



高级数据链路控制规程 HDLC

目 录

1. 数据链路控制规程.....	3
1. 1 数据链路结构.....	3
1. 2 数据链路控制规程功能.....	3
1. 3 数据链路控制规程分类.....	4
2. HDLC 基本概念.....	5
2. 1 主站、从站、复合站.....	5
2. 2 HDLC 链路结构.....	5
2. 3 操作方式.....	6
3. HDLC 帧结构.....	7
4. 控制字段和参数.....	8
4. 1 信息帧 (I 帧)	9
4. 2 监控帧 (S 帧)	9
4. 3 无编号帧 (U 帧)	10
4. 4 P/F 比特的使用	10
5. HDLC 操作.....	12
6. HDLC 规程的特点.....	13

1. 数据链路控制规程

1. 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。

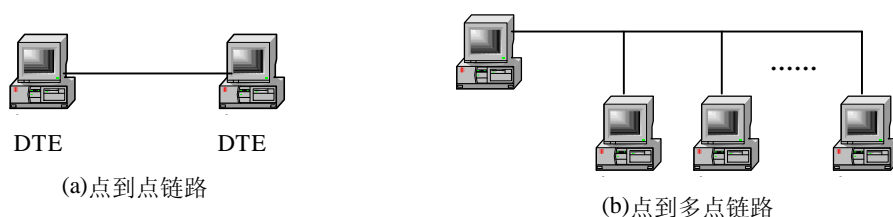


图 1 数据链路结构

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。

1. 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：

I 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。

I 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。

I 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。

I 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。

I 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。

I 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。

I 寻址

在多点链路中，帧必须能到达正确的接收站。

I 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。

1.3 数据链路控制规程分类

为了适应数据通信的需要，ISO、ITU-T 以及一写国家和大的计算机制造公司，先后制定了不同类型的数据链路控制规程。根据帧控制的格式，可以分为面向字符型、面向比特型。

I 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。

I 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。

2. HDLC 基本概念

2.1 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。

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。

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。

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。

2.2 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。

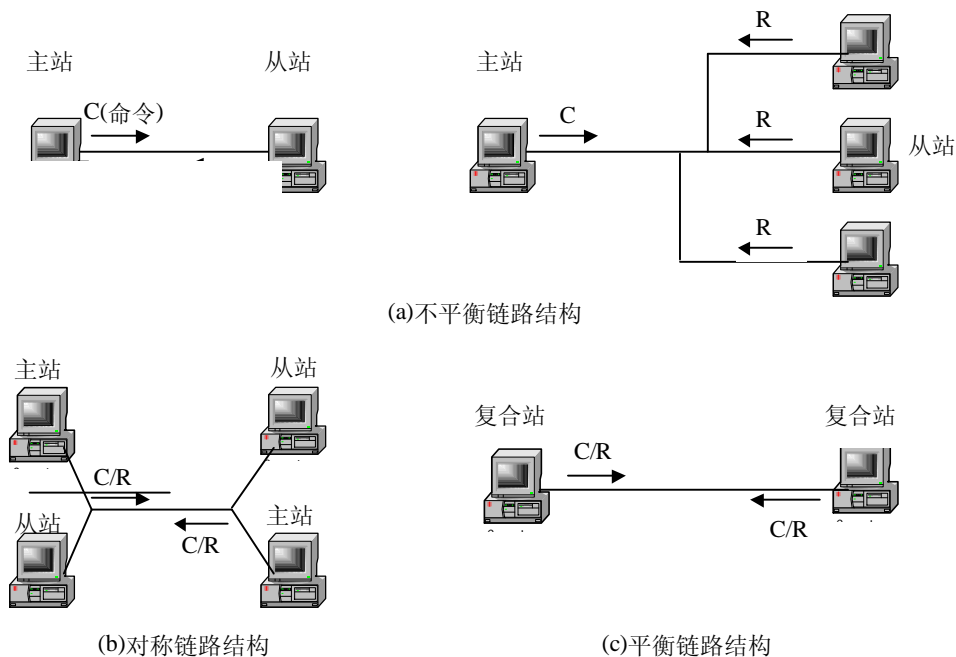


图2 HDLC 链路结构类型

2.3 操作方式

根据通信双方的链路结构和传输响应类型,HDLC 提供了三种操作方式:正常响应方式、异步响应方式和异步平衡方式。

I 正常响应方式 (NRM)

正常响应方式 (NRM) 适用于不平衡链路结构,即用于点-点和点-多点的链路结构中,特别是点-多点链路。这种方式中,由主站控制整个链路的操作,负责链路的初始化、数据流控制和链路复位等。从站的功能很简单,它只有在收到主站的明确允许后,才能发出响应。

