**设计数据链路层的原因**

1、在原始的物理线路上传输数据信号是有差错的。

2、设计数据链路层的主要目的就是在原始的、有差错的物理传输线路的基础上，采取差错检测、差错控制与流量控制等方法，将有差错的物理线路改进成逻辑上无差错的数据链路，向网络层提供高质量的服务。

3、从网络参考模型的角度看，物理层之上的各层都有改善数据传输质量的责任，数据链路层是最重要的一层。[1]

数据链路层的最基本的功能是向该层用户提供透明的和可靠的数据传送基本服务。透明性是指该层上传输的数据的内容、格式及编码没有限制，也没有必要解释信息结构的意义；可靠的传输使用户免去对丢失信息、干扰信息及顺序不正确等的担心。在物理层中这些情况都可能发生，在数据链路层中必须用纠错码来检错与纠错。数据链路层是对物理层传输原始比特流的功能的加强，将物理层提供的可能出错的物理连接改造成为逻辑上无差错的数据链路，使之对网络层表现为一无差错的线路。如果您想用尽量少的词来记住数据链路层，那就是：“帧和介质访问控制”。

帧同步

为了使传输中发生差错后只将有错的有限数据进行重发，数据链路层将比特流组合成以帧为单位传送。每个帧除了要传送的数据外，还包括校验码，以使接收方能发现传输中的差错。帧的组织结构必须设计成使接收方能够 明确地从物理层收到的比特流中对其进行识别，也即能从比特流中区分出帧的起始与终止，这就是**帧同步**要解决的问题。由于网络传输中很难保证计时的正确和一致，所以不可采用依靠时间间隔关系来确定一帧的起始与终止的方法。

（1）字节计数法：这是一种以一个特殊字符表示一帧的起始并以一个专门字段来标明帧内字节数的帧同步方法。接收方可以通过对该特殊字符的识别从比特流中区分出帧的起始并从专门字段中获知该帧中随后跟随的数据字节数，从而可确定出帧的终止位置。面向字节计数的同步规程的典型代表是DEC公司的数字数据通信报文协议DDCMP（Digital Data Communications Message Protocol）。DDCMP采用的帧格式如图3-1。

控制字符SOH标志数据帧的起始。实际传输中，SOH前还要以两个或更多个同步字符来确定一帧的起始，有时也允许本帧的头紧接着上帧的尾，此时两帧间就不必再加同步字符。 count字段共有14位，用以指示帧中数据段中数据的字节数，14位二进制数的最大值为2-1=16383，所以数据最大长度为8×16383=131064。DDCMP协议就是靠这个字节计数来确定帧的终止位置的。DDCMP帧格式中的ACK、SEG、ADDR及FLAG中的第2位，CRC1、CRC2分别对标题部分和数据部分进行双重校验，强调标题部分单独校验的原因是，一旦标题部分中的CONUT字段出错，即失却了帧边界划分的依据，将造成灾难性的后果。由于采用字符计数方法来确定帧的终止边界不会引起数据及其它信息的混淆，因而不必采用任何措施便可实现数据的透明性（即任何数据均可不受限制地传输）。

（2）使用字符填充的首尾定界符法：该法用一些特定的字符来定界一帧的起始与终止，为了不使数据信息位中出现的与特定字符相同的字符被误判为帧的首尾定界符，可以在这种数据字符前填充一个转义控制字符（DLE）以示区别，从而达到数据的透明性。但这种方法使用起来比较麻烦，而且所用的特定字符过份依赖于所采用的字符编码集，兼容性比较差。

数据链路层

数据链路层

（3）使用比特填充的首尾标志法：该法以一组特定的比特模式（如01111110）来标志一帧的起始与终止。本章稍后要详细介绍的HDLC规程即采用该法。为了不使信息位中出现的与特定比特模式相似的比特串被误判为帧的首尾标志，可以采用比特填充的方法。比如，采用特定模式01111110，则对信息位中的任何连续出现的五个“1”，发送方自动在其后插入一个“0”，而接收则做该过程的逆操作，即每接收到连续五个“1”，则自动删去其后所跟的“0”，以此恢复原始信息，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：该法在物理层采用特定的比特编码方法时采用。例如，一种被称作曼彻斯特编码的方法，是将数据比特“1”编码成“高－低”电平对，而将数据比特“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定界帧的起始与终止。局域网IEEE 802标准中就采用了这种方法。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。由于字节计数法中COUNT字段的脆弱性以及字符填充法实现上的复杂性和不兼容性，较普遍使用的帧同步法是比特填充和违法编码法。

差错控制

一个实用的通信系统必须具备发现（即检测）这种差错的能力，并采取某种措施纠正之，使差错被控制在所能允许的尽可能小的范围内，这就是差错控制过程，也是数据链路层的主要功能之一。对差错编码（如奇偶校验码，检查和或CRC）的检查，可以判定一帧在传输过程中是否发生了错误。一旦发现错误，一般可以采用反馈重发的方法来纠正。这就要求接收方收完一帧后，向发送方反馈一个接收是否正确的信息，使发送方所在此作出是不需要重新发送的决定，也即发送方仅当收到接收方已正确接收的反馈信号后才能认为该帧已经正确发送完毕，否则需要重新发送直至正确为止。物理信道的突发噪声可能完全“淹没”一帧，即使得整个数据帧或反馈信息帧丢失，这将导致发送方永远收不到接收方发来的反馈信息，从而使传输过程停滞.为了避免出现这种情况，通常引入计时器(Timer)来限定接收方发回反馈信息的时间间隔，当发送方发送一帧的同时也启动计时器，若在限定时间间隔内未能收到接收方的反馈信息，即计时器超时(Timeout)，则可认为传的帧已出错或丢失，继而要重新发送。由于同一帧数据可能被重复发送多次，就可能引起接收方多次收到同一帧并将其递交给网络层的危险。为了防止发生这种危险，可以采用对发送的帧编号的方法，即赋予每帧一个序号，从而使接收方能从该序号来区分是新发送来的帧还是已经接收但又重新发送来的帧,以此来确定要不要将接收到的帧递交给网络层。数据链路层通过使用计数器和序号来保证每帧最终都被正确地递交给目标网络层一次。

