

资料编码		产品名称	
使用对象	用服工程师	产品版本	
编写部门	交换接入产品技术支持管理部	资料版本	

PPP专题

拟 制：周一帆

日 期：2002-01-05

审 核：

日 期：

审 核：

日 期：

批 准：

日 期：

修 订 记 录

日 期	修订版本	作 者	描 述
2002/01/05	v1.0	周一帆	



深 圳 市 华 为 技 术 有 限 公 司

目 录

第一章 概 述	1
1.1 PPP协议的基本概念	1
1.1.1 PPP协议出现的背景	1
1.1.1.1 SLIP协议的基本概念	1
1.1.1.2 SLIP协议的封装格式	1
1.1.2 PPP协议简介	2
1.2 总结	3
1.3 思考	3
第二章 PPP协议的三组件	4
2.1 PPP协议的组件	4
2.1.1 PPP协议的封装	4
2.1.2 LCP协议	6
2.1.3 NCP协议	7
2.2 总结	7
2.3 思考	7
第三章 PPP链路的建立	8
3.1 PPP链路的建立过程	8
3.1.1 PPP的状态转移图	8
3.1.2 LCP协议	10
3.1.2.1 LCP数据报文的封装方式	10
3.1.2.2 LCP数据报文的分类	12
3.1.2.3 LCP的链路配置报文	12
3.1.2.4 LCP的链路终止报文	15
3.1.2.5 LCP的链路维护报文	15
3.1.3 NCP协议	16
3.1.3.1 IPCP	16
3.2 总结	20
3.3 思考	20
第四章 LCP的可选配置参数	21
4.1 LCP的参数配置选项	21
4.1.1 魔术字 (Magic-Number)	21
4.1.2 认证协议	22
4.1.2.1 PAP认证	22

4.1.2.2 CHAP认证	24
4.1.3 MRU (Maxium Receive Unit)	25
4.2 总结	25
4.3 思考	26
第五章 PPP扩展协议	27
5.1 PPP扩展协议概述	27
5.1.1 MP出现的背景	27
5.1.2 MP (Multilink Protocol) 协议	27
5.2 总结	28
5.3 思考	28
第六章 PPP的状态机	29
6.1 PPP扩展协议概述	29

第一章 概述

1.1 PPP协议的基本概念

1.1.1 PPP协议出现的背景

在提及PPP协议时，不可不提及它的老祖宗SLIP（Serial Line Internet Protocol）协议。虽然它已被淡忘在历史的长河中，但毕竟有过辉煌的日子。它曾经主宰了Internet半边江山，人们不仅可以通过在计算机上安装该协议实现浏览Internet的梦想，而且还可以互连许多网络设备（如路由器与路由器的互连、路由器与主机的互连和主机与主机的互连）。随着网络技术的不断日新月异，特别是计算机技术的发展，人们开始渐渐认识到使用SLIP协议已不能满足日益增长的网络需求，如何在串行点对点的链路上封装IPX、AppleTalk等网络层的协议呢？这就给我们网络专家提出了新的挑战，也为PPP协议的出现提供了契机，PPP由于自身的诸多的优点已成为目前被广泛使用的数据链路层协议。

📖 说明

[如果对SLIP不感兴趣，可直接跳到1.1.2节](#)

1.1.1.1 SLIP协议的基本概念

SLIP协议出现在80年代中期，并被使用在BSD UNIX主机和SUN的工作站上，因为SLIP简单好用，所以后来被大量使用在线路速率从1200bps到19.2Kbps的专用线路和拨号线路上互连主机和路由器，到目前为止仍有问大部分UNIX主机保留对该协议的支持。在80年代末90年代初期，被广泛用于家庭中每台有RS232串口的计算机和调制解调器连接到Internet。SLIP是一种在点对点的串行链路上封装IP数据报的简单协议，它并非是Internet的标准协议。

1.1.1.2 SLIP协议的封装格式

SLIP协议的封装格式必需遵循以下几条原则：

- 通过在被发送IP数据报的尾部增加特殊的END字符（0xC0）从而形成一个简单的SLIP的数据帧，而后该帧会被传送到物理层进行发送。为了防止线路噪声被当成数据报的内容在线路上传输，通常发送端在被传送数

