

HDLC协议

一 HDLC 概述

1.1 HDLC 的发展历史

高级数据链路控制 (High-Level Data Link Control 或简称 HDLC)，是一个在同步网上传输数据、面向比特的数据链路层协议，它是由国际标准化组织(ISO)根据 IBM 公司的 SDLC(Synchronous Data Link Control)协议扩展开发而成的。其最大特点是不需要数据必须是规定字符集，对任何一种比特流，均可以实现透明的传输。1974 年，IBM 公司率先提出了面向比特的同步数据链路控制规程 SDLC (Synchronous Data Link Control)。

随后，ANSI 和 ISO 均采纳并发展了 SDLC，并分别提出了自己的标准：

1* ANSI 的高级通信控制过程 ADCCP (Advanced Data Control Procedure)，

2* ISO 的高级数据链路控制规程 HDLC (High-level Data Link Control)。

从此，HDLC 协议开始得到了人们的广泛关注，并开始应用于通信领域的各个方面。

1.2 HDLC 的特点

HDLC 是面向比特的数据链路控制协议的典型代表，有着很大的优势：

1* HDLC 协议不依赖于任何一种字符编码集；

2* 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；

3* 全双工通信，有较高的数据链路传输效率；

4* 所有帧采用 CRC 检验，对信息帧进行顺序编号，可防止漏收或重份，传输可靠性高；

5* 传输控制功能与处理功能分离，具有较大灵活性。

由于以上特点，目前网络设计及整机内部通讯设计普遍使用 HDLC 数据链路控制协议。

HDLC 已经成为通信领域不可或缺的一个重要协议。

二 数据链路层的控制规程

2.1 数据链路结构

数据链路结构可以分为两种：点-点链路和点-多点链路，如图 1 所示。图中数据链路两端 DTE 称为计算机或终端，从链路逻辑功能的角度常称为站，从网络拓扑结构的观点则称为节点。

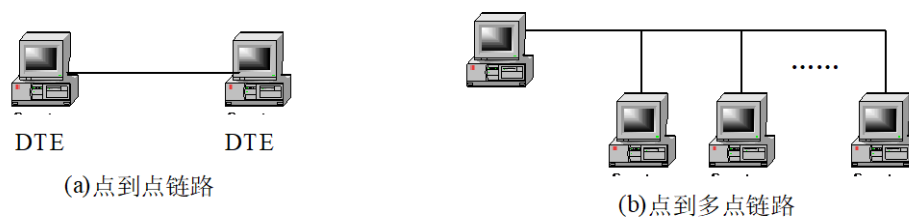


图 1 数据链路结构

在点-点链路中，发送信息和命令的站称为主站，接收信息和命令而发出确认信息或响应的站称为从站，兼有主、从功能可发送命令与响应的站称为复合站。在点-多点链路中，往往有一个站为控制站，主管数据链路的信息流，并处理链路上出现的不可恢复的差错情况，其余各站则为受控站。

2.2 数据链路控制规程功能

数据链路层是 OSI 参考模型的第二层，它在物理层提供的通信接口与电路连接服务的基础上，将易出错的数据电路构筑成相对无差错的数据链路，以确保 DTE 与 DTE 之间、DTE 与网络之间有效、可靠地传送数据信息。为了实现这个目标，数据链路控制规程的功能应包括以下几个部分：

1*帧控制

数据链路上传输的基本单位是帧。帧控制功能要求发送站把网络送来的数据信息分成若干码组，在每个码组中加入地址字段、控制字段、校验字段以及帧开始和结束标志，组成帧来发送；要求接收端从收到的帧中去掉标志字段，还原成原始数据信息后送到网络层。

2*帧同步

在传输过程中必须实现帧同步，以保证对帧中各个字段的正确识别。

3*差错控制

当数据信息在物理链路中传输出现差错，数据链路控制规程要求接收端能检测出差错并予以恢复，通常采用的方法有自动请求重发 ARQ 和前向纠错两种。采用 ARQ 方法时，为了防止帧的重收和漏收，常对帧采用编号发送和接收。当检测出无法恢复的差错时，应通知网络层做相应处理。