流量控制

流量控制并不是数据链路层所特有的功能，许多高层协议中也提供流时控功能，只不过流量控制的对象不同而已。比如，对于数据链路层来说，控制的是相邻两节点之间数据链路上的流量，而对于运输层来说，控制的则是从源到最终目的之间端的流量。由于收发双方各自使用的设备工作速率和缓冲存储的空间的差异，可能出现发送方发送能力大于接收方接收能力的现象，如若此时不对发送方的发送速率（也即链路上的信息流量）作适当的限制，前面来不及接收的帧将被后面不断发送来的帧“淹没”，从而造成帧的丢失而出错。由此可见，流量控制实际上是对发送方数据流量的控制，使其发送率不致超过接收方所能承受的能力。这个过程需要通过某种反馈机制使发送方知道接收方是否能跟上发送方，也即需要有一些规则使得发送方知道在什么情况下可以接着发送下一帧，而在什么情况下必须暂停发送，以等待收到某种反馈信息后继续发送。

链路管理

链路管理功能主要用于面向连接的服务。当链路两端的节点要进行通信前，必须首先确认对方已处于就绪状态，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，在传输过程中则要能维持该连接。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。数据连路层连接的建立维持和释放就称作链路管理。在多个站点共享同一物理信道的情况下（例如在LAN中）如何在要求通信的站点间分配和管理信道也属于数据链路层管理的范畴。

**差错控制**

用以使发送方确定接收方是否正确收到了由它发送的数据信息的方法称为反馈差错控制。通常采用反馈检测和自动重发请求（ARQ）两种基本方法实现。

反馈检测法

反馈检测法也称回送校验或“回声”法，主要用于面向字符的异步传输中，如终端与远程计算机间的通信，这是一种无须使用任何特殊代码的错误检测法。双方进行数据传输时，接收方将接收到的数据（可以是一个字符，也可以是一帧）重新发回发送方，由发送方检查是否与原始数据完全相符。若不相符，则发送方发送一个控制字符（如DEL）通知接收方删去出错的数据，并重新发送该数据；若相符，则发送下一个数据。反馈检测法原理简单、实现容易，也有较高的可靠性，但是，每个数据均被传输两次，信道利用率很低。一般，在面向字符的异步传输中，信道效率并不是主要的，所以这种差错控制方法仍被广泛使用。

自动重发法

（ARQ法）：实用的差错控制方法，应该既要传输可靠性高，又要信道利用率高。为此让发送方将要发送的数据帧附加一定的冗余检错码一并发送，接收方则根据检错码对数据帧进行错误检测，若发现错误，就返回请求重发的答，发送方收到请求重发的应答后，便重新传送该数据帧。这种差错控制方法就称为自动请求法(Automatic Repeat reQuest)，简称ARQ法。ARQ法仅返回很少的控制信息，便可有效地确认所发数据帧是否被正确接收。ARQ法有若干种实现方案，如空闲重发请求(Idle RQ)和连续重请求(Continuous RQ)是其中最基本的两种方案。

**空闲重发请求**

(Idle RQ)：空闲重发请求方案也称停等(stop-and -wait)法，该方案规定发送方每发送一帧后就要停下等待接收方的确认返回，仅当接收方确认正确接收后再继续发送下一帧。空闲重发请求方案的实现过程如下： 发送方每次仅将当前信息帧作为待确认帧保留在缓冲存储器中。当发送方开始发送信息帧时，随即启动计时器。 当接收方检测到一个含有差错的信息帧时，便舍弃该帧。当接收方收到无差错的信息帧后，即向发送方返回一个确认帧。 若发送方在规定时间内未能收到确认帧(即计时器超时)，则应重发存于缓冲器中待确认信息帧。若发送方在规定时间内收到确认帧，即将计时器清零，继而开始下一帧的发送。从以上过程可以看出，空闲RQ方案的收、发双方仅须设置一个帧的缓冲存储空间，便可有效地实现数据重发并保证收接收方接收数据不会重份。空闲RQ方案最主要的优点就是所需的缓冲存储空间最小，因此在链路端使用简单终端的环境中被广泛采用。

