引起数据及其他信息的混淆，因而不必采取其他措施便可实现数据的透明性。

2）使用字符填充的首尾定界符法

这种方法采用一些特定字符来界定一帧的起止，为了不使数据信息位中出现与特定字符相同的字符而被误判为帧的首尾定界符，可以在这种字符前填充一个转义控制字符 DLE 以示区别，从而实现数据的透明性。

3）使用比特填充的首尾定界符法

该方法以一组特定的比特模式（如 01111110）来标识一帧的起止，高级数据链路控制规程 HDLC 采用的就是这种方法。为了区分信息位中出现的与该特定模式相同的比特串，可以采用比特填充的方法。例如，当信息中连续出现 5 个“1”时，发送方自动在其后插入一个“0”，而接收方则做该过程的逆操作，即每收到连续 5 个“1”，则自动删去其后所跟的“0”，实现数据传输的透明性。比特填充很容易由硬件来实现，性能优于字符填充方法。

4）违法编码法。

违法编码法在物理层采用特定的比特编码方法。例如当使用曼彻斯特编码方法时，将“1”编码成“高-低”电平对，将“0”编码成“低-高”电平对，而“高-高”电平对和“低-低”电平对在数据比特中是违法的。可以借用这些违法编码序列来界定帧的起始与终止，局域网 IEEE 802 标准就采用了这种方法。违法编码法不需要任何填充技术，便能实现数据的透明性，但它只适用于采用冗余编码的特殊编码环境。

3.2　差错控制

数据在传输中可能被破坏，因此需要进行检错和纠错。差错主要是由线路本身的电气

特性所产生的随机噪声（也称热噪声）、信号振幅、频率和相位的衰减或畸变、电信号在传输介质上的反射回音效应、相邻线路的串扰、外界的电磁干扰和设备故障等因素造成的。

3.2.1　差错类型和差错控制

1．差错类型

差错可分为单比特差错和突发差错两类，单比特差错是指在传输的数据单元中只有一个比特发生了改变（0 变 1 或 1 变 0）；突发差错是指在传输的数据单元中有两个或两个以上的比特发生了改变，发生差错的比特不一定连续，即可能的情况有连续几位出现差错、发生差错的位间隔一位或若干位后又出现差错。

2．差错控制方法

提高通信可靠性的办法有两种：一种方法是从硬件入手，选用高质量的传输介质并提高信号功率强度，采取最佳的信号编码和调制手段，使传输信号特性与信道特性达到最好的匹配，但这种方法大大增加了通信成本，这也是物理层的事情；另一种方法是在传输过程中进行差错控制，在数据链路层采用编码的方法进行查错或纠错处理。注意数据链路层的编码和物理层的编码是不同的，物理层的编码针对的是单个比特，主要解决传输过程中比特的同步等问题，如曼彻斯特编码。而数据链路层的编码针对的是一组比特，它通过冗余码的技术来检查一组二进制比特串在传输过程是否出现了差错。

3．检错码和纠错码

只具有检错能力的编码称为检错码，既能检错又具有自动纠错能力的编码则称为纠错码。差错控制方式有自动请求重发（Automatic Repeat-reQuest，ARQ）和前向纠错（Forward Error Correction，FEC）两种，ARQ 采用检错码方法实现，它使用冗余技术。所谓冗余技术是在发送方的数据单元中增加一些用于检查差错的附加位，便于接收端进行检错。一旦传输的正确性被确认，这些附加位就被接收端丢弃，并给发送端发送一个确认应答（Acknowledge character，ACK）。当接收端接收到的检错码检测到差错时，就给发送端发送一个否定应答（Negative Acknowledgment，NAK），并要求发送端重发数据。

FEC 采用纠错码方法实现，理论上可以自动纠正任何一种二进制编码中的所有差错，但纠错码比检错码要复杂得多，并且需要足够多的冗余位，实现起来复杂，编码和解码速度慢，效率较低，造价高而且费时。一般用于没有反向信道或线路传输时间长、重发费用较高的场合。大多数纠错技术只纠正一组比特中的一个、两个比特或三个比特的差错，所以在计算机网络中采用的大多数是检错码。后面介绍几种常用的差错控制编码方法。