4*流量控制

流量控制用于克服链路的拥塞。它能对链路上信息流量进行调节，确保发送端发送的数据速率与接收端能够接收的数据速率相容。常用的流量控制方法是滑动窗口控制法。

5*链路管理

数据链路的建立、维持和终止，控制信息的传输方向，显示站的工作状态，这些都属于链路管理的范畴。

6*透明传输

规程中采用的标志和一些字段必须独立于要传输的信息，这就意味着数据链路能够传输各种各样的数据信息，即传输的透明性。

7*寻址

在多点链路中，帧必须能到达正确的接收站。

8*异常状态恢复

当链路发生异常情况时，如收到含义不清的序列或超时收不到响应等，能自动重新启动，恢复到正常工作状态。

2.3 数据链路层协议

数据链路控制规程，根据帧控制的格式，可以分为面向字符型、面向比特型。

1*面向字符型

国际标准化组织制定的 ISO 1745、IBM 公司的二进制同步规程 BSC 以及我国国家标准 GB3543-82 属于面向字符型的规程，也称为基本型传输控制规程。在这类规程中，用字符编码集中的几个特定字符来控制链路的操作，监视链路的工作状态，例如，采用国际 5 号码中的 SOH、STX 作为帧的开始，ETX、ETB 作为帧的结束，ENQ、EOT、ACK、NAK 等字符控制链路操作。面向字符型规程有一个很大的缺点，就是它与所用的字符集有密切的关系，使用不同字符集的两个站之间，很难使用该规程进行通信。面向字符型规程主要适用于中低速异步或同步传输，很适合于通过电话网的数据通信。

2*面向比特型

ITU-T 制定的 X.25 建议的 LAPB、ISO 制定的 HDLC、美国国家标准 ADCCP、IBM 公司的 SDLC 等均属于面向比特型的规程。在这类规程中，采用特定的二进制序列 01111110

作为帧的开始和结束，以一定的比特组合所表示的命令和响应实现链路的监控功能，命令和响应可以和信息一起传送。所以它可以实现不编码限制的、高可靠和高效率的透明传输。面向比特型规程主要适用于中高速同步半双工和全双工数据通信，如分组交换方式中的链路层就采用这种规程。随着通信的发展，它的应用日益广泛。