连续重发请求

(Continuous RQ)：连续重发请求方案是指发送方可以连续发送一系列信息帧，即不用等前一帧被确认便可发送下一帧。这就需要一个较大的缓冲存储空间（称作重发表），用以存放若干待确认的信息帧。每当发送站收到对某信息帧的确认帧后，便从重发表中将该信息帧删除。所以，连续RQ方案的链路传输效率大大提高，但相应地需要更大的缓冲存储空间。连续RQ方案的实现过程如下：发送方连续发送信息帧而不必等待确认帧的返回。发送方在重发表中保存所发送的每个帧的拷贝。重发表按先进先出(FIFO)队列规则操作。接收方对每一个正确收到的信息帧返回一个确认帧。每一个确认帧包含一个唯一的序号，随相应的确认帧返回。接收方保存一个接收次序表，它包含最后正确收到的信息帧的序号。当发送方收到相应信息帧的确认帧后，从重发表中删除该信息帧。当发送方检测出失序的确认帧(即第n号信息帧和第n+2号信息帧的确认帧已返回，而n+1号的确认帧未返回)后，便重发未被确认的信息帧。实际操作过程中，两节点间采用双工方式将确认帧插在双方的发送信息帧中来传送的。上面的连续RQ过程是假定在不发生传输差错的情况下描述的。如果差错出现，如何进一步处理可以有两种策略，即Go-back-N和选择重发。 Go-back-N是当接收方检测出失序的信息帧后，要求发送方重发最后一个正确接收的信息帧之后的所有未被确认的帧，或者当发送方发送了n帧后，若发现该n帧的前一帧在计时器超时后仍未返回其确认信息，则该帧被判定为出错或丢失。对接收方来说，因为这一帧出错，就不能以正确的序号向它的高层递交数据，对其后发送来的n帧也可能都不能接收而丢弃，因此，发送方发现这种情况，就不得不重新发送该出错帧及其后的n帧，这就是Go-back-N(退回N)法名称的由来。Co-back-N法操作过程如图3-2所示。图中假定发送完8号帧后，发现2号帧的确认返回在计时器超时后还未收到，则发送方只能退回从2号帧开始重发。Go-back-N可能将已正确传送到目的方的帧再传一遍，这显然是一种浪费。另一种更好的策略是当接收方发现某帧出错后，其后继续送来的正确的帧虽然不能立即递交给接收方的高层，但接收方仍可收下来，存放在一个缓冲区中，同时要求发送方重新传送出错的那一帧，一旦收到重新传来的帧后，就可与原已存于缓冲区中的其余帧一并按正确的顺序递交高层。这种方法称为选择重发(Selective repeat)，其工作过程如图3-3所示。图中2号帧的否认返回信息NAK2要求发送方选择重发2号帧。显然，选择重发减少了浪费但要求接收方有足够大的缓冲区容量。

流量控制

差错控制是数据链路层功能中的一个部分，另一个重要部分是流量控制。流量控制涉及链路上字符或帧的传输速率的控制，以使接收方在接收前有足够的缓冲存储空间来接受每一个字符或帧。例如，在面向字符的终端－－计算机链路中，若远程计算机为许多台终端服务，它就有可能因不能在高峰时按预定速率传输全部字符而暂时过载。同样，在面向帧的自动重发请求系统中，当待确认帧数量增加时，有可能超出缓冲器存储容量，也造成过载。

XON/XOFF方案

增加缓冲存储空间在某种程度上可以缓解收、发双方在传输速率上的差别，但这是一种被动的和消极的方法，实现起来有诸多的不便和限制。因为一方面系统不允许开设过大的缓冲空间，另一方面对于速率显著失配并且又传送大型文件的场合，仍会出现缓冲存储空间不够。XON/XOFF方案则是一种相比之下更主动、积极的流量控制方法。XON/XOFF方案中使用一对控制字符来实现流量控制，其中XON采用ASCII字符字集中的控制字符DC1、XOFF采用ASCII字符集中的控制字符DC3。当通信链上的接收方发生过过载时便向发送方发送一个XOFF字符后便暂时停止发送数据，等接收方处理完缓冲存储器中中的数据，过载恢复后，再向发送方发送一个XON字符，以通知发送方恢复数据发送。在一次数据传输过程中，XOFF、XON的周期可重复多次，但对用户是透明的。许多异步数据通信软件包均支持XON/XOFF协议。这种方案也可用于计算机向打印机或其它终端设备发送字符，在这种情况下，打印机或终端设备中的控制部件用以控制字符流量。

窗口机制

为了提高信道的有效利用率。如前节所述采用了发送方不等待确认帧返回就连续发送若干帧的方案，这样的发送过程就象一条连续的流水线，故又称为管道(pipelining)技术。由于允许连续发送多个未被确认折帧，帧号就采用多位二进制数才能加以区分。因为凡被发送出去但沿尚未被确认的帧都可能出错或丢失而要求重发，因而这些帧都要保留下来。这就要求发送方有较大的发送缓冲区保留可能要求重发的未被确认的帧。但是缓冲区容量总是有限的，如果接收方不能以发送方的发送速率处理收到的帧，则还是可能用完缓冲容量而暂时过载。为此，可引入类似于空闲RQ方案的调整措施，其本质是在收到一确定帧之前，对发送方可发送的帧的数目加以限制，这是由发送方调整保留在重发表中的待确认帧的数目来实现的。如果接收方来不及对收到的帧进行处理，则接收方停发确认信息，此时送方的重发表增长，当达到重发表限度时，就不再发送新帧，直至再次收到确认信息为止。 为了实现此方案，存放未确认帧的重发表中应设置未确认帧数目的最大限度，这一限度被称为链路的发送窗口。显然，如果窗口设置为1，即发送方缓冲能力公为一个帧，则传输控制方案就回到了空闲RQ方案，此时传输效率很低，故窗口限度应选为使接收方尽量能处理或接受收到的所有帧。当然选择时还必须考虑诸如帧的最大长度、可使用的缓冲存容量以及传输的比特速率等因素。重发表是一个连续序号的列表，对应发送方已发送但尚未确认的那些帧。这些帧的序号有一个最大值即发送窗口的限度。所谓发送窗口就是指示发送方已发送但尚未确认的帧序号队列的界，其上、下界分别称为发送窗口的上、下沿，上、下沿的间距称为窗口尺寸。接收方类似地有接收窗口，它指示允许接收的帧的序号。接收窗口的上、下界也是随时间滑动的。

发送方每次发送一帧后，待确认帧的数目便增1；同样，发送方每收到一个确认信息后，待确认帧的数目便减1。当重发计数值，即待确认帧的数目等于发送窗口时，便停止发送新的帧。一般帧号只取有限位二进制数，到一定时间后就又反复循环，若帧号配3位二进制，则帧号在0～7间循环。如果发送窗口限度取值为2，则发送过程如图3-4所示。图中发送方阴影表示发送窗口，接收方阴影则相应可视作接收窗口。当传送过程进行时，窗口位置一直在滑动，所以也称为滑动窗口（Slidding Window)，或简称为滑窗。