3.2.2　差错控制编码

1．奇偶校验码

奇偶校验码是一种简单但能力有限的检错码，它是通过在信息位的后面附加一个检验位，使得码字中“1”的个数保持为奇数或偶数的编码方法。

奇偶校验码在一维空间上有“水平奇偶校验”和“垂直奇偶校验”码，在二维空间上有“水平垂直奇偶校验码”。

由于奇偶校验码容易实现，所以当信道干扰不太严重及信息位不太长时很有用，特别是

在计算机通信网的数据传送（如计算机串行通信）中经常应用这种检错码。

ASCII 代码（见附录 B）表示一个字符需要 7 位，而计算机内实际表示一个 ASCII 字符时，需要占用 8 个二进制位（一字节），其中 7 位是 ASCII 编码，另外一位作为奇偶校验位（冗余位）。虽然奇偶校验不能提供出错位置，也不具备纠错能力，但实践证明它是一种简单、有效的差错检测方法。

例如发送方要发送“word”这个单词，则这四个字母的 ASCII 代码为：

←－－－－　1110111　1101111　1110010　1100100

发送方向　　w　　o　　r　　d

如果使用水平偶校验方法，则发送数据时保证表示每个字母的二进制位序列中 1 的个数为偶数，实际发送“word”这个单词时，发送的二进制位序列和发送方向为：

←－－－－　11101110　11011110　11100100　11001001

发送方向　　w　　o　　r　　d

每字节的最后一位为冗余位，它保证每字节的 1 的个数为偶数个。接收方针对每字节统计 1 的个数，如果每字节都是偶数个 1，作为无差错传输处理（如果错了两位，还是偶数个 1，也不认为传输有错）。如果有一字节是奇数个 1，表示数据传输过程中受到破坏。

2．循环冗余校验码

奇偶校验码能力有限，如果将它使用在二进制位较多的帧序列中，大部分传输错误是检查不出来的。所以计算机网络中，需要使用检错能力更强的检错码，循环冗余校验码就是其中的一种。

循环冗余校验码（Cyclic Redundancy Check，CRC）又称多项式码，它是一种在计算机网络和数据通信中最常用的检错码。CRC 的基本思路是收发双方选定一个特定的二进制数（后面所述的生成多项式 $G(x)$的系数），发送方将需要发送的数据使用这个特定的二进制数做除法运算，计算出冗余码，然后将冗余码附加在数据后生成一个新的数据帧再发送，接收方对收到的数据同样使用这个二进制特定数做除法运算，以此判断有没有传输错误，以实现差错检查的目的。

下面举例说明 CRC 检验原理和计算方法。

（1）假定数据帧有 k 比特，例如发送方需要发送的信息位 M=1011001，可以将它们作为对应多项式 $F(x)=x^6+x^4+x^3+1$ 的系数，这时 k=7。

（2）双方预先约定一个特定的除数有 p 比特（是一个特定的生成多项式的系数，后面还将介绍四种主要的生成多项式），例如为 p=11001，可以将它们作为对应多项式 $G(x)=x^4+x^3+1$ 的系数，这时 p=5。

（3）CRC 运算就是信息位 M 后附加 n 位冗余码，$n=p-1$，这里 $n=5-1=4$。也就是说，发送方构成一个新的数据帧，共（$k+n=7+4$）11 位发送出去。

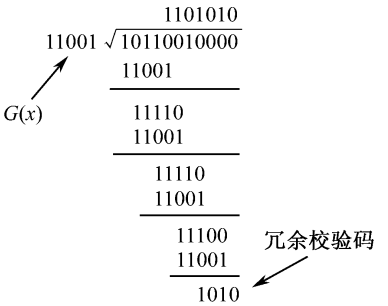
（4）冗余码计算方法如下。因为本例的冗余码有 4 位，所以先在信息位 M=1011001 后添加上 4 个 0，变为“1011001**0000**”。然后将它用双方预先约定的长度为 $n+1$ 的 p（11001）去除（模 2 运算，不借位也不进位）。

（5）除法运算得到一个 n 位余数（余数位数不够 n 位，前面补若干位 0，例如余数为 11 时，前面补两位为 0011），这 n 位余数就是冗余位，计算得到的商没什么用处。本例得到的余数为“1010”，将它替换掉步骤（4）中“1011001**0000**”的后 4 个 0，得到循环冗余校验码“1011001**1010**”。

（6）发送方将 CRC 码“1011001**1010**”发送出去。在接收端，用同样的 p（11001）去除，若能被其整除，表示传输正确，同时将后四位的冗余位丢弃；否则表示数据传输有错，

通知发送方重传。

除法运算如下：



运算时需要注意以下两点：

（1）这里所涉及的运算都是模 2 运算，即“异或”运算（0−0=0，1−1=0，0−1=1，1−0=1）。

（2）因为 $F(x)$对应的多项式比特序列后加上了 4 个 0，所以冗余位也应是 4 位。假如余数位数为“10”，那么冗余位是“0010”。

本例中经相除后得到的余数“1010”就是冗余码，发送方传输前将冗余码“1010”附加到信息“1011001”，因此实际传输的 CRC 码为“10110011010”。