三 HDLC 协议

3.1 HDLC 的基本概念

3.1.1 主站、从站、复合站

HDLC 涉及三种类型的站，即主站、从站和复合站。

1*主站的主要功能是发送命令（包括数据信息）帧、接收响应帧，并负责对整个链路的控制系统的初启、流程的控制、差错检测或恢复等。

2*从站的主要功能是接收由主站发来的命令帧，向主站发送响应帧，并且配合主站参与差错恢复等链路控制。

3*复合站的主要功能是既能发送，又能接收命令帧和响应帧，并且负责整个链路的控制。

3.1.2 HDLC 链路结构

在 HDLC 中，对主站、从站和复合站定义了三种链路结构，如图 2 所示

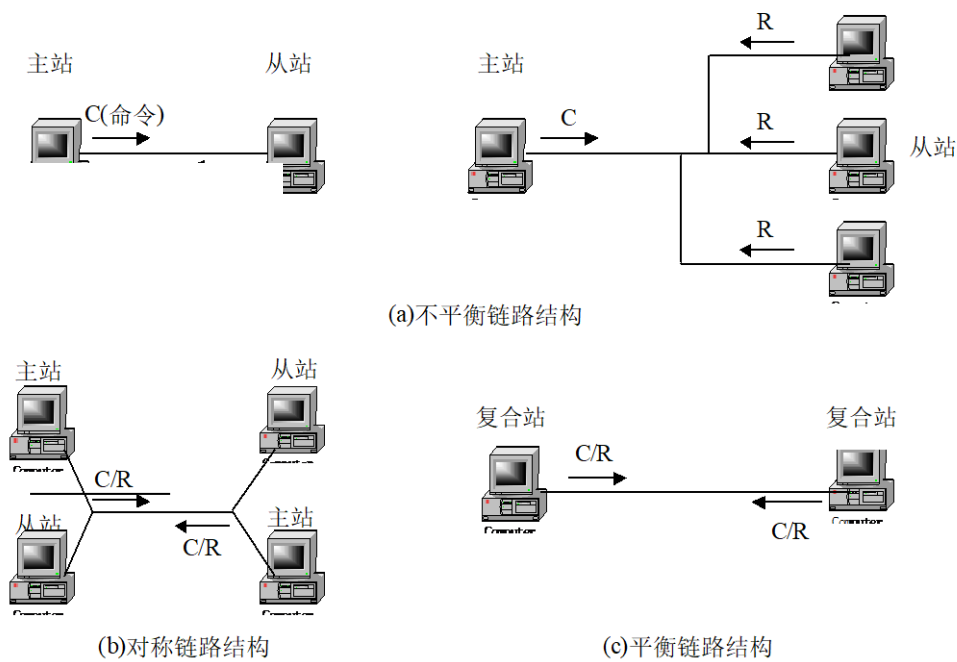


图 2 HDLC 链路结构类型

3.2 HDLC 协议的主要内容

3.2.1 HDLC 帧结构

HDLC 的帧格式如图 3 所示，它由六个字段组成，这六个字段可以分为五中类型，即标志序列（F）、地址字段（A）、控制字段（C）、信息字段（I）、帧校验字段（FCS）。在帧结构中允许不包含信息字段 I。

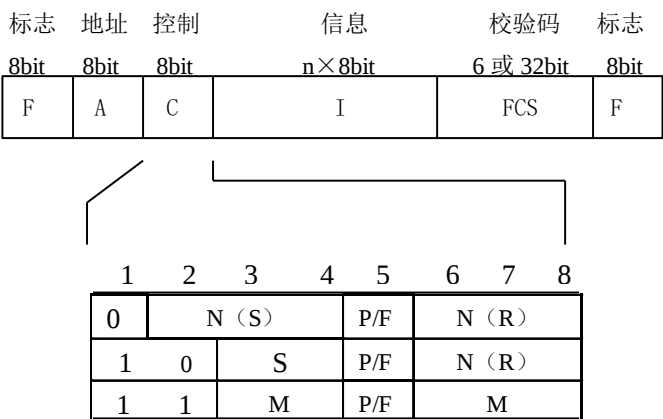


图 3 HDLC 帧结构

1*标志序列（F）

HDLC 指定采用 01111110 为标志序列，称为 F 标志。要求所有的帧必须以 F 标志开始和结束。接收设备不断地搜寻 F 标志，以实现帧同步，从而保证接收部分对后续字段的正确识别。另外，在帧与帧的空载期间，可以连续发送 F，用来作时间填充。

在一串数据比特中，有可能产生与标志字段的码型相同的比特组合。为了防止这种情况产生，保证对数据的透明传输，采取了比特填充技术。当采用比特填充技术时，在信码中连续 5 个“1”以后插入一个“0”；而在接收端，则去除 5 个“1”以后的“0”，恢复原来的数据序列，如图 4 所示。比特填充技术的采用排除了在信息流中出现的标志字段的可能性，保证了对数据信息的透明传输。

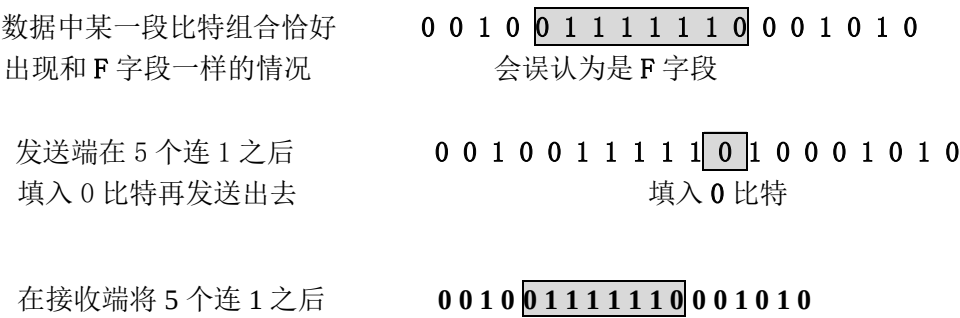


图 4 比特填充

当连续传输两帧时，前一个帧的结束标志字段 F 可以兼作后一个帧的起始标志字段。当暂时没有信息传送时，可以连续发送标志字段，使接收端可以一直保持与发送端同步。

2*地址字段 (A)

