第3章 数据链路层协议

数据链路层处于OSI模型的第2层，介于物理层和网络层之间，设置数据链路层的主要目的是解决物理层传输的不可靠问题，提供功能上和规程上的方法，以便建立、维护和释放网络实体间的数据链路。

数据链路层在物理层提供的服务基础上向网络层提供服务，其最基本的服务是将源节点网络层传递过来的数据可靠地传送到相邻节点。因此，数据链路必须具备一系列相应的功能。数据链路层属于通信子网。

3.1 数据链路层概述

1. 数据链路层模型

所谓链路一般是指相邻节点之间的一条点到点的物理线路，又称为物理链路，但是仅有物理链路并不能实现数据的传输，还需要有相应的通信协议来控制数据的传输。将实现通信协议的硬件和软件加到物理链路上所构成的可以通信的链路称为数据链路，又称为逻辑链路，即通信规程＋物理链路＝数据链路。一条数据链路类似于一个数字管道。当采用多路复用技术时，一条物理链路上可以有多条数据链路。

数据链路层协议定义了一条链路的两个节点间交换的数据单元格式，以及节点发送和接收数据单元的动作。数据链路层模型如图3-1所示。

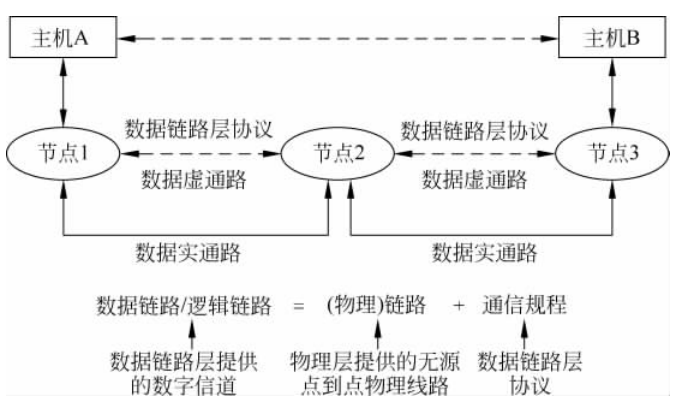


图3-1 数据链路层模型

2. 数据链路层功能

数据链路层利用不可靠的物理线路向网络层提供可靠的数据链路，因此，数据链路层应具有帧的定界功能，搭建一种能够识别帧的开始和结束的结构。帧的结构中可以包含错误检测机制，错误的纠正既可以后向的通过帧的重传完成，也可以前向的通过冗余纠错编码完成。对于某些数据链路连接，还应该能够提供保序和流量控制功能，保证在数据链路层连接上收到的帧能够以和发送方相同的顺序递交给网络层实体，并协调发送方和接收方的节奏，保证发送方的发送速度不会太快以致接收方被淹没。数据链路层具有以下功能。

(1) 成帧。成帧是指帧定界的方法。发送方数据链路层将网络层传递下来的数据分组根据某种定界方法划分成大小一定的帧，并在帧头和帧尾添加可以与传输数据相区别的定界符。接收方数据链路层对所接收到的比特流，应能确定每一帧的开始和结束位置。帧的定界有多种方法，如字符计数法、首尾界符法、首尾标志法等。

(2) 帧的透明传输。成帧方法是在帧中增加了一些控制信息，以确定帧的定界，为了保证帧的透明传输，还需要进行一些特殊的处理，否则将不能正确地区分数据与控制信息，这些特殊的处理对高层来说是透明的。

(3) 流量控制。流量控制就是对发送方发送数据的速度加以控制，以免超过接收方的接收能力而导致数据丢失。

(4) 差错控制。差错控制就是接收方对接收到的数据帧进行校验，如果发生差错，则应该能够对错误帧进行相应处理。数据链路层一般采用在信息位中添加冗余码的方法进行差错校验，接收方利用冗余码可以检测出接收到的帧是否存在差错，如果有错，既可以采用纠错编码前向纠错，也可以将它丢弃，并通知发送方重传出错的数据帧。

(5) 数据链路管理。如果在数据链路上传送数据采用面向连接的方式，则发送方和接收方之间需要有建立、维持和释放数据链路连接的管理功能。

(6) 寻址。数据链路层可以通过编址及识别相应的地址(一般称为硬件地址)，来保证每一帧数据都能传送到规定的目的地，接收方也应能识别出接收到的数据帧来自哪里。

数据链路层通过实现上述功能来提供数据帧的可靠传输，消除物理层传输的不可靠性，对网络层提供一条无差错的、可靠的数据链路。

3. 数据链路层向网络层提供的服务类型

数据链路层可以将源节点的网络层数据可靠地传输到相邻节点的数据链路层，并由其递交给其上的网络层。数据链路层主要提供以下 3 种服务。

(1) 无确认的无连接服务。无确认的无连接服务简单，适用于低误码率环境中的数据传输。大多数局域网的数据链路层都使用无确认的无连接服务。该服务主要包含以下 5 个方面：

① 双方无须建立链路连接；

② 每个帧都带有目的地址；

③ 各帧相互独立传送；

④ 目的节点对收到的帧不做任何应答确认；

⑤ 由高层处理丢失的帧，数据链路层不做处理。

(2) 有确认的无连接服务。有确认的无连接服务是在无确认的无连接服务基础上增加了确认功能,适用于可靠性不高的通信信道。该服务主要包含以下两个方面:

① 目的节点对接收到的每一帧都要向发送方发送确认帧(ACK);

② 发送方利用超时机制处理确认帧,每发送一个数据帧的同时启动一个定时器,若在规定时间内未收到对该帧的肯定确认帧,则发送方启动超时重发机制,重发该数据帧。

(3) 面向连接服务。面向连接服务是指在数据传输之前首先建立数据链路连接,然后所有的数据帧均在该链路中依次按序传输,最后传输结束时再释放该连接。适用于实时传输或对数据传输有较高可靠性要求的环境。面向连接服务主要分为以下 3 个阶段。

① 第一阶段:连接建立阶段。在传送数据之前,首先利用服务原语建立一条连接(即建立数据链路)。

② 第二阶段:连接维持阶段。本阶段完成数据帧的透明传输。所有数据帧都带上各自的编号,传输过程中对每一帧都要进行确认,发送方收到确认后才能发送下一帧。

③ 第三阶段:连接断开阶段。数据传输结束后,妥善释放数据链路。

面向连接服务的建立连接和释放连接过程中所用到的服务原语主要有请求(request)、指示(indication)、响应(response)和确认(confirm)4 种。不同阶段使用的原语并不完全相同。例如,连接建立阶段需要使用 4 种原语,连接维持阶段(数据传输阶段)和连接释放阶段只需使用请求和指示两种原语。

- 连接建立阶段:DL-CONNECT. request,DL-CONNECT. indication,
  DL-CONNECT. response,DL-CONNECT. confirm。
- 连接维持阶段:DL-DATA. request,DL-DATA. indication。
- 连接释放阶段:DL-DISCONNECT. request,DL-DISCONNECT. indication。

3.2 差错控制

物理层的任务是接收一个原始的比特流,并准备将它传输到目的地。在传输过程中传输的比特流的个数和内容可能会发生变化,即产生差错,但目前已有的物理层协议对传输的比特流并不进行任何检测和纠错。也就是说,物理层并不保证这个比特流的正确传输,物理层传输产生的差错将由数据链路层负责检测和纠错。

3.2.1 传输差错

差错是指在数据传输过程中,接收方接收到的数据与发送方发送的数据出现不一致的现象。网络通信过程中,差错是不可避免的,为了保证通信质量,减少差错,系统必须具有差错控制及差错检测机制。

3.2.2 差错控制方法

在数据通信中,差错控制方法基本上分为两大类:自动重传请求(Automatic Request for Repeat,ARQ)和前向纠错(Forward Error Correction,FEC)。

在 ARQ 方式中，接收方发现接收的数据帧出现差错时，用某种方式通知发送方重传该数据帧，直到收到正确的数据帧为止，这是一种后向纠错方法。

在 FEC 方式中，接收方不但能发现接收的数据帧中的差错，而且能确定二进制代码中发生错误的位置，从而进行纠正，这是一种自动纠错方式，也称为前向纠错方法。

能够发现差错的编码称为检错码（又称检验码或校验码），不仅能发现差错而且能自动纠错的编码称为纠错码。

虽然纠错码可以避免出现差错的数据帧的重传，节省带宽和网络资源，但由于纠错码冗余度较大，编码效率低，解码设备复杂等原因，使得纠错码在网络中应用并不广泛。目前，使用最广泛的差错控制方式仍是 ARQ 方式。

1. 检错码

目前，所有的差错检验编码都是采用冗余编码技术。具体方法很多，其差别就是冗余度不同，能检错的位数不同，但核心思想是相同的，都是在有效数据（信息位）发送前，按照某种关系附加上一定的冗余位（冗余位与数据相关，若数据不同，则冗余位也不同），构成一个符合某一规则的码字后再发送。接收方收到带有冗余位的码字后，用同样的方法判断它是否仍然符合原规则，若不符合，则可判定传输过程中出现了差错。

常用的检错码有奇偶校验码和循环冗余校验码。

1）奇偶校验码

奇偶校验码是一种通过增加冗余位使得码字中 1 的个数恒为奇数或偶数的编码方法。在实际使用时又可分为垂直奇偶校验、水平奇偶校验和水平垂直奇偶校验等。

（1）垂直奇偶校验。垂直奇偶校验又称为纵向奇偶校验，它是将要发送的整个信息块分为定长 p 位的若干段（例如 q 段），在每段后面按 1 的个数为奇数或偶数的规律加上一位奇偶校验位，如图 3-2 所示。在信息（$I_{11},I_{21},\cdots,I_{p1},I_{12},I_{22},\cdots,I_{p2},\cdots,I_{1q},\cdots,I_{pq}$）中，每 p 位构成一段（即图中的一列），共有 q 段（即共有 q 列）。每段加上一位奇偶校验冗余位，即图 3-2 中的 $r_i(i=1,2,3,\cdots,q)$。

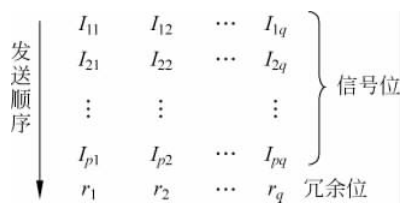


图 3-2　垂直奇偶校验

r_i 的编码规则为：

偶校验：$r_i=I_{1i}+I_{2i}+\cdots+I_{pi}$　　$(i=1,2,3,\cdots,q)$

奇校验：$r_i=I_{1i}+I_{2i}+\cdots+I_{pi}+1$　　$(i=1,2,3,\cdots,q)$

说明：式中的＋为模 2 加，即异或运算，只求本位和，进位丢弃。

图 3-2 中箭头给出了串行发送的顺序，即逐位先后次序为 $I_{11},I_{21},\cdots,I_{p1},r_1,I_{12},I_{22},\cdots,I_{p2},r_2,\cdots,I_{1q},I_{2q},\cdots,I_{pq},r_q$。在编码和校验过程中，用硬件方法或软件方法很容易实现上述连续半加运算，而且可以边发送边产生冗余位；同样，在接收方也可以边接收边进行校验后去掉校验位。

垂直奇偶校验方法能检测出每列中的所有奇数位错，但检测不出偶数位错。对于突发错误来说，奇数位错与偶数位错的发生概率接近于相等，因而对差错的漏检率接近于 1/2。

(2) 水平奇偶校验。为了降低对突发错误的漏检率,可以采用水平奇偶校验方法。水平奇偶校验又称为横向奇偶校验,它是对各个信息段的相应位横向进行编码,产生一个奇偶校验冗余位,如图 3-3 所示。

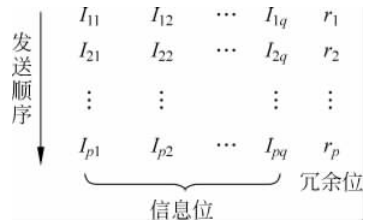


图 3-3 水平奇偶校验

r_i 编码规则为:

偶校验:$r_i = I_{i1} + I_{i2} + \cdots + I_{iq}$ $(i = 1,2,3,\cdots,p)$

奇校验:$r_i = I_{i1} + I_{i2} + \cdots + I_{iq} + 1$ $(i = 1,2,3,\cdots,p)$

若每个信息段就是一个字符的话,这里的 q 就是发送信息块中的字符数。

水平奇偶校验不但可以检测出各段同一位上的奇数位错,而且还能检测出突发长度$<p$ 的所有突发错误。按发送顺序,从图 3-3 中可以看出,突发长度$<p$ 的突发错误必然分布在不同的行中,且每行一位,所以可以检出差错,它的漏检率要比垂直奇偶校验方法低。但是实现水平奇偶校验时,不论采用硬件方法还是软件方法,都不能在发送过程中边发送边产生奇偶校验冗余位,而是必须等待要发送的全部信息块到齐后,才能计算冗余位,也就是说,需要使用能容纳整个数据块大小的缓冲区存放待发送数据,因此它的编码和检测实现起来都要复杂一些。

(3) 水平垂直奇偶校验。同时进行水平奇偶校验和垂直奇偶校验就构成了水平垂直奇偶校验,又称为纵横奇偶校验,如图 3-4 所示。

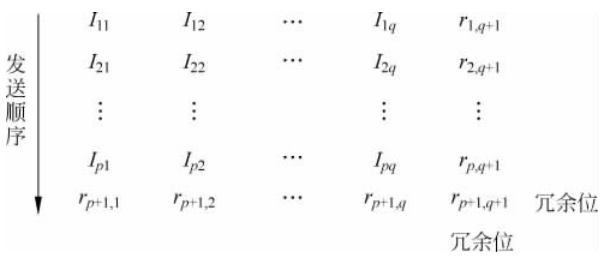


图 3-4 水平垂直奇偶校验

若水平垂直都采用偶校验,则 r_{ij} 的编码规则为:

$$r_{i,q+1} = I_{i1} + I_{i2} + \cdots + I_{iq} \qquad (i = 1,2,3,\cdots,p)$$

$$r_{p+1,j} = I_{1j} + I_{2j} + \cdots + I_{pj} \qquad (j = 1,2,3,\cdots,q)$$

$$r_{p+1,q+1} = r_{p+1,1} + r_{p+1,2} + \cdots + r_{p+1,q} = r_{1,q+1} + r_{2,q+1} + \cdots + r_{p,q+1}$$

水平垂直奇偶校验能检测出所有 3 位或 3 位以下的错误(因为此时至少在某一行或某一列上有一位错)、奇数位错、突发长度$\leqslant p+1$ 的突发错以及很大一部分偶数位错。测量表明,这种方式的编码可使误码率降至原误码率的百分之一到万分之一。水平垂直奇偶校验不仅可检错,还可用来纠正部分差错。例如,数据块中仅存在 1 位错时,便能确定出错码的位置就在某行和某列的交叉处,从而可以纠正它。

2) 循环冗余校验码

循环冗余校验方法是数据通信中差错检测的重要方法,它对随机错码和突发错码均能以较低的冗余度进行严格检查。其方法是:在发送方产生一个循环冗余校验码,附加在信

息位后面一起发送到接收方，接收方将收到的信息按发送方形成循环冗余校验码同样的算法进行校验以检测是否出错。

循环冗余校验码(Cyclic Redundancy Check，CRC)也称为多项式码，简称 CRC 校验码。其原理如下。

(1) 确定信息多项式 $M(x)$。将待发送的二进制位串看成是一个多项式的系数，该多项式称为信息多项式 $M(x)$。任何一个由二进制数位串组成的代码都可以和一个只含有 0 和 1 两个系数的多项式建立一一对应关系，一个 k 位数据帧可以看成是从 x^{k-1} 到 x^0 的 k 次多项式的系数序列，这个多项式的阶数为 $k-1$，最高位(最左边)是 x^{k-1} 项的系数，下一位是 x^{k-2} 的系数，以此类推。

例如，若信息位为 1011011(7 位)，则 $M(x)=1*x^6+0*x^5+1*x^4+1*x^3+0*x^2+1*x^1+1*x^0=x^6+x^4+x^3+x+1$；同样，若 $M(x)=x^5+x^4+x^2+1$，则对应的二进制位串为 110101。

(2) 确定一个素多项式 $G(x)$。$G(x)$ 又称为生成多项式，生成多项式的作用是和信息多项式进行计算产生余数多项式。生成多项式的最高位和最低位必须是 1。目前，国际标准中生成多项式有以下几类。

- CRC-12：$x^{12}+x^{11}+x^3+x^2+x+1$
- CRC-16：$x^{16}+x^{15}+x^2+1$
- CRC-CCITT-1：$x^{16}+x^{12}+x^5+1$
- CRC-32：$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

(3) 计算余数多项式 $R(x)$。设 $G(x)$ 为 r 阶，发送方计算 $x^rM(x)/G(x)$(模 2 除法)，得到余数多项式 $R(x)$，商舍掉。依据群论的相关理论，可以证明 $R(x)$ 具有发现错误的能力。

(4) 形成码元多项式 $C(x)$。发送方将 $R(x)$ 附在 $M(x)$ 之后，组成码元多项式 $C(x)$，然后将其发送出去。

(5) 接收方检验。当接收方收到码元多项式 $C'(x)$ 后，计算 $C'(x)/G(x)$(模 2 除法)，得到新的余数多项式 $R'(x)$。如果 $R'(x)=0$，则认为传输没有错误，否则，可以确定传输中产生了差错。

目前，CRC 校验已有成熟的硬件完成，因此校验速度很快，冗余度也不大，是应用最广泛的一种校验码。

设信息位为 m 位，生成多项式 $G(x)$ 为 r 阶，则计算 CRC 校验码的方法可以简化如下：

① 在信息码尾部附加 r 个 0，使其成为 $m+r$ 位二进制位串，即相应的多项式为 $x^rM(x)$；

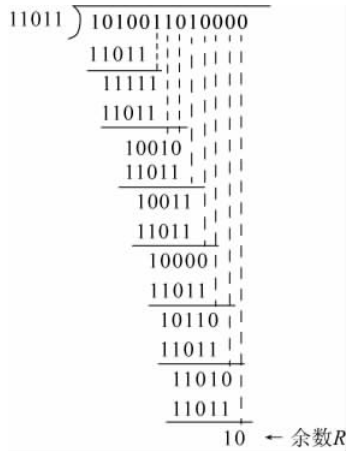
② 按模 2 除法用 $G(x)$ 对应的位串去除 $x^rM(x)$ 对应的位串，得到余数 $R(x)$ 所对应的位串；

③ 按模 2 加法从 $x^rM(x)$ 对应的位串中加上得到的余数 $R(x)$ 所对应的位串，结果就是要传送的带 CRC 校验码的数据。

例 3-1 设信息位 $M=101001101$，生成多项式 $G(x)=x^4+x^3+x+1$，试计算信息 M 的 CRC 校验码。

解：已知 $r=4$，生成多项式 $G(x)$ 对应的位串为 11011。$x^rM(x)$ 对应的位串为

1010011010000。利用短除法计算如下：



余数 R 为 0010(因为 $r=4$，所以余数 R 补足 4 位)，因此，信息 $M=101001101$ 的 CRC 校验码为：

$$1010011010000+0010=101001101\mathbf{0010}$$

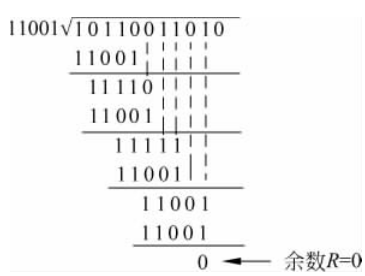
例 3-2 在数据传输过程中，若接收方收到发送方发送的信息为 10110011010，其生成多项式 $G(x)=x^4+x^3+1$，问接收方收到的数据是否正确？(写出判断依据和推演过程)

解：由题意知，在数据通信过程中采用的是循环冗余校验码(CRC 码)进行数据检错。发送方在发送的数据块中加入足够的冗余位以满足检错需要。

用数据多项式与生成多项式 $G(x)$ 进行运算得到校验和(余数)，将校验和附加在数据帧尾部，并使带有校验和的帧所对应的多项式能被 $G(x)$ 除尽，然后将带有校验和的数据帧发送出去，当接收方接收时，用 $G(x)$ 去除它，若余数为 0，则表示传输正确，否则表示传输出错。

