物理层

**物理层**（或称物理层，Physical Layer）是[计算机网络](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%BD%91%E7%BB%9C" \t "_blank)[OSI模型](https://baike.baidu.com/item/OSI%E6%A8%A1%E5%9E%8B)中最低的一层。物理层规定:为传输数据所需要的物理链路创建、维持、拆除，而提供具有机械的，电子的，功能的和规范的特性。简单的说，物理层确保原始的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)可在各种物理媒体上传输。[局域网](https://baike.baidu.com/item/%E5%B1%80%E5%9F%9F%E7%BD%91)与[广域网](https://baike.baidu.com/item/%E5%B9%BF%E5%9F%9F%E7%BD%91)皆属第1、2层。

物理层是[OSI](https://baike.baidu.com/item/OSI)的第一层，它虽然处于最底层，却是整个开放系统的基础。物理层为设备之间的[数据通信](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%80%9A%E4%BF%A1" \t "_blank)提供传输媒体及互连设备，为[数据传输](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E8%BE%93)提供可靠的环境。如果您想要用尽量少的词来记住这个第一层，那就是“[信号](https://baike.baidu.com/item/%E4%BF%A1%E5%8F%B7)和[介质](https://baike.baidu.com/item/%E4%BB%8B%E8%B4%A8)”。

OSI采纳了各种现成的协议，其中有RS-232、RS-449、X.21、V.35、ISDN、以及FDDI、IEEE802.3、IEEE802.4、和IEEE802.5的物理层协议。

物理层要解决的主要问题：

（1）物理层要尽可能地屏蔽掉物理设备和传输媒体，通信手段的不同，使数据链路层感觉不到这些差异，只考虑完成本层的协议和服务。

（2）给其服务用户（数据链路层）在一条物理的传输媒体上传送和接收比特流（一般为串行按顺序传输的比特流）的能力，为此，物理层应该解决物理连接的建立、维持和释放问题。 （3）在两个相邻系统之间唯一地标识数据电路。[2]

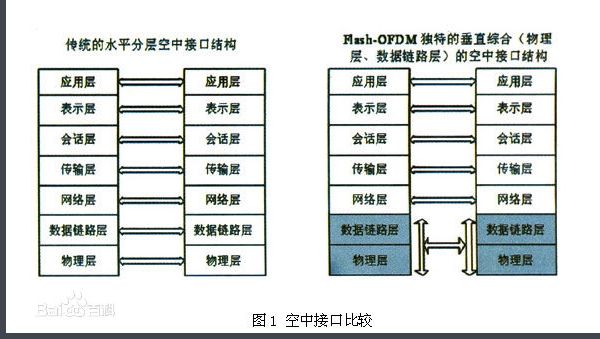
物理层主要功能：为数据端设备提供传送数据通路、传输数据。

1.为数据端设备提供传送数据的通路，[数据通路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%80%9A%E8%B7%AF" \t "_blank)可以是一个[物理媒体](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%AA%92%E4%BD%93)，也可以是多个物理媒体连接而成。一次完整的[数据传输](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E8%BE%93)，包括激活物理连接，传送数据，终止物理连接。所谓激活，就是不管有多少物理媒体参与，都要在通信的两个[数据终端设备](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%BB%88%E7%AB%AF%E8%AE%BE%E5%A4%87)间连接起来，形成一条通路。

2.传输数据，物理层要形成适合数据传输需要的[实体](https://baike.baidu.com/item/%E5%AE%9E%E4%BD%93" \t "_blank)，为数据传送服务。一是要保证数据能在其上正确通过，二是要提供足够的[带宽](https://baike.baidu.com/item/%E5%B8%A6%E5%AE%BD)（带宽是指每秒钟内能通过的[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9" \t "_blank)（BIT）数），以减少信道上的拥塞。传输数据的[方式](https://baike.baidu.com/item/%E6%96%B9%E5%BC%8F" \t "_blank)能满足点到点，一点到多点，串行或并行，半双工或全双工，同步或[异步传输](https://baike.baidu.com/item/%E5%BC%82%E6%AD%A5%E4%BC%A0%E8%BE%93)的需要。

3. 完成物理层的一些管理工作。

物理层的媒体包括架空明线、平衡电缆、[光纤](https://baike.baidu.com/item/%E5%85%89%E7%BA%A4" \t "_blank)、无线信道等。通信用的互连设备指DTE和DCE间的互连设

物理层

备。DTE即数据[终端](https://baike.baidu.com/item/%E7%BB%88%E7%AB%AF" \t "_blank)设备，又称[物理设备](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E8%AE%BE%E5%A4%87)，如计算机、终端等都包括在内。而DCE则是[数据通信](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%80%9A%E4%BF%A1" \t "_blank)设备或电路连接设备，如调制解调器等。[数据传输](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E8%BE%93)通常是经过DTE──DCE，再经过DCE──DTE的路径。互连设备指将DTE、DCE连接起来的装置，如各种插头、插座。LAN中的各种粗、细[同轴电缆](https://baike.baidu.com/item/%E5%90%8C%E8%BD%B4%E7%94%B5%E7%BC%86" \t "_blank)、T型接、插头，接收器，发送器，[中继器](https://baike.baidu.com/item/%E4%B8%AD%E7%BB%A7%E5%99%A8" \t "_blank)等都属物理层的媒体和[连接器](https://baike.baidu.com/item/%E8%BF%9E%E6%8E%A5%E5%99%A8)

数据链路层

[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层是[OSI参考模型](https://baike.baidu.com/item/OSI%E5%8F%82%E8%80%83%E6%A8%A1%E5%9E%8B)中的第二层，介乎于[物理层](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%B1%82)和[网络层](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B1%82)之间。[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层在[物理层](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%B1%82)提供的服务的基础上向[网络层](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B1%82)提供服务，其最基本的服务是将源自网络层来的数据可靠地传输到相邻节点的目标机网络层。为达到这一[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)，[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)必须具备一系列相应的[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)，主要有：如何将数据组合成[数据块](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%9D%97)，在数据链路层中称这种数据块为帧（frame），帧是数据链路层的传送[单位](https://baike.baidu.com/item/%E5%8D%95%E4%BD%8D)；如何[控制帧](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%B8%A7)在[物理信道](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E4%BF%A1%E9%81%93)上的传输，包括如何处理传输差错，如何调节发送[速率](https://baike.baidu.com/item/%E9%80%9F%E7%8E%87)以使与接收方相匹配；以及在两个网络实体之间提供数据链路通路的建立、维持和释放的管理。

移动[通信系统](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E7%B3%BB%E7%BB%9F" \t "_blank)中Uu口协议的第二层，也叫层二或L2。

基本功能

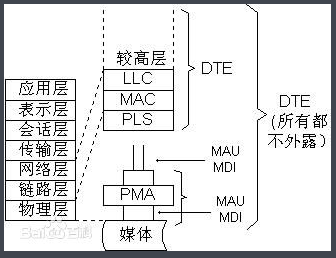
设计数据链路层的原因

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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层的最基本的[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)是向

数据链路层

该层[用户](https://baike.baidu.com/item/%E7%94%A8%E6%88%B7)提供[透明](https://baike.baidu.com/item/%E9%80%8F%E6%98%8E)的和可靠的[数据传送](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E9%80%81)基本服务。[透明性](https://baike.baidu.com/item/%E9%80%8F%E6%98%8E%E6%80%A7)是指该层上传输的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)的内容、格式及编码没有限制，也没有必要解释[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)结构的意义；可靠的传输使[用户](https://baike.baidu.com/item/%E7%94%A8%E6%88%B7)免去对丢失[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)、干扰信息及顺序不正确等的担心。在[物理层](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%B1%82)中这些[情况](https://baike.baidu.com/item/%E6%83%85%E5%86%B5)都可能发生，在[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层中必须用[纠错码](https://baike.baidu.com/item/%E7%BA%A0%E9%94%99%E7%A0%81)来检错与纠错。[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层是对[物理层](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%B1%82)传输原始[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)流的[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)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](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B1%82)表现为一无差错的[线路](https://baike.baidu.com/item/%E7%BA%BF%E8%B7%AF)。如果您想用尽量少的词来记住[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层，那就是：“[帧](https://baike.baidu.com/item/%E5%B8%A7)和[介质](https://baike.baidu.com/item/%E4%BB%8B%E8%B4%A8)访问控制”。

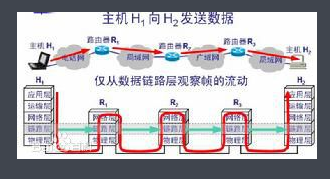
帧同步

为了使传输中发生差错后只将有错的有限[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)进行重发，[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层将[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)流组合成以帧为[单位](https://baike.baidu.com/item/%E5%8D%95%E4%BD%8D" \t "_blank)传送。每个帧除了要传送的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)外，还包括[校验码](https://baike.baidu.com/item/%E6%A0%A1%E9%AA%8C%E7%A0%81)，以使接收方能发现传输中的差错。帧的[组织](https://baike.baidu.com/item/%E7%BB%84%E7%BB%87)结构必须设计成使接收方能够 明确地从[物理层](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%B1%82)收到的[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)流中对其进行识别，也即能从[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)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](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E4%BC%A0%E8%BE%93)中很难保证计时的正确和一致，所以不可采用依靠[时间](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4)间隔[关系](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB)来确定一帧的起始与终止的方法。

（1）[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82)计数法：这是一种以一个特殊[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)表示一帧的起始并以一个专门[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)来标明帧内字节数的[帧同步](https://baike.baidu.com/item/%E5%B8%A7%E5%90%8C%E6%AD%A5)方法。接收方可以通过对该特殊[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)的识别从[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)流中区分出帧的起始并从专门[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)中获知该帧中随后跟随的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82)数，从而可确定出帧的终止[位置](https://baike.baidu.com/item/%E4%BD%8D%E7%BD%AE)。面向[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82)计数的同步规程的典型[代表](https://baike.baidu.com/item/%E4%BB%A3%E8%A1%A8)是DEC公司的数字[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)通信报文协议DDCMP（Digital Data Communications Message Protocol）。DDCMP采用的[帧格式](https://baike.baidu.com/item/%E5%B8%A7%E6%A0%BC%E5%BC%8F" \t "_blank)如图3-1。