地址字段表示链路上站的地址。在使用不平衡方式传送数据时（采用 NRM 和 ARM），地址字段总是写入从站的地址；在使用平衡方式时（采用 ABM），地址字段总是写入应答站的地址。

地址字段的长度一般为 8bit，最多可以表示 256 个站的地址。在许多系统中规定，地址字段为“11111111”时，定义为全站地址，即通知所有的接收站接收有关的命令帧并按其动作；全“0”比特为无站地址，用于测试数据链路的状态。因此有效地址共有 254 个之多，这对一般的多点链路是足够的。但考虑在某些情况下，例如使用分组无线网，用户可能很多，可使用扩充地址字段，以字节为单位扩充。在扩充时，每个地址字段的第 1 位用作扩充指示，即当第 1 位为“0”时，后续字节为扩充地址字段；当第 1 位为“1”时，后续字节不是扩充地址字段，地址字段到此为止。

3*控制字段 (C)

控制字段用来表示帧类型、帧编号以及命令、响应等。从图 5-11 可见，由于 C 字段的构成不同，可以把 HDLC 帧分为三种类型：信息帧、监控帧、无编号帧，分别简称 I 帧 (Information)、S 帧 (Supervisory)、U 帧 (Unnumbered)。在控制字段中，第 1 位是“0”为 I 帧，第 1、2 位是“10”为 S 帧，第 1、2 位是“11”为 U 帧，它们具体操作复杂，在后面予以介绍。另外控制字段也允许扩展。

4*信息字段 (I)

信息字段内包含了用户的数据信息和来自上层的各种控制信息。在 I 帧和某些 U 帧中，具有该字段，它可以是任意长度的比特序列。在实际应用中，其长度由收发站的缓冲器的大小和线路的差错情况决定，但必须是 8bit 的整数倍。

5*帧校验序列字段 (FCS)

帧校验序列用于对帧进行循环冗余校验，其校验范围从地址字段的第 1 比特到信息字段的最后一比特的序列，并且规定为了透明传输而插入的“0”不在校验范围内。

3.2.2 HDLC 的帧类型

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）

监控帧用于监视和控制数据链路，完成信息帧的接收确认、重发请求、暂停发送求等功能。监控帧不具有信息字段。监控帧共有 4 种，表 1 是这 4 种监控帧的代码、名称和功能。

表 1 监控帧的名称和功能

记忆符	名 称	比特		功 能
		b ₂	b ₃	
RR	接收准备好	0	0	确认，且准备接受下一帧，已收妥 N(R) 以前的各帧
RNR	接收未准备好	1	0	确认，暂停接收下一帧，N(R) 含义同上
REJ	拒绝接收	0	1	否认，否认 N(R) 起的各帧，但 N(R) 以前的帧已收妥
SREJ	选择拒绝接收	1	1	否认，只否认序号为 N(R) 的帧

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）

无编号帧用于数据链路的控制，它本身不带编号，可以在任何需要的时刻发出，不影响带编号的信息帧的交换顺序。它可以分为命令帧和响应帧。用 5 个比特位（即 M₁、

M₂) 来表示不同功能的无编号帧。HDLC 所定义的无编号帧名称和代码见表 2。

表 2 无编号帧的名称和代码

记忆符	名 称	类型		M ₁	M ₂
		命令	响应	b ₃ b ₄	b ₆ b ₇ b ₈
SNRM	置正常响应模式	C		0 0	0 0 1
SARM/DM	置异步响应模式/断开方式	C	R	1 1	0 0 0
SABM	置异步平衡模式	C		1 1	1 0 0
SNRME	置扩充正常响应模式	C		1 1	0 1 1
SARME	置扩充异步响应模式	C		1 1	0 1 0
SABME	置扩充异步平衡模式	C		1 1	1 1 0
DISC/RD	断链/请求断链	C	R	0 0	0 1 0
SIM/RIM	置初始化方式/请求初始化方式	C		1 0	0 0 0
UP	无编号探询	C		0 0	1 0 0
UI	无编号信息	C		0 0	0 0 0
XID	交换识别	C	R	1 1	1 0 1
RESET	复位	C		1 1	0 0 1
FRMR	帧拒绝		R	1 0	0 0 1
UA	无编号确认		R	0 0	1 1 0