本题中，如果接收信息/$G(x)$，余数为 0，则收到的数据正确，否则出错。

因为 $G(x)=x^4+x^3+1$，其对应的位串为 11001，所以 10110011010/11001 的模 2 除的推演过程如下：



10110011010/11001 的模 2 除的余数 $R=0$，因此，接收数据正确。

2. 纠错码

在数据通信过程中，解决差错问题的另一种方法就是在每个待发送的数据块上附加足够的冗余信息，如果出错，使接收方能够推导出发送方实际送出的应该是什么样的比特串。

海明码是一种可以纠正一位差错的编码。对于 m 位数据位(信息位)，若增加 r 位冗余

位(校验位),则组成总长度为 n 位($n=m+r$)的编码,称为 n 位码字。为了能纠正单比特错,m 和 r 之间应该满足一定的关系。

对于 m 位数据位,其有效码字有 2^m 个,对于每一个有效码字,均附加一个固定的 r 位的冗余位,形成一个特定的 $n(n=m+r)$位码字。当且仅当其中一位改变时,都可以形成 n 个无效但可以纠错的码字(知道出错的位置),即有 $n+1$ 个可识别的码字(1 个有效码字,n 个无效但可识别的码字)。

对于 2^m 个有效码字,共有 $2^m(n+1)$个可识别的码字,2^n 个可识别及不可识别的码字,因此有 $2^m(n+1)\leqslant 2^n$,将 $n=m+r$ 代入,得:

$$m+r+1\leqslant 2^r$$

为了纠正单比特错,m 和 r 应该满足上述关系式。

海明码的编码方法是将码字内的各位从最左边开始按顺序依次编号,第 1 位为 1,第 2 位为 2,……,第 n 位为 n,其中编号为 2^i 的位(1,2,4,8,…)为海明码的校验位,即海明码校验位不是附加在数据位的头或尾,而是分散在数据位中,分别占用 2^i 位置。其余位顺序填入 m 位数据。每个校验位的取值应使得包括自身在内的一些位的集合服从规定的奇偶性,因此,海明码利用的原理仍是奇偶检验原理。下面举例说明海明码的形成方法。

例如,设信息位 $m=7$,则由 $m+r+1\leqslant 2^r$,得 $r=4$,所以海明码长 $n=11$ 位,设海明码字为 $x1x2x3x4x5x6x7x8x9x10x11$,其中 $x1$、$x2$、$x4$ 和 $x8$ 为海明校验码。计算海明码的校验位方法可以采用如图 3-5 所示的形式,海明码校验位编码(校验表达式)计算表达式如下:

$$x1=x3+x5+x7+x9+x11$$
$$x2=x3+x6+x7+x10+x11$$
$$x4=x5+x6+x7$$
$$x8=x9+x10+x11$$

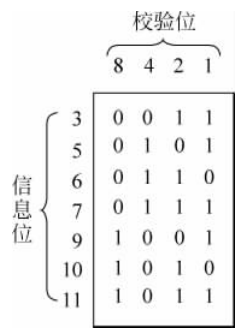


图 3-5　海明码校验码形成示意图

接收方验证收到的信息是否正确,采用的方法是重新计算海明码校验位 xi',但采用海明码监督表达式进行计算。海明码监督表达式如下:

$$x1'=x1+x3+x5+x7+x9+x11$$
$$x2'=x2+x3+x6+x7+x10+x11$$
$$x4'=x4+x5+x6+x7$$
$$x8'=x8+x9+x10+x11$$

若 $xi'=0(i=1,2,4,8,\cdots)$,则表示该信息传输正确,xi'中任何一位不为 0,则表示该信息传输出错,出错位为 xi'值,如 $xi'=110$,则表示出错位是 $x6$。如果信息位出错则需纠正,校验位出错则不需纠正。

例 3-3　假定传送信息位 M 为 1001011,求它的海明码。

解:已知信息位 M 的位数 $m=7$,设冗余位为 r 位,根据公式:

$$m+r+1\leqslant 2^r$$

计算得:$r=4$

海明码：**x1 x2** x3 **x4** x5 x6 x7 **x8** x9 x10 x11

x1 x2 1 **x4** 0 0 1 **x8** 0 1 1

计算海明码校验位：

$$x1 = x3 + x5 + x7 + x9 + x11 = 1 + 0 + 1 + 0 + 1 = 1$$
$$x2 = x3 + x6 + x7 + x10 + x11 = 1 + 0 + 1 + 1 + 1 = 0$$
$$x4 = x5 + x6 + x7 = 0 + 0 + 1 = 1$$
$$x8 = x9 + x10 + x11 = 0 + 1 + 1 = 0$$

所以，计算得到海明码为：**10**1**1**001**0**011。

例 3-4 假定传送信息位 M 为 8 位，接收方收到的信息为 110010100000，试判断该传输是否出错？如果出错是否需要纠正？并求出发送方发送的原始信息。

解：已知信息位 M 的位数 $m=8$，设冗余位为 r 位，根据公式：

$$m + r + 1 \leqslant 2^r$$

计算得：$r=4$

接收方收到的信息为 1 1 0 0 1 0 1 0 0 0 0 0

x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12

判断接收方收到的信息是否正确，需要根据海明码监督表达式计算 xi'，若 $xi'=0(i=1,2,4,8,\cdots)$，则表示该信息传输正确，否则出错。

海明码监督表达式计算如下：

$$x1' = x1 + x3 + x5 + x7 + x9 + x11 = 1 + 0 + 1 + 1 + 0 + 0 = 1$$
$$x2' = x2 + x3 + x6 + x7 + x10 + x11 = 1 + 0 + 0 + 1 + 0 + 0 = 0$$
$$x4' = x4 + x5 + x6 + x7 + x12 = 0 + 1 + 0 + 1 + 0 = 0$$
$$x8' = x8 + x9 + x10 + x11 + x12 = 0 + 0 + 0 + 0 + 0 = 0$$

因为得到的计算结果 $x8'\ x4'\ x2'\ x1'$ 为：0001，不为 0。因此，该传输存在错误。

出错位为 $x1$，而 $x1$ 是校验位，因此，不需纠错。

发送方发送的原始信息 01010000。

海明码属于分组码，分组码是一组固定长度的码组，一般用符号(n,k)表示，其中 n 是码组的总位数，又称为码组的长度(码长)，k 是码组中信息码元的数目，$r=n-k$ 为码组中的监督码元数目。通常用于前向纠错。

在采用纠错码或检错码的编码方案中，基本的数据处理单元通常称为码字(codeword)，由数据比特和冗余比特构成。两个码字之间对应比特取值不同的比特个数称为这两个码字的海明距离。例如，10101 和 00110 从第一位开始依次有第 1 位、第 4 位、第 5 位不同，则海明距离为 3。海明距离表明：假设两个码字的海明距离为 d，则需要 d 个比特差错才能将其中一个码字转换成另一个码字。

在一个有效编码集中，任意两个码字的海明距离的最小值(d_{min})称为该编码集的海明距离。一种编码的检错能力和纠错能力取决于它的海明距离。为了检测 d 个比特错，需要使用海明距离为 $d+1$ 的编码方案，因为在这种编码方案中，d 个单比特错不可能将一个有效码字改编成另一个有效码字。当接收方接收到一个无效码字时，就知道已经发生了传输错误。同样，为了纠正 d 个比特错，需要使用海明距离为 $2d+1$ 的编码方案。因为在这种编码方案中，合法码字之间的距离足够远，即使发生了 d 个比特错，仍然更接近于原始码字

而不是其他码字，从而可以唯一确定原来的码字以达到纠错的目的。

d_{min}与分组码的纠、检错能力存在以下关系：

- 当 $d_{min} \geqslant e+1$ 时，可检出 e 个错误；
- 当 $d_{min} \geqslant 2t+1$ 时，具有纠正 t 个错误的能力；
- 当 $d_{min} \geqslant t+e+1(e>t)$时，具有同时检出 e 个错误、纠正 t 个错误的能力。

3.3　数据链路层成帧机制

数据链路层采用一定的方法将比特流划分成离散的数据帧，这就是数据链路层的成帧机制。数据链路层常用的成帧方法有 3 种，即字符计数法、带填充字符的首尾界符法和带填充位的首尾标志法。

1. 字符计数法

字符计数法是在帧头部使用一个字段来标明帧内字符数，如图 3-6 所示。

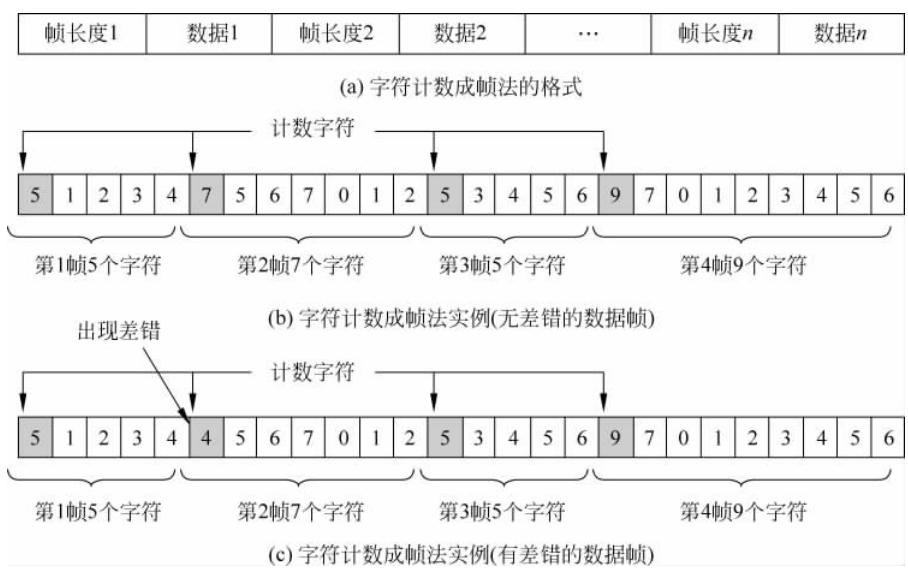


图 3-6　字符计数成帧法

说明：帧长度字段表示该帧所含的字节数，帧长度字符也包含在内。

字符计数法的工作原理是当接收方收到数据帧时，根据帧长度，即可知道帧的起始位和结束位。所面临的问题是当计数字符由于传输差错而发生变化时(在图 3-6(c)中，第 2 帧的计数字符由 7 变成了 4)，则其后所有帧的起始位和结束位均与发送方不一致，即接收方与发送方失去了同步，即使接收方通过校验和知道此帧出错，也无法确定下一帧从哪里开始，且无法向发送方请求重传，因为接收方无法确定应该回跳多少字符开始重传。计数字段一旦出错，将无法再同步，这是字符计数成帧法的致命缺点。因此，字符计数成帧法很少使用。

2. 带填充字符的首尾界符法

带填充字符的首尾界符法采用的方法是每一帧以 ASCII 字符序列 DLE STX 开头,以 DLE ETX 结束,DLE 代表 Data Link Escape,STX 代表 Start of Text,ETX 代表 End of Text。采用这种方法,接收方主机一旦丢失帧边界,它只需查找 DLE STX 或 DLE ETX 字符序列就可以找到它所在的位置,即以特定的字符序列为控制字段,避免了出错后再同步的问题。

当传送的是目标程序或浮点数据的二进制数据时,DLE STX 或 DLE ETX 可能会出现在用户的数据中,因此会使接收方误认为帧开始或帧结束而产生错误。解决方法是字符填充法,就是发送方的数据链路层在用户数据中出现的每一个 DLE 字符前再插入一个 DLE 字符,接收方的数据链路层将数据中两个连续的 DLE 字符的第一个去掉,恢复数据的原始状态。如果只有单个 DLE 字符出现,就可以断定是帧的边界,因为数据中的 DLE 是成对出现的,如图 3-7 所示。带填充字符的首尾界符法缺点是依赖于字符集、不通用、扩展性差,这种方法现在也基本不再使用。

图 3-7 带填充字符的首尾界符成帧法

3. 带填充位的首尾标志法

带填充位的首尾标志法允许数据帧包含任意个数的比特,采用统一的帧格式,以特定的位序列进行帧的同步和定界。

带填充位的首尾标志法工作原理如下。

(1) 帧的开始:特定位模式,即 01111110,称为帧开始标志字节。

(2) 帧的结束:特定位模式,即 01111110,称为帧结束标志字节。

(3) 工作原理:为了解决透明传输比特,采用 0 比特插入技术,即发送方的数据链路层在数据段中遇到 5 个连续 1 时,自动在其后插入 1 个 0,这就是所谓的位填充技术。当接收方收到连续 5 个 1,且后面跟着 1 个 0 时,自动将此 0 删去。如果接收方收到连续 5 个 1,且后面跟着的还是 1 时,则再看下一位,如果下一位是 0,则认为是帧结束标志字节,如果下一位是 1,则可断定是数据出错了,因为数据中不可能连续出现 7 个 1。

带填充位的首尾标志成帧法如图 3-8 所示。对于通信双方计算机的网络层来说,位填充技术和字符填充技术都是透明的。

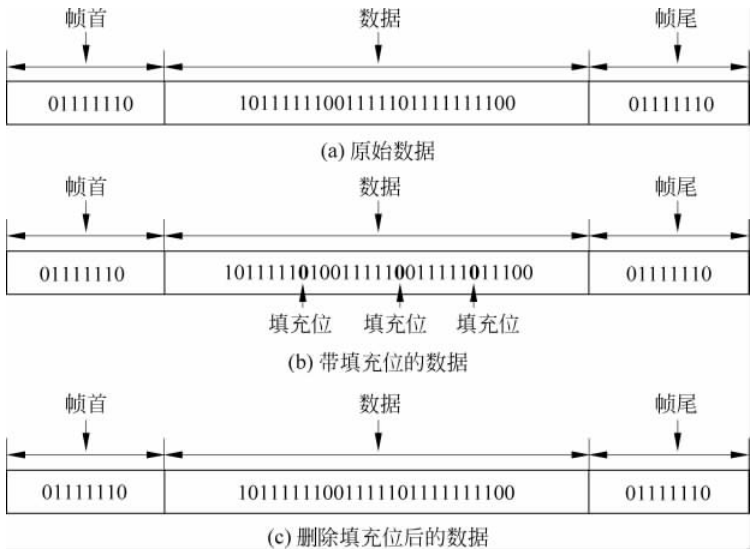


图 3-8　带填充位的首尾标志成帧法

3.4　数据链路层协议机制

数据链路层协议机制是指链路及协议的基本机制，即数据无差错地传输。因为链路级流量及差错控制决定了通信链路和网络的性能，所以如何能够使数据无差错地传输到对方，是数据链路层协议机制研究的核心问题。链路级的流量及差错控制技术通常有 3 种，即停止—等待 ARQ 协议、后退 N 帧 ARQ 协议，选择重传 ARQ 协议。其中后退 N 帧 ARQ 协议和选择重传 ARQ 协议采用了滑动窗口技术。

3.4.1　停止—等待 ARQ 协议

停止—等待协议(stop-and-wait)是最简单但也是最基础的数据链路层协议。在介绍停止—等待 ARQ 协议之前，先来看两种理想信道的数据传输。

数据传输环境为主机 A 向主机 B 点对点传输数据。

1. 一种无限制、理想化的数据传输

为了说明数据链路层协议的作用，先做以下两个假定。

假定 1：物理链路是理想的传输信道，所传送的任何数据既不会出现差错也不会丢失。

假定 2：发送方和接收方一直处于就绪状态，缓冲空间无限大，不管发送方以多快的速率发送数据，接收方总能及时接收并上交主机。即接收方向主机交付数据的速率永远不会低于发送方发送数据的速率(或者说接收方不会被发送方过快的发送速率淹没)。

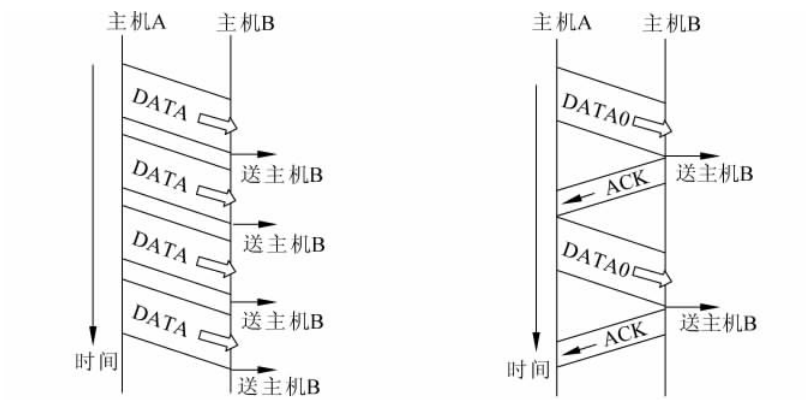
显然，在这种无限制、理想化的信道上传输数据永远都会准确及时到达。因此，在这种无限制理想信道上传输数据是不需要数据链路层协议的，但同时这种无限制理想信道也是不存在的。

2. 具有最简单流量控制的数据链路层协议

既然无限制的理想信道不存在,现在进一步再考虑以下情况:

保留第 1 个假定,去掉第 2 个假定,即主机 A 向主机 B 传输数据的信道仍然是无差错的理想信道,数据链路之间的交互信道也从不会损坏,数据传输过程中既不会出错也不会丢失,但是不能保证接收方向主机交付数据的速率永远不低于发送方发送数据的速率,也就是说如何防止发送方发送数据过快,以致接收方来不及处理是传输中需要解决的问题。为了使接收方的接收任何情况下都不会溢出,通常的解决方法是:接收方及时向发送方提供一个反馈,发送方根据接收方的反馈信息确定自己的发送速度。当接收方把数据帧正确交给主机 B 的网络层后,及时向发送方回复一个确认帧,表示已正确收到某一数据帧,允许发送方继续发送下一数据帧;发送方每发送完一个数据帧后,则等待一段时间直到收到对该数据帧的确认帧。

这种由接收方来控制发送方发送数据的速度是计算机网络中流量控制的一个基本方法。具有最简单流量控制的数据链路层协议就是发送方每发送一帧就暂停,等待接收方的确认,如图 3-9 所示。



(a) 不需要任何数据链路层协议的数据传输　(b) 具有最简单流量控制的数据链路层协议

图 3-9　具有最简单流量控制的数据链路层协议

3. 停止—等待 ARQ 协议

具有最简单流量控制的数据链路层协议是在保留第 1 个假定的情况下完成的,但是第 1 个假定通常也是不存在的,即无差错的理想信道通常是不存在的,数据在信道中传输出错是很有可能的。因此,下面将研究去掉上述两个假定的数据传输问题。在有差错的信道中传输数据,且不能保证接收方向主机交付数据的速率永远不低于发送方发送数据的速率。这里需要解决的问题是差错控制和流量控制。

自动重传请求(ARQ)协议是应用最广泛的一种差错控制技术,它包括对无错接收的 PDU 的肯定确认和对未确认的 PDU 的自动重传。ARQ 协议实现的前提条件是:

- 一个单独的发送方向一个单独的接收方发送信息;
- 接收方能够向发送方发送确认帧;

- 信息帧和确认帧都包含检错码；
- 发生了错误的信息帧和确认帧将被丢弃。

1）停止—等待 ARQ 协议的原理

在实际的数据传输过程中，由于传输信道不理想和外界干扰的存在，出现传输差错是不可避免的。传输差错可能导致数据帧或确认帧错误、数据帧或确认帧丢失，致使发送操作不能继续进行，或接收方重复接收数据等情况发生。数据传输中可能出现的几种情况如图 3-10 所示。