图3-4中滑动窗口的状态变化过程可叙述如下（假设发送窗口为2，接收窗口为1）。

初始态，发送方没有帧发出，发送窗口前后沿相等。接收窗口限度为1，它允许接收0号帧。

发送方已发送0号帧，此时发作口打开（即前沿加1），窗口对准0号，表示已发出但尚未收到确认返回信息。接收窗口状态同前，指示允许接收0帧。

发送方在未收到0帧的确认返回信息前，继续发送1号帧。发送窗口状态不变。

接收方已收到0帧，窗口滑动一格，表示准备接收1号帧。发送窗口状态不变。

发送方已收到0号帧的确认返回信息，发送窗口后沿加1，表示从重发表中删除0号帧，接收窗口状态不变。

发送方继续发送2帧，发送窗口前沿加1，表示2号帧也纳入待确认之列。接收窗口状态仍不变。

接收方已收到1号帧，接收窗口滑动一格，表示准备接收2号帧。发送窗口状态不变。

发送方收到接收方发来的1号帧收毕的确认信息，发送窗口后沿加1，表示从重发表中删除最早进入的1号帧。接收窗口状态不变。 一般说来，凡是在一定范围内到达的帧，那怕不按顺序，接收方也要接收下来。若把这个范围看成是接收窗口的话，则接收窗口的大小应该是大于1的，而Go-back-N正是接收窗口等于1的一个特例。选择重发也可以看作是一种滑动窗口协议，只不过其发送窗口和接收窗口都大于1。若从滑动窗口的观点来统一看待空闲RQ、Go-back-N及选择重发三种协议，它们的差别公在于各自窗口的大小不同而已：

空闲RQ：发送窗口=1，接收窗口=1

Go-back-N：发送窗口>1，接收窗口=1

选择重发：发送窗口>1，接收窗口>1

若帧序号采用3位二进制编码，则最大序号为SMAX=2^3－1=7。对于有序接收方式，发送窗口最大尺寸选为SMAX；对于无笆接收方式，发送窗口最大尺寸至多是序号范围的一半。管理超时控制的计时器应等于发送缓冲器数，而不是序号空间的大小。实际上，每一个缓冲器应对应一个计时器，当计时器超时时，该对应缓冲器的内容重发。按收方必须设置的缓冲器数应该等于接收窗口尺寸，而不是序号空间的大小。

链路控制规程

数据链路控制协议也称链路通信规程，也就是OSI参考模型中的数据链路层协议。链路控制协议可分为异步协议和同步协议两大类。

数据链路层的主要协议有：

(1)点对点协议（Point-to-Point Protocol);

(2)以太网（Ethernet);

(3)高级数据链路协议(High-Level Data Link Protocol);

(4) 帧中继（Frame Relay);

(5) 异步传输模式(Asynchronous Transfer Mode);

异步协议

以字符为独立的信息传输单位，在每个字符的起始处开始对字符内的比特实现同步，但字符与字符之间的间隔时间是不固定的（即字符之间是异步的）。由于发送器和接收器中近似于同 一频率的两个约定时钟，能够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，而不需要每个比特再用其它方法同步。异步协议中因为每个传输字符都要添加诸如起始位、校验位及停止位等冗余位，故信道利用率很低，一般用于数据速率较低的场合。

同步协议

同步协议是以许多字符或许多比特组织成的数据块－－帧为传输单位，在帧的起始处同步，使帧内维持固定的时钟。实际上该固定时钟是发送端通过某种技术将其混合在数据中一并发送出去的，供接收端从输入数据中分离出时钟来，实现起来比较复杂，这个功能通常是由调解器来完成。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。同步协议又可分为面向字符的同步协议、面向比特的同步协议及面向字节计数的同步协议三种类型。

二进制协议

面向字符的同步协议是最早提出的同步协议，其典型代表是IBM公司的二进制同步通信(Binary Synchronous Communication、BISYNC或BSC)协议，通常，也称该协议为基本型协议。随后，ANSI和ISO都提出类似的相应标准。ISO的标准称为数据通信系统的基本型控制过程(Basic mode control procedures for data communication Systems)，即ISO 1745标准。任何链路层协议均可由链路建立、数据传输和链路拆除三部分组成。为实现建链、拆链等链路管理以及同步等各种功能，除了正常传输的数据块和报文外，还需要一些控制字符。 BSC协议用ASC2或EBCDIC字符集定义的传输控制（TC）字符来实现相应功能。这些传输控制字符的标记、名称及ASC2码值和EBCDIC码值见表3.1。 各传输控制字符的功能如下：

SOH(Start of Head)： 序始或标题开始，用于表示报文（块）的标题信息或报头的开始。

STX(Start of TEXT)：文始，标志标题信息的结束和报文（块）文本的开始。

ETX(End of Text)： 文终，标志报文（块）文本的结束。

EOT(End of Transmission)： 送毕，用以表示一个或多个文本块的结束，并拆除链路。

ENQ(Enquire)：询问，用以请求远程站给出响应，响应可能包括站的身份或状态。

ACK(Acknowledge)： 确认，由接收方发出一肯定确认，作为对正确接收来自发送方的报文（块）的响应。

DLE(Data Link Escape)： 转义，用以修改紧跟其后的有限个字符的意义。用于在BSC中实现透明方式的数据传输，或者当10个传输控制字符不够用时提供新的转义传输控制字符。