3.2.3 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，通俗地讲就是以主节点方式操作，还是以从节点方式操作，或者是二者兼备。

在链路上用于控制目的的节点称为主节点，其他的受主节点控制的节点称为从节点。主节点负责对数据流进行组织，并且对数据上的差错实施恢复。由主节点发往从节点的帧称为命令帧，而由从节点返回主节点的帧称为响应帧。连有多个节点的链路通常使用轮询技术，轮询其他节点的节点为主节点，而在点到点链路中每个节点均可为主节点。在一个节点点连接多条链路的情况下，该节点对于一些链路而言可能是主节点，而对另外一些链路而言有可能是从节点。

HDLC中常用的操作方式有3种：

1*正常响应方式 NRM

正常响应方式 NRM (Normal Response Mode) 一种非平衡数据链路操作方式, 有时也称为非平衡正常响应方式。该操作方式使用于面向终端的点到点或一点到多点的链路。在这种操作方式下, 传输过程由主节点启动, 从节点只有收到主节点某个命令帧后, 才能作为响应向主节点传输信息。响应信息可以由一个或多个帧组成, 若信息由多个帧组成, 则应指出哪一帧是最后一帧。主节点负责管理整个链路, 且具有轮询、选择从节点及向从节点发送命令的权利, 同时也负责对超时、重发及各类恢复操作的控制。

2*异步响应方式 ARM

异步响应方式 ARM (Asynchronous Response Mode) 也是一种非平衡数据链路操作方式, 与NRM不同的是, ARM下的传输过程由从节点启动。从节点主动发送给主节点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, 由从节点来控制超时和重发。该方式对采用轮询方式的多节点链路来说是比不可少的。

3*异步平衡方式 ABM

异步平衡方式ABM (Asynchronous Balanced Mode) 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有较高的信息传输量。在这种操作方式下, 任何时候任何节点都能启动传输操作, 每个节点即可以作为主节点又可以作为从节点, 即每个节点都是组合节点。各个节点都有相同的一组协议, 任何节点都可以发送或接受命令, 也可以给出应答, 并且各节点对差错恢复过程都负有相同的责任。

整个数据通信一般分为 3 个阶段: 数据链路建立阶段、信息帧传送阶段、数据链路释放阶段。第 2 阶段的完成需要用到信息帧和监控帧, 第 1、3 阶段的完成需要用到无编号帧。

图 5 画出了多点链路的建立和释放。主站 A 先向从站 B 发出置正常响应模式 SNRM 的命令, 并将 P 置 1, 要求 B 站作出响应。B 站同意建立链路后, 发送无编号确认 UA 的响应, 将 F 置 1。A 站和 B 站在将其状态变量 V(S) 和 V(R) 进行初始化后, 就完成了数据链路的建立。接着 A 站开始与 C 站建立链路。