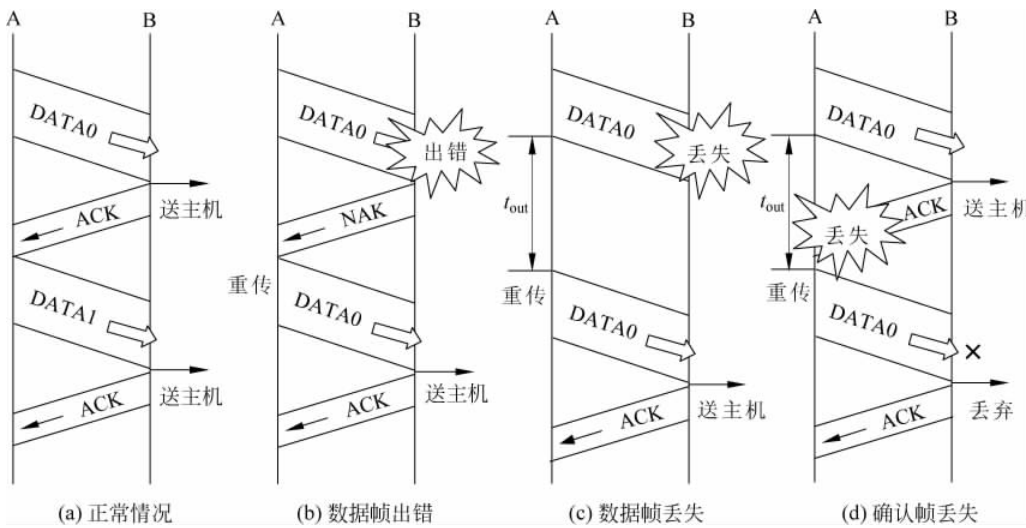


图 3-10　数据传输的几种情况

图 3-10(a)表示正常情况，数据传输过程中未出现任何差错。当接收方收到一个正确的数据帧后立即交付给主机 B，同时向主机 A 发送一个确认帧(ACK)，当主机 A 收到确认帧(ACK)后再发送一个新的数据帧，实现了接收方对发送方的流量控制。

如果传输过程中出现了差错，则可能出现图 3-10(b)、图 3-10(c)、图 3-10(d)的情况。

图 3-10(b)表示接收方主机 B 收到的数据帧 DATA0 经过 CRC 检测出错，则主机 B 丢弃该数据帧，同时给主机 A 发送一个否认帧(NAK)，希望主机 A 重传该数据帧。如果多次出现差错，主机 A 则需要多次重传该数据帧，直到从主机 B 接收到确认帧(ACK)为止。在此之前，主机 A 的缓冲区将一直保存着 DATA0 的副本。但如果通信线路质量太差，主机 A 在重传一定次数(事先约定好)后仍收不到主机 B 的确认帧(ACK)，则不再重传，而是将该情况向上一层报告。

图 3-10(c)表示接收方主机 B 未收到主机 A 发送的数据帧(数据帧丢失)的情况，此时主机 A 将永远接收不到主机 B 发送的确认帧(ACK)，而主机 A 会一直等待下去，发生了死锁现象。为了避免死锁问题，主机 A 会采用超时重传技术，所谓超时重传技术就是主机 A 每发送完一个数据帧，就启动一个超时计时器(又称定时器)，若到了超时计时器所设置的重传时间 t_{out}，主机 A 仍收不到主机 B 的确认帧或否认帧，则主机 A 会自动重传该数据帧，这样就避免了由于数据帧丢失造成的死锁现象的发生。一般可将重传时间设为“从发完数据帧到收到确认帧所需的平均时间”的 2 倍，重传若干次后仍不能成功，则报告差错。

图 3-10(d)与图 3-10(c)相似,主机 B 收到了主机 A 发送的数据帧 DATA0,但由于主机 B 发送给主机 A 的确认帧(ACK)丢失,对主机 A 来说同样会超时重传,但接收方主机 B 将再次收到重传的 DATA0 数据帧,产生了重复帧问题。为了解决重复帧问题,发送方为每个数据帧带上不同的发送序号,每发送一个新的数据帧就按一定规则修改它的发送序号。若主机 B 收到发送序号相同的数据帧,表示出现了重复帧,则应丢弃该重复帧,但此时主机 B 仍须向主机 A 发送确认帧(ACK),因为主机 A 并不知道主机 B 是否已收到 DATA0 帧,以保证协议正常执行。

发送序号所占的比特数是有限的,经过一段时间后,发送序号就会重复。而序号占用的比特数越少,数据传输的额外开销也就越小。因此,既要解决序号重复问题,又要尽量减少数据传输的额外开销,需要对发送序号所占用的比特数给出一个适当值。对于停止—等待 ARQ 协议,由于每发送一个数据帧就暂停,等待应答。也就是说,数据链路中同一时刻只有一个数据帧存在,因此用一个比特来编号就可以了。一个比特可表示 0 和 1 两种不同的序号。数据帧中的发送序号 N(S)以 0 和 1 交替的方式出现在数据帧中。每发送一个新的数据帧时,发送序号只要与上次发送的不一样即可,这样接收方就能够区分出是新的数据帧还是重传的数据帧了。

停止—等待 ARQ 协议的优点是比较简单。缺点是通信信道的利用率不高,即信道远远不会被数据比特填满。虽然物理层在传输比特时会出差错,但由于数据链路层的停止—等待 ARQ 协议采用了有效的检错重传机制,数据链路层对上面的网络层可以提供无差错的可靠传输服务。

2) 停止—等待 ARQ 协议算法

停止—等待 ARQ 协议算法分为发送方算法和接收方算法两类,算法如下。

(1) 发送方算法。

① $V(S)=0$(发送状态变量初始化)。

② 从主机取出一个数据帧→将数据帧送入发送缓冲区,同时,$N(S)=V(S)$(将发送状态变量值写入数据帧的发送序号)。

③ 从发送缓冲区取出数据帧并发送→启动定时器(设置重传时间 t_{out})。

④ 等待应答:转入⑤或⑥。

⑤ 若定时器时间未到:收到应答帧(ACK 或 NAK)。

- 肯定应答(收到 ACK):$V(S)=1-V(S)$,转入②,发送下一帧。
- 否定应答(收到 NAK):转入③,重传该数据帧。

⑥ 若定时器时间已到:未收到应答帧,则转入③,重传该数据帧。

⑦ 若重传次数<设定值,重传该帧;否则,信道故障,通信终止。

(2) 接收方算法。

① $V(R)=0$(接收状态变量初始化,数值等于待接收数据帧的发送序号)。

② 等待接收数据帧。

③ 接收到数据帧,进行差错校验。

- 数据帧正确:转入④。
- 数据帧出错:丢弃该数据帧,发送 NAK,转入②。

④ 重复帧判断:判断 $N(S)=V(R)$?(是否是希望接收的数据帧序号)

- 是：不是重复帧，接受该数据帧，并将收到的数据帧中的数据部分交付上一层，更改欲接收的新数据帧序号 $V(R)=1-V(R)$。
- 否：是重复帧，丢弃该数据帧。

⑤ 发送确认帧(ACK)应答，转入②。

停止—等待 ARQ 协议算法流程图如图 3-11 所示。

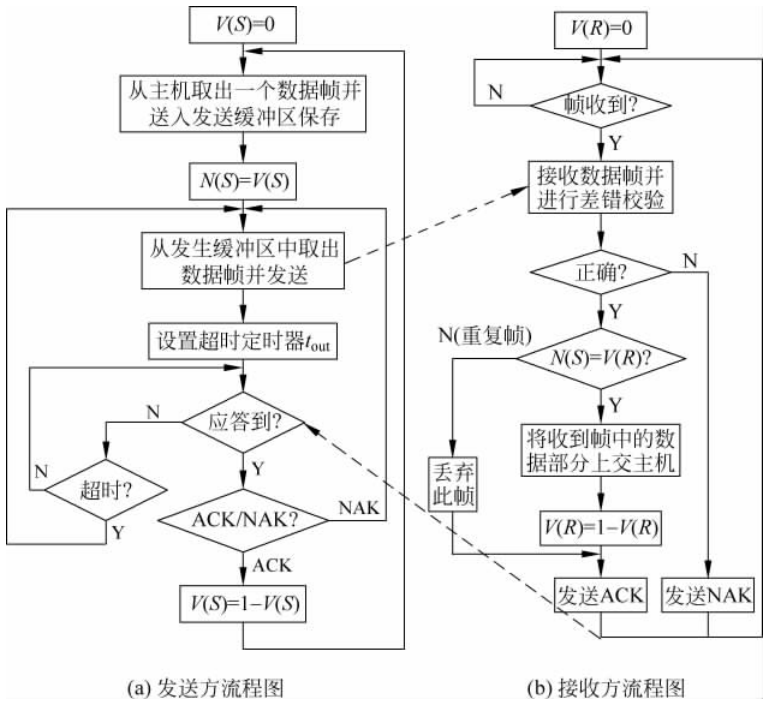


图 3-11　停止—等待 ARQ 协议算法流程图

例 3-5　若信道速率为 4kbps，采用停止—等待协议，传播时延 $t_p=20$ms，确认帧长度和处理时间均可忽略。问帧长为多少才能使信道利用率至少达到 50%。

解：停止—等待协议中的时间关系如图 3-12 所示。

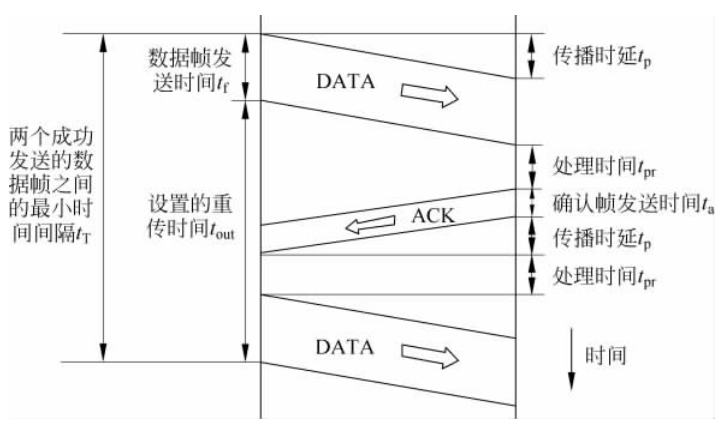


图 3-12　停止—等待协议中数据帧和确认帧的发送时间关系

在确认帧长度和处理时间均可忽略的情况下,要使信道利用至少达到50%,必须使数据帧的发送时间等于2倍的单程传播时延,即 $t_f = 2t_p$。

已知:帧长、信道速率和数据帧发送时间关系为

$$t_f = l_f / C$$

其中,C 为信道容量(又称为带宽或信道速率),l_f 为帧长(单位为比特)。

所以,求得数据帧长:

$$l_f = C \times t_f \geqslant C \times t_p = 4000 \times 0.04 = 160(\text{b})$$

故,帧长为160b才能使信道利用率至少达到50%。

3.4.2 滑动窗口协议

停止—等待ARQ协议要求任一时刻只能传输一帧,当信道很长时,就会出现利用率严重低下的情况,为了提高信道的有效利用率,则做如下改进:允许发送方连续发送 N 帧,让发送的数据帧在信道上按前后次序排列起来,并同步向前推进,犹如一条流水的管道,故又称为管道技术。允许连续发送或接收数据帧的范围称为滑动窗口,N 称为窗口的大小。当 $N=1$ 时,就是停止—等待ARQ协议。

滑动窗口协议的基本原理就是在任意时刻,发送方都维持一个允许发送的连续帧的序号范围,称为发送窗口 W_T;同时,接收方也维持一个允许接收的连续帧的序号范围,称为接收窗口 W_R。发送窗口和接收窗口序号的上下界不一定相同,甚至大小也可以不同。发送方窗口内的序列号代表已经被发送,但是还没有被确认的帧,或允许被发送(尚未发送)的帧,接收方窗口内的序列号代表准备接收的帧。

下面以一个例子(假设发送窗口尺寸为2,接收窗口尺寸为1)为例说明滑动窗口的工作过程,如图3-13所示。

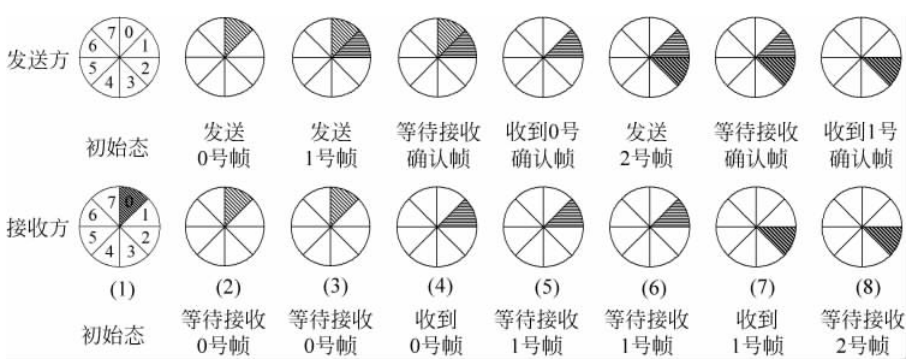


图3-13 滑动窗口示意图

如图3-13所示的滑动窗口工作过程如下。

(1) 初始态:发送方没有帧发出,发送窗口前后沿重合。接收方0号窗口打开,等待接收0号帧。

(2) 发送方发送0号帧:发送方打开0号窗口,等待接收0号帧的确认帧。接收窗口状态不变(等待接收0号帧)。

(3) 发送方发送1号帧:发送方打开0号和1号窗口,等待接收0号和1号帧的确认

帧。此时，发送方打开的窗口数已达规定限度，在未收到新的确认返回帧之前，发送方暂停发送新的数据帧。此时接收窗口状态不变(等待接收 0 号帧)。

(4) 接收方收到 0 号帧：接收方收到 0 号帧并检验正确，同时发送 0 号确认帧，接收方关闭 0 号窗口，打开 1 号窗口，等待接收 1 号帧。此时发送窗口状态不变。

(5) 发送方收到 0 号确认帧：发送方正确收到接收方发来的 0 号帧确认帧，发送方关闭 0 号窗口，表示从重发表(缓冲区)中删除 0 号帧。此时接收窗口状态不变(等待接收 1 号帧)。

(6) 发送方发送 2 号帧：发送方打开 2 号窗口，发送 2 号帧，并等待 2 号确认帧。此时，发送方打开的窗口又已达规定限度，在未收到新的确认帧之前，发送方暂停发送新的数据帧。此时接收窗口状态不变(等待接收 1 号帧)。

(7) 接收方接收 1 号帧：接收方正确收到 1 号帧，发送 1 号确认帧，关闭 1 号窗口，打开 2 号窗口，准备接收 2 号帧。此时发送窗口状态不变。

(8) 发送方接收 1 号确认帧：发送方正确收到 1 号确认帧，发送方关闭 1 号窗口，表示从重发表(缓冲区)中删除 1 号帧。此时接收窗口状态不变。

若从滑动窗口的观点来统一看待停止—等待 ARQ 协议、后退 N 帧 ARQ 协议及选择重传 ARQ 协议，它们的差别仅在于各自窗口尺寸的大小不同而已。

停止—等待 ARQ 协议：发送窗口＝1，接收窗口＝1；

后退 N 帧 ARQ 协议：发送窗口＞1，接收窗口＝1；

选择重传 ARQ 协议：发送窗口＞1，接收窗口＞1。

1. 后退 N 帧 ARQ 协议

后退 N 帧 ARQ 协议是指发送方发送完一个数据帧后，不必停下来等待接收方的应答，可以连续发送若干帧；若在发送过程中收到接收方的肯定应答，则可以继续发送，若收到对其中某一帧的否认应答，则重发否认帧开始的其后所有的后续帧，即后退 N 帧 ARQ(Go Back-N ARQ)。

后退 N 帧 ARQ 协议的简单工作过程如图 3-14 所示，假定数据帧 DATA2 传输出错。

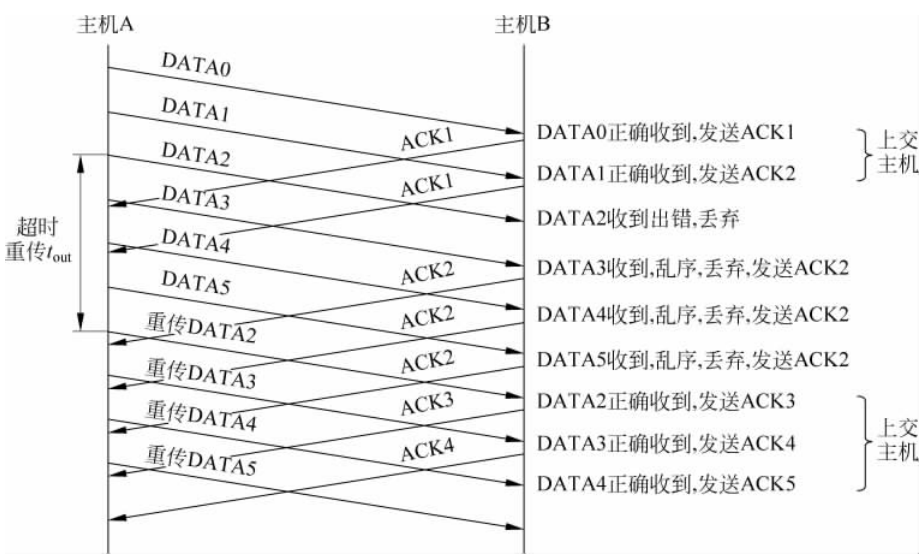


图 3-14　后退 N 帧 ARQ 协议的简单工作过程

图 3-14 所示的后退 N 帧 ARQ 协议工作原理如下。

- 接收方只按序接收数据帧。虽然在有差错的 2 号帧之后接着又收到了正确的 3 个数据帧,但接收方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,但仍重复发送已发送过的最后一个确认帧(防止确认帧丢失)。
- ACK1 表示确认收到 0 号帧(DATA0),并期望下次收到 1 号帧;ACK2 表示确认收到 1 号帧(DATA1),并期望下次收到 2 号帧;以此类推。在协议的流量控制方式中,确认序号 $N(R)$一般表示接收方希望接收的下一帧序号,同时也对 $N(R)-1$ 帧及其以前各帧的接收确认。
- 主机 A 每发送完一个数据帧时都要设置该帧的超时计时器。如果在超时时间内收到该帧的确认帧,就立即将超时计时器清零,继续发送后续的数据帧。若在所设置的超时时间结束时仍未收到该帧的确认帧,则重传相应的数据帧,此时需重新设置超时计时器。
- 在重传 2 号数据帧时,虽然主机 A 已经发送完了 5 号帧,但仍必须将 2 号帧及其以后的各帧全部进行重传。这就是后退 N 帧的 ARQ 协议的含义,当出现差错必须重传时,要向后退 N 个帧,然后再开始重传。

从以上描述可以看出,在后退 N 帧 ARQ 协议中,如果发送方一直没有收到对方的确认,那么它不能无限制地发送其他帧。因为当未被确认数据帧的数目太多时,只要有一帧出现差错,就会有很多数据帧需要重传,必然白白花费较多的时间,因而增大了网络开销。而且对所有发送出去的大量数据帧都要进行编号,每个数据帧的发送序号编码也会占用较多的位数,这样又增加了一些开销。因此,在后退 N 帧 ARQ 协议中,应当对已发送但尚未确认的数据帧数目加以限制。采取的措施就是使用发送窗口来对发送方进行流量控制。

在后退 N 帧 ARQ 协议中,同时会有多个等待确认的帧,因此,逻辑上需要多个超时计时器,每一个等待确认的帧都需要一个计时器,它们是相互独立各不相关的。对后退 N 帧 ARQ 协议的修改及其工作流程如下。

(1) 发送方。

① 每发送一帧,都启动超时计时器,不等待接收应答帧,连续发送后续的若干帧,即使在发送过程中收到肯定应答也不停止发送;

② 若收到对编号为 i 的帧的肯定应答帧 ACK(i),则登记;

③ 对编号为 i 的帧,若收到否定应答帧 NAK(i),或超时未收到应答,则将发送指针调整为 i,从帧 i 开始,按步骤①的方式重新开始发送。

(2) 接收方。

① 对接收到的数据帧进行检错、排序;

② 若接收到的帧编号不在接收窗口中或收到重复帧,则丢弃;

③ 若收到错误帧(位错),则发送否定应答帧 NAK(i);

④ 将收到的正确帧保存到相应编号的缓冲区中(同时实现了排序),发送肯定应答帧 ACK(i)。

假设一次可连续发送的帧数为 N(发送窗口 W_T),在发送完 N 帧后等待接收应答;收到否定应答 NAK(i)或超时后,发送窗口调整为 $i\sim i+N-1$,重发窗口中的若干帧。发送

方和接收方可以分别独立设置发送窗口 W_T 和接收窗口 W_R 的大小。

- 发送方设置发送窗口 W_T。发送窗口用来对发送方进行流量控制。发送窗口的大小 W_T 是指在未收到对方确认帧时发送方最多可以发送多少个数据帧，如图 3-15 所示。

图 3-15　后退 N 帧 ARQ 协议发送窗口的变化

- 接收方设置接收窗口 W_R。在后退 N 帧 ARQ 协议中，接收窗口的大小 $W_R=1$。只有当收到的帧的序号与接收窗口一致时才能接收该帧。否则，就丢弃它。接收方每收到一个序号正确的帧时，接收窗口 W_R 就向前(向右方)滑动一个帧的位置。同时发送对该帧的确认。后退 N 帧 ARQ 协议可以采用累积确认方法。

后退 N 帧 ARQ 协议接收窗口 W_R 的变化如图 3-16 所示。

图 3-16　后退 N 帧 ARQ 协议接收窗口变化示意图

滑动窗口的重要特性主要有以下几个方面。

- 只有在接收窗口向前滑动时(与此同时也发送了确认帧),发送窗口才有可能向前滑动。收发双方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。
- 当发送窗口和接收窗口的大小都等于1时,就是停止—等待 ARQ 协议。
- 发送窗口的最大值:当用 n 个比特进行编号时,则只有在发送窗口的大小 $W_T \leqslant 2^n - 1$ 时,后退 N 帧 ARQ 协议才能正确运行。

连续发送数据帧提高了信道的利用率,但后退 N 帧协议又导致某些已正确接收的数据帧也会重传,反过来又降低了发送效率。因此,后退 N 帧 ARQ 协议适用于误码率较低的环境,此时,后退 N 帧 ARQ 协议优于停止—等待 ARQ 协议;反之则不一定。

例 3-6 一个 3000km 的 T1 干线被用来传送采取后退 N 帧重传滑动窗口协议的长度都是 64B 的数据链路帧。如果传播速度是 6μs/km,那么序列号应该是多少位?