据报的开始处也传一个END字符。如果线路上的确存在噪声，则该数据报起始位置的END字符将结束这份错误的报文，这样当前正确的数据报文就能正确的传送了，而前一个含有无意义报文的数据帧会在对端的高层被丢弃。

- 当被传送的IP数据报文中含有END字符时，则需要对该字符进行转意（就是使用其它字符来表示），可使用连续传输的两个字节来代替它（如0xdb和0xdc）。如果当被转意后的字符也包含在数据报中，则也需要对其进行同样的操作，直至不出现歧义为止。下图为SLIP数据帧的封装格式：

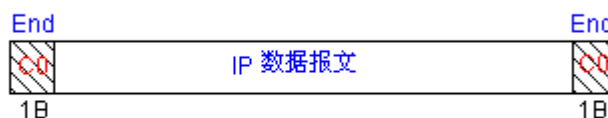


图1-1 SLIP数据帧格式

SLIP简单封装方式的缺陷：

- 从上图可以看出SLIP帧的封装格式非常简单，通信双方无需在数据报发送前协商任何配置参数选项（在PPP协议中需协商配置参数选项），所以双方IP层通信前必需先获知对方的IP地址，才能进行网络层的通信，否则链路层发送的数据帧在被送到对方网络层时将无法进行转发。
- 由于数据帧中也没有类似于以太网、HDLC和PPP等数据链路层协议中定义的协议域字段，因此SLIP仅支持一种网络层协议（IP协议）同一时刻在串行链路上发送。
- SLIP协议没有在数据帧的尾部加上CRC校验和，如果由于线路噪声的干扰影响传送数据包的内容是无法在对端的数据链路层中发现的，必须交由上层的应用软件来处理。

正是由于上面的诸多缺点，导致了SLIP很快的被后面要讲的PPP协议所替代。

1.1.2 PPP协议简介

PPP提供了一种在点对点的链路上封装多协议数据报（IP、IPX和AppleTalk）的标准方法。它不仅能支持IP地址的动态分配和管理；同步（面向位的同步数据块的传送）或异步（起始位+数据位+奇偶校验位+停止位）物理层的传输；网络层协议的复用；链路的配置、质量检测 and 纠错；而且还支持多种配置参数选项的协商。

PPP协议主要包括三部分：LCP（Link Control Protocol）链路控制协议、NCP（Network Control Protocol）和PPP的扩展协议（如Multilink Protocol，

详见第五章)。随着网络技术的不断发展,网络带宽已不再是瓶颈,所以PPP扩展协议的应用也就越来越少,因此往往人们在叙述PPP协议时经常会忘记它的存在。而且大部分网络教材上会将PPP的认证也作为PPP协议的一个主要部分,实际上这是一个错误概念的引导。PPP协议默认是不进行认证配置参数选项的协商,它只作为一个可选的参数,当点对点线路的两端需要进行认证时才需配置。当然在实际应用中这个过程是不可忽略的,例如我们使用计算机上网时,需要通过PPP协议与NAS设备互连,在整个协议的协商过程中,我们需要输入用户名和密码。因此当别人说PPP协议主要包括LCP、认证和NCP协议三个部分时,你不要认为他的说法有误,而只是不够准确罢了。

1.2 总结

- PPP协议由于自身诸多的优点取代了SLIP协议,从而成为目前被广泛使用的数据链路层协议
- SLIP协议归咎其简单数据包的封装方式,使其仅能在点对点的链路上封装单一的网络层协议(IP协议)
- PPP协议包括LCP协议、NCP协议和PPP扩展协议
- RFC1661文档中说明了PPP协议缺省是不进行PAP和CHAP认证

1.3 思考

- 1、当SLIP协议封装的IP数据报文中存在END字符时,发送该数据帧的网络设备会对该数据报文做什么样的处理?
- 2、SLIP协议没有引入CRC校验机制,那么它是如何保证数据发送的正确性的?
- 3、PPP协议不仅可以支持同步物理层传输,而且还支持异步物理层传输,请比较一下两者的区别?
- 4、PPP协议和SLIP协议的区别,可从封装格式,IP地址分配等方面考虑?