NAK(Negative Acknowledge)： 否认，由接收方发出的否定确认，作为对未正确接收来自发送方的响应。

SYN(Synchronous)： 同字符，在同步协议中，用以实现节点之间的字符同步，或用于在列数据传输时保持该同步。

ETB(End of Transmission Block)： 块终或组终，用以表示当报文分成多个数据块时， 一个数据块的结束。

BSC 协议将在链路上传输的信息分为数据报文和监控报文又分为正向监控和反向监控两种。每一种报文中至少包含一个传输控制字符，用以确定报文中信息的性质或实现某种控制作用。

数据报文和文本组成。文本是要传送的有用数据信息，而报文是与文本传送及处理有关的辅助信息，报头有时也可不用，对于不超过长度限制的报文可只用一个数据块作为一个传输单位。接收方对于每一个收到的数据块都要给予确认，发送方收到返回的确认后，才能发送下一个数据块。BSC协议为数据块格式可以有5种，如图3.5所示。

BSC协议中所有发送的数据均跟在至少两个SYT字符之后，以使接收方能实现字符同步。报头字段用以说明数据文字段的包识别符（序号）及地址。所有数据块在块终限定符（ETX或ETB）之后不有块验字符BCC（Block Check Charracter），BCC可以是垂直奇偶校验或16位CRC，校验范围自STX始，至ETX或ETB止。

当发送的报文是二进制数据而不是字符串时，二进制数据中形同传输控制字符的比特串将会引传输混乱。为使二进制数据中允许与传输控制字符相同的数据(即数据的透明性），可在各帧中真正的传输控制字符（SYN除外）前加上DLE转义字符，在发送时，若文本中也出现与DLE字符相同的二进制比特串，则可插入一个外加的DLE字符加以标记。 在接收端则进行同样的检测，若发现单个的DLE字符，则知其后的DLE为数据，在进一步处理前将其中一个删去。

正、反向监控报文有四种格式。

(1)肯定确认和选择响应：

SYN | SYN | ACK

(2)否定确认和选择响应:

SYN | SYN | NAK

(3)轮询/选择请求:

SYN | SYN | P/S前缀 | 站地址 | ENQ

(4)拆链:

SYN | SYN | EOT

监控报文一般由单个传输控制字符或由若干个其它字符引导的单个传输控制字符组成。引导字符统称为前缀，它包含识别符（序号）、地址信息、状态信息以及其它所需的信息。ACK和NAK监控报文的作用，首先作为对先前所发数据块是否正确接收的响应，因而包含识符（序号）；其次，用作对选择监控信息的响应，以ACK表示所选站能接收数据块，而NAK表示不能接收。ENQ用作轮询和选择监控报文，在多结构中，轮询或选择的站地址在ENQ字符前。EOT监控报文有用以标志报文的结束，并在两站点间除逻辑链路。

面向字符的同步协议的最大缺点，是它和特定的字符编码集关系过于密切，不利于兼容性。为了实现数据的透明性而采用的字符填充法，实现起来比较麻烦，且也依赖于采用的字符编码集。另外，由于BSC是一个半双工协议，它的链路传输效率很低，即使物理连路支持全双工传输，BSC也不能加以运用。不过，由于BSC协议需要的缓冲存储容量最小，因而在面向终端的网络系统中仍然广泛使用。

高级控制协议

七十年代初，IBM公司率先提出了面向比特的同步数据链路控制规程SDLC。随后，ANSI和ISO均采纳并发展了SDLC，并分别提出了自己的标准：ANSI的高级通信控制过程ADCCP（Advanced Data Control Procedure），ISO的高级数据链路控制规程HDLC。链路控制协议着重于对分段成物理块或包的数据的逻辑传输，块或包由起始标志引导并由终止标志结束，也称为帧。帧是每个控制、每个响应以及用协议传输的所有信息的媒体的工具。所有面向比特的数据链路控制协议均采用统一的帧格式，不论是数据还是单独的控制信息均以帧为单位传送。

每个帧前、后均有一标志码01111110、用作帧的起始、终止指示及帧的同步。标志码不允许在帧的内部出现，以免引起畸意。为保证标志码的唯一性但又兼顾帧内数据的透明性，可以采用“0比特插入法”来解决。该法在发送端监视除标志码以外的所有字段，当发现有连续5个“1”出现

数据链路层图3-6

数据链路层图3-6

时，便在其后添插一个“0”，然后继续发后继的比特流。在接收端，同样监除起始标志码以外的所有字段。当连续发现5个“1”出现后，若其后一个比特“0”则自动删除它，以恢复原来的比特流；若发现连续6个“1”，则可能是插入的“0”发生差错变成的“1”，也可能是收到了帧的终止标志码。后两种情况，可以进一步通过帧中的帧检验序列来加以区分。“0比特插入法”原理简单，很适合于硬件实现。 在面向比特的协议的帧格式中，有一个8比特的控制字段，可以用它以编码方式定义丰富的控制命令和应答，相当于起到了BSC协议中众多传输控制字符和转义序列的功能。作为面向比特的数据链路控制协议的典型，HDLC具有如下特点：协议不依赖于任何一种字符编码集；数据报文可透明传输，用于实现透明传输的“0比特插入法”易于硬件实现；全双工通信，不必等待确认便可连续发送数据，有较高的数据链路传输效率；所有帧均采用CRC校验，对信息帧进行编号，可纺止漏收或重份，传输可靠性高；传输控制功能与处理功能分离，具有较大灵活性和较完善的控制功能。由于以上特点，网络设计普遍使用HDLC作为数据链路管制协议。