解:为了有效运行,序列空间(实际上就是发送窗口大小)必须足够大,以允许发送方在收到第1个确认应答之前可以不断发送。

由题意,得传播时间为

$$6 \times 3000 = 18\,000(\mu s) = 18ms$$

因为 T1 速率为 1.544Mbps,因此,发送 64B 的数据帧需花费时间为

$$64 \times 8 \div 1.544 \approx 333(\mu s) = 0.333ms$$

因此,第一个帧从开始发送起,18.3ms 后完全到达接收方。确认应答又花了回程 18ms 加上很少的(可以忽略)发送时间,就可以完全收到。

这样,加在一起的总的时间是 36.3ms。发送方应该有足够的窗口空间,从而能够连续发送 36.3ms。

为充满整个传输信道所需要的数据帧数为

$$36.3 \div 0.3 = 121$$

又 $121 \leqslant 128 = 2^7$,因此,序列号应该是 7 位。

2. 选择重传 ARQ 协议

在后退 N 帧 ARQ 协议中,接收方若发现错误帧就不再接收后续的帧,即使是正确到达的帧也会被丢弃,这显然是一种浪费。因此,另一种效率更高的改进策略是接收方开辟一个大小适当的缓冲区,当接收方发现接收的某数据帧出错后,发送对该数据帧的否认帧,而对其后继续到来的数据帧正常接收检验,因为不能立即上交给接收方的高层,所以将存放在缓冲区中,等待发送方重新传送出错的那一帧。一旦正确收到重传的出错帧后,就将已存放在缓冲区中的后续帧一起按正确的顺序递交高层,且只对最高序号的帧进行确认,这种方法就称为选择重传 ARQ 协议(Selective Repeat ARQ,SR ARQ)。显然,选择重传 ARQ 协议在某帧出错时减少了后面所有帧都需要重传的浪费,但对接收方提出了更高的要求,接收方要开辟一个足够大的缓冲区来暂存未按顺序正确接收到的帧。选择重传 ARQ 协议的工作过程如图 3-17 所示,假定 2 号数据帧出错。

下面对选择重传 ARQ 协议的发送窗口及接收窗口进行简单讨论。

(1) 发送窗口 W_T。发送窗口 W_T 的大小表示在未收到对方确认应答帧之前,发送方可

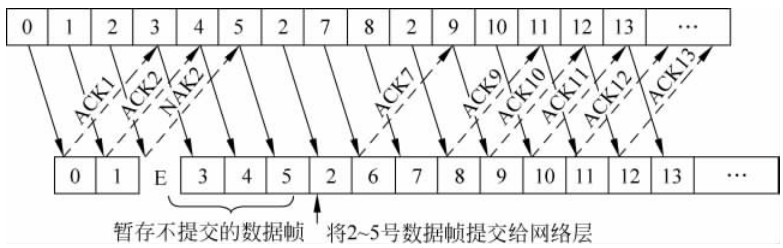


图 3-17　选择重传 ARQ 协议的工作过程

以连续发送数据帧的最大数目，且只有序号在窗口内的帧才可以发送。

（2）接收窗口 W_R。接收窗口 W_R 的大小表示接收方可以连续接收数据帧的最大数目。接收方仅在收到的数据帧的发送序号落入接收窗口内的情况下才接收该数据帧，若接收到的数据帧的发送序号落在接收窗口之外，则一律丢弃。

假定用 n 比特对滑动窗口编号，则要求下式成立：

$$W_T + W_R \leqslant 2^n \qquad (n \text{ 为序号的位数})$$

因为接收窗口 W_R 最少为 1，所以发送窗口 W_T 的最大值为：

$$W_T \leqslant 2^n - 1$$

但一般要求发送窗口 W_T 最大值不超过总窗口大小的一半，即 $W_T \leqslant 2^n/2$，原因是可能会产生编号回绕问题，但也与具体的应答方式有关。

例如，假设帧的序列号位数 $n=3$，若发送窗口 W_T 取最大值 $N=2^3-1=7$ 帧，接收方的接收窗口 $W_R=7$，现在发送方连续发送了序号为 0～6 号数据帧（共 7 帧），接收方正确接收了 0～6 号数据帧，若出现图 3-18(d)中的情况：接收方发送了对 0～6 号数据帧的确认帧，但却丢失了。这时，发送窗口就会发生错误，错误窗口示意图如图 3-18 所示。

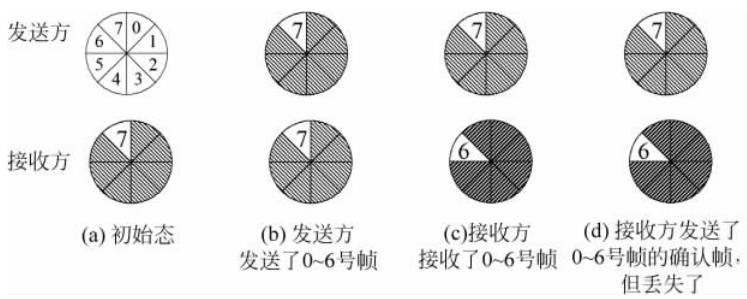


图 3-18　错误窗口示意图

在图 3-18 中，当接收方正确接收了 0～6 号帧后，立即发送了 0～6 号帧的确认帧，且接收窗口整体向前移动了一个位置，等待接收 7、0、1、2、3、4、5 帧。但是，接收方发给发送方的确认帧在返回过程中丢失了，因此发送窗口并不改变，仍然是保存 0～6 号帧，如图 3-18(d)所示，等待接收确认帧。最后当发送方的 0 号帧的 t_{out} 到达时，发送方就会重发保存在缓存中的 0～6 号帧（重复发送），而此时接收窗口是 7、0～5，因此当重发帧到达接收方的时候，经检查帧的发送序号落在接收窗口内，重发帧被当做新帧接收，发生了错误窗口问题。导致这一问题的原因是，在接收窗口向前滑动时，新窗口中的帧序号和旧窗口的帧序号重叠了，致使接收方无法区分接收的帧是重发帧（如果确认帧丢失）还是新帧（如果确认帧被收到）。

解决这一问题的关键是保证不出现上述的窗口重叠现象,采用的方法是接收窗口和发送方的最大窗口设置为相等,且小于等于序号范围的一半,即 $W_T \leqslant 2^n/2$,图 3-18 中 $W_T=4$。

例 3-7 在选择重传 ARQ 协议中,假设序列号为 3 位,发送窗口 $W_T=6$,接收窗口 $W_R=3$。试找出一种情况,使得在此情况下协议不能正确工作。

解:假定发送端发送完 0～5 号共 6 个数据帧。因发送窗口已满,发送暂停。再假定 6 个数据帧中的 0 号帧正确到达接收方,1 号帧丢失,并且随后的 2、3、4 和 5 号帧均正确到达接收方,那么接收方在把 0 号帧提交给上层协议模块后,因需要等待接收发送方重传的 1 号帧,必须缓存正确接收的 2、3、4 和 5 号帧。然而由于 $W_R=3$,接收方没有足够容量的缓存空间同时存储这 4 个帧,只能把最后到达的 5 号帧丢弃。这种情况的发生,表明在选择重传 ARQ 协议中,若序列号为 3 位,发送窗口 $W_T=6$,接收窗口 $W_R=3$,协议不能正确工作。

3.5 局域网协议

局域网(Local Area Network,LAN)是指在一个局部的地理范围内(如一个学校、工厂和机关内),将各种计算机、外部设备和数据库等互相连接起来组成的计算机通信网。局域网通常是封闭型的,可以由办公室内的两台计算机组成,也可以由一个公司内的上千台计算机组成。但它可以通过数据通信网或专用数据电路,与远方的局域网、数据库或处理中心相连接,构成一个大范围的信息处理系统。局域网可以实现硬件资源(如服务器、打印机、扫描仪等)共享和软件资源(应用软件、文件管理等)共享,还可以实现办公自动化(如工作组内的日程安排、电子邮件和传真通信服务)等。

3.5.1 局域网体系结构

电子和电气工程师协会(Institute of Electrical and Electronics Engineers,IEEE)于 1980 年 2 月成立了局域网标准化委员会,该委员会制定了局域网标准,称为 IEEE 802 标准。IEEE 802 标准中规定局域网体系结构由物理层和数据链路层组成,且数据链路层又分为介质访问控制(Media Access Control,MAC)子层和逻辑链路控制(Logical Link Control,LLC)子层,其结构如图 3-19 所示。图中的○是服务访问点(SAP)。

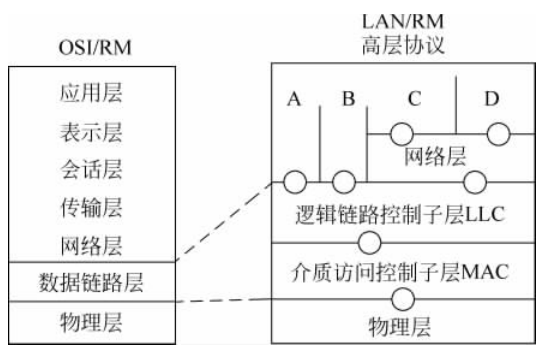


图 3-19 IEEE 802 局域网体系结构

局域网组网的一个显著特点是网上所有计算机使用一条共享信道进行广播式通信，这是与点对点链路组成的广域网通信方式的重要区别。因此，局域网协议需要解决的一个重要问题就是网上多个节点如何接入一条共享信道，即介质访问控制（MAC 访问控制）问题。

在 IEEE 802 标准中，为了使数据帧的传送独立于所采用的物理介质和介质访问控制方法，数据链路层划分为 LLC 和 MAC 两个子层，其中 LLC 定义的是传输中与具体网络无关的部分，即 LLC 子层与介质无关；而所有与传输介质相关的部分都集中在 MAC 子层，即仅让 MAC 子层依赖于物理介质。也就是说 LLC 子层隐藏了各种 IEEE 802 网络的差异，向网络层提供统一的帧格式和接口，提供面向连接和无连接的服务、差错控制和流量控制。这也是 IEEE 802 局域网的数据链路层划分为 LLC 和 MAC 两个子层的一个重要原因。

LLC 子层与 MAC 子层的区别在于，在 LLC 子层上看不到具体的局域网，局域网对 LLC 子层是透明的，只有在 MAC 子层才能看见所连接的局域网采用的是什么标准。

IEEE 802 局域网数据链路层通信过程除了 MAC 地址外还定义了 SAP 地址。SAP 地址就是 LLC 服务访问点，作为 LLC 子层的地址，提供对网络层的接口，标识网络层的通信进程。因此，IEEE 802 局域网中的寻址分为两步：第一步利用 MAC 帧的 MAC 地址信息找到网络中的某一个节点；第二步利用 LLC 帧的 SAP 地址找到该节点中高层的某一个进程。

局域网中物理层和数据链路层的主要功能有以下几个方面。

（1）物理层完成信号的编码与译码、产生和去除双方同步所使用的前同步码（又称前导码或前缀）、比特在介质中的发送与接收等功能。

（2）MAC 子层完成成帧、寻址，实现介质访问控制和 CRC 差错检验等功能。

（3）LLC 子层完成建立和释放数据链路层的逻辑连接、提供与高层的接口、差错控制、给帧加上序号等功能。

目前，IEEE 802 已经公布的标准主要有：

- IEEE 802.1　概述、LAN 体系结构和网络互联，以及网络管理和性能测量。
- IEEE 802.1A　概述及系统结构。
- IEEE 802.1B　网络管理和网络互联。
- IEEE 802.2　逻辑链路控制协议（LLC）的定义。
- IEEE 802.3　以太网介质访问控制协议（CSMA/CD，带有冲突检测的载波侦听多路访问）及物理层技术规范。
- IEEE 802.4　令牌总线网（Token Bus）的介质访问控制协议及物理层技术规范。
- IEEE 802.5　令牌环网（Token Ring）的介质访问控制协议及物理层技术规范。
- IEEE 802.6　城域网（MAN）介质访问控制协议分布式队列双总线（Distributed Queue Dual Bus，DQDB）及物理层技术规范。
- IEEE 802.7　宽带技术咨询组，提供有关宽带网络访问方法、物理层技术规范及宽带联网的技术咨询。
- IEEE 802.8　光纤技术咨询组，提供光纤分布数字接口 FDDI 及有关光纤联网的技术咨询。
- IEEE 802.9　综合声音数据的局域网（IVD LAN）介质访问控制协议及物理层技术规范，提供综合数据/话音 LAN 标准。

- IEEE 802.10 网络安全技术咨询组,定义了网络互操作的认证和加密方法,可互操作的 LAN 的安全机制。
- IEEE 802.11 无线局域网(WLAN)的介质访问控制协议及物理层技术规范。
- IEEE 802.11 1997 年,原始标准(2Mbps,工作在 2.4GHz)。
- IEEE 802.11a 1999 年,物理层补充(54Mbps,工作在 5GHz)。
- IEEE 802.11b 1999 年,物理层补充(11Mbps,工作在 2.4GHz)。
- IEEE 802.11c 符合 802.1D 的媒体接入控制层桥接(MAC Layer Bridging)。
- IEEE 802.11d 根据各国无线电规定做的调整。
- IEEE 802.11e 对服务等级(Quality of Service,QoS)的支持。
- IEEE 802.11f 基站的互联性(Inter-Access Point Protocol,IAPP),2006 年 2 月被 IEEE 批准撤销。
- IEEE 802.11g 2003 年,物理层补充(54Mbps,工作在 2.4GHz)。
- IEEE 802.11h 2004 年,无线覆盖半径的调整,室内(indoor)和室外(outdoor)信道(5GHz 频段)。
- IEEE 802.11i 2004 年,无线网络的安全方面的补充。
- IEEE 802.11j 2004 年,根据日本规定做的升级。
- IEEE 802.11l 预留及准备不使用。
- IEEE 802.11m 维护标准,互斥及极限。
- IEEE 802.11n 更高传输速率的改善,基础速率提升到 72.2Mbps,可以使用双倍带宽 40MHz,此时速率提升到 150Mbps。支持多输入多输出技术(Multi-Input Multi-Output,MIMO)。
- IEEE 802.11k 该协议规范规定了无线局域网络频谱测量规范。该规范的制订体现了无线局域网络对频谱资源智能化使用的需求。
- IEEE 802.11p 这个通信协定主要用在车用电子的无线通信上。它设置上是从 IEEE 802.11 来扩充延伸,来符合智能型运输系统(Intelligent Transportation Systems,ITS)的相关应用。
- IEEE 802.12 需求优先的介质访问控制协议(100VG AnyLAN),100Base-VG ANY LAN 高速网络访问方法及物理层技术规范。
- IEEE 802.14 电缆电视(Cable-TV)的宽带通信标准。
- IEEE 802.15 无线个人区域网(WPAN)规范,采用蓝牙技术的无线个人网(Wireless Personal Area Networks,WPAN)技术规范。
- IEEE 802.15.1 无线个人网络。
- IEEE 802.15.4 低速无线个人网络。
- IEEE 802.16 宽带无线连接工作组,开发 2~66GHz 的无线接入系统空中接口。
- IEEE 802.17 弹性分组环(Resilient Packet Ring,RPR)工作组,制定了单性分组环网访问控制协议及有关标准。
- IEEE 802.18 宽带无线局域网技术咨询组(Radio Regulatory)。
- IEEE 802.19 多重虚拟局域网共存(Coexistence)技术咨询组。

- IEEE 802.20　移动宽带无线接入(Mobile Broadband Wireless Access,MBWA)工作组,制定宽带无线接入网的解决。
- IEEE 802.21　媒介独立换手(Media Independent Handover)。
- IEEE 802.22　无线区域网(Wireless Regional Area Network)。
- IEEE 802.23　紧急服务工作组(Emergency Service Work Group)。

以上部分标准之间的关系如图 3-20 所示。

<table>
<tr><td colspan="8">IEEE 802.10　网络安全与加密</td></tr>
<tr><td colspan="8">IEEE 802.1A　LAN体系结构</td></tr>
<tr><td colspan="8">IEEE 802.1B　LAN寻址、互连与管理</td></tr>
<tr><td colspan="8">IEEE 802.2　逻辑链路控制LLC</td></tr>
<tr><td>802.3
CSMA/CD</td><td>802.4
令牌总线</td><td>802.5
令牌环</td><td>802.6
城域网</td><td>802.7
宽带LAN</td><td>802.8
FDDI</td><td>802.9
语音数据</td><td>802.11
WLAN</td></tr>
<tr><td>物理层</td><td>物理层</td><td>物理层</td><td>物理层</td><td>物理层</td><td>物理层</td><td>物理层</td><td>物理层</td></tr>
</table>

图 3-20　IEEE 802 标准关系图

3.5.2　IEEE 802 协议族

1. 逻辑链路控制子层(LLC)

IEEE 802.2 协议规定了逻辑链路控制(LLC)子层的规范。LLC 是局域网体系结构的最高层,该子层主要提供 LLC 用户之间通过受控的 MAC 链路进行数据交换的手段。为了满足特定的可靠性及效率的需要,规定了不同形式的 LLC 服务。

1) LLC 的工作原理

发送节点的网络层使用 LLC 访问原语将分组传给 LLC,LLC 子层为其加上 LLC 头,其中包含了序列号和确认号,然后将封装后的内容做为数据传给 MAC 子层,由 MAC 子层加入 FCS(CRC 校验),形成 MAC 帧。接收方进行相反的过程。

2) LLC 提供的服务

LLC 标准向 LLC 子层的用户提供无确认的无连接服务、有确认的无连接服务和连接服务 3 种服务形式。其中,无确认的无连接服务支持点到点、多点及广播等不同工作方式;有确认的无连接服务和连接服务只支持点到点工作方式。

3) LLC 提供的服务原语

LLC 提供的服务都用原语来定义,逻辑链路控制原语如表 3-1 所示。这些原语及其参数在提供 LLC 服务的 LLC 实体及被 LLC 服务访问点(SAP)标志的 LLC 用户之间进行交换。