第二章 PPP协议的三组件

2.1 PPP协议的组件

首先简单介绍一下PPP协议的三组件：PPP协议的封装方式、LCP协议的协商过程和NCP协议的协商过程，然后再结合具体的LCP和NCP数据报的封装格式和两个阶段实际数据报文的交换过程，进一步理解PPP的LCP和NCP协商阶段的具体内容。

2.1.1 PPP协议的封装

我们知道ISO参考模型共分七层，自下而上分别是物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。通常会依据协议所完成的功能将它与这七层进行对照，PPP协议就属于数据链路层协议。

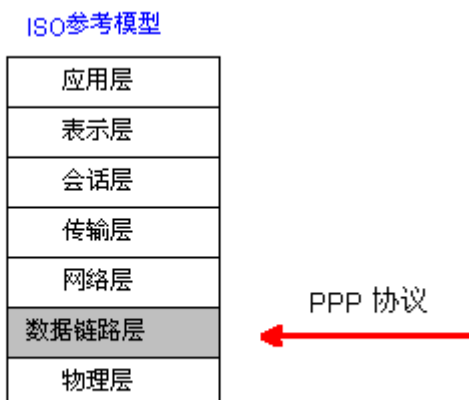


图2-1 PPP协议在网络模型中的位置

我们在提及PPP协议的报文封装格式时，不可不先提一下HDLC协议。HDLC也是最常用的数据链路层协议，它是从SDLC协议衍进过来的，许多常用的数据链路层协议的封装方式都是基于HDLC的封装格式的，同样PPP协议也不例外，它也采用了HDLC的定界帧格式。下图为PPP数据帧的封装格式：

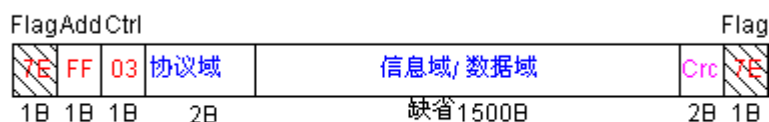


图2-2 PPP数据帧格式

以下为对PPP数据帧封装格式的一点说明：

- 每一个PPP数据帧均是以一个标志字节起始和结束的，该字节为0x7E。
- 紧接在起始标志字节后的一个字节是地址域，该字节为0xFF。我们熟知网络是分层的，且对等层之间进行相互通信，而下层为上层提供服务。当对等层进行通信时首先需获知对方的地址，而对不同的网络，在数据链路层则表现为需要知道对方的MAC地址、X.121地址、ATM地址等；在网络层则表现为需要知道对方的IP地址、IPX地址等；而在传输层则需要知道对方的协议端口号。例如如果两个以太网上的主机希望能够通信的话，首先发送端需获知对端的MAC地址。但由于PPP协议是被运用在点对点的链路上的特殊性，它不像广播或多点访问的网络一样，因为点对点的链路就可以唯一标示对方，因此使用PPP协议互连的通信设备的两端无须知道对方的数据链路层地址，所以该字节已无任何意义，按照协议的规定将该字节填充为全1的广播地址。
- 同地址域一样，PPP数据帧的控制域也没有实际意义，按照协议的规定通信双方将该字节的内容填充为0x03。
- 就PPP协议本身而言，我们最关心的内容应该是它的协议域和信息域。协议域可用来区分PPP数据帧中信息域所承载的数据报文的内容。协议域的内容必须依据ISO 3309的地址扩展机制所给出的规定。该机制规定协议域所填充的内容必须为奇数，也即是要求低字节的最低位为“1”，高字节的最低位为“0”。如果当发送端发送的PPP数据帧的协议域字段不符合上述规定，则接收端会认为此数据帧是不可识别的，那么接收端会

向发送端发送一个Protocol-Reject报文，在该报文尾部将完整地填充被拒绝的报文。协议域的具体取值如下表所示：

	协议域类型	说明
ISO 标 准	0x0*** - 0x3***	信息域中承载的是网络层的数据报文
	0x4*** - 0x7***	信息域中承载的是与NCP无关的低整流量
	0x8*** - 0xb***	信息域中承载的是网络控制协议（NCP）的数据报文
	0xc*** - 0xf***	信息域中承载的是链路控制协议（LCP）的数据报文
最典 型的 几种 取值	0xc021	信息域中承载的是链路控制协议（LCP）的数据报文
	0xc023	信息域中承载的是PAP协议的认证报文
	0xc223	信息域中承载的是CHAP协议的认证报文
	0x8021	信息域中承载的是网络控制协议（NCP）的数据报文
	0x0021	信息域中承载的是IP数据报文