[控制字符](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%AD%97%E7%AC%A6)[SOH](https://baike.baidu.com/item/SOH)标志[数据帧](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%B8%A7" \t "_blank)的起始。实际传输中，SOH前还要以两个或更多个同步[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)来确定一帧的起始，有时也允许本帧的头紧接着上帧的[尾](https://baike.baidu.com/item/%E5%B0%BE)，此时两帧间就不必再加同步字符。 count[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)共有14位，用以指示帧[中数据](https://baike.baidu.com/item/%E4%B8%AD%E6%95%B0%E6%8D%AE" \t "_blank)段中数据的[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82)数，14位[二进制](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6" \t "_blank)数的最大值为2-1=16383，所以数据最大[长度](https://baike.baidu.com/item/%E9%95%BF%E5%BA%A6" \t "_blank)为8×16383=131064。DDCMP协议就是靠这个[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82" \t "_blank)计数来确定帧的终止[位置](https://baike.baidu.com/item/%E4%BD%8D%E7%BD%AE)的。DDCMP[帧格式](https://baike.baidu.com/item/%E5%B8%A7%E6%A0%BC%E5%BC%8F)中的ACK、SEG、ADDR及FLAG中的第2位，CRC1、CRC2分别对标题部分和[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)部分进行双重校验，强调标题部分单独校验的[原因](https://baike.baidu.com/item/%E5%8E%9F%E5%9B%A0)是，一旦标题部分中的CONUT[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)出错，即失却了帧边[界](https://baike.baidu.com/item/%E7%95%8C" \t "_blank)划分的依据，将造成[灾难](https://baike.baidu.com/item/%E7%81%BE%E9%9A%BE)性的后果。由于采用[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)计数[方法](https://baike.baidu.com/item/%E6%96%B9%E6%B3%95)来确定帧的终止边界不会引起[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)及其它[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6" \t "_blank)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E9%80%8F%E6%98%8E%E6%80%A7" \t "_blank)。但这种方法使用起来比较麻烦，而且所用的特定[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)过份依赖于所采用的字符[编码](https://baike.baidu.com/item/%E7%BC%96%E7%A0%81)集，[兼容性](https://baike.baidu.com/item/%E5%85%BC%E5%AE%B9%E6%80%A7)比较差。

数据链路层

（3）使用[比特填充](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9%E5%A1%AB%E5%85%85" \t "_blank)的首尾标志法：该法以一组特定的比特[模式](https://baike.baidu.com/item/%E6%A8%A1%E5%BC%8F)（如01111110）来标志一帧的起始与终止。本章稍后要详细介绍的[HDLC](https://baike.baidu.com/item/HDLC" \t "_blank)规程即采用该法。为了不使[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)位中出现的与特定[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)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](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C" \t "_blank)，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](https://baike.baidu.com/item/%E5%8E%9F%E5%A7%8B" \t "_blank)信息，实现[数据传输](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E8%BE%93)的透明性。[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)填充很容易由[硬件](https://baike.baidu.com/item/%E7%A1%AC%E4%BB%B6)来实现，性能优于[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)填充方法。

（4）违法[编码法](https://baike.baidu.com/item/%E7%BC%96%E7%A0%81%E6%B3%95" \t "_blank)：该法在物理层采用特定的[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)编码方法时采用。例如，一种被称作[曼彻斯特编码](https://baike.baidu.com/item/%E6%9B%BC%E5%BD%BB%E6%96%AF%E7%89%B9%E7%BC%96%E7%A0%81)的方法，是将[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)“1”编码成“高－低”[电平](https://baike.baidu.com/item/%E7%94%B5%E5%B9%B3)对，而将数据[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)“0”编码成“低－高”电平对。而“高－高”电平对和“低－低”电平对在[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)比特中是违法的。可以借用这些违法[编码序列](https://baike.baidu.com/item/%E7%BC%96%E7%A0%81%E5%BA%8F%E5%88%97)来定界帧的起始与终止。[局域网](https://baike.baidu.com/item/%E5%B1%80%E5%9F%9F%E7%BD%91)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](https://baike.baidu.com/item/%E6%8A%80%E6%9C%AF" \t "_blank)，便能实现[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)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](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82)计数法中COUNT[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)的脆弱性以及[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)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](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)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](https://baike.baidu.com/item/%E8%8C%83%E5%9B%B4)内，这就是[差错控制](https://baike.baidu.com/item/%E5%B7%AE%E9%94%99%E6%8E%A7%E5%88%B6)过程，也是[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层的主要[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)之一。对差错编码（如[奇偶校验码](https://baike.baidu.com/item/%E5%A5%87%E5%81%B6%E6%A0%A1%E9%AA%8C%E7%A0%81" \t "_blank)，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](https://baike.baidu.com/item/%E9%94%99%E8%AF%AF)。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

数据链路层图3.1

反馈一个接收是否正确的[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)，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](https://baike.baidu.com/item/%E4%BF%A1%E5%8F%B7)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](https://baike.baidu.com/item/%E7%89%A9%E7%90%86)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF%E5%B8%A7" \t "_blank)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](https://baike.baidu.com/item/%E6%83%85%E5%86%B5" \t "_blank)，通常引入[计时器](https://baike.baidu.com/item/%E8%AE%A1%E6%97%B6%E5%99%A8)(Timer)来限定接收方发回反馈信息的[时间](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4" \t "_blank)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)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](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C)层的[危险](https://baike.baidu.com/item/%E5%8D%B1%E9%99%A9)。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](https://baike.baidu.com/item/%E5%BA%8F%E5%8F%B7)，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](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B1%82" \t "_blank)。[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)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](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B1%82)一次。

流量控制

[流量控制](https://baike.baidu.com/item/%E6%B5%81%E9%87%8F%E6%8E%A7%E5%88%B6)并不是[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层所特有的[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)，许多高层[协议](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层来说，控制的是相邻两节点之间数据链路上的[流量](https://baike.baidu.com/item/%E6%B5%81%E9%87%8F)，而对于[运输层](https://baike.baidu.com/item/%E8%BF%90%E8%BE%93%E5%B1%82)来说，控制的则是从源到最终[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)之间端的流量。由于收发[双方](https://baike.baidu.com/item/%E5%8F%8C%E6%96%B9)各自使用的设备工作[速率](https://baike.baidu.com/item/%E9%80%9F%E7%8E%87)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](https://baike.baidu.com/item/%E7%8E%B0%E8%B1%A1)，如若此时不对发送方的发送速率（也即[链路](https://baike.baidu.com/item/%E9%93%BE%E8%B7%AF)上的[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)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](https://baike.baidu.com/item/%E8%BF%87%E7%A8%8B)需要通过某种反馈[机制](https://baike.baidu.com/item/%E6%9C%BA%E5%88%B6)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](https://baike.baidu.com/item/%E6%83%85%E5%86%B5)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)后继续发送。

链路管理

链路[管理](https://baike.baidu.com/item/%E7%AE%A1%E7%90%86" \t "_blank)[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)主要用于面向连接的服

数据链路层

务。当链路两端的[节点](https://baike.baidu.com/item/%E8%8A%82%E7%82%B9" \t "_blank)要进行[通信](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1)前，必须首先确认对方已处于就绪[状态](https://baike.baidu.com/item/%E7%8A%B6%E6%80%81)，并交换一些必要的[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)以对帧序号[初始化](https://baike.baidu.com/item/%E5%88%9D%E5%A7%8B%E5%8C%96" \t "_blank)，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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)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](https://baike.baidu.com/item/%E6%83%85%E5%86%B5)下（例如在[LAN](https://baike.baidu.com/item/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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF" \t "_blank)层管理的[范畴](https://baike.baidu.com/item/%E8%8C%83%E7%95%B4)。

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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)的方法称为[反馈差错控制](https://baike.baidu.com/item/%E5%8F%8D%E9%A6%88%E5%B7%AE%E9%94%99%E6%8E%A7%E5%88%B6)。通常采用反馈检测和自动重发请求（[ARQ](https://baike.baidu.com/item/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](https://baike.baidu.com/item/%E5%9B%9E%E5%A3%B0)”法，主要用于面向[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6" \t "_blank)的[异步传输](https://baike.baidu.com/item/%E5%BC%82%E6%AD%A5%E4%BC%A0%E8%BE%93)中，如终端与远程[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA)间的通信，这是一种无须使用任何特殊[代码](https://baike.baidu.com/item/%E4%BB%A3%E7%A0%81)的错误检测法。双方进行[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)传输时，接收方将接收到的数据（可以是一个[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6" \t "_blank)，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](https://baike.baidu.com/item/%E5%8E%9F%E5%A7%8B%E6%95%B0%E6%8D%AE" \t "_blank)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)，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](https://baike.baidu.com/item/%E5%8E%9F%E7%90%86)简单、实现容易，也有较高的可靠性，但是，每个[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)均被传输两次，信道利用率很低。一般，在面向[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)的异步传输中，[信道](https://baike.baidu.com/item/%E4%BF%A1%E9%81%93)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

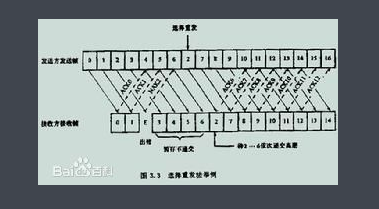
（ARQ法）：实用的差错控制方法，应该既要传输[可靠性](https://baike.baidu.com/item/%E5%8F%AF%E9%9D%A0%E6%80%A7" \t "_blank)高，又要信道利用率高。为此让发送方将要发送的数据帧附加一定的冗余检错码一并发送，接收方则根据检错码对数据帧进行错误检测，若发现错误，就返回请求重发的答，发送方收到请求重发的应答后，便重新传送该数据帧。这种差错控制方法就称为自动请求法(Automatic Repeat reQuest)，简称ARQ法。ARQ法仅返回很少的控制[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)，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](https://baike.baidu.com/item/%E6%96%B9%E6%A1%88" \t "_blank)，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](https://baike.baidu.com/item/%E8%A7%84%E5%AE%9A" \t "_blank)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)帧作为待确认帧保留在[缓冲存储器](https://baike.baidu.com/item/%E7%BC%93%E5%86%B2%E5%AD%98%E5%82%A8%E5%99%A8)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)帧时，便舍弃该帧。当接收方收到无差错的[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)帧后，即向发送方返回一个确认帧。 若发送方在规定[时间](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4" \t "_blank)内未能收到确认帧(即计时器超时)，则应重发存于[缓冲器](https://baike.baidu.com/item/%E7%BC%93%E5%86%B2%E5%99%A8" \t "_blank)中待确认[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)帧。若发送方在规定[时间](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4)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](https://baike.baidu.com/item/%E7%A9%BA%E9%97%B4" \t "_blank)，便可有效地实现[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)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](https://baike.baidu.com/item/%E4%BC%98%E7%82%B9" \t "_blank)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](https://baike.baidu.com/item/%E7%8E%AF%E5%A2%83)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)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