HDLC的操作方式

HDLC是通用的数据链路控制协议，当开始建立数据链路时，允许选用特定的操作方式。所谓链路操作方式，通俗地讲就是某站点以主站方式操作，还是以从站方式操作，或者是二者兼备。在链路上用于控制目的站称为主站，其它的受主站控制的站称为从站。主站负责对数据流进行组织，半且对链路上的差错实施恢复。由主站发往从站的帧称为命令帧，而由由站返回主站的帧称响应帧。连有多个站点的链路通常使用轮询技术，轮询其它站的站称为主站，而在点到点燃链路中每个站均可为主站。主站需要比从站有更多的逻辑功能，所以当终端与主机相连时，主机一般总是主站。在一个站连接多条链中的情况下，该站对于一些链路而言可能是主站，而对另外一些链路而言又可能是从站。有些可兼备主站和从站的功能，这站称为组合站，用于组合站之间信息传输的协议是对称的，即在链路上主、从站具有同样的传输控制功能，这又称作平衡操作，在计算机网络中这是一个非常重要的概念。相对的，那种操作时有主站、从站之分的，且各自功能不同的操作，称非平衡操作。

HDLC中常用的操作方式有以下三种：(1)正常响应方式NRM是一种非平衡数据链路操作方式，有时也称非平衡正常响应方式。该操作方式适用于面向终端的点到点或一点与多点的链路。在这种操作方式，传输过程由主站启动，从站只有收到主站某个命令帧后，才能作为响应向主站传输信息。响应信息可以由一个或多个帧组成，若信息 由多个帧组成，则应指出哪一个是最后一帧。主站负责管理整个链路，且具有轮询、选择从站及向从站发送命令的权利，同时也负责对超时、重发及各类恢复操作的控制。NRM操作方式见图3.7(a)。（2）异步响应方式ARM，异步响应方式ARM也是一种非平衡数据链路操作方式，与NRM不同的是，ARM的传输过程由从站启动。从站主动发送给主站的一个或一组帧中可包含有信息，也可以是仅以控制为目的而发的帧。在这种操作方式下，由从站来控制超时和重发。该方式对采用轮询方式的多站莲路来说是必不可少的。ARM操作方式见图3.7(b)。（3）异步平衡方式ABM，异步平衡方式ABM是一种允许任何节点来启动传输的操作方式。为了提高链路传输效率，节点之间在两个方向上都需要的较高的信息传输量。在这种操作方式下任何时候任何站都能启动传输操作，每个站既可作为主站又可作为从站，每个站都是组合站。各站都有相同的一组协议，任何站都可以发送或接收命令，也可以给出应答，并且各站对差错恢复过程都负有相同的责任。

HDLC的帧格式

在HDLC中，数据和控制报文均以帧的标准格式传送。HDLC中的帧类似于BSC的字符块，但BSC协议中的数据报文和控制报文是独立传输的，而HDLC中的命令应以统一的格式按帧传输。HDLC的完整的帧由标志字段（F）、地址字段（A）、控制字段（C）、信息字段（I）、帧校验序列字段（FCS）等组成，其格式见图3.8。

（1）标志字段（F）：标志字段为01111110的比特模式，用以标志帧的起始和前一帧的终止。标志字段也可以作为帧与帧之间的填充字符。通常，在不进行帧传送的时刻，信道仍处于激活状态，在这种状态下，发方不断地发送标志字段，便可认为一个新的帧传送已经开始。采用“0比特插入法”可以实现0数据的透明传输。

（2）地址字段（A）：地址字段的内容取决于所采用的操作方式。在操作方式中，有主站、从站、组合站之分。每一个从站和组合站都被分配一个唯一的地址。命令帧中的地址字段携带的是对方站的地址，而响应帧中的地址字段所携带的地址是本站的地址。某一地址也可分配给不止一个站，这种地址称为组地址，利用一个组地址传输的帧能被组内所有拥有该组一焉的站接收。但当一个站或组合站发送响应时，它仍应当用它唯一的地址。还可用全“1”地址来表示包含所有站的地址，称为广播地址，含有广播地址的帧传送给链路上所有的站。另外，还规定全“0”地址为无站地址，这种地址不分配给任何站，仅作作测试。

（3）控制字段（C）：控制字段用于构成各种命令和响应，以便对链路进行监视和控制。发送方主站或组合站利用控制字段来通知被寻址的从站或组合站执行约定的操作；相反，从站用该字段作对命令的响应，报告已完成的操作或状态的变化。该字段是HDLC的关键。控制字段中的第一位或第一、第二位表示传送帧的类型，HDLC中有信息帧（I帧）、监控帧（S帧）和无编号帧（U帧）三种不同类型的帧。控制字段的第五位是P/F位，即轮询/终止（Poll/Final）位。

（4）信息字段（I）：信息字段可以是任意的二进制比特串。比特串长度未作限定，其上限由FCS字段或通信站的缓冲器容量来决定，国际上用得较多的是1000~2000比特；而下限可以为0，即无信息字段。但是，监控帧（S帧）中规定不可有信息字段。（5）帧校验序列字段（FCS）：帧校验序列字段可以使用16位CRC，对两个标志字段之间的整个帧的内容进行校验。FCS的生成多项式CCITT V4.1建议规定的X^16+X^12+X^5+1。

HDLC的帧类型

HDLC有信息帧（I帧）、监控帧（S帧）和无编号帧（U帧）三种不同类型的帧。每一种帧中的控制字段的格式及比特定义见图3.9。

（1）信息帧（I帧）：信息帧用于传送有效信息或数据，通常简称I帧。I帧以控制字第一位为“0”来标志。信息帧的控制字段中的N（S）用于存放发送帧序号，以使发送方不必等待确认而连续发送多帧。N（R）用于存放接收方下一个预期要接收的帧的序号，N（R）=5，即表示接收方下一帧要接收5号帧，换言之，5号帧前的各帧接收到。N（S）和N（R）均为3位二进制编码，可取值0～7。

（2）监控帧（S帧）：监控帧用于差错控制和流量控制，通常简称S帧。S帧以控制字段第一、二位为“10”来标志。S帧带信息字段，只有6个字节即48个比特。S帧的控制字段的第三、四位为S帧类型编码，共有四种不同编码，分别表示：