4) LLC 提供的操作类型

对应着 LLC 提供的 3 种服务,IEEE 802.2 定义了 LLC 向高层提供的 3 种操作类型。

(1) 操作类型 1:支持无确认的无连接服务。

(2) 操作类型 2:支持连接服务。

(3) 操作类型 3:支持有确认的无连接服务。

表 3-1 逻辑链路控制原语

服务类型	原语及参数
无确认的无连接服务	DL-UNITDATA. request(源地址,目的地址,数据,优先级)
	DL-UNITDATA. indication(源地址,目的地址,数据,优先级)
连接服务	DL-CONNECT. request(源地址,目的地址,优先级)
	DL-CONNECT. indication(源地址,目的地址,优先级)
	DL-CONNECT. response(源地址,目的地址,优先级)
	DL-CONNECT. confirm(源地址,目的地址,优先级)
	DL-DATA. request(源地址,目的地址,数据)
	DL-DATA. indication(源地址,目的地址,数据)
	DL-DISCONNECT. request(源地址,目的地址)
	DL-DISCONNECT. indication(源地址,目的地址,理由)
	DL-RESET. request(源地址,目的地址)
	DL-RESET. indication(源地址,目的地址,理由)
	DL-RESET. response(源地址,目的地址)
	DL-RESET. confirm(源地址,目的地址)
	DL-CONNECTION-FLOWCONTROL. request(源地址,目的地址,数据量)
	DL-CONNECTION-FLOWCONTROL. indication(源地址,目的地址,数据量)
有确认的无连接服务	DL-DATA-ACK. request(源地址,目的地址,数据,优先级,服务类别)
	DL-DATA-ACK. indication(源地址,目的地址,数据,优先级,服务类别)
	DL-DATA-ACK-STATUS. indication(源地址,目的地址,优先级,服务类别,状态)
	DL-REPLY. request(源地址,目的地址,数据,优先级,服务类别)
	DL-REPLY. indication(源地址,目的地址,数据,优先级,服务类别)
	DL-REPLY-STATUS. indication(源地址,目的地址,数据,优先级,服务类别,状态)
	DL-REPLY-UPDATE. request(源地址,数据)
	DL-REPLY-UPDATE-STATUS. indication(源地址,状态)

一个单独的站(节点或进程)可以支持一种或一种以上的服务形式,并因此使用一种或一种以上的协议。根据 LLC 支持若干服务的组合不同,可将 LLC 站划分为Ⅰ、Ⅱ、Ⅲ、Ⅳ4种站类别,LLC 站类别支持的服务如表 3-2 所示。

表 3-2 LLC 可容许的站类别支持的服务

操作类型	LLC 站类别			
	Ⅰ	Ⅱ	Ⅲ	Ⅳ
操作类型 1	√	√	√	√
操作类型 2		√		√
操作类型 3			√	√

从表 3-2 可以看出,LLC 所有容许的站类别都支持操作类型 1,它保证了局域网中所有的站都具有一种共同的服务形式(即支持无确认的无连接服务),它主要用于管理操作。除此之外,各站类别仅支持其用户所需的服务,因而可使实现的规模达到最小,节省资源。

5) LLC 协议数据单元(LLC PDU)

LLC 协议都使用相同的 PDU 格式,包括控制字段、数据字段、两个 LLC 地址字段(目

标服务访问点 DSAP 和源服务访问点 SSAP)。LLC PDU 格式如图 3-21 所示。

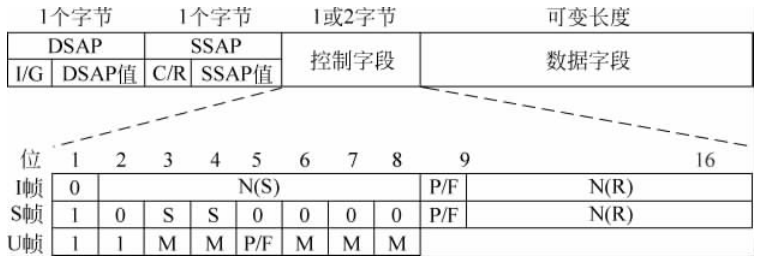


图 3-21　LLC PDU 格式

LLC 地址(LLC 的服务访问点)是一个逻辑地址,是一个层次系统的上下相邻层之间进行通信的接口,LLC 子层为网络层的各种协议提供服务,而网络层可能运行不同协议。为区分网络层上不同协议的数据,提供了服务访问点机制,即 LLC 的服务访问点提供了多个高层协议进程共同使用一个 LLC 层实体进行通信的机制。在一个网络节点上,一个 LLC 层实体可能同时为多个高层协议提供服务。因此,LLC 协议定义了一种逻辑地址 SAP 及其编码机制,允许多个高层协议进程使用不同的 SAP 地址来共享一个 LLC 层实体进行通信,而不会发生冲突。SAP 机制还允许高层协议进程同时使用多个 SAP 进行通信,但在某一时刻一个 SAP 只能由一个高层协议进程使用,一次通信结束并释放了 SAP 后,才能被其他高层协议进程使用。

LLC 服务访问点分为目标服务访问点 DSAP 和源服务访问点 SSAP。两个 LLC 地址段(DSAP 和 SSAP)都包括 1 个 7 位地址和 1 个控制位。

(1) DSAP 段:控制位(I/G)指出该地址为单个地址(I)还是组地址(G),如果 I/G=0 表示 DSAP 为单个地址,I/G=1 表示 DSAP 为组地址。组地址只用于无确认的无连接服务中。全 1 的组地址为全局 DSAP 地址,该地址为所有工作站的 DSAP。

(2) SSAP 段:控制位(C/R)指出此 PDU 为命令帧(C)还是响应帧(R),C/R=0 表示该帧为命令帧,否则为响应帧,用于确定控制字段 P/F 等位的含义。

控制字段指出帧类型(I 帧、S 帧或 U 帧)及各种控制功能。其长度可为 8 位(无编号 U 帧)或 16 位(信息 I 帧和监控 S 帧)。N(S)表示发送帧序号,N(R)表示接收方的应答序号;S 是监控功能位;M 是修改功能位;P/F 是探询/终结位。

① 信息帧(I 帧)。信息帧用于传输数据并捎带应答,第 1 位为 0。

② 监控帧(S 帧)。监控帧用于响应和流量控制,第 1、第 2 位固定为 10。

③ 无编号帧(U 帧)。无编号帧用于无编号信息和控制信息的传输,第 1、第 2 位固定为 11。

具体应用参见 3.6.1 节。

数据字段部分是由高层传入的数据,不加任何变换直接封装在 LLC PDU 数据字段中。

2. 介质访问控制子层(MAC)

介质访问控制子层所要完成的主要任务是为使用该介质的每个设备隔离来自同一通信通道上的其他设备的交通。交通隔离有时域和频域两种方法,同时也提供把时间或频率资源按一定规则分配给网络上每个设备的方法。

1) 介质访问控制

(1) 多路复用。

多路复用是指把来自 n 个输入通道的信息通过某种方法复合到一个输出通道上的技术。多路分解是相反的过程,即在一个输入复用通道上的信息位被分离和传送到 n 个输出通道。多路复用技术主要有以下 4 种。

① 频分多路复用(Frequency-division Multiplexing,FDM)。频分多路复用是指载波带宽被划分为多种不同频带的子信道,每个子信道可以并行传送一路信号的一种多路复用技术。在通信系统中,信道所能提供的带宽通常比传送一路信号所需的带宽宽得多。如果一个信道只传送一路信号是非常浪费的,为了能够充分利用信道的带宽,可以采用频分复用的方法。在频分复用系统中,信道的可用频带被分成若干个互不交叠的频段,每路信号用其中一个频段传输,接收方可以用滤波器将它们分别滤出来,然后分别解调接收。FDM 常用于模拟传输的宽带网络中。

② 时分多路复用(Time-Division Multiplexing,TDM)。时分多路复用是将整个传输时间分为许多时间间隔(Slot Time,TS,时间片,又称为时隙),将输入的多路信号按时间进行分割,每个时间片被一路信号占用,不同的信号在不同的时间内传送。因此,TDM 就是通过在时间上交错发送每一路信号的一部分来实现在一条电路上传送多路信号的。因为数字信号是有限个离散值,所以,TDM 技术广泛应用于包括计算机网络在内的数字通信系统,而模拟通信系统的传输一般采用 FDM。

③ 波分多路复用(Wave-Division Multiplexing,WDM)。波分多路复用是指将两种或多种不同波长的光载波信号在发送方经复用器(亦称合波器)汇合在一起,并耦合到光线路的同一根光纤中进行传输的技术;在接收方,经解复用器(亦称分波器或称去复用器)将各种波长的光载波分离,然后由光接收机作进一步处理以恢复原信号。这种在同一根光纤中同时传输两个或众多不同波长光信号的技术,称为波分多路复用。此外,利用光耦合器和可调的光滤波器还可以实现光交换,或将在一根光纤上输入的光信号向多根输出光纤上转发。

④ 码分多路复用(Code Division Multiple Access,CDMA)。码分多路复用又称码分多址,是指将多路信号按照经过特殊挑选的不同码型组合在一起共享同一时间和同一频率输出到一条信道上的技术。虽然码分多路复用的各个信号可以在同样的时间使用同样的频带进行通信,但由于各用户使用的是经过特殊挑选的不同码型,因此各用户的信号之间不会造成干扰。可以把码分多路复用比喻成在一个大房间里同时进行多组会话,不同组的人,分别用不同的语言交谈,讲英语的人只接收英语,讲法语的人只接收法语,其他声音当作噪音置之不理。因此,码分多路复用的关键就是能够提取出所需的信号,同时把收到的其他信号当作随机噪声丢弃。

(2) 随机访问型的介质访问控制。

随机访问型的介质访问控制协议属于争用型协议。也就是说,为了在一个多点共享的通信介质上进行数据交换,并不是采取有集中控制的方式解决发送信息的次序问题,而是让各个节点以随机的方式发送信息,竞争使用共享的通信介质。典型的协议主要有以下 3 种。

① ALOHA 协议。ALOHA 协议又称 ALOHA 技术或 ALOHA 网,是世界上最早的无线电计算机通信网。它是 1968 年美国夏威夷大学的一项研究计划的名字,是由该校 Norman Amramson 等为他们的地面无线分组网设计的,是 20 世纪 70 年代初研制成功的

一种使用无线广播技术的分组交换计算机网络，也是最早最基本的无线数据通信协议。ALOHA 协议分为纯 ALOHA 协议（最初的 ALOHA 协议）和时隙 ALOHA 协议（改进的 ALOHA 协议）两种。

- 纯 ALOHA 协议。纯 ALOHA 协议的思想很简单，数据传输采用广播方式，用两个 24kbps 的信道分别传送数据和应答信号。它对用户发送数据时间不加任何限制，根据需要，任何时间都可以发送。但要求发送节点在发送数据后侦听一段时间，侦听时间等于电波传到最远的节点再返回本节点所需时间（信号的往返时间）。如果在侦听的时间段里收到接收节点发来的应答信号，说明本次发送成功。否则，说明发送失败，重发该数据帧。如果反复几次都失败，就停止发送。接收节点对收到的数据帧进行校验，如果正确无误，则立即发出应答（应答帧采用另一频率传输）；若收到的数据帧不正确，比如有噪音干扰，或其他节点同时也在发送数据，发生了冲突，破坏了这个数据帧，则接收节点丢弃该数据帧，不发送应答信息。发送节点在规定时间内收不到应答会自动重传。纯 ALOHA 方式虽然简单，但是性能并不理想。随着通信负载的增加，冲突机会也急剧增加，据统计，信道的最高吞吐率大约只有 18%。
- 时隙 ALOHA 协议。时隙 ALOHA 协议是一种改进的 ALOHA 协议，又称为分槽 ALOHA。其方法是将信道的使用划分为等长的时间片（slot，或称时槽），每个节点所发送的数据帧到达目的地的最大时延就等于时间片长度，网络采用集中同步方式，用统一的时钟来控制用户的数据发送时间。要发送数据帧的节点只能在各个时间片的起始时刻发送，避免了用户发送数据的随意性，也避免了两个数据帧部分冲突的情况，如果冲突，则在数据帧起始处就产生冲突，整个数据帧完全冲突，不会产生部分数据冲突，因而减少了数据帧冲突的概率。据统计，时隙 ALOHA 协议可将信道吞吐率提高到 37%。

② CSMA 协议。ALOHA 协议的最大问题就是发送数据的盲目性，即发送数据前不知信道是否空闲，因此造成了很大一部分冲突，为了解决这一问题引入了 CSMA 协议。载波侦听多路访问（Carrier Sense Multiple Access，CSMA）协议在发送数据前使用了一种检测介质是否正在被使用的机制。如果一个要发送的节点“听到”在介质上有分组在传送，为避免冲突，该节点在发送之前必须等待。采用 CSMA 协议，需要有“侦听”到介质忙或闲时节点如何处理的算法。常用的有以下 3 种算法。

- 1-持续 CSMA（1-persistent CSMA）。当一个节点要传送数据时，它首先侦听信道，看是否有其他节点正在传送。如果信道正忙，它就持续等待并一直侦听，直到侦听到信道空闲时，就立即将数据送出。若发生冲突，节点就等待一个随机长的时间，然后重新开始侦听信道。此协议被称为 1-持续 CSMA，因为节点一旦发现信道空闲，其发送数据的概率为 1。
- 非持续 CSMA（non-persistent CSMA）。在该协议中，节点发送数据之前，首先侦听信道的状态，如果没有其他节点在发送，它就开始发送。如果信道正在使用之中，则该节点不再继续侦听信道，而是等待一个随机时间后，再重新侦听信道的忙闲状态，它比 1-持续 CSMA“理智”。
- P-持续 CSMA（P-persistent CSMA）。该协议主要用于时隙信道。与前两种协议一样，节点在发送数据之前，首先侦听信道，如果信道空闲，便以概率 p 发送数据，以概率 q=1−p 将本次数据发送推迟到下一个时隙。如果下一时隙仍然空闲，便再次以

概率 p 发送数据而以概率 q 将本次发送推迟到下下个时隙。此过程一直重复,直到发送成功或者另外一个节点开始发送为止。在后一种情况下,该节点的动作与发生冲突时一样(即等待一个随机时间后重新开始)。若节点一开始就侦听到信道忙,它就等到下一个时隙,然后重新开始上述过程。

③ CSMA/CD 协议。CSMA/CD(Carrier Sense Multiple Access/Collision Detect)是带有冲突检测机制的载波侦听多路访问协议。冲突检测机制(CD)是指在分组被发送到信道之后比较在介质上的能量的大小,如果能量值大于该发送设备所使用的能量,那么就可以断定信道中发生了冲突。反之也就可以认为没有发生冲突。采用 CSMA/CD 协议对最大物理介质长度有一定的限制,其原因之一就是要保证在传输介质上信号的强度足够强,以便能被检测到。如果两个节点距离太远,那么信号强度会因太弱而不能被检测出来,从而发现不了冲突。在星型物理网络中,冲突检测集中在集线器(Hub)上。如果集线器任意时刻在多于一个输入端口上检测到活动,那么就认为产生了冲突,然后集线器会向所有的端口发送称为冲突存在的特别信号(JAM 信号,拥塞信号)。只要在介质上有多个活动存在,集线器就继续发送这种信号。当集线器上的其他设备接收到 JAM 信号时,它们暂停发送。除了检测冲突,CSMA/CD 还能从冲突中恢复。一旦发生了冲突,参与冲突的两个发送设备紧接着再次发送是没有意义的。如果它们这样做,将会再次冲突,从而陷入发送→冲突→发送→冲突……的无休止的循环中。为了解决这个问题,CSMA/CD 规定,首先检测到冲突的节点发送一个短的 JAM 信号,当所有的节点都检测到 JAM 信号时,它们立即停止发送尝试,然后参与冲突的设备,使用二进制指数后退算法,在再次尝试发送之前等待一个随机长度的时间。

所谓的二进制指数后退算法是指如果网络中发生了冲突,则参与冲突的节点后退,等待一个随机的时间长度后再发送,推迟的时间必须是时隙(slot time)的整数倍。时隙是冲突处理的时间单位,它大于物理层往返传输时间,其值跟网络的具体实现有关,如在基带类型 10BASE5 中,该值是 512 位。延长多少时隙取决于均匀分布的随机参数 r,$0 \leqslant r \leqslant 2^k$,其中 $k=\min(n,10)$,n 为重传次数。用来产生随机值 r 的算法应使任何两个节点产生相关值的可能性最小。每当该节点在重传数据之后又检测到冲突时,都要把后退的时间长度加倍,因此,称为指数后退。大多数后退算法例行程序的实现规定,一旦尝试发送的次数达到 16,则节点就会放弃发送,并向上层报告错误。

CSMA/CD 最普遍的实现是以太网和 IEEE 802.3 网络,它们都使用带有二进制指数后退算法的 1-持续机制。

④ CSMA/CA 协议。在某些网络(如无线局域网)上系统不能够检测冲突,因为发送设备的功率要比接收设备的功率强得多。在这种情况下,冲突检测不可行,那么设计一个能够帮助避免冲突的系统则更有意义。因此人们开发了一种带有冲突避免的 CSMA 协议,即 CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance)协议。当前流行的冲突避免方法主要有以下两种。

- 采用 P-持续机制和空闲时间管理相结合的方法。当一个设备检测到传输介质空闲时,该设备在它可以竞争访问介质之前必须等待一个指定的帧际间隔(Inter Frame Space,IFS)时间。帧际间隔也可以用于传输优先级的确定,如果一个设备被分配一个较小的帧际间隔值,那么它就有更多的机会得到对传输介质的访问。
- 信道预约方法。发送方激发接收方,使其发送一个短帧,接收方覆盖范围的所有节点都会监测到这个短帧,知道接收节点有数据要传输。因此,接收方覆盖范围内的

其他节点在接收方有数据帧到来期间不会发送自己的帧，相当于节点在发送数据帧前对信道进行了预约。

例 3-8　试举例说明什么是隐藏终端问题？试给出一种解决这一问题的办法。

解：如图 3-22 所示，画出了 4 个无线站点。其中 A 和 B 的无线电波范围相互重合并且可能相互干扰。C 可能干扰 B 和 D 但不会干扰 A。现在假定 A 向 B 发送数据，C 在侦听，因为 A 在 C 的电波范围之外，所以 C 听不到 A，它会错误地认为它也可以发送数据。如果 C 确实也在此时开始发送数据，它就会干扰 B，从而破坏了从 A 传到 B 的数据帧。由于可能的竞争者相距太远，导致基站不能监测到的问题有时被称为隐藏终端问题，即本例中 C 对 A 来说是它的隐藏终端。

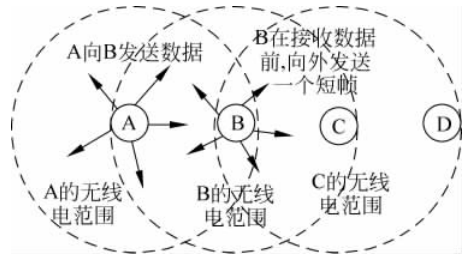


图 3-22　隐藏终端问题

为了解决这个问题，人们为无线局域网设计了称为“带有冲突避免的多路访问协议(CSMA/CA)”，它被采用为 IEEE 802.11 无线局域网标准的基础。其基本思想是：发送方发送之前先激发接收方，使其发送一短帧，因此在接收方周围的站点就会监测到这个短帧，从而使得它们在接收方有数据帧到来期间不会发送自己的帧来干扰接收方。

(3) 轮询访问型介质访问控制。

轮询访问型介质访问控制是一种利用令牌传递协议完成数据传输的机制。在令牌传递网络中，各节点访问共享介质不再是竞争访问，而是事先确定好一个顺序，按顺序依次访问。这个顺序就是一个逻辑连接环，并设置一个令牌，让其在环中依次移动。如果一个设备要发送数据帧，当它等到令牌到达时就可以发送数据，也就是持有令牌的设备允许发送数据，当该设备结束发送时，令牌被传递给环中的下一个设备。这种方法给了所有设备对介质访问的机会，并且消除了冲突。在一个令牌传递网络中，传输介质的物理拓扑不必是一个环，但是为了把对介质访问的许可从一个设备传递到另一个设备，令牌在设备之间的传递通路在逻辑上必须是一个环。所以，逻辑连接环指的是在不同设备之间令牌循环传递的过程。

2) 以太网及 IEEE 802.3

(1) 以太网概况。

以太网(Ethernet)指的是由 Xerox(施乐)公司创建并由 Xerox、Intel 和 DEC 公司联合开发的基带局域网规范，是应用最为广泛的局域网。

1972 年 Bob Metcalfe 在 Xerox 公司的 PARC 计算机实验室工作时，主要研究任务是如何将他们的第一台个人计算机 Alto 和第一台激光打印机 EARS 互连起来。1972 年底，Metcalfe 和同事 David Boggs 开发出第一个实验性的局域网系统，实验系统的数据传输速率达到 2.94Mbps。