说明

协议域类型中的*号表示可从（0-F）中任意取值


- 信息域缺省时最大长度不能超过1500字节，其中包括填充域的内容（在图2-1中并未表示，因为它属于信息域的一部分），1500字节大小等于PPP协议中配置参数选项MRU（Maximum Receive Unit）的缺省值，在实际应用当中可根据实际需要进行信息域最大封装长度选项的协商。信息域如果不足1500字节时可被填充，但不是必须的，如果填充则需通信双方的两端能辨认出有用与无用的信息方可正常通信。

我们通常在通信设备的配置过程中，遇到最多的也是MTU（Maximum Transmit Unit）。对于一个设备而言，它网络的层次均使用MTU和MRU两个值，一般情况下设备的MRU会比MTU稍大几个字节，但这需根据各厂商的设备而定。

- CRC校验域主要是对PPP数据帧传输的正确性进行检测的，当然在数据帧中引入了一些传输的保证机制是好的，但可以反过来说，同样我们会引入更多的开销，这样可能会增加应用层交互的延迟，对于这个字节的使用我们可以参考一下X.25协议和FR协议就知道了。

2.1.2 LCP协议


为了能适应复杂多变的网络环境，PPP协议提供了一种链路控制协议来配置和测试数据通信链路，它能用来协商PPP协议的一些配置参数选项；处理不同大小的数据帧；检测链路环路、一些链路的错误；终止一条链路。

 说明

[详细内容请见3.1.2节LCP协议](#)

2.1.3 NCP协议

PPP的网络控制协议根据不同的网络层协议可提供一族网络控制协议，常用的有提供给TCP/IP网络使用的IPCP网络控制协议和提供给SPX/IPX网络使用的IPXCP网络控制协议等，但最为常用的是IPCP协议，当点对点的两端进行NCP参数配置协商时，主要是用来通信双方的网络层地址。

 说明

[详细内容请见3.1.3节NCP协议](#)

2.2 总结

- PPP协议的三组件包括PPP协议的封装方式、LCP协议和NCP协议
- PPP协议是数据链路层协议，它的数据帧封装格式非常类似于HDLC
- PPP协议可通过协议域来区分数据域中净载荷的数据类型
- PPP协议通过LCP协议完成数据链路的配置和测试
- PPP协议通过NCP协议完成点对点通信设备之间网络层通信所需参数的配置

2.3 思考

1、PPP数据帧中的地址域被填充为0xFF（广播地址），在实际应用当中该域已没有任何意义，请想一下为什么使用PPP协议通信的设备不需要类似于以太网的数据链路层寻址机制？

2、PPP协议数据域缺省的最大值是多少？

3、当发送端发送的PPP数据帧的协议域不被接收方识别时，接收方将如何处理这个数据帧？

4、IPCP协议的主要功能？

第三章 PPP链路的建立

3.1 PPP链路的建立过程

3.1.1 PPP的状态转移图

数据通信设备（在本文中指路由器）的两端如果希望通过PPP协议建立点对点的通信，无论哪一端的设备都需发送LCP数据报文来配置链路（测试链路）。一旦LCP的配置参数选项协商完后，通信的双方就会根据LCP配置请求报文中所协商的认证配置参数选项来决定链路两端设备所采用的认证方式。协议缺省情况下双方是不进行认证的，而直接进入NCP配置参数选项的协商，直至所经历的几个配置过程全部完成后，点对点的双方就可以开始通过已建立好的链路进行网络层数据报文的传送了，整个链路就处于可用状态。只有当任何一端收到LCP或NCP的链路关闭报文时（一般而言协议是不要求NCP有关闭链路的能力的，因此通过情况下关闭链路的数据报文是在LCP协商阶段或应用程序会话阶段发出的）；物理层无法检测到载波或管理人员对该链路进行关闭操作，都会将该条链路断开，从而终止PPP会话。以下为PPP协议整个链路过程需经历阶段的状态转移图：