I 异步响应方式 (ARM)

异步响应方式 (ARM) 也适用于不平衡链路结构。它与 NRM 不同的是:在 ARM 方式中,从站可以不必得到主站的允许就可以开始数据传输。显然它的传输效率比 NRM 有所提高。

I 异步平衡方式 (ABM)

异步平衡方式 (ABM) 适用于平衡链路结构。链路两端的复合站具有同等的能力,不管哪个复合站均可在任意时间发送命令帧,并且不需要收到对方复合站发出的命令帧就可以发送响应帧。ITU-T X.25 建议的数据链路层就采用这种方式。

除三种基本操作方式,还有三种扩充方式,即扩充正常响应方式 (SNRM)、扩充异步

响应方式（SARM）、扩充异步平衡方式（SABM）它们分别与基本方式相对应。

3. 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。

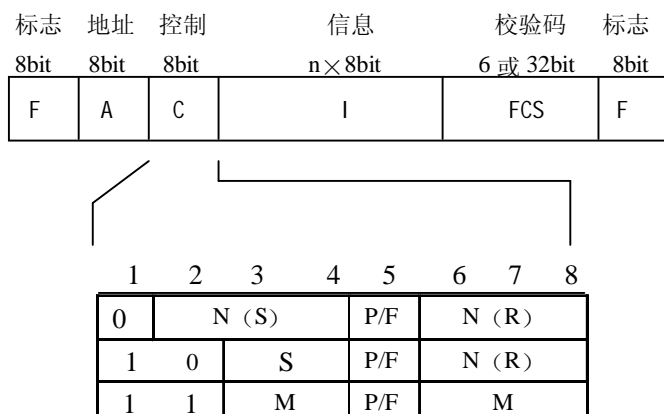


图 3 HDLC 帧结构

I 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。

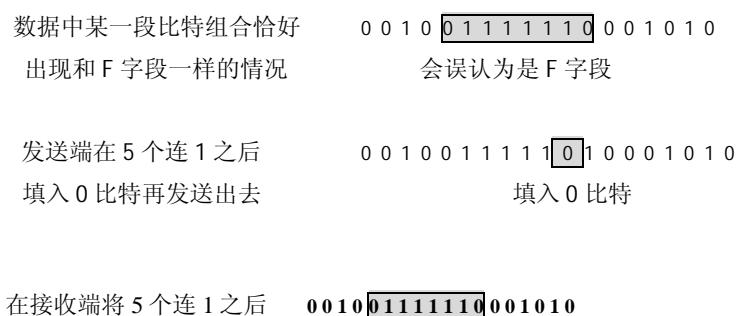


图 4 比特填充

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。

I 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“1111111”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。

I 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。

I 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。

I 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。

4. 控制字段和参数

控制字段是 HDLC 的关键字段，许多重要的功能都靠它来实现。控制字段规定了帧的类型，即 I 帧、S 帧、U 帧，控制字段的格式如图 3 所示，其中

N(S) 发送帧序列编号

N(R)	期望接收的帧序列编号，且是对 N(R) 以前帧的确认
S	监控功能比特
M	无编号功能比特
P/F	查询/结束(Poll/Final)比特，作为命令帧发送时的查询比特，以 P 位出现； 作为响应帧发送时的结束比特，以 F 位出现。

下面对三种不同类型的帧分别予以介绍。

4.1 信息帧 (I 帧)

I 帧用于数据传送，它包含信息字段。在 I 帧控制字段中 $b_1 \sim b_3$ 比特为 N(S)， $b_5 \sim b_7$ 比特为 N(R)。由于是全双工通信，所以通信每一方都各有一个 N(S) 和 N(R)。这里要特别强调指出：N(R) 带有确认的意思，它表示序号为 N(R)-1 以及在这以前的各帧都已经正确无误地收妥了。

为了保证 HDLC 的正常工作，在收发双方都设置两个状态变量 V(S) 和 V(R)。V(S) 是发送状态变量，为发送 I 帧的数据站所保持，其值指示待发的一帧的编号；V(R) 是接收状态变量，其值为期望所收到的下一个 I 帧的编号。可见这两个状态变量的值确定发送序号 N(S) 和接收序号 N(R)。

在发送站，每发送一个 I 帧， $V(S) \rightarrow N(S)$ ，然后 $V(S)+1 \rightarrow V(S)$ 。在接收站，把收到的 N(S) 与保留的 V(R) 作比较，如果这个 I 帧可以接收，则 $V(R)+1 \rightarrow N(R)$ ，回送到发送站，用于对前面所收到的 I 帧的确认。N(R) 除了可以用 I 帧回送之外，还可以用 S 帧回送，这一点从图 5-11 中可以看出，在 I 帧和 S 帧的控制字段中具有 N(R)。