数据链路层图3-2

若干待确认的[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)帧。每当发送站收到对某[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E6%95%88%E7%8E%87" \t "_blank)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)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](https://baike.baidu.com/item/%E6%8B%B7%E8%B4%9D)。重发表按先进先出(FIFO)[队列](https://baike.baidu.com/item/%E9%98%9F%E5%88%97)规则操作。接收方对每一个正确收到的[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E5%BA%8F%E5%8F%B7)。当发送方收到相应[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)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](https://baike.baidu.com/item/%E6%96%B9%E5%BC%8F" \t "_blank)将确认帧插在双方的发送[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E6%83%85%E5%86%B5" \t "_blank)下描述的。如果差错出现，如何进一步处理可以有两种[策略](https://baike.baidu.com/item/%E7%AD%96%E7%95%A5)，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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)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](https://baike.baidu.com/item/%E9%AB%98%E5%B1%82" \t "_blank)递交[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)，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](https://baike.baidu.com/item/%E6%83%85%E5%86%B5)，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](https://baike.baidu.com/item/%E5%90%8D%E7%A7%B0" \t "_blank)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](https://baike.baidu.com/item/%E7%BC%93%E5%86%B2%E5%8C%BA)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)NAK2要求发送方选择重发2号帧。显然，选择重发减少了[浪费](https://baike.baidu.com/item/%E6%B5%AA%E8%B4%B9" \t "_blank)但[要求](https://baike.baidu.com/item/%E8%A6%81%E6%B1%82)接收方有足够大的缓冲区[容量](https://baike.baidu.com/item/%E5%AE%B9%E9%87%8F)。

流量控制

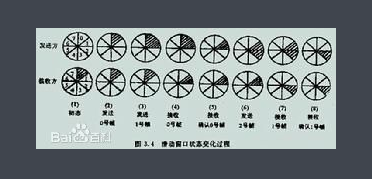
差错控制是[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)层[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)中的一个部分，另一个重要部分是流量控制。流量控制涉及链路上[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)或帧的传输速率的控制，以使接收方在接收前有足够的缓冲存储空间来接受每一个字符或帧。例如，在面向[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)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](https://baike.baidu.com/item/%E6%96%87%E4%BB%B6)的场合，仍会出现缓冲存储空间不够。[XON/XOFF](https://baike.baidu.com/item/XON%2FXOFF)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](https://baike.baidu.com/item/ASCII" \t "_blank)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](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)后便暂时停止发送[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)，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](https://baike.baidu.com/item/%E9%80%9A%E7%9F%A5" \t "_blank)发送方恢复数据发送。在一次[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)传输过程中，XOFF、XON的[周期](https://baike.baidu.com/item/%E5%91%A8%E6%9C%9F" \t "_blank)可重复多次，但对[用户](https://baike.baidu.com/item/%E7%94%A8%E6%88%B7)是[透明](https://baike.baidu.com/item/%E9%80%8F%E6%98%8E)的。许多异步[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)通信[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6)包均支持XON/XOFF协议。这种方案也可用于计算机向[打印机](https://baike.baidu.com/item/%E6%89%93%E5%8D%B0%E6%9C%BA" \t "_blank)或其它终端设备发送[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)，在这种[情况](https://baike.baidu.com/item/%E6%83%85%E5%86%B5)下，打印机或[终端设备](https://baike.baidu.com/item/%E7%BB%88%E7%AB%AF%E8%AE%BE%E5%A4%87)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

数据链路层图3-4

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](https://baike.baidu.com/item/%E7%AE%A1%E9%81%93" \t "_blank)(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](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6" \t "_blank)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](https://baike.baidu.com/item/%E7%BC%93%E5%86%B2%E5%AE%B9%E9%87%8F)而暂时过载。为此，可引入类似于空闲RQ方案的调整[措施](https://baike.baidu.com/item/%E6%8E%AA%E6%96%BD" \t "_blank)，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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)，此时送方的重发表增长，当达到重发[表](https://baike.baidu.com/item/%E8%A1%A8)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](https://baike.baidu.com/item/%E7%BC%93%E5%86%B2%E8%83%BD%E5%8A%9B" \t "_blank)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](https://baike.baidu.com/item/%E9%95%BF%E5%BA%A6" \t "_blank)、可使用的缓冲存容量以及传输的[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)速率等[因素](https://baike.baidu.com/item/%E5%9B%A0%E7%B4%A0)。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](https://baike.baidu.com/item/%E7%AA%97)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](https://baike.baidu.com/item/%E5%B0%BA%E5%AF%B8)。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](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)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](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3" \t "_blank)时，便停止发送新的帧。一般帧号只取有限位二[进制](https://baike.baidu.com/item/%E8%BF%9B%E5%88%B6)数，到一定[时间](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4)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](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3" \t "_blank)限度取值为2，则发送过程如图3-4所示。图中发送方[阴影](https://baike.baidu.com/item/%E9%98%B4%E5%BD%B1" \t "_blank)表示[发送窗口](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3)，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](https://baike.baidu.com/item/%E4%BD%8D%E7%BD%AE)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](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3" \t "_blank)为2，接收窗口为1）。

初始态，发送方没有帧发出，[发送窗口](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3)前后沿相等。接收窗口限度为1，它允许接收0号帧。

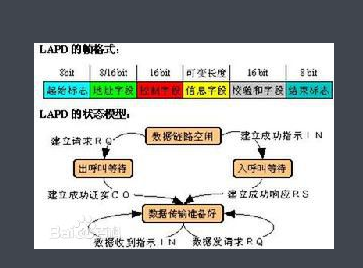
发送方已发送0号帧，此时发作口打开（即前沿加1），窗口对准0号，表示已发出但尚未收到确认返回信息。接收窗口状态同前，指示允许接收0帧。

发送方在未收到0帧的确认返回[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)前，继续发送1号帧。[发送窗口](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3" \t "_blank)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](https://baike.baidu.com/item/%E5%8F%91%E9%80%81" \t "_blank)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](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3" \t "_blank)前沿加1，表示2号帧也纳入待确认之列。接收

数据链路层

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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)，[发送窗口](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3)后沿加1，表示从重发表中[删除](https://baike.baidu.com/item/%E5%88%A0%E9%99%A4" \t "_blank)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](https://baike.baidu.com/item/%E6%BB%91%E5%8A%A8%E7%AA%97%E5%8F%A3%E5%8D%8F%E8%AE%AE" \t "_blank)，只不过其[发送窗口](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3)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](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3" \t "_blank)=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](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6%E7%BC%96%E7%A0%81" \t "_blank)，则最大序号为SMAX=2^3－1=7。对于有序接收方式，[发送窗口](https://baike.baidu.com/item/%E5%8F%91%E9%80%81%E7%AA%97%E5%8F%A3" \t "_blank)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](https://baike.baidu.com/item/OSI%E5%8F%82%E8%80%83%E6%A8%A1%E5%9E%8B)中的数据链路层协议。链路控制协议可分为[异步协议](https://baike.baidu.com/item/%E5%BC%82%E6%AD%A5%E5%8D%8F%E8%AE%AE)和[同步协议](https://baike.baidu.com/item/%E5%90%8C%E6%AD%A5%E5%8D%8F%E8%AE%AE)两大类。

[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)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](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)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](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)实现同步，但字符与字符之间的间隔[时间](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4)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](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4)内保持同步，所以可以用[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)起始处同步的时钟来采样该字符中的各[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)，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](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)都要添加诸如起始位、[校验位](https://baike.baidu.com/item/%E6%A0%A1%E9%AA%8C%E4%BD%8D)及停止位等[冗余位](https://baike.baidu.com/item/%E5%86%97%E4%BD%99%E4%BD%8D)，故信道利用率很低，一般用于[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)速率较低的场合。

同步协议

同步协议是以许多[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6" \t "_blank)或许多[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)[组织](https://baike.baidu.com/item/%E7%BB%84%E7%BB%87)成的[数据块](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%9D%97)－－帧为传输[单位](https://baike.baidu.com/item/%E5%8D%95%E4%BD%8D)，在帧的起始处同步，使帧内维持固定的[时钟](https://baike.baidu.com/item/%E6%97%B6%E9%92%9F)。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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)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](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)通常是由调解器来完成。由于采用帧为传输[单位](https://baike.baidu.com/item/%E5%8D%95%E4%BD%8D)，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](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)。同步协议又可分为面向[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)的同步协议、面向[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)的同步协议及面向[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82)计数的同步协议三种[类型](https://baike.baidu.com/item/%E7%B1%BB%E5%9E%8B)。

二进制协议

面向[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6" \t "_blank)的同步协议是最早提出的同步协议，其典型[代表](https://baike.baidu.com/item/%E4%BB%A3%E8%A1%A8)是[IBM](https://baike.baidu.com/item/IBM)公司的[二进制](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6" \t "_blank)[同步通信](https://baike.baidu.com/item/%E5%90%8C%E6%AD%A5%E9%80%9A%E4%BF%A1)(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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%80%9A%E4%BF%A1%E7%B3%BB%E7%BB%9F" \t "_blank)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)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](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)，除了正常传输的[数据块](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%9D%97)和报文外，还需要一些[控制字符](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%AD%97%E7%AC%A6)。 BSC协议用ASC2或EBCDIC[字符集](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6%E9%9B%86)定义的传输控制（TC）字符来实现相应[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD" \t "_blank)。这些传输[控制字符](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%AD%97%E7%AC%A6)的[标记](https://baike.baidu.com/item/%E6%A0%87%E8%AE%B0)、名称及ASC2码值和EBCDIC码值见表3.1。 各传输[控制字符](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%AD%97%E7%AC%A6)的[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)或报头的开始。

STX(Start of TEXT)：文始，标志标题[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)的结束和[报文](https://baike.baidu.com/item/%E6%8A%A5%E6%96%87)（块）[文本](https://baike.baidu.com/item/%E6%96%87%E6%9C%AC" \t "_blank)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](https://baike.baidu.com/item/%E8%AF%A2%E9%97%AE" \t "_blank)，用以请求远程站给出[响应](https://baike.baidu.com/item/%E5%93%8D%E5%BA%94)，响应可能包括站的[身份](https://baike.baidu.com/item/%E8%BA%AB%E4%BB%BD)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](https://baike.baidu.com/item/%E8%BD%AC%E4%B9%89)，用以修改紧跟其后的有限个[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6" \t "_blank)的意义。用于在BSC中实现[透明](https://baike.baidu.com/item/%E9%80%8F%E6%98%8E" \t "_blank)方式的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)传输，或者当10个传输[控制字符](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%AD%97%E7%AC%A6" \t "_blank)不够用时提供新的转义传输控制字符。