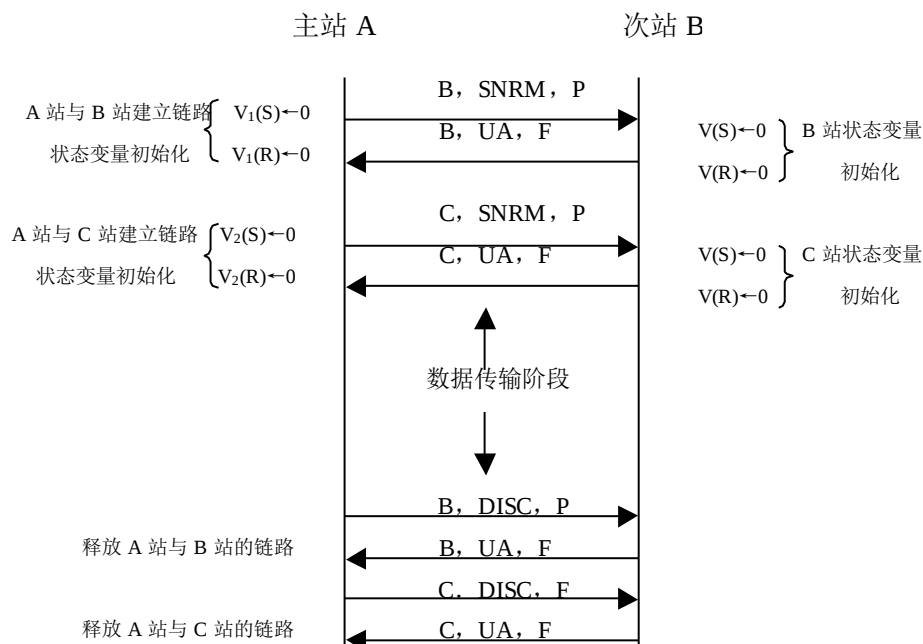


图 5 多点链路的建立和释放

当数据传送完毕后，A 站分别向 B 站和 C 站发出断链命令 DISC，B 站、C 站用无编号确认帧 UA 响应，数据链路就释放了。

图 6 为点对点链路中两个站都是复合站的情况。复合站中的一个站先发出置异步平衡模式 SABM 的命令，对方回答一个无编号响应帧 UA 后，即完成了数据链路的建立。由于两个站是平等的，任何一个站均可在数据传送完毕后发出 DISC 命令提出断链的要求，对方用 UA 帧响应，完成数据链路的释放。

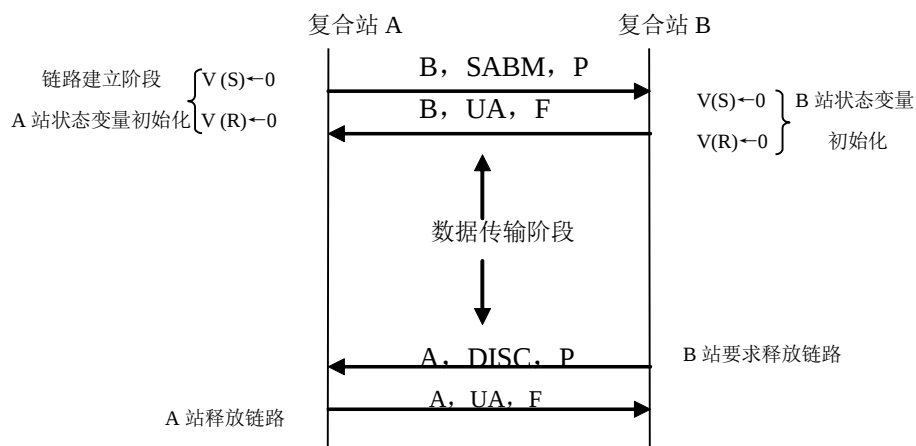


图 6 复合站的链路建立和释放

3.3 HDLC 规程的特点

与面向字符的基本型传输控制规程相比较，HDLC 具有以下特点：

1*透明传输

HDLC 对任意比特组合的数据均能透明传输。“透明”是一个很重要的术语，它表示：某一个实际存在的事物看起来好象不存在一样。“透明传输”表示经实际电路传送后的数据信息没有发生变化。因此对所传送数据信息来说，由于这个电路并没有对其产生什么影响，可以说数据信息“看不见”这个电路，或者说这个电路对该数据信息来说是透明的。这样任意组合的数据信息都可以在这个电路上传送。

2*可靠性高

在 HDLC 规程中，差错控制的范围是除了 F 标志的整个帧，而基本型传输控制规程中不包括前缀和部分控制字符。另外 HDLC 对 I 帧进行编号传输，有效地防止了帧的重收和漏收。