V(S)、V(R) 和 N(S)、N(R) 都各占 3bit，即序号采用模 8 运算，使用 0~7 八个编号。在有些场合，如卫星通信模 8 已经不能满足要求了，这时可以把控制字段扩展为两个字节，N(S)、N(R) 和 V(S)、V(R) 都用 7bit 来表示，即增加到模 128。

4.2 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、暂停发送请求等功能。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

上面四种监控帧中，前三种用在返回 N 连续 ARQ 方法中，最后一种只用于选择重发 ARQ 方式中。

S 帧中没有包含用户的数据信息字段，它只有 48bit 长，显然不需要 N(S)，但 S 帧中 N(R) 特别有用，它具体含义随不同的 S 帧类型而不同。其中 RR 帧和 RNR 帧相当于确认信息 ACK，REJ 帧相当于否认信息 NAK。同时应当注意到，RR 帧和 RNR 帧还具有流量控制的作用，RR 帧表示已经作好表示接收帧的准备，希望对方继续发送，而 RNR 帧则表示希望对方停止发送（这可能是由于来不及处理到达的帧或缓冲器已存满）。

4. 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，而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 M_1 、 M_2 ）来表示不同功能的无编号帧。HDLC 所定义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	无编号探测	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

4. 4 P/F 比特的使用

值得注意的是在 HDLC 的各类帧中，均带有查询/结束（P/F）比特。在不同的数据传送方式中，P/F 比特的用法是不一样的，

在 NRM 方式中，从站不能主动向主站发送信息，从站只有收到主站发出的 P 比特为 1（对从站的查询）的命令帧以后才能发送响应帧。若从站有数据发送，则在最后一个数据帧中将 F 比特置 1；若无数据发送，则应在回答的 S 帧中将 F 比特置 1。

在 ARM 或 ABM 方式中，任何一个站都可以在主动发送的 S 帧和 I 帧中将 P 比特置 1。对

方站收到P=1 的帧后，应尽早地回答本站的状态并将F 比特置 1。

下面结合图 5 的例子具体说明 P/F 比特的使用方法。图中主站 A 和从站 B、C 连成多点链路，传送帧的一些主要参数按照“地址，帧名和序号，P/F”的先后顺序标注。这里的地址是指地址字段中应填入的站地址；帧名是指帧的名称，如 RR、I；序号是指监控帧中的 N(R) 或信息帧中的 N(S)N(R)，如 RR4、I31（第 1 个数字是 N(S)，第 2 个数字是 N(R)）。P/F 是在其为 1 时才写上 P 或 F，表明此时控制字段的第 5 比特为 1。

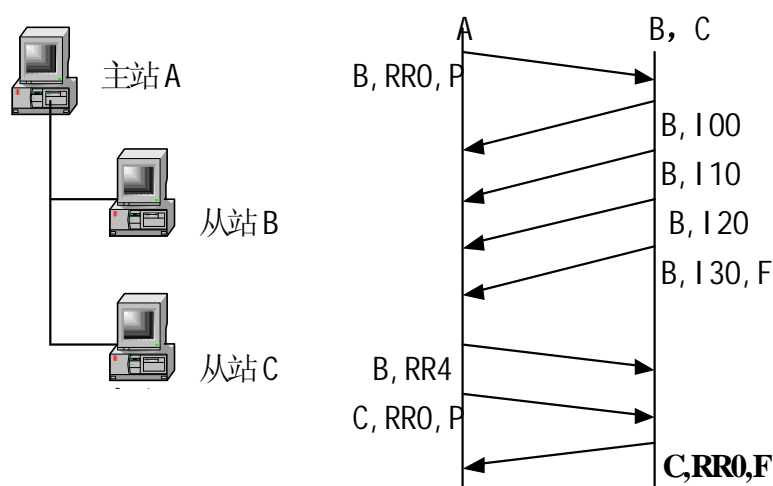


图 5 P/F 比特的使用方法

主站 A 先询问 B 站：“B 站，若有信息，请立刻发送”。这时 A 站发送的帧是 RR 监控帧，并将 N(R)置 0，表示期望收到对方的 0 号帧。因此在图 5-13 中将这样的帧记为“B, RR0, P”。对主站的这一命令，B 站响应以连续 4 个信息帧，其序号 N(S)从 0 到 3。最后在第 4 个信息帧中将 F 置 1，表示“我要发送的信息已发完”。这个帧记为“B, I30, F”。A 站在收到 B 站发来的 4 个信息帧后，发回确认帧 RR4（这时 N(R)=4）。我们注意到这时 P/F 比特并未置 1，所以 B 站收到 RR4 后不必应答。接下去 A 站轮询 C 站，P=1，虽然这时 C 站没有数据发送，但也必须立即应答。C 站应答也是 RR 帧，表示目前没有信息帧发送，F=1 表明这是回答对方命令的一个响应。