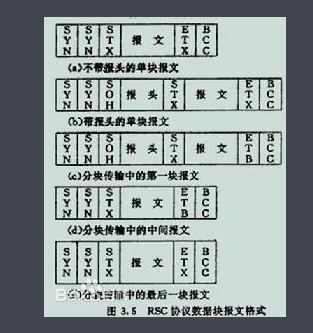
NAK(Negative Acknowledge)： 否认，由接收方发出的否定确认，作为对未正确接收来自发送方的响应。

SYN(Synchronous)： 同字符，在同步协议中，用以实现节点之间的[字符同步](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6%E5%90%8C%E6%AD%A5)，或用于在列[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%9D%97)时， 一个数据块的结束。

BSC 协议将在链路上传输的[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)分为[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)报文和监控报文又分为正向[监控](https://baike.baidu.com/item/%E7%9B%91%E6%8E%A7)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](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%AD%97%E7%AC%A6)，用以确定报文中[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)的性质或实现某种控制[作用](https://baike.baidu.com/item/%E4%BD%9C%E7%94%A8)。

[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)报文和文本组成。文本是要传送的有用[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)，而报文是与文本传送及处

数据链路层图3.5

理有关的辅助[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF" \t "_blank)，[报头](https://baike.baidu.com/item/%E6%8A%A5%E5%A4%B4)有时也可不用，对于不超过[长度](https://baike.baidu.com/item/%E9%95%BF%E5%BA%A6)限制的报文可只用一个[数据块](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%9D%97)作为一个传输[单位](https://baike.baidu.com/item/%E5%8D%95%E4%BD%8D)。接收方对于每一个收到的[数据块](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%9D%97)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%9D%97" \t "_blank)格式可以有5种，如图3.5所示。

BSC协议中所有发送的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)均跟在至少两个[SYT](https://baike.baidu.com/item/SYT)[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)之后，以使接收方能实现字符同步。报头[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)用以说明[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)文字段的包识别符（序号）及地址。所有[数据块](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%9D%97" \t "_blank)在块终[限定符](https://baike.baidu.com/item/%E9%99%90%E5%AE%9A%E7%AC%A6" \t "_blank)（ETX或ETB）之后不有块验[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6" \t "_blank)BCC（Block Check Charracter），BCC可以是垂直[奇偶校验](https://baike.baidu.com/item/%E5%A5%87%E5%81%B6%E6%A0%A1%E9%AA%8C)或16位[CRC](https://baike.baidu.com/item/CRC)，校验范围自STX始，至ETX或ETB止。

当发送的报文是[二进制](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6)[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)而不是[字符串](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6%E4%B8%B2)时，[二进制](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6)数据中形同传输[控制字符](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%AD%97%E7%AC%A6)的[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)串将会引传输混乱。为使二进制[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)中允许与传输[控制字符](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%AD%97%E7%AC%A6)相同的数据(即数据的[透明性](https://baike.baidu.com/item/%E9%80%8F%E6%98%8E%E6%80%A7)），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](https://baike.baidu.com/item/%E8%BD%AC%E4%B9%89%E5%AD%97%E7%AC%A6)，在发送时，若文本中也出现与DLE字符相同的二进制[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)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](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)，则知其后的DLE为[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)，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](https://baike.baidu.com/item/%E6%A0%BC%E5%BC%8F)。

(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

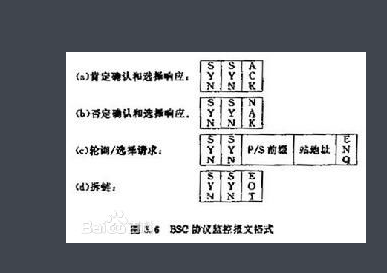
监控报文一般由单个传输[控制字符](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%AD%97%E7%AC%A6)或由若干个其它字符引导的单个传输控制字符组成。引导[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)统称为前缀，它包含识别符（序号）、地址[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)、[状态信息](https://baike.baidu.com/item/%E7%8A%B6%E6%80%81%E4%BF%A1%E6%81%AF)以及其它所需的信息。[ACK](https://baike.baidu.com/item/ACK)和[NAK](https://baike.baidu.com/item/NAK)监控报文的作用，首先作为对先前所发[数据块](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%9D%97)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)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](https://baike.baidu.com/item/%E7%BC%BA%E7%82%B9)，是它和特定的[字符编码](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6%E7%BC%96%E7%A0%81)集[关系](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB)过于密切，不利于兼容性。为了实现[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)的[透明性](https://baike.baidu.com/item/%E9%80%8F%E6%98%8E%E6%80%A7)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](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6%E7%BC%96%E7%A0%81)集。另外，由于BSC是一个半双工协议，它的链路[传输效率](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E6%95%88%E7%8E%87" \t "_blank)很低，即使物理连路支持[全双工传输](https://baike.baidu.com/item/%E5%85%A8%E5%8F%8C%E5%B7%A5%E4%BC%A0%E8%BE%93" \t "_blank)，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](https://baike.baidu.com/item/%E7%BB%88%E7%AB%AF)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](https://baike.baidu.com/item/%E5%85%AC%E5%8F%B8)率先提出了面向[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)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](https://baike.baidu.com/item/ADCCP)（Advanced Data Control Procedure），ISO的[高级数据链路控制规程](https://baike.baidu.com/item/%E9%AB%98%E7%BA%A7%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF%E6%8E%A7%E5%88%B6%E8%A7%84%E7%A8%8B)HDLC。[链路控制协议](https://baike.baidu.com/item/%E9%93%BE%E8%B7%AF%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE" \t "_blank)着重于对分段成物理块或包的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)的媒体的工具。所有面向[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)的数据[链路控制协议](https://baike.baidu.com/item/%E9%93%BE%E8%B7%AF%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE)均采用统一的[帧格式](https://baike.baidu.com/item/%E5%B8%A7%E6%A0%BC%E5%BC%8F)，不论是数据还是单独的控制[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)均以帧为[单位](https://baike.baidu.com/item/%E5%8D%95%E4%BD%8D" \t "_blank)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)的透明性，可以采用“0[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)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](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)，当发现有连续5个“1”出现

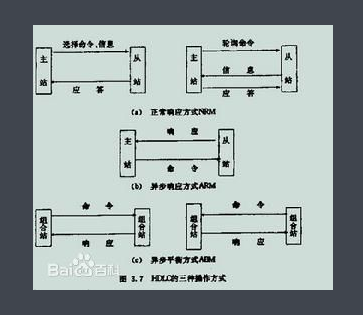
数据链路层图3-6

时，便在其后添插一个“0”，然后继续发后继的[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)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](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)“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](https://baike.baidu.com/item/%E6%83%85%E5%86%B5)，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](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)的协议的[帧格式](https://baike.baidu.com/item/%E5%B8%A7%E6%A0%BC%E5%BC%8F)中，有一个8比特的控制[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)，可以用它以[编码方式](https://baike.baidu.com/item/%E7%BC%96%E7%A0%81%E6%96%B9%E5%BC%8F)定义丰富的控制命令和应答，相当于起到了[BSC协议](https://baike.baidu.com/item/BSC%E5%8D%8F%E8%AE%AE)中众多传输[控制字符](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%AD%97%E7%AC%A6)和转义序列的[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)。作为面向[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)的数据[链路控制协议](https://baike.baidu.com/item/%E9%93%BE%E8%B7%AF%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE)的典型，HDLC具有如下特点：协议不依赖于任何一种[字符编码](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6%E7%BC%96%E7%A0%81)集；数据报文可[透明传输](https://baike.baidu.com/item/%E9%80%8F%E6%98%8E%E4%BC%A0%E8%BE%93)，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](https://baike.baidu.com/item/%E5%85%A8%E5%8F%8C%E5%B7%A5%E9%80%9A%E4%BF%A1)，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](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E6%95%88%E7%8E%87)；所有帧均采用CRC校验，对[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)管制协议。

**HDLC的操作方式**

HDLC是通用的数据[链路控制协议](https://baike.baidu.com/item/%E9%93%BE%E8%B7%AF%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE" \t "_blank)，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](https://baike.baidu.com/item/%E9%80%9A%E4%BF%97)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%B5%81)进行[组织](https://baike.baidu.com/item/%E7%BB%84%E7%BB%87)，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](https://baike.baidu.com/item/%E7%AB%99)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](https://baike.baidu.com/item/%E8%BD%AE%E8%AF%A2)[技术](https://baike.baidu.com/item/%E6%8A%80%E6%9C%AF)，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](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91)[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)，所以当终端与主机相连时，[主机](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA)一般总是主站。在一个站连接多条链中的[情况](https://baike.baidu.com/item/%E6%83%85%E5%86%B5)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](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)，这站称为组合站，用于组合站之间[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)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](https://baike.baidu.com/item/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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)。响应[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E8%BD%AE%E8%AF%A2)、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

数据链路层图3.7

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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)，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](https://baike.baidu.com/item/%E8%BD%AE%E8%AF%A2)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](https://baike.baidu.com/item/%E5%BC%82%E6%AD%A5%E5%B9%B3%E8%A1%A1%E6%96%B9%E5%BC%8F)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](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E6%95%88%E7%8E%87)，节点之间在两个[方向](https://baike.baidu.com/item/%E6%96%B9%E5%90%91)上都需要的较高的[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)和控制报文均以帧的[标准](https://baike.baidu.com/item/%E6%A0%87%E5%87%86)格式传送。HDLC中的帧类似于BSC的[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)块，但BSC协议中的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)报文和控制报文是[独立](https://baike.baidu.com/item/%E7%8B%AC%E7%AB%8B)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](https://baike.baidu.com/item/%E5%9C%B0%E5%9D%80%E5%AD%97%E6%AE%B5)（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](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)（F）：标志字段为01111110的[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)模式，用以标志帧的起始和前一帧的终止。标志[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)也可以作为帧与帧之间的填充[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)。通常，在不进行帧传送的[时刻](https://baike.baidu.com/item/%E6%97%B6%E5%88%BB)，信道仍处于[激活](https://baike.baidu.com/item/%E6%BF%80%E6%B4%BB)状态，在这种状态下，发方不断地发送标志[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)，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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)的[透明传输](https://baike.baidu.com/item/%E9%80%8F%E6%98%8E%E4%BC%A0%E8%BE%93)。