1973 年 5 月 22 日，Metcalfe 与 Boggs 在 Alto Ethernet 中提出了以太网工作原理设计方案。他们受 19 世纪物理学家解释光在空间中传播的介质“以太(Ether)”的影响，将这种局域网命名为 Ethernet(以太网)，寓意为“无所不在的网络”。Ethernet 的核心技术是共享总线的介质访问控制方法 CSMA/CD，用于解决多个节点共享总线的发送权问题。

1977 年，Metcalfe 申请了相关专利，但他放弃收取专利费，任何人都可免费使用这些专利。

1980年,Xerox、Intel、DEC公司合作,制定了以太网物理层、数据链路层规范,命名为DIX规范。该规范规定:

- 以太网为总线拓扑结构的局域网;
- 使用同轴电缆作为传输介质,遵循同轴电缆组网的限制性规定;
- 使用CSMA/CD访问控制方式;
- 使用曼彻斯特编码,数据传输速率为10Mbps;
- 具有物理层和数据链路层的功能。

1981年,DIX 2.0发布;1982年,IEEE 802委员会以DIX 2.0为基础(几乎未作修改),发布了IEEE 802.3协议,成为现在以太网的通用标准。

1995年,100Mbps以太网标准发布;1998年,1Gbps以太网标准发布;2002年,10Gbps以太网标准发布。

以太网的发展非常迅速,每隔几年就会发布新版本标准,但从2002年至今却一直没有新版本公布。除了技术原因外,有一个重要原因就是业界在讨论以太网该向哪个方向发展。目前有两种观点,一种观点认为,应该延续局域网的模式,向100Gbps方向发展;另一种观点认为,应该与广域网统一,向40Gbps方向发展。尽管如此,以太网仍得到了广泛的应用,现在已经成为局域网的代名词。而Bob Metcalfe对以太网的产生做出了重大贡献,被称为以太网之父。

(2) IEEE 802.3协议。

IEEE 802.3协议得到广泛使用,其内容基本上就是原来的以太网DIX 2.0规范,所以IEEE 802.3协议也常被称为以太网协议。

① 访问控制方式。IEEE 802.3协议的拓扑结构为总线型,访问控制方式为CSMA/CD,使用截断的二进制指数后退算法确定随机延迟时间,发送或重发时选择在时间片开始时刻进行,不跨越时间片,以减少冲突机会。时间片的长度为51.2μs,总线长度不超过2500m。

② 数据编码。IEEE 802.3协议的物理层采用曼彻斯特编码,IEEE 802.3u(百兆)采用4B/5B编码;IEEE 802.3z(千兆)主要采用8B/10B编码;IEEE 802.3ae(万兆)主要采用64B/66B编码。

③ MAC帧结构。早期MAC帧规定的载荷是LLC帧,但这种格式现在已不再使用。现在普遍使用的帧格式是直接封装IP包的格式,如图3-23所示。

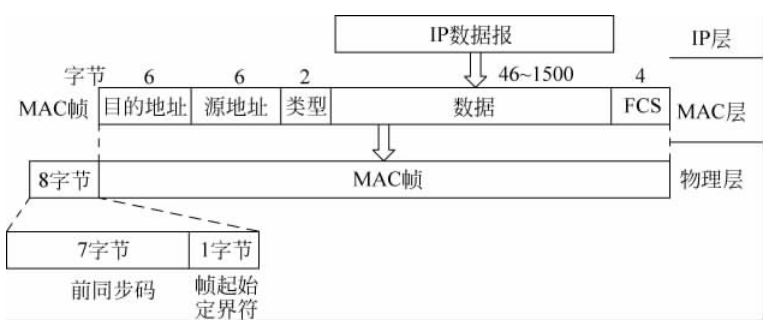


图3-23 IEEE 802.3帧格式

- 目的地址、源地址：6 字节的物理地址，称为 MAC 地址或硬件地址。
- 类型：指出上层协议的类型或数据部分的长度。当该字段值＝0800H 时，表示数据部分是 IP 包；当该字段值＝8137H 时，表示数据部分是 IPX 包；当该字段值≥8000H 时，表示帧的类型，其含义由高层定义和解释；当该字段值＜0800H 时，该字段值为数据部分的长度。
- 数据：上层传递下来的用户数据。由于 CSMA/CD 规定帧的最短长度为 64B，而 MAC 帧的协议信息（MAC 头部）为 18B，所以数据部分最少 46B，最多 1500B。
- FCS：为 CRC 校验和。

当 MAC 帧交给物理层发送时，物理层首先发送 8B 的插入信号，包括 7B 的前同步码和 1B 的帧起始定界符。7B 的前同步码的每个 B 都是 10101010，帧开始定界符是 10101011。接收方硬件在收到 6 位交替的 1、0 及 2 位 1 后，即判断为一个帧的开始，从下一位开始作为帧的正常内容接收并放到缓冲区中。

④ 帧校验和 FCS。帧校验和（FCS）按 CRC-32 生成 4B 的 CRC 校验和。其生成多项式为：

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

⑤ MAC 地址结构。MAC 地址是 6B（尽管协议规定可以为 2B，但 2B 地址基本无用），其结构如图 3-24 所示。

图 3-24　MAC 地址格式

MAC 地址的高 24 位表示设备生产商，全球唯一，称为机构唯一标识符（OUI）。OUI 由 IEEE 或 ISO 分配，并规定第一字节的最低位为 G/I 位，规定其值为 0 表示单播地址，为 1 表示组播地址，次低位为 G/L 位，其值为 0，表示全局管理地址，为 1 表示本地管理地址。

MAC 地址的低 24 位表示扩展标识符，由生产商自行分配，通常表示生产商生产的产品序号。

MAC 地址通常用十六进制书写，记为 XX-XX-XX-XX-XX-XX，如，

02-60-8C-12-03-5B

其中，02-60-8C 是设备生产商标识号，12-03-5B 是该设备的编号。

MAC 地址一般封装到网卡或网络设备中，不能更改。

数据通信过程中，MAC 帧结构中的 MAC 地址分为以下 3 类。

- 单播地址：目的地址为单播地址时，MAC 帧发送给单一节点。
- 组播地址：目的地址为组播地址时，MAC 帧发送给一组节点。
- 广播地址：目的地址为全 1，表示 MAC 帧发送给所有节点。

⑥ 寻址方式。源节点以广播方式发送一个帧（若采用交换机，则由交换机使用交换方式发送）。目的节点的底层硬件（如网卡）首先无条件接收帧，然后根据目的地址来确定是否保留所接收的帧是否送给高层处理。其处理规则如下。

- 如果所接收的数据帧的目的地址为广播地址（为全 1），则保留该帧并送高层处理。
- 如果所接收的数据帧的目的地址为单播地址，且目的地址为本节点地址时，则保留该帧并送至高层处理；若目的地址不是本节点地址时，则丢弃该帧。

- 如果所接收的数据帧的目的地址为组播地址,且其 OUI 部分与本节点地址的 OUI 部分相同,则保留该帧并送至高层处理。

⑦ 帧的发送与接收流程。发送帧的处理流程如图 3-25 所示,接收帧的处理流程如图 3-26 所示。

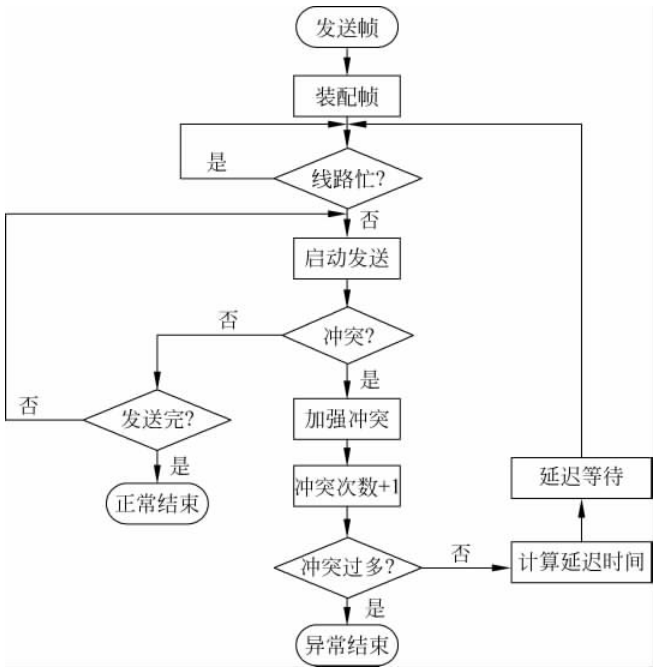


图 3-25 以太网发送数据流程

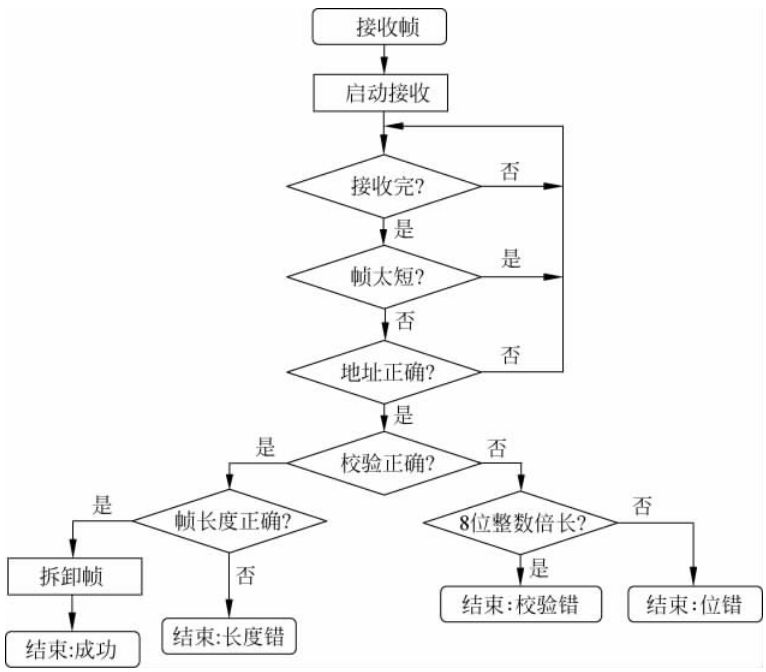


图 3-26 以太网接收数据流程

3.6 广域网协议

广域网(Wide Area Network,WAN)又称为远程网。从网络发展的过程看,首先出现的是广域网,其次是局域网,最后是其他类型的网络,如城域网、个域网等。由于出现的时间不同,各种网络设计的目标不同,因此所采用的技术不同,适用的环境也不同。

广域网是指将跨地区的计算机互联在一起组成的计算机网络。广域网除了直接连接分散的、独立的计算机外,常被用来连接多个局域网,而 Internet 是连接多个广域网、局域网和分散的计算机所组成的网际网。

广域网由通信子网和资源子网两部分构成。通信子网是由通信链路、通信节点等网络设备组成的通信网,主要使用分组交换技术。通常所说的广域网,不特殊指明时,一般指通信子网。广域网通常是由各个国家的电信部门运营和管理,在广大的地域内为不同单位的用户提供公共服务。广域网由一些节点交换机及连接链路组成,节点交换机执行分组的存储转发功能,它们并不关心数据的内容,只是提供在节点间移动数据的交换设施,直到它们到达目的地。进行通信的端点设备一般被称为站。站可以是计算机、终端、电话或其他通信设备,目前更多的是路由器。局域网可以通过路由器连接到电信部门管理的广域网交换机上。每个站都连接到一个通信节点,所有通信节点的集合称为通信网络。节点之间都是点到点的连接,但为了提高网络的可靠性,一个节点交换机往往与多个节点交换机相连。

常用的广域网协议有高级数据链路控制规程(HDLC)、点到点协议(PPP)和串行链路通信协议(SLIP)等。

3.6.1 高级数据链路控制规程

高级数据链路控制规程(High Level Data Link Control,HDLC)是由国际标准化组织 ISO 制定的面向位的有序的数据链路控制协议。HDLC 不仅使用广泛,而且还是其他许多重要数据链路控制协议的基础,它们的格式与 HDLC 中使用的格式相同或相似,使用的机制也相似。

1. HDLC 协议概述

为了适应不同配置、不同操作方式和不同传输距离的数据通信链路,HDLC 定义了 3 种站类型、两种链路配置和 3 种数据传输方式。

(1) HDLC 定义的 3 种站类型分别是主站、从站和复合站。

① 主站:主站控制数据链路(通道),负责控制链路上的操作。它向信道上的从站发送命令帧,并依次接收来自从站的响应帧。如果这条链路是多点共享的,则主站负责跟连接在该链路上的每一个从站维持一个单独的会话,即主站为链路上的每个从站维护一条独立的逻辑链路。

② 从站:又称为次站,在主站的控制下操作。从站不发送命令帧,只能响应主站的命令帧,以响应帧配合主站的工作,从站只维持一个与主站的会话。

③ 复合站:复合站复合了主站和从站双重功能,复合站既可以发送命令帧和响应帧,

也接收来自另一个复合站的命令帧和响应帧。它维持着一条与另一个复合站之间的会话。

(2) 两种链路配置是非平衡型配置和平衡型配置。

① 非平衡配置：由一个主站及一个或多个从站组成，以点对点或多点共享、半双工或全双工、交换型或非交换型等方式工作。主站负责控制每个从站，并负责建立设置方式。这种结构之所以称为非平衡的，是因为一个主站可以与多个从站互连，而一个从站只能与一个主站相连。

② 平衡配置：由两个复合站组成，两个复合站点对点互连，信道可以是半双工或全双工、可以是交换型或非交换型的。两个复合站在信道上处于同等的地位，可以互相发送未经邀请的数据帧。每个站都有同等的链路控制责任。

(3) 3 种数据传输方式是正常响应方式、异步响应方式和异步平衡方式。

① 正常响应方式(Normal Response Mode，NRM)：用于非平衡配置，主站可以初始化到从站的数据传输，而从站只能通过传输数据来响应主站的命令。从站在得到主站明确的许可后启动一次可以包含数据的响应传输，在从站的响应传输期间，通道被从站占用，从站可以在此期间发送一个或多个帧。在发送完最后一个帧之后，从站必须再等待，直到得到主站明确的许可后才可以再次发送。

NRM 主要用于多点线路，多个终端连接到一个主计算机上的情况。主计算机对每个终端进行轮询(Polling)，并采集数据。NRM 有时也可用于点对点的链路，特别是当计算机通过链路连接到一台终端或其他外设时。

② 异步平衡方式(Asynchronous Balanced Mode，ABM)：用于平衡配置。异步平衡方式提供了在两个复合站之间的平衡型数据传输方式。一个复合站不需要得到另一个复合站的许可就能启动发送。对于点对点结构，异步方式比通常响应方式效率更高，因为异步方式不需要轮询。

ABM 是 3 种方式中使用最广泛的一种，由于没有用于轮询的额外开销，所以它利用全双工点对点链路，效率非常高。

③ 异步响应方式(Asynchronous Response Mode，ARM)：用于非平衡配置。每当发现链路空闲时，不论是主站还是从站，都可以启动发送。也就是说，允许从站在未得到主站明确许可的情况下启动发送，但主站要对线路全权负责，包括初始化、差错恢复以及链路的逻辑断开等。传送可以包含一个或多个数据帧，也可以包含反映从站状态变化的控制信息。这种工作方式可以降低开销，因为从站不需要轮询序列就可以发送数据。多点配置时以一种竞争的方式进行操作，连接在一起的工作站都可以自由地发送，两个站同时传输将会引起数据破坏。当同时传输的可能性很小时，竞争方式才是一种成功的操作。显然，有一部分应用可能需要在多点配置的异步响应方式中操作，HDLC 并不禁止使用这种方式。ARM 很少被使用，它主要应用于从站需要发起传输的某些特殊场合。

HDLC 的站类型、链路配置及数据传输方式的关系如图 3-27 所示。

2. HDLC 帧格式

HDLC 帧格式如图 3-28 所示。

1) 标志(F)

标志字段值是 01111110，标识帧的开始和结束。标志位串在缓冲区中并不存在，是发

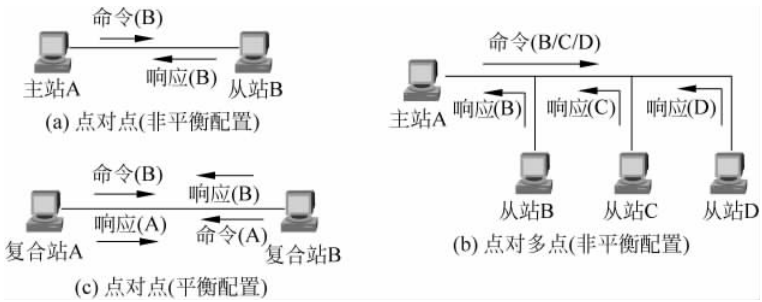


图 3-27　HDLC 的站类型、链路配置及数据传输方式的关系

位	8	8	8	≥0	16	8
	标志 01111110	地址域	控制域	信息域	帧校验序列 FCS	标志 01111110

图 3-28　HDLC 帧格式

送方硬件设备在发送 HDLC 帧之前自动产生并发送的，只在传输过程中存在。

由于位串 01111110 有可能在帧中间出现，导致接收方错误地判断帧的结束。为了避免出现这种情况，需要使用位填充法(0 比特插入技术)对帧进行透明化处理。

2) 地址(A)

地址字段值为从站的地址。点对点的链路中不需要这个地址域，但是为了统一，所有帧都含有地址域。该值在命令帧中为接收方(从站)的地址，在响应帧中为发送方(从站)的地址。

地址域通常为 8 位，但可以扩展。扩展方式是：每个 8 位组中的第 1 位作为扩展标志，若是 0，表示后续的 8 位也是地址；若为 1，表示这是最后一个 8 位组。8 位组中的其他 7 位共同组成地址部分，如图 3-29 所示。

0	7位	0	7位	…	1	7位

图 3-29　扩展地址

全 1 的 8 位地址称为广播地址，表示该帧要发送到所有从站，所有从站都应接收这个帧。

3) 信息(INFO)

信息域为任意数据，长度可变，但其位数必须是 8 的整数倍。

4) 帧校验序列(FCS)

帧校验序列又称为帧校验和，它是按 CCITT-CRC-16(生成多项式为 $x^{16}+x^{15}+x^2+1$)生成的 CRC 校验和，FCS 只对地址、控制和信息 3 部分计算校验和。

5) 控制域(C)

HDLC 定义了 3 种类型的帧，每种类型都具有不同的控制域格式。3 种类型的帧分别是信息帧、监控帧和无编号帧，其格式如图 3-30 所示。

3 种帧类型由第一位或前两位确定。

(1) 信息帧：又称为 I 帧，用于发送数据。

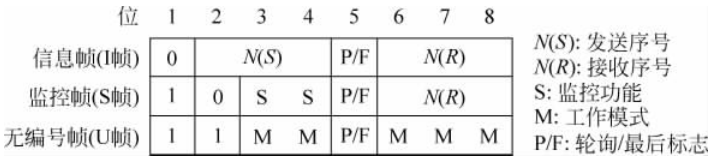


图 3-30 控制域格式

(2) 监控帧：又称为 S 帧，用于执行链路监控功能。主要用于应答、流量控制和差错控制，如对帧的确认、要求重发或请求帧传送暂停等。监控帧都不包含要传送的数据信息，不需要发送序号 $N(S)$，但需要接收序号 $N(R)$。

(3) 无编号帧：又称为 U 帧，用于提供附加的链路控制功能，如确定工作模式和链路控制等。U 帧不含编号字段，也不改变信息帧流动的顺序，只是利用修正功能位 M 来规定各种附加的命令和响应功能。

(4) 各部分的含义。

① $N(S)$：发送帧的序号。

② $N(R)$：接收序号，表示编号小于 $N(R)$的帧已正确收到，下一次期望接收帧的编号为 $N(R)$。

③ P/F：轮询/最后标志位，在主站发出的询问从站是否有信息发送的帧中，该位表示询问(Poll)；在从站发出的响应帧中，该位表示最后(Final)。P/F 的值在正常响应方式下，主站发出的信息帧中 P/F 置 1，询问从站有无数据发送。从站如果有数据发送，则开始发送，其中最后一个帧的 P/F 位置 1，表示一批数据发送完毕，其他帧的 P/F 为 0。在异步响应方式和异步平衡方式下，P/F 位用于控制监控帧和无编号帧的交换过程，不表示询问和结束。