接收端接收数据后，用同样的 $G(x)$系数“11001”去除接收到的比特序列，如果整除（余数为 0）则表示传输正确，同时接收端将附加的冗余码“1010”丢弃；如果没整除（余数不为 0）则表示传输错误，接收端发送“NAK” 给发送端，发送端重发数据。

表 3.1　（7,3）CRC 码

信息位	（7,3）CRC 码
000	0000000
001	0011101
010	0100111
011	0111010
100	1001110
101	1010011
110	1101001
111	1110100

【例 3.1】 在（7,3）码中，信息码有 3 位，可分别表示十进制数据 0～7，设 $G(x)$为 $x^4+x^3+x^2+1$，对应的生成多项式比特序列为“11101”，由 5 位组成，因此冗余校验码应是 4 位，通过计算，得到表 3.1 所示的（7,3）CRC 码。

本来 7 位二进制码的排列中，可以表示 $2^7+1=128$ 个码字，但表中所得（7,3）CRC 码只有 8 个。实际上这 8 个是从 128 个码字中，按照一致校验方程组挑选出来的。CRC 码有如下特性：

（1）封闭性。表中任两个 CRC 码的对应位进行模 2 相加后得到的结果，仍然是表 3.1 中 8 个码字中的一个。

（2）循环性。表中任意一个（7,3）CRC 码字循环右移一位或多位后，仍是表中 8 个（7,3）CRC 码字中的一个。

循环冗余校验码中的“循环”也来自它的循环性，循环冗余校验码在数据传输中得到了最广泛的应用。使用这种冗余编码的实质在于，传输信息符号时，不使用全部编码组合，而只使用其中的一部分，这部分编码具有某种事先确定的性质。当在接收端出现不使用的编码组合（禁用码）时，说明在某一位或若干位中发生了错误。CRC 码还有纠错功能，但网络中不使用其纠错功能，仅用其强大的检错功能，检出错误后要求重发。

目前广泛使用的生成多项式主要有四种：

$CRC_{12}=x^{12}+x^{11}+x^3+x^2+1$

$CRC_{16}=x^{16}+x^{15}+x^2+1$

$CRC_{16}=x^{16}+x^{12}+x^5+1$

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

因为除法运算易于用移位寄存器和模 2 加法器实现，因此循环冗余校验码的编译码过程通常采用硬件实现，可以达到较高的处理速度。随着集成电路工艺的发展，循环冗余码的产生和校验均由集成电路产品完成，发送端能够自动生成 CRC 码，接收端可自动校验，速度大大提高。Ethernet 采用的是 32 位 CRC 码，它可以用专用的芯片实现。

3．海明码

海明码是 R.Hamming 于 1950 年首次提出的，它是一种可以纠正单比特差错的编码，它也是通过增加冗余位进行纠错的。

设信息位为 k 位，现增加 r 位冗余位，则构成 $n=k+r$ 位码字。若希望用 r 个监督关系式产生 r 个校正因子来判断码字在传输后是否出错，并确定出错位的位置，则要求满足下列关系式：

$$2^r \geqslant n+1 \quad 或 \quad 2^r \geqslant k+r+1$$

例如，当 $k=4$ 时，为了满足上述不等式，则需要 $r\geqslant 3$。现取 $r=3$，则 $n=k+r=7$。也就是说，在 4 位信息位 $a_6a_5a_4a_3$ 后面加上 3 位冗余位 $a_2a_1a_0$，构成 7 位码字 $a_6a_5a_4a_3a_2a_1a_0$。其中 a_2、a_1 和 a_0 分别可通过 4 位信息位中的某几位按模 2 相加的方法得到。在校验时，a_2、a_1 和 a_0 就分别和 4 个信息位构成 3 个不同的监督关系式。

如果传输后没有错误，监督关系式 S_2、S_1 和 S_0 的值应该全为“0”，即 $S_2S_1S_0$ 的值为“000”时表示传输无错。如果 a_0 传输后出错，可以设定 $S_0=1$，而 $S_2=S_1=0$，即 $S_2S_1S_0$ 的值为“001”时，表示 a_0 传输出错。$S_2S_1S_0$ 值与出错码位置的对应见表 3.2，当然也可以规定成其他形式的对应关系。