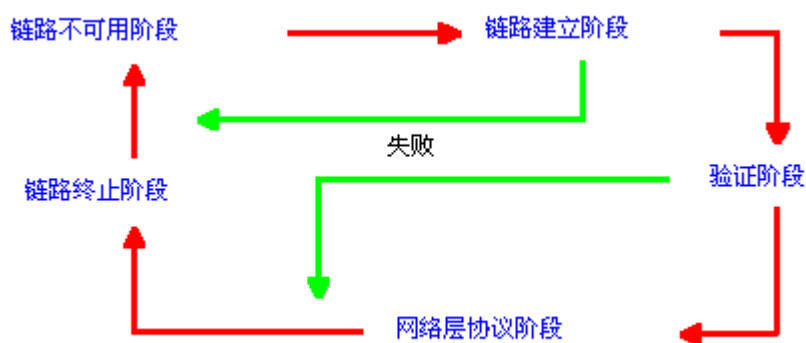


图3-1 PPP的状态转移图

在点对点链路的配置、维护和终止过程中，PPP需经历以下几个阶段：

- **链路不可用阶段**，有时也称为物理层不可用阶段，PPP链路都需从这个阶段开始和结束。当通信双方的两端检测到物理线路激活（通常时检测到链路上有载波信号）时，就会从当前这个阶段跃迁至下一个阶段（即链路建立阶段）。先简单提一下链路建立阶段，在这个阶段主要是通过LCP协议进行链路参数的配置，LCP在此阶段的状态机也会根据不同的

事件发生变化。当处于在链路不可用阶段时，LCP的状态机是处于initial（初始化状态）或starting（准备启动状态），一旦检测到物理线路可用，则LCP的状态机就要发生改变。当然链路被断开后也同样会返回到这个阶段，往往在实际过程中这个阶段所停留的时间是很短的，仅仅是检测到对方设备的存在。

- **链路建立阶段**，也是PPP协议最关键和最复杂的阶段。该阶段主要是发送一些配置报文来配置数据链路，这些配置的参数不包括网络层协议所需的参数。当完成数据报文的交换后，则会继续向下一个阶段跃迁，该下一个阶段既可是验证阶段，也可是网络层协议阶段，下一阶段的选择是依据链路两端的设备配置的（通常是由用户来配置，但对NAS或BAS设备的PPP模块缺省就需要支持PAP或CHAP中的一种认证方式）。在此阶段LCP的状态机会发生两次改变，前面我们说了当链路处于不可用阶段时，此时LCP的状态机处于initial或starting，当检测到链路可用时，则物理层会向链路层发送一个UP事件，链路层收到该事件后，会将LCP的状态机从当前状态改变为Request-Sent（请求发送状态），根据此时的状态机LCP会进行相应的动作，也即是开始发送Config-Request报文来配置数据链路，无论哪一端接收到了Config-Ack报文时，LCP的状态机又要发生改变，从当前状态改变为opened状态，进入Opened状态后收到Config-Ack报文的一方则完成了当前阶段，应该向下一个阶段跃迁。同理可知，另一端也是一样的，但须注意的一点是在链路配置阶段双方是链路配置操作过程是相互独立的。如果在该阶段收到了非LCP数据报文，则会的将这些报文丢弃。在实际配置当中在该阶段可能会遇到很多情况，在LCP协议章节中会详细介绍可能遇到的情况，但最好结合一些troubleshooting案例能更好的帮助理解。
- **验证阶段**，多数情况下的链路两端设备是需要经过认证后才进入到网络层协议阶段，缺省情况下链路两端的设备是不进行认证的。在该阶段支持PAP和CHAP两种认证方式，验证方式的选择是依据在链路建立阶段双方进行协商的结果。然而，链路质量的检测也会在这个阶段同时发生，但协议规定不会让链路质量的检测无限制的延迟验证过程。在这个阶段仅支持链路控制协议、验证协议和质量检测数据报文，其它的数据报文都会被丢弃。如果在这个阶段再次收到了Config-Request报文，则又会返回到链路建立阶段。
- **网络层协议阶段**，一旦PPP完成了前面几个阶段，每种网络层协议（IP、IPX和AppleTalk）会通过各自相应的网络控制协议进行配置，每个NCP协议可在任何时间打开和关闭。当一个NCP的状态机变成Opened状态时，则PPP就可以开始在链路上承载网络层的数据包报文了。如果在个阶段收到了Config-Request报文，则又会返回到链路建立阶段。
- **网络终止阶段**，PPP能在任何时候终止链路。当载波丢失、授权失败、链路质量检测失败和管理员人为关闭链路等情况均会导致链路终止。链