④ S：共 2 位，表示 4 种方式，如表 3-3 所示，列出了 4 种监控帧的名称和功能说明。

表 3-3 HDLC 的 4 种监控帧的名称及功能

监控帧中的 S 位		帧 名	功 能
第 3 位	第 4 位		
0	0	RR(接收准备就绪)	准备接收下一帧，确认已正确接收了序号为 $N(R)-1$ 及以前各帧
0	1	RNR(接收未就绪)	暂停接收下一帧，确认已正确接收了序号为 $N(R)-1$ 及以前各帧
1	0	REJ(拒绝)	从 $N(R)$开始的所有帧都被否认，确认已正确接收了序号为 $N(R)-1$ 及以前各帧
1	1	SREJ(选择拒绝)	只否认序号为 $N(R)$的帧，确认已正确接收了序号为 $N(R)-1$ 及以前各帧

REJ 是一种否定应答 NAK，REJ 中的序号 $N(R)$表示所否认的帧号。这种否认帧捎带有确认信息，即确认 $N(R)-1$ 及其以前各帧均已正确收到。

RR 帧和 RNR 帧具有流量控制的作用。RR 表示已做好接收帧的准备，希望对方发送。RNR 帧表示接收未准备好，希望对方暂停发送，当准备好接收后，再次发送 RR 帧通知发送方开始发送。

⑤ M：共 5 位，可定义 32(2^5)种工作模式，目前只定义了其中一部分。如表 3-4 所示给出了命令帧控制域的设置，如表 3-5 所示给出了响应帧控制域的设置。

表 3-4　无编号帧(U 帧)命令帧控制域

1	1	M	M	P/F	M	M	M	名称	功　能
1	1	0	0	P	0	0	1	SNRM	置正常响应方式
1	1	1	1	P	0	0	0	SARM	置异步响应方式
1	1	1	1	P	1	0	0	SABM	置异步平衡响应方式
1	1	1	1	P	0	1	1	SNRME	置扩充的正常响应方式
1	1	1	1	P	0	1	0	SARME	置扩充的异步响应方式
1	1	1	1	P	1	1	0	SABME	置扩充的异步平衡响应方式
1	1	1	0	P	0	0	0	SIM	置初始化方式
1	1	1	1	P	0	1	0	DISC	置断开连接
1	1	1	1	P	0	0	0	UI	无编号信息帧
1	1	1	1	P	1	0	0	UP	无编号探询
1	1	0	0	P	0	0	1	REST	复位
1	1	0	0	P	1	0	1	XID	交换标志命令

表 3-5　无编号帧(U 帧)响应帧控制域

1	1	M	M	P/F	M	M	M	名称	功　能
1	1	0	0	F	1	1	0	UA	无编号确认
1	1	1	1	F	0	0	0	DM	断开连接应答
1	1	1	0	F	0	0	0	RIM	请求初始化
1	1	0	0	F	0	0	0	UI	无编号信息帧
1	1	1	0	F	0	0	1	FRMR	帧拒绝
1	1	1	1	F	1	0	1	XID	交换标志
1	1	1	0	F	0	1	0	RD	请求断开连接

3. HDLC 的操作

HDLC 提供的是面向连接服务，其操作包括在两个站之间交换的信息帧、监控帧和无编号帧。HDLC 的操作涉及了以下 3 个阶段。

- 连接建立(初始化)：通信双方中的一方(主站)初始化数据链路，协商各种选项，使得帧能够以有序的方式进行交换。
- 数据传送：通信双方有序交换用户数据和控制信息，并实施流量控制和差错控制。
- 连接拆除：通信结束时，双方中一方发出结束信号来终止操作。

下面用一个例子来说明 HDLC 的工作过程。

1) 链路的建立与拆除

如图 3-31 所示，主机 A 向主机 B 发出设置异步平衡方式 ABM 命令，并启动定时器。主机 B 在收到 ABM 命令后，返回一个无编号应答帧 UA，并设置必要的参数，如将局部变量和计数器设置为初值。主机 A 在接收到该无编号应答帧 UA 后，完成自身的变量和计数器设置，并停止定时器。这时逻辑连接建立完成，双方可以开始数据帧的传输。假定主机 A

发送 ABM 命令,定时器超时后还没有收到主机 B 的响应,则主机 A 重新发送设置平衡方式命令 ABM 帧,这一过程将不断被重复,直至收到一个无编号应答帧 UA 或非连接方式响应帧 DM,或者在重传次数超过了规定的次数后,放弃尝试,并向管理实体报告操作失败。

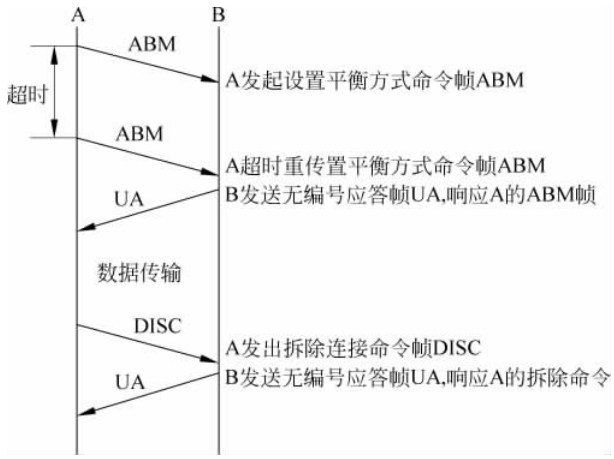


图 3-31 链路的建立和拆除过程示例

当数据发送完毕,主机 A 发出拆除连接命令帧 DISC,主机 B 用无编号应答帧 UA 来确认响应。链路的建立和拆除响应均采用无编号帧完成。

2) 数据传输

数据传输一般采用信息帧来实现全双工的数据传输。如图 3-32 所示的是利用信息帧实现双向数据交换示例,图中标记为 I,$N(S)$,$N(R)$,其中 I 表示此帧为信息帧;$N(S)$表示发送数据帧的序号;$N(R)$表示希望接收数据帧的序号。如 I,1,2 表示发送一个信息帧,发送序号为 1,希望接收对方发送的数据帧序号为 2。

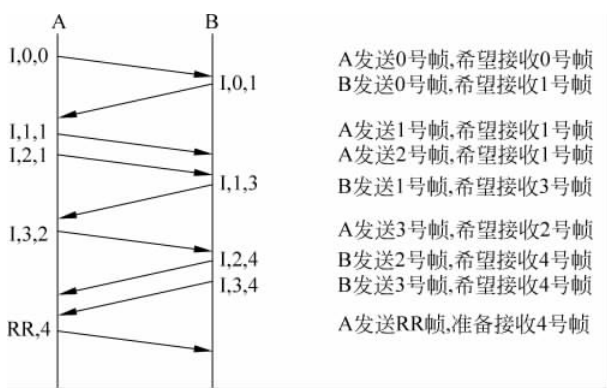


图 3-32 利用信息帧实现双向数据交换示例

当一方在没有收到对方发出的任何数据的情况下连续发送若干个信息帧时,它的接收序号只是在不断地重复(如从 A 到 B 的方向上有 I,1,1 和 I,2,1)。如果一方在没有发出任何数据帧的情况下连续收到若干个信息帧,那么它发出的下一帧中的接收序号必须反映出这一累积效果(如从 B 到 A 的方向上有 I,1,3,连续接收了 A 发出的两帧)。在此示例中,只是使用了信息帧,实际上数据交换时可能会涉及使用监控帧等。图 3-32 中主机 A 最后发

送一个监控帧 RR,4 表示接收就绪,准备接收主机 B 的 4 号帧。

3) 流量控制

当接收方处理信息帧的速度或高层用户接收信息帧的速度小于发送方发送信息帧的速度时,就需要进行流量控制。HDLC 中实现流量控制的方法是接收方使用接收未就绪(RNR)命令来阻止发送方发送新的信息帧。

如图 3-33 所示是流量控制的一个示例,主机 A 发出了一个"RNR,4"帧,要求主机 B 暂停发送信息帧,并捎带确认已正确收到 3 号帧及以前的所有帧。主机 B 收到 RNR 帧后,通常会每隔一段时间就向忙站(图中的主机 A)发出询问,通过发送一个 P 位为 1 的 RR 帧来实现,请求对方用 RR 帧或者用 RNR 帧来响应。当忙状态清除后,主机 A 会返回一个 RR 帧,这时主机 B 就可以继续发送信息帧。

图 3-33　流量控制示例

4) 差错控制

差错控制主要有拒绝恢复和超时恢复两种。

(1) 拒绝恢复。如图 3-34 所示是拒绝恢复的示例,图中主机 A 连续发送了编号为 3、4、5 的信息帧,其中编号为 4 的信息帧出现差错或丢失;当主机 B 接收到编号为 5 的信息帧时,因为顺序不对(缺少 4 号信息帧),而将 5 号信息帧丢弃,并发送一个拒绝接收帧"REJ,4"。主机 A 收到 REJ 帧后将再次发送从编号 4 开始的所有信息帧。

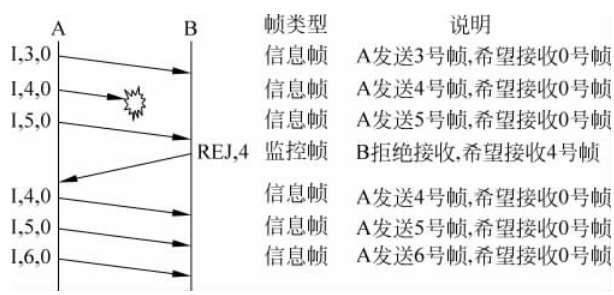


图 3-34　拒绝恢复示例

(2) 超时恢复。如图 3-35 所示是超时恢复的示例,主机 A 连续发送了 3 号和 4 号信息帧,但 4 号帧在传输过程中丢失。主机 A 在发送时启动了一个定时器,在等待应答时 t_{out} 时间到,超时启动恢复过程。主机 A 用 P 位为 1 的 RR 命令帧来询问主机 B,以判断主机 B 所处的状态。主机 B 收到该 RR 帧,由于 P 位为 1,因此需要强制应答,所以,主机 B 会发出一个包含 $N(R)$ 的响应帧(RR 帧或 RNR 帧)给主机 A,主机 A 根据它继续处理。

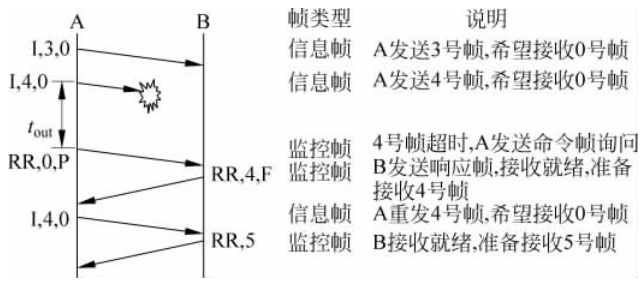


图 3-35 超时恢复示例

例 3-9 如图 3-36 所示,通信的两个站采用 HDLC 协议,交换的帧用"地址+帧名+$N(S)$值+P/F+$N(R)$值"的形式表示,P 和 $\overline{P}$ 分别表示 P 位置成 1 和 0,F 和 $\overline{F}$ 分别表示 F 位置成 1 和 0,在帧中不使用的段用—(短划线)表示。

请根据给出的一个帧序列回答下列问题:

(1) 它们使用的是 HDLC 的哪一种通信方式?

(2) 它们所进行的是半双工还是全双工通信?

(3) 由 B 站发往 A 站的帧 B.I(2)F(3)是命令帧还是响应帧?在发送此帧时,B 已经成功地收到了由 A 发往 B 的第几号帧?

(4) 在帧序列中用长方形表示的空白中正确的帧格式应该是什么?

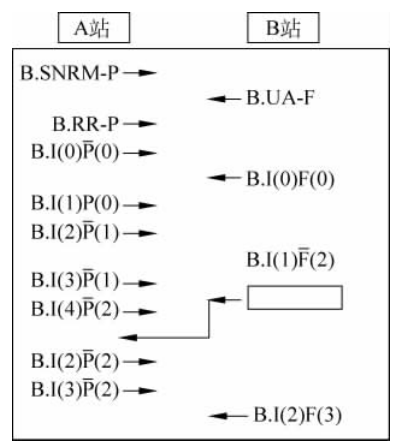


图 3-36 两个站采用 HDLC 协议交换的一个帧序列

解:(1) 它们使用的是 HDLC 的通常响应通信方式。

(2) 它们所进行的是全双工通信。

(3) 由 B 站发往 A 站的帧 B.I(2)F(3)是响应帧,在发此帧时,B 已经成功地收到了由 A 发往 B 的第 2 号帧。

(4) 在帧序列中用长方形表示的空白中正确的帧格式是 B.REJ-F(2)。

3.6.2 点到点协议(PPP)

HDLC 协议在历史上起过很大的作用,但随着互联网的快速发展,现在全世界使用得最多的数据链路层协议是点到点协议(Point-to-Point Protocol,PPP)。PPP 是使用串行线路通信的面向字节的协议。它既可以在异步线路上使用,也可以在同步线路上使用;不仅用于拨号 Modem 链路,还用于租用的路由器到路由器的线路。

1. PPP 协议概述

用户接入互联网的方法一般有两种:一种是孤立的计算机通过远程拨号或虚拟拨号方式接入互联网;另一种是通过局域网接入 Internet。前者包括电话拨号、xDSL、CableModem 等主要方式,后者包括以太网、WLAN 等主要方式。前一种接入方式现在普遍使用 PPP 协议。

PPP 协议是 IETF 于 1992 年制定的，针对 PPP 的应用环境，其应满足的条件如下。

(1) 简单。不需要复杂的流量控制、差错控制等功能，也不需要序号，只需要实现最基本的功能。

(2) 封装成帧。规定特殊的字符作为帧的开始和结束标志，同时保证能正确地区分数据与帧的定界标志，保证数据的透明传输。

(3) 支持多种网络层协议。能支持多种网络层协议。

(4) 支持多种类型链路。能在多种链路上运行，如同步或异步、高速或低速、电或光等链路。

(5) 差错检测。可进行差错检测，丢弃错帧。

(6) 检测连接状态。能及时自动检测链路的工作状态。

(7) 可设置最大传送单元。可针对不同的链路设置最大传送单元 MTU 的值(帧中数据部分的长度)。

(8) 支持网络层地址协商。支持网络层通过协商配置并识别网络地址。

(9)支持数据压缩协商。提供协商使用数据压缩算法的方法。

由于流量控制、差错控制已在 TCP 中实现，为使 PPP 协议简单化，因此 PPP 没有纠错功能；不进行流量控制；不需要帧序号；不支持多点链路；使用全双工方式传输数据。

2. PPP 协议的组成

PPP 协议由 3 个部分组成，即 HDLC 数据封装协议、链路控制协议和网络控制协议。它们分别提供下列 3 个方面的功能。

(1) 一种成帧方法。PPP 提供的成帧方法与 HDLC 相似，定义了将 IP 数据报封装到串行链路的方法，明确地定界一个帧的结束和下一个帧的开始，其帧格式允许进行差错检测。PPP 既支持异步链路(无奇偶检验的 8b 数据)，也支持面向位串的同步链路。IP 数据报是 PPP 中信息部分，其长度受最大传送单元 MTU 的限制，MTU 的默认值是 1500B。

(2) 一个链路控制协议(Link Control Protocol，LCP)。PPP 定义了一个链路控制协议(LCP)，LCP 负责线路的建立、配置、测试和协商选项，并在链路不再需要时，稳妥地释放。

(3) 一套网络控制协议(Network Control Protocol，NCP)。PPP 定义了一套网络控制协议(NCP)，NCP 是一组协议，其中的每一个协议支持不同的网络层协议，如 IP、IPX、Appletalk 等，它提供了协商网络层选项的方式。PPP 被设计成允许同时使用多个网络层协议，对于所支持的每一个网络层协议都有一个不同的网络控制协议，用来建立和配置不同的网络层协议。

3. PPP 协议的帧格式

PPP 协议的帧格式与 HDLC 帧格式相似，如图 3-37 所示。PPP 帧的前 3 个域和最后 2 个域与 HDLC 的格式是一样的。

字节	1	1	1	2或1	长度可变≤1500	2或4	1
	标志域 01111110	地址域 11111111	控制域 00000011	协议域	信息域	校验和域 FCS	标志域 01111110

图 3-37　PPP 协议的帧格式

(1) 标志域。标志为 0x7E,即 01111110,与 HDLC 相同。

(2) 地址域。固定为 0xFF,即 11111111,表示所有站都可以接收这个帧。因为 PPP 只用于点对点链路,地址域实际上不起作用。

(3) 控制域。设置为 0x03,即 00000011,表示 PPP 帧不使用编号。作为默认条件,PPP 不提供使用序列号和确认应答的可靠传输。在有噪声的环境中,如无线网络中,可以使用带编号方式的可靠传输(通过 LCP 协商确定)。

(4) 协议域。协议域的作用是说明在信息域中承载的是什么种类的分组。这是 PPP 与 HDLC 的不同之处。PPP 已经为 LCP、NCP、AppleTalk 和其他协议定义了相应的代码。常用的有:

① 0x0021:表示 PPP 帧的信息域是 IP 数据报。

② 0x002b:表示 PPP 帧的信息域是 IPX 数据。

③ 0x0029:表示 PPP 帧的信息域是 AppleTalk 数据。

④ 0xc021:表示 PPP 帧的信息域是 PPP 链路控制数据(LCP)。

⑤ 0x8021:表示 PPP 帧的信息域是 IP 控制协议。

⑥ 0x802b:表示 PPP 帧的信息域是 IPX 控制协议。

⑦ 0x8029:表示 PPP 帧的信息域是 AppleTalk 控制协议。

协议字段的默认长度是 2B,但可以通过 LCP 协商变成 1B。

(5) 信息域。信息域是网络层传送过来分组(如 IP 数据报等),长度是可变的,可以协商一个最大值。PPP 协议是面向字节的协议。因此,所有 PPP 帧的长度都是整数个字节。如果在线路建立期间没有协商长度,就采用默认长度 1500B。如果需要,在载荷的后面可以有填充。由于信息域中的内容有可能出现和标志域中一样的位串组合,因此需要使用一种方法避免出现这种情况。具体方法与传输方式有关。

① 字节填充法。当 PPP 使用异步传输时(面向字符),使用字节填充法来消除信息中可能出现的 0x7E 字节。具体方法如下:

- 将信息域中出现的每个 0x7E 字节转变成 2 字节序列:0x7D,0x5E。
- 若信息域中出现 0x7D 字节,则将其转变成 2 字节序列:0x7D,0x5D。
- 若信息域中出现 ASCII 码的控制字节(即数值小于 0x20 的字节),则在该字节前面加上一个 0x7D 字节,同时将该字节的编码加以改变。具体需要变换的字节及其变换规则如下。

  a. 0x03(ETX):变换为 0x7D,0x23。

  b. 0x11(XON):变换为 0x7D,0x31。

  c. 0x13(XOFF):变换为 0x7D,0x33。

② 位填充法。当 PPP 协议用于 SONET/SDH 链路时,使用同步传输而不是异步传输。此时,PPP 协议采用 0 比特插入技术来消除信息中可能出现的 0x7E,保证数据的透明传输。

(6) 校验和。校验和字段通常是 2B,但也可以通过协商使用 4B 的校验和。PPP 协议对收到的每一个帧,使用硬件进行 CRC 检验。若发现有差错,则丢弃该帧。因此,PPP 协议可保证链路级无差错接收。

在 PPP 中不提供使用序号和确认的可靠传输。主要原因如下。

① 若使用能够实现可靠传输的数据链路层协议，开销要增大。而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。

② 在互联网环境下，PPP 的信息域中放入的数据是 IP 数据报。假设网络采用能实现可靠传输且十分复杂的数据链路协议，然而，当数据帧在路由器中从数据链路层递交到网络层后，还是有可能因网络拥塞而丢弃。因此，在数据链路层的可靠传输并不能保证网络层的传输可靠。