表 3.2　$S_2S_1S_0$ 值与出错码位置的对应

$S_2S_1S_0$ 值	000	001	010	100	011	101	110	111
出错码	无错	a_0	a_1	a_2	a_3	a_4	a_5	a_6

由表 3.2 可见，a_2、a_4、a_6 中某一位传输错误都应使 $S_2=1$，由此可以得到监督关系式：

$$S_2=a_2\oplus a_4\oplus a_5\oplus a_6$$

同理可得：

$$S_1=a_1\oplus a_3\oplus a_5\oplus a_6 \qquad (3.1)$$

$$S_0=a_0\oplus a_3\oplus a_4\oplus a_6$$

发送端在进行编码时，信息位 a_6、a_5、a_4 和 a_3 的值是随机的，冗余位 a_2、a_1 和 a_0 的值根据信息位的取值按监督关系式生成，它需要将式（3.1）中的 S_2、S_1 和 S_0 取值为零，即

$$\begin{aligned} a_2\oplus a_4\oplus a_5\oplus a_6&=0\\ a_1\oplus a_3\oplus a_5\oplus a_6&=0 \qquad (3.2)\\ a_0\oplus a_3\oplus a_4\oplus a_6&=0 \end{aligned}$$

由此可求得

$$\begin{aligned} a_2&=a_4\oplus a_5\oplus a_6\\ a_1&=a_3\oplus a_5\oplus a_6 \qquad (3.3)\\ a_0&=a_3\oplus a_4\oplus a_6 \end{aligned}$$

发送端就是根据式（3.3）计算出冗余位的。由信息位算得的海明码冗余位见表 3.3。

例如，十六进制数字 A（1010）的海明码为 1010010，发送端按海明码发送，接收端收到这个码字后，按监督关系式计算出 S_2、S_1 和 S。的值，若全为“0”，则没有出错；若不全为“0”，在某一位出错的情况下，可查表 3.2 来判定是哪一位错，从而纠正之。例如码字“0010101”传输中发生一位错，在接收端收到的为“0011101”，通过监督关系式可算得 S_2=0、

S_1=1 和 S_0=1，由表 3.2 可查得 $S_2S_1S_0$=011 对应于 a_3 错，因而可将“0011101”纠正为“0010101”。

表 3.3　由信息位算得的海明码冗余位

信息位 $a_6a_5a_4a_3$	冗余位 $a_2a_1a_0$	信息位 $a_6a_5a_4a_3$	冗余位 $a_2a_1a_0$
0000	000	1000	111
0001	011	1001	100
0010	101	1010	010
0011	110	1011	001
0100	110	1100	001
0101	101	1101	010
0110	011	1110	100
0111	000	1111	111

3.3　流量控制和链路管理

流量控制指限制发送方的数据发送流量，使得发送速率不至于超过接收方所能处理的能力范围上限，而导致接收方数据帧的“淹没”。当发送方发送速率大于接收方的接收速率，或接收方缓存中帧已满还来不及处理时，就会被发送方源源不断发送来的帧所“淹没”，从而造成帧的丢失而出错。当接收方缓存将满时，必须通知发送方暂停发送，直到接收方又能接收数据。

流量控制实际上是一组规则，使得发送方知道在什么情况下可以接着发送下一帧，什么情况下必须暂停发送，等待收到某种反馈信息后再继续发送等。

常用的流量控制方法是停止等待和滑动窗口等机制。

3.3.1　停止等待

停止等待流量控制机制中，发送方每发出一帧就等待接收方返回的一个确认帧（ACK），只有当接收到确认帧后，才发送下一帧，否则继续等待。这种发送和等待交替的过程不断重复，直到发送方发送了一个传输结束帧（EOT），完成一次数据传输，停止等待流量控制机制如图 3.1 所示。

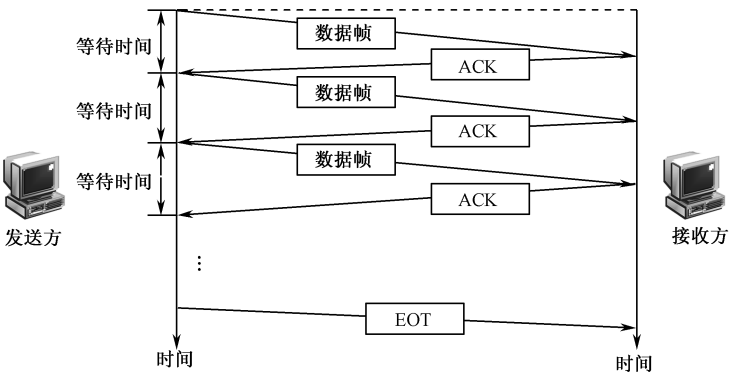


图 3.1　停止等待流量控制机制

停止等待流量控制机制较为简单，它要求在发送新的一帧时上一次发送的帧必须得到校验并确认。它的缺点是效率比较低，因为在下一帧发送之前，每一帧必须穿越所有的路径到达接收方，接收方经过校验后再将确认帧传输回来。如果发送方和接收方设备之间的距离较长，而每传输一帧需等待 ACK 帧，所花费的时间将大大增加总传输时间，因此停止等待方式传输速度很慢。