（2）地址[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)（A）：地址字段的[内容](https://baike.baidu.com/item/%E5%86%85%E5%AE%B9)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](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)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](https://baike.baidu.com/item/%E5%9C%B0%E5%9D%80)，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

数据链路层

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](https://baike.baidu.com/item/%E5%B9%BF%E6%92%AD%E5%9C%B0%E5%9D%80)，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](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)（C）：控制字段用于构成各种[命令](https://baike.baidu.com/item/%E5%91%BD%E4%BB%A4)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](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)来通知被[寻址](https://baike.baidu.com/item/%E5%AF%BB%E5%9D%80)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](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)是HDLC的关键。控制[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)中的第一位或第一、第二位表示传送帧的类型，HDLC中有[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)帧（I帧）、监控帧（S帧）和[无编号帧](https://baike.baidu.com/item/%E6%97%A0%E7%BC%96%E5%8F%B7%E5%B8%A7)（U帧）三种不同类型的帧。控制[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)的第五位是P/F位，即轮询/[终止](https://baike.baidu.com/item/%E7%BB%88%E6%AD%A2)（Poll/Final）位。

（4）[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)（I）：信息字段可以是任意的[二进制](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6)比特串。[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)串[长度](https://baike.baidu.com/item/%E9%95%BF%E5%BA%A6)未作限定，其上限由FCS[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)或通信站的缓冲器容量来决定，[国际](https://baike.baidu.com/item/%E5%9B%BD%E9%99%85)上用得较多的是1000~2000比特；而下限可以为0，即无[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)字段。但是，监控帧（S帧）中规定不可有[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)。（5）[帧校验序列](https://baike.baidu.com/item/%E5%B8%A7%E6%A0%A1%E9%AA%8C%E5%BA%8F%E5%88%97)[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)（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](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)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](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)的格式及[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)定义见图3.9。

（1）[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)帧（I帧）：[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)帧用于传送有效信息或[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)，通常简称I帧。I帧以控制字第一位为“0”来标志。[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)帧的控制[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)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](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6)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](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5" \t "_blank)第一、二位为“10”来标志。S帧带信息[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5" \t "_blank)，只有6个[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82)即48个[比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9)。S帧的控制[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)的第三、四位为S[帧类型](https://baike.baidu.com/item/%E5%B8%A7%E7%B1%BB%E5%9E%8B)编码，共有四[种](https://baike.baidu.com/item/%E7%A7%8D)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](https://baike.baidu.com/item/%E8%BD%AE%E8%AF%A2" \t "_blank)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

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。

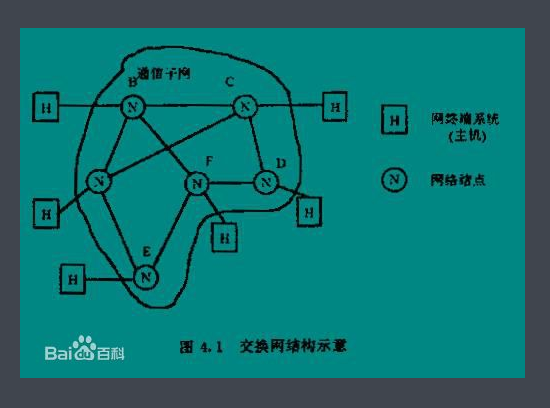
可以看出，接收就绪RR型S帧和接收未就绪RNR型S帧有两个主要[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)：首先，这两种类型的S帧用来表示从站已准备好或未准备好接收[信息](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF)；其次，确认编号小于N（R）的所有接收到的I帧。拒绝REJ和选择拒绝SREJ型S帧，用于向对方站指出发生了差错。REJ帧用于GO-back-N策略，用以请求重发N（R）以前的帧已被确认，当收到一个N（S）等于REJ型S帧的N（R）的I帧后,REJ状态即可清除。SREJ帧用于[选择重发策略](https://baike.baidu.com/item/%E9%80%89%E6%8B%A9%E9%87%8D%E5%8F%91%E7%AD%96%E7%95%A5)，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](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)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](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)，这些控制功能5个M位（M1、M2、M3、M4、M5，也称修正位）来[定义](https://baike.baidu.com/item/%E5%AE%9A%E4%B9%89)。5个M位可以定义32种附加的命令[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD)或32种[应答](https://baike.baidu.com/item/%E5%BA%94%E7%AD%94)功能，但许多是空缺的。

网络层

网络层是[OSI参考模型](https://baike.baidu.com/item/OSI%E5%8F%82%E8%80%83%E6%A8%A1%E5%9E%8B)中的第三层，介于[传输层](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E5%B1%82)和[数据链路层](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF%E5%B1%82)之间，它在数据链路层提供的两个相邻端点之间的数据帧的传送功能上，进一步管理网络中的[数据通信](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%80%9A%E4%BF%A1)，将数据设法从源端经过若干个中间[节点](https://baike.baidu.com/item/%E8%8A%82%E7%82%B9)传送到[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)端，从而向运输层提供最基本的端到端的[数据传送](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E9%80%81)服务。主要内容有：虚电路分组交换和[数据报](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8A%A5)分组交换、[路由选择](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E9%80%89%E6%8B%A9)[算法](https://baike.baidu.com/item/%E7%AE%97%E6%B3%95)、阻塞控制方法、[X.25协议](https://baike.baidu.com/item/X.25%E5%8D%8F%E8%AE%AE)、综合业务数据网（ISDN）、[异步传输模式](https://baike.baidu.com/item/%E5%BC%82%E6%AD%A5%E4%BC%A0%E8%BE%93%E6%A8%A1%E5%BC%8F" \t "_blank)（ATM）及网际互连[原理](https://baike.baidu.com/item/%E5%8E%9F%E7%90%86)与实现。

网络层的[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84" \t "_blank)是实现两个[端系统](https://baike.baidu.com/item/%E7%AB%AF%E7%B3%BB%E7%BB%9F)之间的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)透明传送，具体功能包括[寻址](https://baike.baidu.com/item/%E5%AF%BB%E5%9D%80)和[路由选择](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E9%80%89%E6%8B%A9)、连接的建立、保持和终止等。它提供的服务使[传输层](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E5%B1%82)不需要了解网络中的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)传输和[交换技术](https://baike.baidu.com/item/%E4%BA%A4%E6%8D%A2%E6%8A%80%E6%9C%AF)。如果您想用尽量少的词来记住网络层，那就是“路径选择、[路由](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1" \t "_blank)及[逻辑寻址](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E5%AF%BB%E5%9D%80)”。

网络层

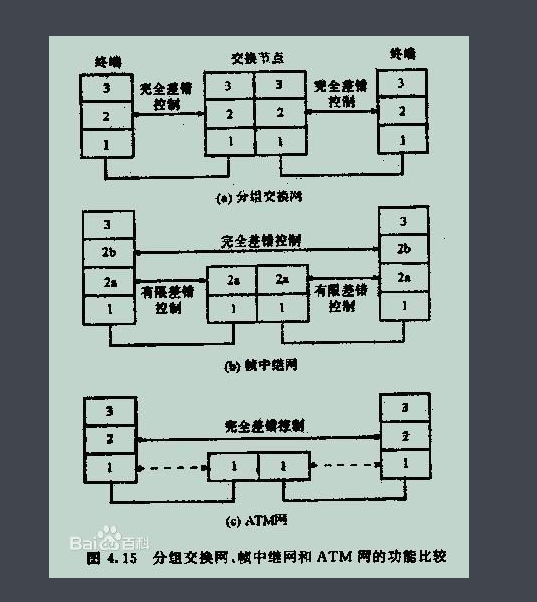
为了说明网络层的功能，如图4.1所示的交换[网络拓扑结构](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E6%8B%93%E6%89%91%E7%BB%93%E6%9E%84" \t "_blank)，它是由若干个网络[节点](https://baike.baidu.com/item/%E8%8A%82%E7%82%B9)按照任意的拓扑结构相互连接而成的。网络层关系到[通信子网](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E5%AD%90%E7%BD%91)的运行控制，体现了网络应用环境中资源子网[访问](https://baike.baidu.com/item/%E8%AE%BF%E9%97%AE)通信子网的[方式](https://baike.baidu.com/item/%E6%96%B9%E5%BC%8F)。网络层从物理上来讲一般分布地域宽广，从逻辑上来讲功能复杂，因此是[OSI模型](https://baike.baidu.com/item/OSI%E6%A8%A1%E5%9E%8B)中面向[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)通信的下三层（也即[通信子网](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E5%AD%90%E7%BD%91" \t "_blank)）中最为复杂也最关键的一层。

## 网络层协议

TCP/IP网络层的核心是[IP协议](https://baike.baidu.com/item/IP%E5%8D%8F%E8%AE%AE" \t "_blank)，它是[TCP/IP](https://baike.baidu.com/item/TCP%2FIP)协议族中最主要的协议之一。IP协议非常[简单](https://baike.baidu.com/item/%E7%AE%80%E5%8D%95" \t "_blank)，仅仅提供不可靠、无连接的传送服务。IP协议的主要功能有：无连接[数据报](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8A%A5" \t "_blank)传输、数据报[路由选择](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E9%80%89%E6%8B%A9)和差错控制。与IP协议配套使用实现其功能的还有[地址解析协议](https://baike.baidu.com/item/%E5%9C%B0%E5%9D%80%E8%A7%A3%E6%9E%90%E5%8D%8F%E8%AE%AE" \t "_blank)ARP、[逆地址解析协议](https://baike.baidu.com/item/%E9%80%86%E5%9C%B0%E5%9D%80%E8%A7%A3%E6%9E%90%E5%8D%8F%E8%AE%AE" \t "_blank)RARP、因特网报文协议ICMP、因特网组管理协议IGMP。

### IP地址

TCP/[IP网络](https://baike.baidu.com/item/IP%E7%BD%91%E7%BB%9C)使用32位长度的地址以标识一台[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA)和同它相连的网络，它的[格式](https://baike.baidu.com/item/%E6%A0%BC%E5%BC%8F)为：IP地址=网

网络层

络地址+主机地址。IP地址是通过它的格式分类的，它有四种格式：A类、B类、C类、D类。如下所示

格式位数[主机地址](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA%E5%9C%B0%E5%9D%80)：A类0网络（7位）主机地址（24位）、