3*传输效率高

在 HDLC 中，额外的开销比特少，允许高效的差错控制和流量控制。

4*适应性强

HDLC 规程能适应各种比特类型的工作站和链路。

5*结构灵活

在 HDLC 中，传输控制功能和处理功能分离，层次清楚，应用非常灵活。

3.4 HDLC 存在的问题

3.4.1“0”位插入 / 删除技术

如上所述，SDLC / HDLC 协议规定以 01111110 为标志字节，但在信息场中也完全有可能有同一种模式的字符，为了把它与标志区分开来，所以采取了“0”位插入和删除技术。具体作法是发送端在发送所有信息（除标志字节外）时，只要遇到连续 5 个“1”，就自动插入一个“0”当接收端在接收数据时（除标志字节）如果连续接收到 5 个“1”，就自动将其后的一个“0”删除，以恢复信息的原有形式。这种“0”位的插入和删除过程是由硬件自动完成的，比上述面向字符的“数据透明”容易实现。

3.4.2 SDLC / HDLC 异常结束

若在发送过程中出现错误，则 SDLC / HDLC 协议用异常结束（Abort）字符，或称失效序列使本帧作废。在 HDLC 规程中 7 个连续的“1”被作为失效字符，而在 SDLC 中失效字符是 8 个连续的“1”。当然在失效序列中不使用“0”位插入 / 删除技术。

四 HDLC 协议的应用与发展前景

4.1 HDLC 的应用特点

1*应用场合

就系统结构而言，HDLC适用于点到点或点到多点式的结构；就工作方式而言，HDLC适用于半双工或全双工；就传输方式而言，HDLC只用于同步传输；在传输速率方面考虑，HDLC常用于中高速传输。

2*传输效率

HDLC开始发送一帧后，就要连续不断地发完该帧；HDLC可以同时确认几个帧；HDLC中的每个帧含有地址字段A，在多点结构中，每个从节点只接收含有本节点地址的帧，因此主节点在选中一个从节点并与之通讯的同时，不用拆链，便可以选择其他的节点通讯，即可以同时与多个节点建立链路。

由于以上特点，HDLC具有较高的传输效率。

3*传输可靠性

HDLC中所有的帧（包括响应帧）都有FCS，帧按窗口序号顺序编号，传输可靠性比异步通讯高。

4*数据透明性

HDLC采用“0比特插入法”对数据进行透明传输，传输信息的比特组合模式无任何限制，处理简单。

5*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，实现起来方便。

6*链路控制

HDLC利用改变一帧中的控制字段的编码格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。

4.2 HDLC协议的发展前景

HDLC作为面向比特的数据链路控制协议的典型代表，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，可防止漏收或重份，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。由于以上的优点，HDLC协议发展和普及的非常快，目前网络设计普遍使用HDLC作为数据链路层协议，HDLC协议的发展前景也被业内人士看好。

五 结论

本文通过对数据链路层的HDLC协议进行综述介绍，主要内容包括对HDLC的发展数据链路控制协议，HDLC协议的主要内容、存在的技术标准以及HDLC的应用和发展前景等进行了研究。HDLC作为面向比特的数据链路控制协议的典型代表，有着很大的优势：HDLC协议不依赖于任何一种字符编码集；并且数据报文可透明传输，用于实现透明传输的“0比特插入法”也易于硬件实现；另外它采用的是全双工通信，有较高的数据链路传输效率；它的所有帧采用CRC检验，对信息帧进行顺序编号，可防止漏收或重份，传输可靠性比较高；最后其传输控制功能与处理功能分离，具有较大灵活性。

由于以上特点，HDLC的应用得很广泛。目前网络设计普遍使用HDLC作为数据链路层协议。所以说HDLC已经成为通信领域非常重要的一个协议。

六 参考文献

- [1]谢希仁.计算机网络.[M].北京：清华大学出版社.2004.2.
- [2]张德民.数据通信.[M].北京：科学技术文献出版社.1997.8.
- [3]王承恕，通信网新技术，人民邮电出版社，2006年1月第1版
- [4]朱近康，CDMA 通信技术，人民邮电出版社，2001年9月第1版
- [5]秦国主编.现代通信网概述[M].北京：人民邮电出版社，2004
- [6]方旭明等，下一代无线因特网技术：无线 Mesh 网络，人民邮电出版社，2006年5月第1版
- [7]付新虎，史浩，郑喜凤，丁铁夫。HDLC 协议在通信系统中的应用[J].电子技术，2005（8）