4. PPP 协议的工作过程

当用户拨号接入 ISP 时，路由器对拨号做出确认，并建立一条物理连接，这时，主机向路由器发送一系列的 LCP 帧（封装成多个 PPP 帧）。这些帧及其响应帧选择了将要使用的 PPP 协议参数。然后进行网络层配置，NCP 给新接入的主机分配一个临时的 IP 地址。此时，主机进入已连入的互联网中。

当用户通信完毕时，首先，NCP 释放网络层连接，并收回原来分配出去的 IP 地址；其次，LCP 释放数据链路层连接；最后，释放物理层的连接。

上述过程可用如图 3-38 所示的状态图来描述。“链路静止”是 PPP 链路的起始和终止状态，此时，物理层连接尚未建立。当 PPP 检测到调制解调器的载波信号，并建立物理连接后，PPP 就进入“链路建立”状态。这时，LCP 开始协商一些配置选项，即发送 LCP 的配置请求帧。这是一个 PPP 帧，其协议字段设置为 0xc021（表示数据部分是 LCP），信息域包含特定的配置请求。

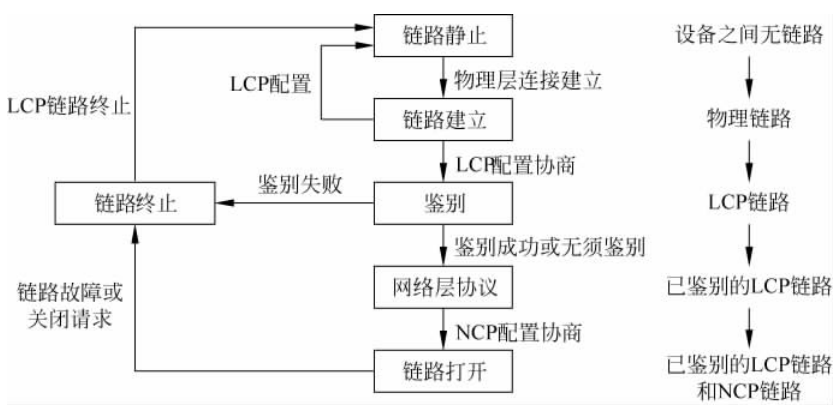


图 3-38　PPP 协议工作过程

链路的另一端可以发送以下 3 种响应。

（1）配置确认帧：所有选项都接受。

（2）配置否认帧：所有选项都理解，但不能接受。

（3）配置拒绝帧：选项有的无法识别或不能接受，需要协商。

LCP 配置选项包括链路上的最大帧长度、所使用的鉴别协议，以及不使用 PPP 帧中的地址和控制字段等。

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。

PPP链路的两端相互交换网络层特定的网络控制分组。如果在PPP链路上运行的是IP协议,则使用IP控制协议IPCP来对PPP链路的每一端配置IP协议模块(如分配IP地址)。与LCP帧封装成PPP帧一样,IPCP分组也封装成PPP帧(其中协议字段为0x8021)在PPP链路上传送。当网络层配置完毕后,链路就进入数据通信的"链路打开"状态。此时,两个PPP端点还可以发送回送请求LCP帧和回送应答LCP帧,以检查链路的状态。数据传输结束后,链路的一端发出终止请求LCP帧,请求终止链路连接,当收到对方发来的终止确认LCP帧后,就转入到"链路终止"状态,当载波停止后,则回到"链路静止"状态。

PPP是一个适用于Modem、HDLC位串行线路、SONET和其他物理层的多协议成帧机制。它支持错误检测、选项协商、头部压缩(可选)和使用HDLC成帧的可靠传输。

本章要点

本章主要阐述了数据链路层的基本概念及成帧机制,数据链路层的差错控制和流量控制,着重讲述了数据链路层的停止—等待ARQ协议、后退N帧ARQ协议和选择重传ARQ协议的工作原理。IEEE 802协议的LLC子层和MAC子层、以太网、HDLC协议、PPP协议的格式、工作原理及工作过程等内容。

习题

一、单项选择题

1. 不属于数据链路层协议考虑的范畴是_____。
   A. 控制对物理传输介质的访问
   B. 相邻节点间的可靠数据传输
   C. 为终端节点隐藏物理拓扑的细节
   D. 定义数据格式
2. 在ISO/OSI网络体系结构中,属于数据链路层提供的功能是_____。
   A. 数据功能　　B. 帧同步　　C. 路由选择　　D. 端—端通信
3. 在通信过程中产生的传输差错是由随机差错与_____共同组成的。
   A. 字节差错　　B. 连接差错　　C. 突发差错　　D. 字符差错
4. 3比特连续ARQ协议,发送窗口的最大值为_____。
   A. 2　　B. 3　　C. 7　　D. 8
5. 最常用的差错检测方法有奇偶校验和_____。
   A. 海明码　　B. 纠错码　　C. 循环冗余码　　D. 归零码
6. 前向纠错的实现是_____。
   A. 错误监测码　　B. 按字节计算的错误编码
   C. 按位计算的错误编码　　D. 差错纠正码

7. CRC-16 标准规定的生成多项式为 $G(x)=x^{16}+x^{15}+x^2+1$，它产生的校验码是_____位。

A. 2　　B. 4　　C. 16　　D. 32

8. 若信息码字为 11100011，生成多项式为 $G(x)=x^5+x^4+x+1$，则计算出的 CRC 校验码为_____。

A. 01101　　B. 11010　　C. 001101　　D. 0011010

9. 要检查出 d 位错，码字之间的海明距离最小值应为_____。

A. d　　B. $d+1$　　C. $d-1$　　D. $2d+1$

10. 要纠正 d 位错，码字之间的海明距离最小值应为_____。

A. d　　B. $d+1$　　C. $d-1$　　D. $2d+1$

11. 接收方发现有差错时，设法通知发送方重发，直到收到正确的码字为止，这种差错控制方法为_____。

A. 前向纠错　　B. 冗余校验

C. 混合差错控制　　D. 自动重发请求

12. 在_____差错控制方式中，只会重新传送出错的数据帧。

A. 连续工作　　B. 停止—等待

C. 选择重发　　D. 后退 N 帧

13. 采用简单的停止—等待协议时，应该采用_____位来表示数据帧序号。

A. 不需要　　B. 1　　C. 2　　D. 8

14. 数据链路层采用后退 N 帧协议，发送方已经发送了编号为 0～7 的帧，当计时器超时时，若发送方只收到 0、2、3 号帧的确认，则发送方需要重传的帧数是_____。

A. 2　　B. 3　　C. 4　　D. 5

15. 在 IEEE 802 标准中，LLC 层的标准是_____。

A. IEEE 802.1　　B. IEEE 802.2　　C. IEEE 802.3　　D. IEEE 802.4

16. 下列地址中，正确的以太网物理地址是_____。

A. 00-06-08-A6　　B. 202.196.1.1

C. 001　　D. 00-60-08-00-A6-38

17. IEEE 802.11 协议为提高信道利用率，采用_____协议。

A. CSMA/CD　　B. ALOHA　　C. CSMA/CA　　D. 时隙 ALOHA

18. CSMA/CD 中一旦某个站点检测到冲突，它就立即停止发送，其他站点_____。

A. 都处于待发送状态　　B. 都会相继竞争发送权

C. 都会接收到阻塞信号　　D. 仍有可能继续发送帧

19. HDLC 是一种_____协议。

A. 面向比特的同步链路控制　　B. 面向字节的异步链路控制

C. 面向字符的同步链路控制　　D. 面向比特的异步链路控制

20. 在 HDLC 帧格式中标志序列(F)是_____。

A. 1111 1111　　B. 1111 1110　　C. 0111 1111　　D. 0111 1110

21. HDLC 协议采用的帧同步方法为_____。

A. 字节计数法　　B. 使用字符填充的首尾定界法

C. 使用比特填充的首尾定界法　　D. 违法编码法

22. 采用 HDLC 传输比特串 011111111000001,比特填充后输出为_____。

A. 0101111111000001　　B. 0111110111000001

C. 0111101111000001　　D. 0111111011000001

23. 下列_____不是 HDLC 帧格式中控制字段(C)定义的帧类型。

A. 数据帧　　B. 监控帧　　C. 无编号帧　　D. 编号帧

24. HDLC 的_____域定义帧的起始和结束。

A. 标志　　B. 地址　　C. 控制　　D. FSC

25. _____是 HDLC 协议中的数据传输方式。

A. 通常响应方式　　B. 异步响应方式

C. 异步平衡方式　　D. 上面 3 项都是

26. PPP 协议是属于_____的协议。

A. 物理层　　B. 数据链路层　　C. 网络层　　D. 运输层

27. 在 PPP 链路上建立连接的第一步是_____。

A. 初始的 PPP 节点向最近的 PPP 邻居发送一个会话启动消息

B. 在 PPP 链路激活以前,路径上的路由器与身份验证程序进行协商

C. PPP 节点为动态地址分配进行通告或查询服务器为地址分配进行通告

D. 初始节点为配置数据链路而发送链路控制协议(LCP)帧

28. 在 PPP 帧中,_____字段标识封装的是 IPX 还是 IP 数据报。

A. 标识　　B. 控制　　C. 协议　　D. 帧校验序列

29. PPP 会话的建立包括_____个阶段。

A. 1　　B. 2　　C. 3　　D. 4

30. PPP 使用 NCP 的目的是_____。

A. 识别不同的物理层协议　　B. 识别不同的数据链路层协议

C. 识别不同的网络层协议　　D. 身份验证

二、综合应用题

1. 数据链路层的主要功能有哪些? 主要协议标准有哪些?

2. 假设数据位为 11011,生成多项式为 $G(x)=x^3+x+1$,计算 CRC 校验码。

3. 若 A 与 B 通信,双方协议中采用 CRC 校验,约定生成多项式是 $G(x)=x^6+x^5+x^3+x^2+1$,若 B 收到的信息是 1001100100110011,则该信息有无差错? 为什么?

4. 由于传输信道的失真或噪声等影响,信号在传输过程中会发生差错。因此如何发现差错并进一步纠正差错是十分重要的,请描述检错、纠错的基本原理。设有一种编码,它有 m 个信息位和 r 个校验位,如果需要纠正所有单比特错,当 $m=7$ 时,r 最少应为多少?

5. 一个 12 位的海明码到达接收方时的十六进制值是 0xE4F,那么,原始发送方发送的信息的十六进制是多少? 假定传输差错不超过 1 位。

6. 数据链路层协议几乎总是把 CRC 放在数据帧的尾部,而不是放在头部,为什么?

7. 卫星信道数据率为 1Mbps。取卫星信道的单程传播时延为 0.25s。每一个数据帧长都是 2000b。忽略误码率、确认帧长和处理时间,忽略帧首部长度对信道利用率的影响。试计算下列情况下的信道利用率:

(1) 停止—等待协议。

(2) 连续 ARQ 协议，$W_T=7$。

(3) 连续 ARQ 协议，$W_T=127$。

(4) 连续 ARQ 协议，$W_T=250$。

8. 证明：当用 n 个比特进行编号时，若接收窗口的大小为 1，则只有在发送窗口的大小 $W_T \leqslant 2^n-1$ 时，连续 ARQ 协议才能正确运行。

9. 试画图说明数据链路层流量控制的机制。

10. 试比较停止—等待 ARQ 协议、后退 N 帧 ARQ 协议和选择重传 ARQ 协议的异同。

11. 一个 2Mbps 的网络，线路长度为 1km，传输速度为 20m/ms，分组大小为 100B，应答帧大小可以忽略。若采用停止—等待协议，问实际速率是多少？信道利用率是多少？若采用滑动窗口技术，问最小序号位数多少？

12. 在选择重传协议中，当帧的序号字段为 3b，且接收窗口与发送窗口尺寸相同时，发送窗口的最大尺寸为多少？

13. 试分析 CSMA/CD 介质访问控制技术的工作原理。

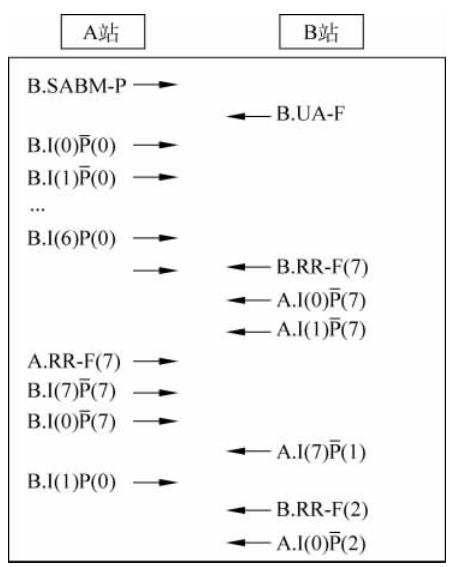
14. 试分析 CSMA/CD 协议是否完全避免碰撞？为什么？

15. 试分析以太网发送数据和接收数据的流程是怎样的？

16. 画出 HDLC 帧格式并说明各字段的意义。

17. 试举例说明 HDLC 的工作过程。

18. 下面所示为 A 站与 B 站两个节点的通信过程，A 站和 B 站都是采用 HDLC 协议的复合站，交换的帧用“地址＋帧名＋$N(S)$值＋P/F＋$N(R)$值”的形式表示，P 和 $\overline{P}$ 分别表示 P 位置成 1 和 0，F 和 $\overline{F}$ 分别表示 F 位置成 1 和 0，在帧中不使用的段用—(短划线)表示。



请根据给出的一个帧序列回答下列问题：

(1) 它们使用的是 HDLC 的哪一种通信方式？

(2) 序列中使用的 I 帧和 RR 帧是命令还是响应?

(3) 信息帧使用的编号规则的模数是几?

(4) 从发往 A 的帧 A.I(1)$\overline{P}$(7)中可以推断在发此帧时,B 已经成功地收到了由 A 发往 B 的第几号帧?

(5) 序列中属于无编号帧类型的有哪几个?

19. 试画图说明 PPP 协议的工作原理。

20. 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 5E。试问真正的数据是什么(用十六进制写出)?

实验 验证以太网(IEEE 802.3)

一、实验目的

1. 掌握以太网的报文格式
2. 掌握 MAC 地址的作用
3. 掌握 MAC 广播地址的作用
4. 掌握 LLC 帧报文格式
5. 掌握协议编辑器和协议分析器的使用方法
6. 掌握协议栈发送和接收以太网数据帧的过程

二、实验准备

1. 实验环境

本实验采用网络结构一。各主机打开协议分析器,验证网络结构一的正确性。

2. OSI 模型和 TCP/IP 协议族

(1) 层次模型的思想。

(2) OSI 模型和 TCP/IP 模型。

① OSI 模型:物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

② TCP/IP 协议族:物理层、数据链路层、网络层、传输层和应用层。

3. 以太网

(1) 以太网的物理地址。以太网上的每一个主机都有自己的网络接口卡(NIC)。网络接口卡通常安装在主机内部,并为主机提供一个 6B 的物理地址,如:44-45-53-54-00-00。在遵循 IEEE 802 标准的以太网网络中,将这个物理地址称为“MAC 地址”。MAC 地址是唯一的,任意两个不同的网络接口卡都具有不同的 MAC 地址。以太网 MAC 地址可分为 3 类:单播地址(Unicast)、广播地址(Broadcast)和多播地址(Multicast)。单播地址是一对一的,该地址是特定主机的 MAC 地址;广播地址是一对全体的,该地址为全 1,指明数据帧是发送给所有主机的;多播地址是一对多的,指明数据帧是发送给一部分主机的。

(2) 以太网访问模式。以太网使用 CSMA/CD 作为其访问模式。当多个节点被连接到一条链路上时,叫做多点链路或广播链路。这时就需要一个协议来协调链路的访问,使得同一时刻只有一个节点访问链路。如果发生同一时刻多个节点使用链路的情况,则称为链路发生了冲突。带有冲突检测的载波侦听多路访问(CSMA/CD)是这样一种方案。发送主机在传输过程中仍继续监听信道,以检测是否存在冲突。如果发生冲突,信道上可以检测到超

过发送主机本身发送的载波信号的幅度，由此判断出冲突的存在。一旦检测到冲突，就立即停止发送，并向总线上发一串阻塞信号，用于通知总线上其他各有关主机。

(3) 以太网的帧格式有 MAC 帧格式、LLC 帧格式、LLC 地址与 MAC 地址。在 MAC 帧的帧首中，有目的 MAC 地址和源 MAC 地址，它们都是 6B 长。在 LLC 帧的帧首中，则设有 DSAP 和 SSAP，该地址是逻辑地址，表示的是数据链路层的不同访问服务点。LLC 地址与 MAC 地址是两个不同的概念，在局域网中，一个主机上的多个服务访问点可以利用同一条数据链路。从这一点可以看出，LLC 子层带有 OSI 网络层的某些功能。

三、实验内容

1. 领略真实的 MAC 帧

2. 理解 MAC 地址的作用

3. 编辑并发送 MAC 广播帧

4. 编辑并发送 LLC 帧

四、实验步骤

1. 领略真实的 MAC 帧

本实验主机 A 和主机 B(主机 C 和主机 D，主机 E 和主机 F)一组进行。

(1) 主机 B 启动协议分析器，新建捕获窗口进行数据捕获并设置过滤条件(提取 ICMP 协议)。

(2) 主机 A ping 主机 B，查看主机 B 协议分析器捕获的数据包，分析 MAC 帧格式。

(3) 将主机 B 的过滤器恢复为默认状态。

2. 理解 MAC 地址的作用

本实验主机 A、B、C、D、E、F 一组进行。

(1) 主机 B、D、E、F 启动协议分析器，打开捕获窗口进行数据捕获并设置过滤条件(源 MAC 地址为主机 A 的 MAC 地址)。

(2) 主机 A ping 主机 C。

(3) 主机 B、D、E、F 停止捕获数据，在捕获的数据中查找主机 A 所发送的 ICMP 数据帧，并分析该帧内容。

3. 编辑并发送 MAC 广播帧

本练习主机 A、B、C、D、E、F 一组进行。

(1) 主机 E 启动协议编辑器。

(2) 主机 E 编辑一个 MAC 帧。

① 目的 MAC 地址：FFFFFF-FFFFFF。

② 源 MAC 地址：主机 E 的 MAC 地址。

③ 协议类型或数据长度：大于 0x0600，但不要使用 0x0800，即不使用 IP 协议。

④ 数据字段：编辑一个长度为 46～1500B 之间的数据。

(3) 主机 A、B、C、D、F 启动协议分析器，打开捕获窗口进行数据捕获并设置过滤条件(源 MAC 地址为主机 E 的 MAC 地址)。

(4) 主机 E 发送已编辑好的数据帧。

(5) 主机 A、B、C、D、F 停止捕获数据，查看捕获到的数据中是否含有主机 E 所发送的数据帧。

4. 编辑并发送 LLC 帧

本实验主机 A 和 B(主机 C 和主机 D,主机 E 和主机 F)一组进行。

(1) 主机 A 启动协议编辑器,并编写一个 LLC 帧。

① 目的 MAC 地址:主机 B 的 MAC 地址。

② 源 MAC 地址:主机 A 的 MAC 地址。

③ 协议类型和数据长度:001F。

④ 控制字段:填写 02(注:回车后变成 0200,该帧变为信息帧,控制字段的长度变为 2B)。

⑤ 用户定义数据/数据字段:AAAAAAABBBBBBBCCCCCCCDDDDDD(注:长度为 27 个 B)。

(2) 主机 B 启动协议分析器并开始捕获数据。

(3) 主机 A 发送编辑好的 LLC 帧。

(4) 主机 B 停止捕获数据,在捕获到的数据中查找主机 A 所发送的 LLC 帧,分析该帧内容。

(5) 将第 1 步中主机 A 已编辑好的数据帧修改为“无编号帧”(前两个比特位为 1),用户定义数据/数据字段修改为 AAAAAAABBBBBBBCCCCCCCDDDDDDD(注:长度为 28 个 B),重做第(2)、(3)、(4)步。

五、思考题

1. 根据实验理解集线器(共享设备)和交换机(交换设备)的区别?

2. 如何编辑 LLC 无编号帧和 LLC 数据帧?

3. 为什么 IEEE 802 标准将数据链路层分割为 MAC 子层和 LLC 子层?

4. 为什么以太网有最短帧长度的要求?

5. MAC 帧字节数是多少? 捕获的 MAC 帧长度为多少? 为什么?

6. MAC 地址的作用是什么?