3.3.2　滑动窗口

滑动窗口流量控制机制，发送方在收到确认帧前可以发送若干帧。帧可以直接依次发送，即链路上可能同时承载多个数据帧，从而充分有效地使用了链路的能力。接收方只对其中一些帧进行确认，使用一个 ACK 帧来对多个数据帧的接收进行确认。

滑动窗口是发送方和接收方创建的一个额外缓冲区（发送方的发送窗口和接收方的接收窗口），窗口可以存储若干数据帧，窗口在数据传输过程中根据控制向前滑动，从而控制数据传输过程，并且发送方在收到接收方的确认之前对能够传输的帧数目也进行了限制。

为了记录哪些帧已经被传输以及接收了哪些帧，滑动窗口引入了一个基于窗口大小的标识机制。帧以模 n 的方式标识，即帧编号从 0 到 n–1。例如 n=8，则帧编号为“0、1、2、3、4、5、6、7、0、1、2、3、4、5、6、7、0、1…”。同时规定滑动窗口的大小为 n–1（本例的窗口大小为 7），也就是说窗口能覆盖的帧数为所有编号的帧数减 1。通俗来说，当帧编号为 0～7 共 8 个数时，窗口大小为 7，它不能覆盖所有编号，滑动窗口如图 3.2 所示。

图 3.2　滑动窗口

接收方可以不等窗口被填满就在任意一点对数据帧进行确认，并且只要接收方窗口未满，发送方就可以继续传输。当接收方发出一个确认帧时，这个确认帧中还包含了将要接收的下一帧编号。例如接收方发出已接收 1 号帧的确认帧，其中就包含了将要接收 2 号帧。这时发送方收到含有编号 2 的确认帧时，就知道了编号 1 前的所有数据帧均已经被接收了。

由于收发两端的窗口最多存储 n–1 个帧，所以发送方在收到接收方的一个确认帧前，最多可以发送 n–1 个帧。

1．发送方窗口

发送窗口用来对发方进行流量控制，窗口大小指明了在收到对方 ACK 之前最多可以发送的数据帧数，窗口内的帧是可以连续发送的。

发送方传输开始前，窗口有 n–1 个帧。随着数据帧的发送，窗口的左边界向内移动，窗口不断缩小。例如在接收到最近一次确认帧时已经发送了 3 个帧，那么窗口中剩余的帧数是 n–1–3。一旦收到一个确认帧。窗口右边界根据确认帧确认的数据帧个数自动对窗口进行相同数目的扩展。

例如一个大小为 7 的发送方窗口，如图 3.3 所示。假设发送方已发送了 0～3 号共 4 个帧，窗口的左边界也向右移动了 4 个位置，如果还没有收到确认帧，则发送方窗口内就只有 3 个帧（4、5 和 6 号）。假设这时收到了编号为 3 的确认帧，就知道已经有 3 个帧（0、1 和 2 号，而 3 号帧接收方正在接收）正确传输到对方，同时发送方扩展其窗口右边界，将缓冲区中后面 3 个帧（7、0 和 1）包含到窗口中，这时发送方窗口包含 6 个帧（4、5、6、7、0 和 1 号）。

注意：发送方将数据帧发送出去时，滑动窗口左边界向右收缩窗口。而当收到确认帧时，发送方滑动窗口右边界向右扩展。

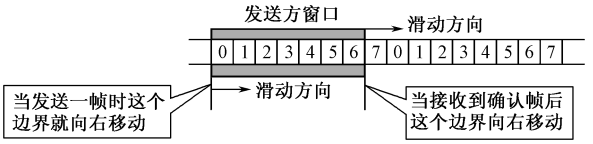


图 3.3　发送方窗口

2．接收方窗口

接收窗口可以控制哪些数据帧可以接收，只有数据帧的序号在接收窗口之内的才可以被接收，接收过的数据帧将被丢弃。一般接收方收到一个有序且无差错的帧后，接收窗口向前滑动，并准备接收下一帧，这时可以向发送方发出一个确认帧。为了提高效率，接收方可以采用累计确认或捎带确认的方式。捎带确认是在双向数据传输的情况下，将确认信息放在自己也要发送的数据帧的首字段中捎带过去。