B类10网络（14位）主机地址（16位）、C类110网络（21位）主机地址（8位）、D类1110多路通信[地址](https://baike.baidu.com/item/%E5%9C%B0%E5%9D%80" \t "_blank)（28位）、未来的格式11110将来使用。这样，[A类地址](https://baike.baidu.com/item/A%E7%B1%BB%E5%9C%B0%E5%9D%80" \t "_blank)空间为0-127，最大网络数为126，最大主机数为16,777,124；[B类地址](https://baike.baidu.com/item/B%E7%B1%BB%E5%9C%B0%E5%9D%80" \t "_blank)空间为128-191，最大网络数为16384，最大主机数为65,534；[C类地址](https://baike.baidu.com/item/C%E7%B1%BB%E5%9C%B0%E5%9D%80" \t "_blank)空间为192-223，最大网络数为2,097,152，最大主机数为254；D类[地址空间](https://baike.baidu.com/item/%E5%9C%B0%E5%9D%80%E7%A9%BA%E9%97%B4" \t "_blank)为224-254。 C类地址空间分配概况。分配区域地址空间：多区域192.0.0.0~193.255.255.255、[欧洲](https://baike.baidu.com/item/%E6%AC%A7%E6%B4%B2" \t "_blank)：194.0.0.0~195.255.255.255、其他：196.0.0.0~197.255.255.255、[北美](https://baike.baidu.com/item/%E5%8C%97%E7%BE%8E" \t "_blank)：197.0.0.0~199.255.255.255、中南美：200.0.0.0~201.255.255.255、[太平洋](https://baike.baidu.com/item/%E5%A4%AA%E5%B9%B3%E6%B4%8B" \t "_blank)地区：202.0.0.0~203.255.255.255、其他：204.0.0.0~205.255.255.255、其他：206.0.0.0~207.255.255.255。注：其中“多区域”表示执行该计划前已经分配的地址[空间](https://baike.baidu.com/item/%E7%A9%BA%E9%97%B4" \t "_blank)；“其他”表示已指定[名称](https://baike.baidu.com/item/%E5%90%8D%E7%A7%B0" \t "_blank)的地区之外的地理区划。

特殊格式的IP地址：[广播地址](https://baike.baidu.com/item/%E5%B9%BF%E6%92%AD%E5%9C%B0%E5%9D%80" \t "_blank)：当网络或主机标志符[字段](https://baike.baidu.com/item/%E5%AD%97%E6%AE%B5)的每位均设置为1时，这个地址编码标识着该[数据报](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8A%A5" \t "_blank)是一个广播式的通信，该数据报可以被发送到网络中所有的[子网](https://baike.baidu.com/item/%E5%AD%90%E7%BD%91)和主机。例如，地址128.2.255.255意味着网络128.2上所有的主机。本[网络地址](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%9C%B0%E5%9D%80" \t "_blank)：IP地址的主机[标识符](https://baike.baidu.com/item/%E6%A0%87%E8%AF%86%E7%AC%A6" \t "_blank)字段也可全部设置为0，表示该地址作为“本主机”地址。网络标识符字段也可全部设置为0，表示“本网络”。如，128.2.0.0表示网络地址为128.2的网络。使用网络标识符字段全部设置为0的IP地址在一台主机不知道网络的IP地址时时是很有用的。私有的[IP地址](https://baike.baidu.com/item/IP%E5%9C%B0%E5%9D%80" \t "_blank)：在有些情况下，一个机构并不需要连接到Internet或另一个专有的网络上，因此，无须遵守对IP地址进行申请和登记的规定。该机构可以使用任何的地址。在[RFC1597](https://baike.baidu.com/item/RFC1597)中，有些IP地址是用作私用地址的：A类地址：10.0.0.0到10.255.255.255。B类地址：172.16.0.0到172.31.255.255.255。C类地址：192.168.0.0到192.168.255.255。

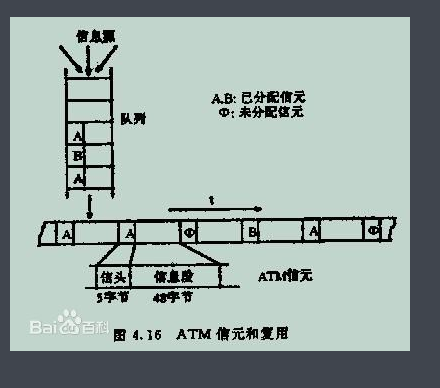
### 地址解析

[ARP](https://baike.baidu.com/item/ARP)协议是“AddressResolutionProtocol”（地址解析协议）的缩写。在[局域网](https://baike.baidu.com/item/%E5%B1%80%E5%9F%9F%E7%BD%91" \t "_blank)中，网络中实际传输的是“帧”，帧里面是有目标主机的MAC地址的。在以太网中，一个主机要和另一个主机进行直接通信，必须要知道目标主机的MAC地址。但这个目标[MAC](https://baike.baidu.com/item/MAC" \t "_blank)地址是如何获得的呢？它就是通过[地址解析](https://baike.baidu.com/item/%E5%9C%B0%E5%9D%80%E8%A7%A3%E6%9E%90" \t "_blank)协议获得的。所谓“地址解析”就是主机在发送[帧](https://baike.baidu.com/item/%E5%B8%A7" \t "_blank)前将目标IP地址转换成目标MAC地址的过程。ARP协议的基本[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD" \t "_blank)就是通过目标设备的IP地址，查询目标设备的MAC地址，以保证通信的顺利进行。协议属于[链路层](https://baike.baidu.com/item/%E9%93%BE%E8%B7%AF%E5%B1%82" \t "_blank)的协议在以太网中的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)帧从一个主机到达网内的另一台主机是根据48位的以太网地址（[硬件](https://baike.baidu.com/item/%E7%A1%AC%E4%BB%B6" \t "_blank)地址）来确定接口的，而不是根据32位的IP地址。内核（如[驱动](https://baike.baidu.com/item/%E9%A9%B1%E5%8A%A8" \t "_blank)）必须知道[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84" \t "_blank)端的硬件地址才能发送[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)。当然，[点对点](https://baike.baidu.com/item/%E7%82%B9%E5%AF%B9%E7%82%B9)的连接是不需要ARP协议的。 ARP协议的[数据结构](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%BB%93%E6%9E%84" \t "_blank)：

以下是引用片段：

typedefstructarphdr

{

网络层

unsignedshortarp\_hrd;/\*硬件类型\*/

unsignedshortarp\_pro;/\*协议类型\*/

unsignedchararp\_hln;/\*硬件地址长度\*/

unsignedchararp\_pln;/\*[协议地址](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE%E5%9C%B0%E5%9D%80)长度\*/

unsignedshortarp\_op;/\*ARP操作类型\*/

unsignedchararp\_sha[6];/\*发送者的硬件地址\*/

unsignedlongarp\_spa;/\*发送者的协议地址\*/

unsignedchararp\_tha[6];/\*目标的硬件地址\*/

unsignedlongarp\_tpa;/\*目标的协议地址\*/

}ARPHDR,\*PARPHDR; 为了[解释](https://baike.baidu.com/item/%E8%A7%A3%E9%87%8A" \t "_blank)ARP协议的[作用](https://baike.baidu.com/item/%E4%BD%9C%E7%94%A8" \t "_blank)，就必须理解[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)在网络上的传输过程。这里举一个[简单](https://baike.baidu.com/item/%E7%AE%80%E5%8D%95)的PING[例子](https://baike.baidu.com/item/%E4%BE%8B%E5%AD%90)。

假设我们的计算机IP地址是192.168.1.1，要执行这个命令：ping192.168.1.2.该命令会通过[ICMP协议](https://baike.baidu.com/item/ICMP%E5%8D%8F%E8%AE%AE" \t "_blank)发送ICMP[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)包。

该过程需要经过下面的步骤：

1、应用程序构造[数据包](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%8C%85" \t "_blank)，该示例是产生ICMP包，被提交给内核（网络驱动程序）；

2、内核检查是否能够转化该IP地址为MAC地址，也就是在本地的ARP缓存中查看IP-MAC对应表；

3、如果存在该IP-MAC对应关系，那么跳到[步骤](https://baike.baidu.com/item/%E6%AD%A5%E9%AA%A4)9；如果不存在该IP-MAC对应关系，那么接续下面的步骤；

4、内核进行ARP[广播](https://baike.baidu.com/item/%E5%B9%BF%E6%92%AD)，[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)地的MAC地址是FF-FF-FF-FF-FF-FF，ARP命令类型为REQUEST（1），其中包含有自己的MAC地址；

5、当192.168.1.2主机接收到该ARP[请求](https://baike.baidu.com/item/%E8%AF%B7%E6%B1%82)后，就发送一个ARP的REPLY（2）命令，其中包含自己的MAC地址；

6、本地获得192.168.1.2主机的IP-MAC地址对应关系，并保存到ARP缓存中；

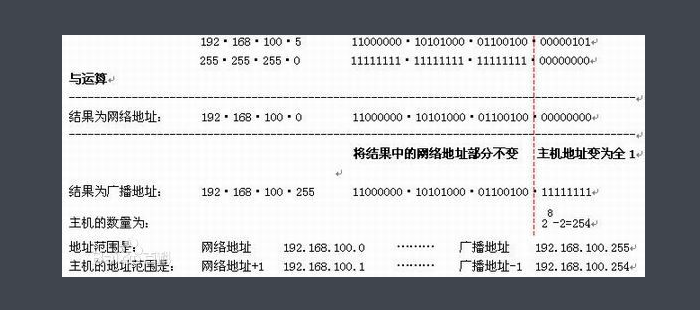
7、[内核](https://baike.baidu.com/item/%E5%86%85%E6%A0%B8" \t "_blank)将把IP转化为MAC地址，然后封装在以太网头结构中，再把数据发送出去；

使用arp-a命令就可以查看本地的ARP缓存内容，所以，执行一个本地的[PING命令](https://baike.baidu.com/item/PING%E5%91%BD%E4%BB%A4" \t "_blank)后，ARP[缓存](https://baike.baidu.com/item/%E7%BC%93%E5%AD%98)就会存在一个目的IP的[记录](https://baike.baidu.com/item/%E8%AE%B0%E5%BD%95" \t "_blank)了。当然，如果你的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)包是发送到不同[网段](https://baike.baidu.com/item/%E7%BD%91%E6%AE%B5)的[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)地，那么就一定存在一条[网关](https://baike.baidu.com/item/%E7%BD%91%E5%85%B3)的IP-MAC地址对应的记录。知道了ARP协议的作用，就能够很清楚地知道，[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)包的向外传输很依靠ARP协议，当然，也就是依赖ARP缓存。要知道，ARP协议的所有操作都是内核自动完成的，同其他的应用程序没有任何关系。同时需要注意的是，ARP协议只使用于本网络。