有了 P/F 比特，使 HDLC 规程使用起来更加灵活。在两个复合站全双工通信时，任何一方都可随时使 P=1，这时对方就要立即回答 RR 帧，并置 F=1，这样就可以收到对方的确认了。如果不使用 P/F 比特，则收方不一定马上发出确认帧，比如收方可以在发送自己的信息帧时，利用 N(R)把确认信息发出。

5. HDLC 操作

在图 5 中讨论了主站 A 和从站 B、C 交换信息的情况，这只是整个数据通信的中间阶段，在这个阶段之前还有一个数据链路的建立阶段，在数据传送完毕后，还必须有一个数据链路的释放阶段。也就是说 HDLC 执行数据传输控制功能，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 6 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。

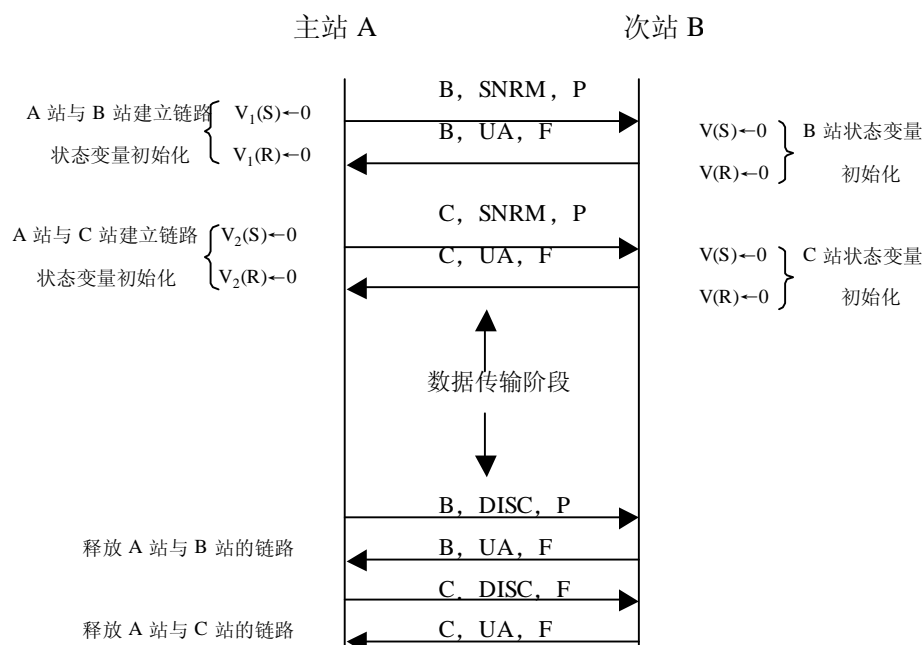


图 6 多点链路的建立和释放

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 7 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。

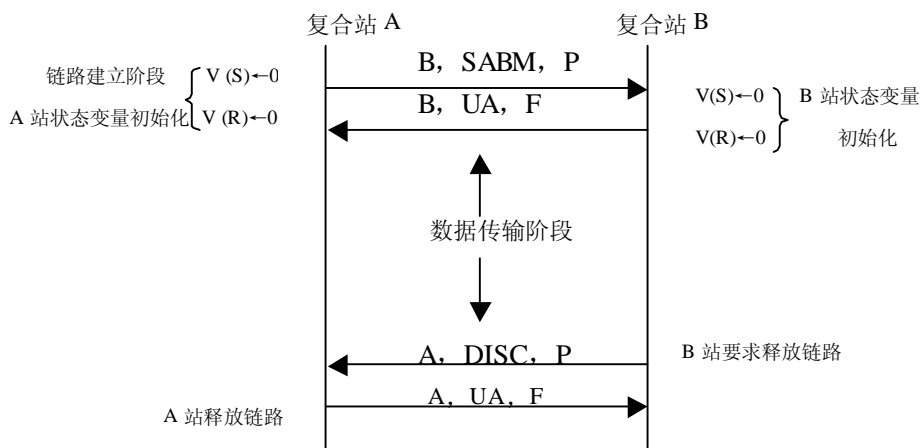


图 7 复合站的链路建立和释放

6. 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：

I 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。

I 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。

I 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。

I 适应性强

HDLC 规程能适应各种比特类型的工作站和链路。

I 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。

最后需要指出，一般的应用极少需要使用 HDLC 的全集，而选用 HDLC 的子集。当使

用某一厂商的 HDLC 时，一定要弄清该厂商所选用的子集是什么。