00——接收就绪（RR），由主站或从站发送。主站可以使用RR型S帧来轮询从站，即希望从站传输编号为N（R）的I帧，若存在这样的帧，便进行传输；从站也可用RR型S帧来作响应，表示从站希望从主站那里接收的下一个I帧的编号是N（R）。

01——拒绝（REJ），由主站或从站发送，用以要求发送方对从编号为N（R）开始的帧及其以后所有的帧进行重发，这也暗示N（R）以前的I帧已被正确接收。

10——接收未就绪（RNR），表示编号小于N（R）的I帧已被收到，但正处于忙状态，尚未准备好接收编号为N（R）的I帧，这可用来对链路流量进行控制。

数据链路层图3.9

数据链路层图3.9

11——选择拒绝（SREJ），它要求发送方发送编号为N（R）单个I帧，并暗示它编号的I帧已全部确认。

可以看出，接收就绪RR型S帧和接收未就绪RNR型S帧有两个主要功能：首先，这两种类型的S帧用来表示从站已准备好或未准备好接收信息；其次，确认编号小于N（R）的所有接收到的I帧。拒绝REJ和选择拒绝SREJ型S帧，用于向对方站指出发生了差错。REJ帧用于GO-back-N策略，用以请求重发N（R）以前的帧已被确认，当收到一个N（S）等于REJ型S帧的N（R）的I帧后,REJ状态即可清除。SREJ帧用于选择重发策略，当收到一个N（S）等SREJ帧的N（R）的I帧时，SREJ状态即应消除。

（3）无编号帧（U帧）：无编号帧因其控制字段中不包含编号N（S）和N（R）而得名，简称U帧。U帧用于提供对链路的建立、拆除以及多种控制功能，这些控制功能5个M位（M1、M2、M3、M4、M5，也称修正位）来定义。5个M位可以定义32种附加的命令功能或32种应答功能，但许多是空缺的。[3]

比较

适用场合

就系统结构而言，HDLC适用于点到点或点到多点式的结构，BSC同样也能适用于这些结构；就工作方式而方，HDLC适用于半双工或全双工，而BSC则更适用于半双工方式（也可扩充为全双工）；就传输方式而言，BSC和HDLC两者都只用于同步传输。在传输速率方面，BSC和HDLC虽然都没有限制，但由于它们各自的特点所定，通常BSC用于低、中速传输，而HDLC则常用于中、高速传输。

传输效率

HDLC开始发送一帧后，就要连续不断地发完该帧，而BSC的同一数据块中的

数据链路层

数据链路层

不同字符之间可能有时间间隔，这些间隔用SYN字符填充。HDLC可以同时确认几个帧，而BSC则在发完一数据块后必须要等待确认（即“停一等”方式）。HDLC中的每个帧都含有地址字段A，在多点结构中，每个从站只接收含有本站地址的帧，因此，主站在选中一个从站并与之通信的同时，不用拆链，便可选择其它的站通信，即同时与多个站建立链路。而在BSC中，从建链开始，两站之间的链路通道就一直保持到传输结束为止。由于以上特点，HDLC的传输效率高于BSC的传输效率。

传输可靠性

HDLC中所有的帧（包括响应帧）都有FCS，在BSC的监控报文中只有字符校验能力而无块校验能力。HDLC中的I帧按窗口序号顺序编号，BSC的数据块不编号。由于以上特点，HDLD的传输可靠性比BSC高。

数据透明性

HDLC采用“0比特插入法”对数据实现透明传输，传输信息的比特组合模式无任何限制。BSC用DLE字符填充法来实现透明传输，依赖于采用的字符编码集，且处理复杂。

信息传输格式

HDLC采用统一的帧格式来实现数据、命令、响应的传输，实施起来方便。而BSC的格式不统一，数据传送、正反向监控各规定了一套格式，给实施带来了不便

链路控制

HDLC利用改变一帧中的控制字段的编码模式来完成各种规定的链路操作功能，提供的是面向比特的传输功能。BSC则是通过改变控制字符来完成链路操作功能，提供的是面向字符的传输功能。

分类

数据链路层包含 LLC逻辑链路层子层 和MAC介质访问控制子层 两个子层

逻辑控制子层

数据链路层的LLC子层用于设备间单个连接的错误控制，流量控制。

与MAC层不同，LLC和物理媒介全无关系。媒介是CSMA/CD的802.3还是802.5的令牌环都没关系。它在LAN中是独立的802.2。在LLC之上的网络层可以是无连接、响应的无连接或面向连接的不同业务。

LLC用业务接入点SAP访问上层协议，有了SAP，站点就能在LLC层只用一个接口同时与几个高层协议玩。一个SAP是简单的地址或协议ID，内容则为空的LLC帧。LLC协议数据单元(LLCPDU)即LPDU。它包括：DSAP(目的SAP)/SSAP(源SAP);一个定义吞吐量优先级的控制域(Controlfield);和含带数据的信息域。在接收方，DSAP例如协议ID就是消息要被递送的，通常DSAP和SSAP是一样的，因为两端只有在同种协议间才能通信。例如当SAP为AA，代表SNAP(子层接入协议)。SNAP是个非标准化的，或厂商特定的协议，用于接入协议的业务。例如当SAP为06，则代表IP协议；当SAP为FO，代表NetBIOS(网络基本输入/输出协议)。SAP为FF表示广播的Global协议。

SNAP机制

规范种DSAP都只有一个字节，那不足够区分所有协议了。SNAP就来了，而且它支持在LLC帧上传厂商的协议。这样传递的协议就被放入所谓SNAP帧中了。SNAP”ㄍ酚形遄纸冢前三字节为厂商号，后两字节指示协议。