### 逆地址

具有本地[磁盘](https://baike.baidu.com/item/%E7%A3%81%E7%9B%98" \t "_blank)的[系统](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F)引导时，一般是从磁盘上的配置文件中读取IP地址。但是[无盘机](https://baike.baidu.com/item/%E6%97%A0%E7%9B%98%E6%9C%BA" \t "_blank)，如X[终端](https://baike.baidu.com/item/%E7%BB%88%E7%AB%AF)或[无盘工作站](https://baike.baidu.com/item/%E6%97%A0%E7%9B%98%E5%B7%A5%E4%BD%9C%E7%AB%99)，则需要采用其他方法来获得IP地址。网络上的每个系统都具有唯一的硬件地址，它是由网络接口[生产](https://baike.baidu.com/item/%E7%94%9F%E4%BA%A7" \t "_blank)厂家配置的。[无盘系统](https://baike.baidu.com/item/%E6%97%A0%E7%9B%98%E7%B3%BB%E7%BB%9F)的RARP实现过程是从接口卡上读取唯一的硬件地址，然后发送一份RARP请求（一帧在网络上广播的[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "_blank)），请求某个主机[响应](https://baike.baidu.com/item/%E5%93%8D%E5%BA%94" \t "_blank)该无盘系统的IP地址（在RARP应答中）。在概念上这个过程是很[简单](https://baike.baidu.com/item/%E7%AE%80%E5%8D%95" \t "_blank)的，但是实现起来常常比ARP要困难。RARP的正式[规范](https://baike.baidu.com/item/%E8%A7%84%E8%8C%83" \t "_blank)是RFC903[Finlaysonetal.1984]。 RARP的分组格：RARP分组的格式与ARP分组基本一致。它们之间主要的差别是RARP请求或应答的[帧类型](https://baike.baidu.com/item/%E5%B8%A7%E7%B1%BB%E5%9E%8B" \t "_blank)[代码](https://baike.baidu.com/item/%E4%BB%A3%E7%A0%81)为0x8035，而且RARP请求的操作代码为3，应答操作代码为4。对应于ARP，RARP请求以广播[方式](https://baike.baidu.com/item/%E6%96%B9%E5%BC%8F" \t "_blank)传送，而RARP应答一般是[单播](https://baike.baidu.com/item/%E5%8D%95%E6%92%AD" \t "_blank)(unicast)传送的。RARP[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)的设计：虽然RARP在[概念](https://baike.baidu.com/item/%E6%A6%82%E5%BF%B5" \t "_blank)上很[简单](https://baike.baidu.com/item/%E7%AE%80%E5%8D%95)，但是一个RARP服务器的设计与系统相关而且比较复杂。相反，提供一个ARP服务器很[简单](https://baike.baidu.com/item/%E7%AE%80%E5%8D%95" \t "_blank)，通常是TCP/IP在内核中实现的一部分。由于内核知道IP地址和硬件地址，因此当它收到一个询问IP地址的ARP请求时，只需用相应的硬件地址来提供[应答](https://baike.baidu.com/item/%E5%BA%94%E7%AD%94" \t "_blank)就可以了。

作为用户进程的RARP服务器：RARP服务器的复杂性在于，服务器一般要为多个主机（网络上所有的[无盘系统](https://baike.baidu.com/item/%E6%97%A0%E7%9B%98%E7%B3%BB%E7%BB%9F" \t "_blank)）提供硬件地址到IP地址的[映射](https://baike.baidu.com/item/%E6%98%A0%E5%B0%84" \t "_blank)。该映射包含在一个磁盘[文件](https://baike.baidu.com/item/%E6%96%87%E4%BB%B6)中。由于内核一般不读取和分析磁盘文件，因此RARP服务器的功能就由用户[进程](https://baike.baidu.com/item/%E8%BF%9B%E7%A8%8B" \t "_blank)来提供，而不是作为内核的实现的一部分。更为复杂的是，RARP请求是作为一个特殊类型的以太网[数据帧](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%B8%A7" \t "_blank)来传送的。这说明RARP服务器必须能够发送和接收这种类型的[以太网](https://baike.baidu.com/item/%E4%BB%A5%E5%A4%AA%E7%BD%91" \t "_blank)[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)帧。在附录A中，我们描述了SBD分组过滤器、SUN的网络接口栓以及[SVR4](https://baike.baidu.com/item/SVR4" \t "_blank)[数据链路](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF)提供者接口都可用来接收这些数据帧。由于发送和接收这些[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)帧与系统有关，因此RARP服务器的实现是与系统捆绑在一起的。

网络层

每个网络有多个RARP服务器：RARP服务器实现的一个复杂[因素](https://baike.baidu.com/item/%E5%9B%A0%E7%B4%A0" \t "_blank)是RARP请求是在硬件层上进行广播的，这意味着它们不经过[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)进行转发。为了让[无盘系统](https://baike.baidu.com/item/%E6%97%A0%E7%9B%98%E7%B3%BB%E7%BB%9F)在RARP服务器关机的状态下也能引导，通常在一个网络上（例如一根电缆）要提供多个RARP服务器。当服务器的数目增加时（以提供[冗余](https://baike.baidu.com/item/%E5%86%97%E4%BD%99" \t "_blank)备份），网络[流量](https://baike.baidu.com/item/%E6%B5%81%E9%87%8F" \t "_blank)也随之增加，因为每个服务器对每个RARP请求都要发送RARP应答。发送RARP请求的[无盘系统](https://baike.baidu.com/item/%E6%97%A0%E7%9B%98%E7%B3%BB%E7%BB%9F" \t "_blank)一般采用最先收到的RARP应答（对于ARP，我们从来没有遇到这种[情况](https://baike.baidu.com/item/%E6%83%85%E5%86%B5" \t "_blank)，因为只有一台主机发送ARP应答）。另外，还有一种可能发生的情况是每个RARP服务器同时应答，这样会增加以太网发生冲突的[概率](https://baike.baidu.com/item/%E6%A6%82%E7%8E%87" \t "_blank)。

### 因特网报文

ICMP的作用：由于IP协议的两个[缺陷](https://baike.baidu.com/item/%E7%BC%BA%E9%99%B7" \t "_blank)：没有差错控制和查询[机制](https://baike.baidu.com/item/%E6%9C%BA%E5%88%B6)，因此产生了ICMP。ICMP主要是为了提高IP[数据报](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8A%A5)[成功](https://baike.baidu.com/item/%E6%88%90%E5%8A%9F)交付的[机会](https://baike.baidu.com/item/%E6%9C%BA%E4%BC%9A)，在IP数据报传输的过程中进行差错报告和查询，比如[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84" \t "_blank)主机或网络不可到达，报文被丢弃，路由阻塞，查询目的网络是否可以到达等等。

ICMP有两种报文[类型](https://baike.baidu.com/item/%E7%B1%BB%E5%9E%8B" \t "_blank)：差错报告报文和询问报文。差错报告报文：终点不可到达（由于[路由表](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E8%A1%A8" \t "_blank)，硬件故障，协议不可到达，[端口](https://baike.baidu.com/item/%E7%AB%AF%E5%8F%A3)不可达到等原因导致，这时[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8)或[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)主机向源站发送终点不可到达报文）；源站抑制（发生拥塞，平衡IP协议没有[流量](https://baike.baidu.com/item/%E6%B5%81%E9%87%8F" \t "_blank)控制的缺陷）；[超时](https://baike.baidu.com/item/%E8%B6%85%E6%97%B6" \t "_blank)（环路或生存时间为0）；[参数](https://baike.baidu.com/item/%E5%8F%82%E6%95%B0" \t "_blank)问题（IP[数据报](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8A%A5)首部参数有二义性）；改变路由（路由[错误](https://baike.baidu.com/item/%E9%94%99%E8%AF%AF" \t "_blank)或不是最佳）。询问报文：回送请求或回答（用来测试连通性，如：PING[命令](https://baike.baidu.com/item/%E5%91%BD%E4%BB%A4)）；[时间戳](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4%E6%88%B3)请求或回答（用来计算往返时间或同步两者时间）；地址[掩码](https://baike.baidu.com/item/%E6%8E%A9%E7%A0%81" \t "_blank)请求或回答（得到掩码信息）；[路由](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1" \t "_blank)[询问](https://baike.baidu.com/item/%E8%AF%A2%E9%97%AE)或通告（得知网络上的[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)信息）。ICMP是网际(IP)层的协议，它作为IP层[数据报](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8A%A5" \t "_blank)的数据，加上数据报的首部，组成数据报发送出去。

### PING

[应用层](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E5%B1%82)的PING(PacketInterNetGroper)命令用来测试两个主机之间的连通性，PING使用了ICMP回送请求与回送回答报文，属于[ICMP](https://baike.baidu.com/item/ICMP)询问报文，它是应用层直接使用网络层ICMP的一个特例，它没有通过[运输层](https://baike.baidu.com/item/%E8%BF%90%E8%BE%93%E5%B1%82)的TCP或UDP。IP[数据报](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8A%A5)首部的协议字段：IP报文首部的协议字段指出了此数据报是使用的何种协议，以便使[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)主机的网络层能够知道如何管理协议

因特网组管理协议(IGMP)被IP主机用于向所有的直接相邻的[多播](https://baike.baidu.com/item/%E5%A4%9A%E6%92%AD)[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8)报告它们的多播组成员关系。本文档只描述在主机和[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8)之间的确定组成员关系的IGMP应用。作为[多播](https://baike.baidu.com/item/%E5%A4%9A%E6%92%AD)组成员的[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8)应当还能表现为一台主机，甚至能对自己的查询作出响应。IGMP还可以应用在[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8)之间，但这种应用不在这里[描述](https://baike.baidu.com/item/%E6%8F%8F%E8%BF%B0)。就像ICMP一样，IGMP作为整合在IP里面的一部分。所有希望接收IP组播的主机都应当实现IGMP。IGMP[消息](https://baike.baidu.com/item/%E6%B6%88%E6%81%AF)被封装在IP[数据报](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8A%A5)中，IP协议号为2。本文档所描述的所有IGMP消息在发送时[TTL](https://baike.baidu.com/item/TTL)都为1，并在它们的IP首部中含有一个[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8)警告选项。主机所关心的所有IGMP消息都具有以下格式：8位类型+8[位](https://baike.baidu.com/item/%E4%BD%8D)最大响应时间+16位校验和+32位组地址。