在传输开始的时候，接收方窗口有 $n-1$ 个帧空间但不一定包含 $n-1$ 个帧。接收数据帧后，接收方窗口会不断缩小。它表示发送确认帧前窗口中还可接收的帧的数目（剩余的帧数）。一旦发送完一个确认帧，窗口大小就会自动扩展。

例如，一个大小为 7 的接收方窗口，如图 3.4 所示。这时接收窗口包含了 7 个帧，表示目前可以接收 7 个数据帧。如果已接收了 1 个帧（0 号），窗口的左边界向右移动 1 帧的位置，这时接收窗口在发送确认帧之前还可以接收 6 个帧。如果 0 号帧到 3 号帧已经接收但还没有确认，那么窗口就只有 3 帧的空间。

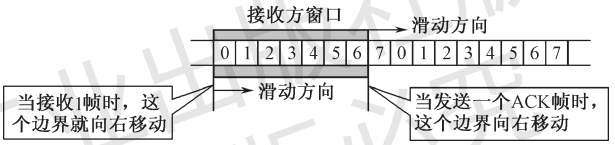


图 3.4　接收方窗口

发送方发送确认帧后，接收方滑动窗口的右边界就会向右扩展，它按照最近确认的帧数来扩展相同数目的位置（窗口的扩展数等于最近的确认帧中包含的编号减去上一确认帧中包含的编号）。例如，上一确认帧中包含的编号为 2，而当前确认帧中包含的编号为 5，则窗口自动扩展 3 个空间；如果上一确认帧中包含的编号为 3，而当前确认帧中包含的编号为 1，则窗口自动扩展 6（1+8–3）个空间。

注意：接收方接收数据帧后，接收方滑动窗口左边界向右收缩窗口。而当发送确认帧时，接收方滑动窗口右边界向右扩展。

3．滑动窗口的流量控制使用

发送方收到接收方的确认后，发送窗口右边界向右移动，同时新的帧会到达发送窗口，已被确认正确收到的帧移到了窗口的外面。所以接收方的确认作为一个依据，控制发送方发送窗口向前滑动。接收方可以根据自己的接收能力来控制确认帧的发送，从而实现传输流量的控制。

由于滑动窗口中使用了确认机制，因此它也兼有差错控制的功能。

前面讲到滑动窗口的大小比模数小 1，原因是为了避免确认帧中包含的编号出现二义性。假设 n=8，窗口大小也为 8，如果这时发送了 0 号帧，又收到编号为 1 的确认帧（ACK 1）。发送方就开始扩展窗口，并继续发送 1、2、3、4、5、6、7 和 0 号帧。如果此时发送方又收到 ACK 1，接收方就不知道是因为网络问题而重发的上一次的 ACK 1，还是最近发送的

8 帧的新的 ACK 1。因此将窗口大小设定为 7（n–1），就不会发生这种情况了。

滑动窗口中，控制传输流量主要采取以下措施：

（1）设置合适的发送窗口大小，一般不超过接收方接收缓冲区的大小。这样发送方发送的数据就不容易“淹没”接收缓冲区。

（2）可变滑动窗口。由接收方根据当前接收缓冲区的大小决定发送方发送窗口的大小，并通知发送方改变发送窗口的大小，TCP 协议流量控制就使用的这种方式。

（3）接收方根据目前可用接收缓冲区的情况，决定发送确认的时机，使发送流量与接收缓冲区的可用容量匹配。

停止等待和滑动窗口等流量控制机制不仅适用于局域网，也适用于城域网和广域网。以太网中帧的传输、TCP 协议等也使用这些流量控制机制。

3.3.3　链路管理

链路的建立、维持和释放称为数据链路层的链路管理。链路管理功能主要用于面向连接的服务，它可以为网络层提供几种不同质量的链路服务。链路两端的节点通信前，必须首先确认对方已处于就绪状态（如发送一个询问帧），并交换一些必要的信息以对帧序号初始化，然后才能建立连接。在传输过程中则要维持该连接。如果出现差错，需要重新初始化，重新自动建立连接，传输完毕后则要释放连接。若传输的正确性被确认，则接收方发一个确认应答 ACK；否则，发送一个否定应答 NAK。