以太网有两种版本：IEEE(802.2、802.3); DIX(DEC、Intel、Xerox)又称V2以太网。

DIX帧在源地址之后是两字节的Type，例如IP。

IEEE以太网帧，在源地址后是帧长度指示，在随后的数据域中才是标准的LPDU封装，包括DSAP/SSAP/控制域/Data。在这个Data中会有协议ID、以太类型指示，例如以太类型806表示地址解析协议ARP。以太网卡通过跳针或软件可以设置需要的以太网版本，DIX通常都设的，因为大约90%的网络都用此版本。请注意网络所有要通信的站点应设成相同版本。

介质访问层

介质访问控制是解决当局域网中共用信道的使用产生竞争时，如何分配信道的使用权问题。

逻辑链路 Logical Links

Logical Links 逻辑链路逻辑链路是实际电路或逻辑电路上交换通信信息的两个端系统之间的一种协议驱动通信会话。协议栈定义了两个系统在某种介质上的通信。在协议栈低层定义可用的多种不同类型的通信协议，如局域网络(LAN)、城域网(MAN)和象X．25或帧中继这样的分组交换网络。逻辑链路在物理链路(可以是铜线、光纤或其他介质)上的两个通信系统之间形成。根据OSI协议模型，这些逻辑链路只在物理层以上存在。你可以认为逻辑链路是存在于网络两个末断系统间的线路。

面向连接的服务 为了保证可靠的通信，需要建立逻辑线路，但在两个端系统间要维持会话。

面向需要应答连接的服务 分组传输并有返回信号的逻辑线路。这种服务产生更大的开销，但更加可靠。

无应答不连接服务 无需应答和预先的传送。在端系统间没有会话。

OSI协议栈中的数据链路层可进一步细分为较低的介质访问控制(MAC)子层和较高的逻辑链路控制(LLC)子层。当它接收到一个分组后，它从MAC子层向上传送。如果有多个网络和设备相连，LLC层可能将分组送给另一个网络。例如，在一个NetWare服务器上，你可能既安装了以太网络适配器又安装了令牌网络适配器，NetWare自动地在连接到适配器的网络间桥接，这样原来在以太网上的分组就可以传送到令牌网上的目的地了，LLC层就象网络段间的交换或链路中继，它将以太网的帧重装成令牌环网的帧。

数据链路层

Uu接口为UE（User Equipment）与

UU口协议结构

UU口协议结构

UTRAN（UMTS Terrestrial Radio Access Network）之

间的接口，是UMTS系统的空中接口，也是最重要的接口。Uu接口可分为三个协议层：物理层（L1）、数据链路层（L2）和网络层（L3）。

L2包括MAC、RLC、PDCP、BMC四个子层，各部分实现功能不同

MAC（媒体接入控制）

MAC（Medium Access Control）子层主要功能是调度，把逻辑信道映射到传输信道，负责根据逻辑信道的瞬时源速率为各个传输信道选择适当的传输格式（TF,Transport Format）。MAC层主要有3类逻辑实体，第一类是MAC-b，负责处理广播信道数据；第二类是MAC-c，负责处理公共信道数据；第三类是MAC-d，负责处理专用信道数据。

MAC子层功能包括：

逻辑信道和传输信道之间的映射；

为每个传输信道选择适当的传送格式；

UE数据流之间的优先级处理；

UE之间采用动态预调度方法的优先级处理；

FACH上几个用户的数据流之间的优先级处理；

公共传输信道上UE的标识；

将高层PDU复接为通过传输信道传送给物理层的传送块，并将来自物理层的传送块复接

为高层PDU；

业务量检测；

动态传输信道类型切换；

透明RLC加密；

接入业务级别选择。

RLC（无线链路控制）

RLC（Radio Link Control）子层不仅承载控制面的数据，而且也承载用户面的数据。RLC子层有三种工作模式，分别是透明模式（TM,Transparent Mode）、非确认模式（UM,Unacknowledged Mode）和确认模式（AM,Acknowledged Mode），针对不同的业务采用不同的模式。

RLC子层功能包括：

数据的分割和重组，串联，填充，用户数据的传送；

错误检测，按序发送高层PDU，副本检测；

流量控制；

非证实数据传送模式序号检查；

协议错误检测和恢复；

加密；

挂起和恢复功能。

BMC（广播/组播控制）

BMC（广播/组播控制）Broadcast/MulticastControl protocol）负责控制多播/组播业务。

BMC子层功能包括：

小区广播消息的存储；

业务量监测和为CBS请求无线资源；

BMC消息的调度；

向UE发送BMC消息；

向高层（NAS）传递小区广播消息。

PDCP（分组数据汇聚协议）

PDCP（分组数据汇聚协议，PacketData Converge Protocol ）负责对IP包的报头进行压缩和解压缩，以提高空中接口无线资源的利用率。

PDCP子层功能包括：

在发送与接收实体中分别执行IP数据流的头部压缩与解压缩；

传输用户数据；

将非接入层送来的PDCP-SDU转发到RLC层，将多个不同的RB复用到同一个RLC实

体。

后来为解决EthernetII与802.3帧的兼容问题推出的Ethernet SNAP格式。

常用的数据链路层

TCP/IP支持多种不同的数据链路层协议，这取决于网络使用的硬件，如以太网、令牌环、FDDI等局域网，串行线路、X.25、帧中继FR、ATM等广域网。

早期，由于各网络公司的竞争，局域网没有形成统一标准，为了将各种局域网融合，将链路层拆分成两层，逻辑链路层（LLC）和媒体接入控制（MAC）（802.3封装：SAP），

90年代，以太网取得垄断地位，以太网成为局域网代名词。数据链路层仅需要MAC层，采用以太网帧格式(Ethernet V2封装：ARPA)。