### IGMP协议

组播协议包括组成员管理协议和组播路由协议。组成员管理协议用于管理组播组成员的加入和离开，组播路由协议负责在路由器之间交互信息来建立组播树。IGMP属于前者，是组播路由器用来维护组播组成员信息的协议，运行于主机和和组播路由器之间。IGMP 信息[封装](https://baike.baidu.com/item/%E5%B0%81%E8%A3%85)在IP[报文](https://baike.baidu.com/item/%E6%8A%A5%E6%96%87)中，其IP的协议号为2。

若一个主机想要接收发送到一个特定组的组播数据包，它需要监听发往那个特定组的所有数据包。为解决Internet上组播数据包的路径选择，主机需通过通知其子网上的组播路由器来加入或离开一个组，组播中采用IGMP来完成这一任务。这样，组播路由器就可以知道网络上组播组的成员，并由此决定是否向它们的网络转发组播数据包。当一个组播路由器收到一个组播分组时，它检查数据包的组播目的地址，仅当接口上有那个组的成员时才向其转发。

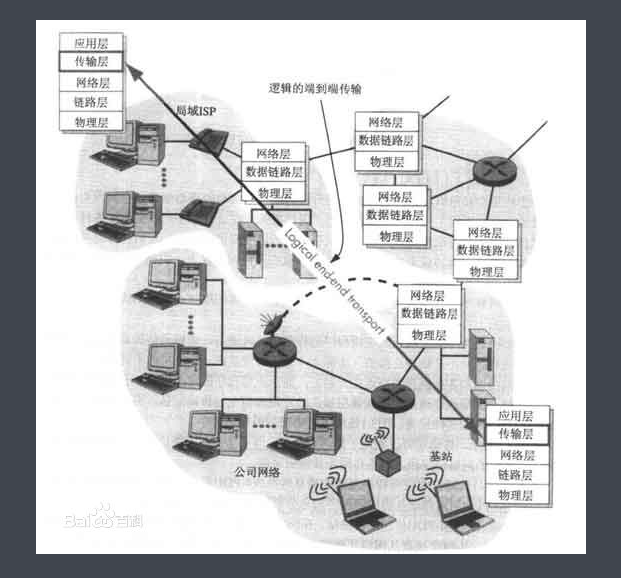
IGMP提供了在转发组播数据包到目的地的最后阶段所需的信息，实现如下双向的功能：

1. 主机通过IGMP通知路由器希望接收或离开某个特定组播组的信息。
2. 路由器通过IGMP周期性地查询局域网内的组播组成员是否处于活动状态，实现所连网段组成员关系的收集与维护。

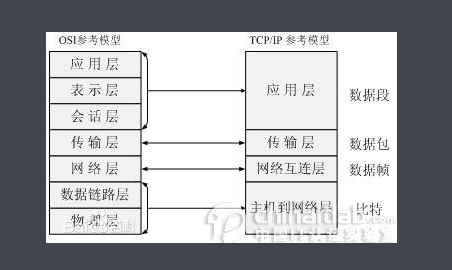
IGMP共有三个版本，即IGMP v1、v2 和 v3。

传输层

传输层（Transport Layer）是ISO OSI协议的第四层协议，实现端到端的数据传输。该层是两台[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA)经过网络进行[数据通信](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%80%9A%E4%BF%A1)时，第一个端到端的层次，具有缓冲作用。当[网络层](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B1%82)服务质量不能满足要求时，它将服务加以提高，以满足[高层](https://baike.baidu.com/item/%E9%AB%98%E5%B1%82)的要求；当网络层服务质量较好时，它只用很少的工作。传输层还可进行复用，即在一个网络连接上创建多个逻辑连接。



传输层在终端用户之间提供透明的数据传输，向上层提供可靠的数据传输服务。传输层在给定的链路上通过流量控、分段/重组和差错控制。一些协议是面向链接的。这就意味着传输层能保持对分段的跟踪，并且重传那些失败的分段。



传输层（Transport Layer）是[OSI](https://baike.baidu.com/item/OSI)中最重要, 最关键的一层,是唯一负责总体的[数据传输](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E8%BE%93)和数据控制的一层.传输层提供端到端的交换数据的机制.传输层对[会话层](https://baike.baidu.com/item/%E4%BC%9A%E8%AF%9D%E5%B1%82)等高三层提供可靠的传输服务,对[网络层](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B1%82)提供可靠的[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)地站点信息。

传输层也称为[运输层](https://baike.baidu.com/item/%E8%BF%90%E8%BE%93%E5%B1%82).传输层只存在于端开放系统中,是介于低3层[通信子网](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E5%AD%90%E7%BD%91)系统和高3层之间的一层,但是很重要的一层.因为它是源端到[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)端对数据传送进行控制从低到高的最后一层.

有一个既存[事实](https://baike.baidu.com/item/%E4%BA%8B%E5%AE%9E)，即世界上各种[通信子网](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E5%AD%90%E7%BD%91)在性能上存在着很大差异.例如电话交换网,[分组交换](https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E4%BA%A4%E6%8D%A2)网,公用数据交换网，[局域网](https://baike.baidu.com/item/%E5%B1%80%E5%9F%9F%E7%BD%91)等通信子网都可互连,但它们提供的[吞吐量](https://baike.baidu.com/item/%E5%90%9E%E5%90%90%E9%87%8F),[传输速率](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E9%80%9F%E7%8E%87),数据延迟通信费用各不相同.对于[会话层](https://baike.baidu.com/item/%E4%BC%9A%E8%AF%9D%E5%B1%82)来说,却要求有一性能恒定的接口.传输层就承担了这一功能.它采用分流/合流，复用/解复用技术来调节上述通信子网的差异,使会话层感受不到.此外传输层还要具备差错恢复，[流量控制](https://baike.baidu.com/item/%E6%B5%81%E9%87%8F%E6%8E%A7%E5%88%B6)等功能,以此对会话层屏蔽通信子网在这些方面的细节与差异.传输层面对的[数据对象](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AF%B9%E8%B1%A1)已不是[网络地址](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%9C%B0%E5%9D%80)和[主机地址](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA%E5%9C%B0%E5%9D%80),而是和会话层的界面端口.

上述功能的最终[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)是为会话提供可靠的,无误的[数据传输](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E8%BE%93).传输层的服务一般要经历传输连接建立阶段,[数据传送](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E9%80%81)阶段,传输连接释放阶段3个阶段才算完成一个完整的服务过程.而在数据传送阶段又分为一般数据传送和加速数据传送两种。传输层服务分成5种类型.基本可以满足对传送质量,传送速度,传送费用的各种不同需要.

## 端口概念

传输层的任务是根据[通信子网](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E5%AD%90%E7%BD%91)的特性，最佳的利用[网络资源](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E8%B5%84%E6%BA%90)，为两个[端系统](https://baike.baidu.com/item/%E7%AB%AF%E7%B3%BB%E7%BB%9F)的[会话层](https://baike.baidu.com/item/%E4%BC%9A%E8%AF%9D%E5%B1%82)之间，提供建立、维护和取消传输连接的功能，负责端到端的可靠[数据传输](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E8%BE%93)。在这一层，信息传送的[协议数据单元](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE%E6%95%B0%E6%8D%AE%E5%8D%95%E5%85%83)称为段或[报文](https://baike.baidu.com/item/%E6%8A%A5%E6%96%87)。

[网络层](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B1%82)只是根据[网络地址](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%9C%B0%E5%9D%80)将源结点发出的[数据包](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%8C%85)传送到[目的](https://baike.baidu.com/item/%E7%9B%AE%E7%9A%84)结点，而传输层则负责将数据可靠地传送到相应的端口。

计算机网络中的[资源子网](https://baike.baidu.com/item/%E8%B5%84%E6%BA%90%E5%AD%90%E7%BD%91)是通信的发起者和接收者，其中的每个设备称为端点；[通信子网](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E5%AD%90%E7%BD%91)提供网络中的通信服务，其中的设备称为结点。[OSI参考模型](https://baike.baidu.com/item/OSI%E5%8F%82%E8%80%83%E6%A8%A1%E5%9E%8B)中用于通信控制的是下面四层，但它们的控制对象不一样。

## 地位

传输层在OSI中的地位和作用

传输层是整个协议[层次结构](https://baike.baidu.com/item/%E5%B1%82%E6%AC%A1%E7%BB%93%E6%9E%84)的核心，是惟一负责总体[数据传输](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E8%BE%93)和控制的一层。

[](https://baike.baidu.com/pic/%E4%BC%A0%E8%BE%93%E5%B1%82/4329536/0/f29faa8f304856aff01f36fb?fr=lemma&ct=single)OSI模型

在[OSI七层模型](https://baike.baidu.com/item/OSI%E4%B8%83%E5%B1%82%E6%A8%A1%E5%9E%8B)中传输层是负责数据通信的最[高层](https://baike.baidu.com/item/%E9%AB%98%E5%B1%82)，又是面向网络通信的低三层和面向信息处理的高三层之间的中间层。因为[网络层](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B1%82)不一定保证服务的可靠，而用户也不能直接对[通信子网](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E5%AD%90%E7%BD%91)加以控制，因此在网络层之上，加一层即传输层以改善传输质量。

传输层利用[网络层](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B1%82)提供的服务，并通过传输层[地址](https://baike.baidu.com/item/%E5%9C%B0%E5%9D%80)提供给[高层](https://baike.baidu.com/item/%E9%AB%98%E5%B1%82)用户传输数据的通信端口，使系统间高层资源的共享不必考虑数据通信方面和不可靠的[数据传输](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BC%A0%E8%BE%93)方面的问题。它的主要功能是：对一个进行的对话或连接提供可靠的传输服务，在通向网络的单一物理连接上实现该连接的复用，在单一连接上提供端到端的[序号](https://baike.baidu.com/item/%E5%BA%8F%E5%8F%B7)与[流量控制](https://baike.baidu.com/item/%E6%B5%81%E9%87%8F%E6%8E%A7%E5%88%B6)、[差错控制](https://baike.baidu.com/item/%E5%B7%AE%E9%94%99%E6%8E%A7%E5%88%B6)及恢复等服务。