在多点共享（广播式）网络中，通信站点间信道的分配和管理也属于数据层链路管理的范畴。例如，Ethernet 中采用的介质访问控制方法 CSMA/CD。CSMA/CD 将在第 4 章进行介绍。

3.4　数据链路协议

数据链路层的“协议”也称为“规程”。在计算机通信的早期，对于经常产生误码的实际链路，只要加上合适的控制规程，就可以使通信变得比较可靠。ARPAnet 使用了 IMP-IMP 协议，而 IBM 使用了 BSC（Binary Synchronous Communication）规程。

数据链路协议主要分为异步协议和同步协议两大类，计算机网络系统主要采用同步协议。

1．异步协议

异步协议以字符（1 字节，8 位）作为信息传输单位，在每个字符的起始处同步，但字符之间的间隔时间是不固定的（字符之间是异步的）。由于发送器和接收器中都有一个近似于同一频率的时钟，它们可以在一段较短的时间内保持同步，所以可以用字符起始处同步的时钟来采样该字符中的各比特，而不需要对该字符内的每个比特同步。前面介绍过的起止式通信规程便是异步协议的典型，它是靠起始位（逻辑 0）和停止位（逻辑 1）来实现字符的定界及字符内比特的同步的。异步协议中由于每个传输字符都要添加诸如起始位、校验位、停止位等冗余位，故信道利用率很低，一般用于数据速率较低的场合，主要是用在调制解调器中。

2．同步协议

同步协议以许多字符或许多比特组成的数据块为传输单位，在帧的起始处同步，使帧内维持固定的时钟。由于采用帧为传输单位，所以同步协议能更有效地利用信道，也便于实现差错控制、流量控制等功能。

同步协议可分为两类：面向比特的协议（Bit-Oriented Protocol）和面向字符（字节）的协议（Character-Oriented Protocol），计算机网络主要使用面向比特的协议。

（1）面向比特的协议

在面向比特的协议中，信息传输以位为单位，链路监控功能通过传输一定的位组合所表示的命令和响应来实现，而且它们可以与信息一起传送。

高级数据链路控制（High-level Data Link Control，HDLC）是一个面向比特的协议，面向比特意味着 HDLC 把帧当作比特流，它支持半双工和全双工通信。由于 HDLC 是 ISO 定义的，所以 HDLC 得到了广泛使用，所有面向比特的协议都与高级数据链路控制 HDLC 有关。

（2）面向字符的协议

面向字符的协议效率比面向比特的协议效率低，例如 BSC 规程，现在很少采用。

3．局域网数据链路层协议

局域网数据链路层协议主要由 IEEE 802 小组制定，它们涵盖了物理层和数据链路层。这些标准主要包括以下 4 个：

① Ethernet（以太网）；

② Token-Ring（令牌环）；

③ Token-Bus（令牌总线）；

④ WLAN（无线局域网）。

这些标准将在后面的章节中陆续讨论。

4．广域网数据链路层协议

广域网是基于交换技术的网络，网络中的中间节点负责将数据转发到下一个节点，节点间的线路利用率高。与局域网相比，广域网数据链路层技术复杂，它需要将数据封装成适合广域网传输的帧，以保证数据的可靠传输。广域网通信子网部分由公共传输系统组成，提供相应服务的一般是电信运营商，如电信、联通、移动等。

广域网数据链路层标准主要有以下 4 个：

① HDLC（高级数据链路控制）；

② X.25（公共分组交换网）；

③ PPP（点到点协议）；

④ Frame Relay（帧中继）。

3.5　本章小结

（1）数据链路层的功能是将物理层传输原始比特流时可能出错的物理连接改造成为逻辑上无差错的数据链路，在节点间实现透明的高可靠性传输，同时为网络层提供有效的服务。主要功能有成帧、差错控制、流量控制等。

（2）物理层的传输单位是比特或位，数据链路层的传输单位是数据帧。

（3）成帧的目的在于：一旦传输出错，只需重传有错的帧，而不必重发全部帧，从而提高效率；将报文分成若干帧后，短帧出错概率小；检查一个较短帧的传输是否出错要比检查一个大的报文更容易，算法也更简单。

（4）局域网 LAN 只用到物理层和数据链路层，以太网网卡也包含这两层的功能。注意：低两层考虑点到点直接连接的情形，

（5）差错控制是指数据链路层采用编码（增加冗余码技术）对传输帧进行查错或纠错处

理，例如奇偶校验码、循环冗余校验码 CRC。它与物理层编码概念不同，物理层编码是指将数据编码成光、电信号，以保证二进制位的传输，例如曼彻斯特编码；数据链路层编码是指在数据帧后增加冗余位，并通过算法检验数据帧传输是否正确，以决定接收方是接收还是请求重发数据帧。

（6）差错控制编码分为检错码和纠错码，相应的控制方式有自动请求重发 ARQ 和前向纠错 FEC 两种。

（7）循环冗余校验码 CRC 是一种检错码，而海明码是一种纠错码。CRC 码有封闭性和循环性两个特性。

（8）流量控制用来限制发送方在等待确认前可以发送的数据流量，主要是因为接收方来不及处理而导致接收方数据帧的“淹没”。常用的流量控制方法是停止等待和滑动窗口机制。

（9）链路的建立、维持和释放称为链路管理，主要用于面向连接的服务。例如 Ethernet 中采用的介质访问控制方法 CSMA/CD。

（10）数据链路控制协议主要分为异步协议和同步协议两大类。异步协议以字符作为信息传输单位；同步协议以许多字符或许多比特组成的数据块为传输单位。

（11）同步协议可分为面向字符（字节）的协议和面向比特的协议两类，面向比特的协议信息传输时以位为单位。

（12）计算机网络主要使用同步协议中的“面向比特的协议”，高级数据链路控制 HDLC 是一个面向比特的协议，所有面向比特的协议都与 HDLC 有关。

习　题

一、选择题

1．数据链路层在数据包前添加链路层的控制信息作为头部信息，形成（　　），再传递到物理层，在物理层传送原始的比特流。

A．帧　　B．信元　　C．数据包　　D．以上都不是

2．HDLC 是一种面向（　　）的链路层协议。

A．字符　　B．比特　　C．信元　　D．数据包

3．在数据链路层是通过（　　）找到本地网络上的主机的。

A．端口号　　B．MAC 地址　　C．默认网关　　D．逻辑网络地址

4．数据链路层可提供的功能有（　　）。

A．对数据分段　　B．提供逻辑地址

C．提供差错控制和流量控制功能　　D．以上都不是

5．对数据链路层的描述正确的是（　　）。

A．实现数据传输所需要的机械、接口、电气等属性

B．实施流量控制、错误检测、链路管理及物理寻址

C．检查网络拓扑结构，进行路由选择和报文转发

D．提供应用软件的接口

6．在（　　），数据传输单元被称为帧。

A．物理层　　B．数据链路层　　C．网络层　　D．传输层

7．采用滑动窗口机制对两个相邻节点 A（发送方）和 B（接收方）的通信过程进行流量控制。现假设发送窗口与接收窗口的大小均为 7，在 A 发送了编号为 0、1、2、3 这 4 个帧

后，B 接收了这 4 帧，但仅应答 0、1 两个帧，此时发送窗口将要发送的帧序号为（　　）（1），接收窗口的左边界对应的帧序号为（　　）（2）。

（1）A．2　　B．3　　C．4　　D．5

（2）A．1　　B．2　　C．3　　D．4

二、填空题

1．差错控制技术主要包括前向纠错机制和_____两种。

2．差错控制编码可以分为_____和_____两种。其中，_____是自动发现差错的编码；_____是指不仅能发现差错而且能自动纠正差错的编码。

3．奇偶校验码又可以分为_____奇偶校验、_____奇偶校验和同时使用这两种方法的奇偶校验。

4．在奇校验方案中，二进制序列“0101101”的校验位为____。

5．帧同步是指数据的接收方应当从接收到的比特中准确地区分帧的_____。

6．数据链路层的传送单元是_____。

7．常用的帧同步方法有_____、_____、_____和_____4 种。

8．局域网中进行差错控制时，广泛使用的校验编码是_____校验。

9．CRC 码有_____和_____两个特性。

10．海明码是一种_____（检错码/纠错码）。

11．数据链路层同步协议可分为面向_____和面向_____两类。高级数据链路控制 HDLC 是一个面向_____的协议。

三、问答题

1．数据链路层主要有哪些功能？

2．什么是帧同步？常用的帧同步方法有哪几种？

3．提高通信可靠性的方法有哪两种？检错码和纠错码有什么不同？

4．什么是 ARQ，什么是 FEC？它们各有什么特点？说出它们是检错码还是纠错码。

5．循环冗余校验码 CRC 有什么特性？

6．如果有一个数据比特序列为“100101110010”，CRC 校验中的生成多项式为：$G(x)=x^4+x^2+1$，请计算 CRC 校验码比特序列。

7．数据链路控制协议主要分为哪两大类？

8．说出你所知道的局域网数据链路层协议和广域网数据链路层协议。