路建立阶段可能通过交换LCP的链路终止报文来关闭链路，当链路关闭时，链路层会通知网络层做相应的操作，而且也会通过物理层强制关断链路。对于NCP协议，它是没有也没有必要去关闭PPP链路的。

3.1.2 LCP协议

3.1.2.1 LCP数据报文的封装方式

LCP数据报文是在链路建立阶段被交换的，它作为PPP的净载荷被封装在PPP数据帧的信息域中，此时PPP数据帧的协议域固定填充0xC021，但在链路建立阶段的整个过程中信息域的内容是在变化的，它包括很多种类型的报文，所以这些报文也要通过相应的字段来区分。LCP数据报文的一般封装方式如下图所示：



图3-2 LCP报文的封装格式

- 代码域的长度为一个字节，主要是用来标识LCP数据报文的类型的。在链路建立阶段时，接收方收到LCP数据报文的代码域无法识别时，就会向对端发送一个LCP的代码拒绝报文（Code-Reject报文）。根据RFC的规定，到目前为止LCP共包括以下几种类型的数据报文：

代码值	报文类型	代码	报文类型
0x01	Config-Request	0x07	Code-Reject
0x02	Config-Ack	0x08	Protocol-Reject
0x03	Config-Nak	0x09	Echo-Request
0x04	Config-Reject	0x0A	Echo-Reply
0x05	Terminate-Request	0x0B	Discard-Request
0x06	Terminate-Ack	0x0C	Reserved

- 标识域也是一个字节，其目的是用来匹配请求和响应报文。一般而言在进入链路建立阶段时，通信双方无论哪一端都会连续发送几个配置请求报文（Config-Request报文），而这几个请求报文的数据域可能是完全一

样的，而仅仅是它们的标志域不同罢了。通常一个配置请求报文的ID是从0x01开始逐步加1的，当对端接收到该配置请求报文后，无论使用何种报文（回应报文可能是Config-Ack、Config-Nak和Config-Reject三种报文中的一种）来回应对方，但必须要求回应报文中的ID要与接收报文中的ID一致，当通信设备收到回应后就可以将该回应与发送时的进行比较来决定下一步的操作。

说明

后面教材中的所有例子，均可参考以下色标的含义

范例中报文色标含义如下图所示：








	PPP数据帧的起始/结束标志字节
	PPP数据帧的地址域和控制域
	PPP数据帧的协议域
	LCP数据报文的类型域
	LCP数据报文的配置参数选项或报文内容
	LCP数据报文标识域
	LCP数据报文的长度域

图3-3 报文色标含义定义

例1：假设点对点通信的一端发送了一个Config-Request报文，报文内容如下：

7E FF 03 C0 21 01 01 00 17 02 06 00 0A 00 00 05 06 00 0B 42 CB 07 02 08 02 0D 03 06 7E

从报文中可以看出这个配置请求报文包括5个配置参数选项。

当对端正确接收到了该报文后，应该回应一个Config-Ack报文，报文内容如下：

7E FF 03 C0 21 02 01 00 17 02 06 00 0A 00 00 05 06 00 0B 42 CB 07 02 08 02 0D 03 06 7E

该报文中唯一修改的内容就是代码域（02表示是Config-Ack报文），标识域与原报文中的一样。

- 长度域的内容 = 总字节数据（代码域+标志域+长度域+数据域）。长度域所指示字节数之外的字节将被当作填充字节而忽略掉，而且该域的内容不能超过MRU的值。
- 数据域的内容依据不同LCP数据报文的内容也是不一样的。

说明

数据域的详细内容请见3.1.2.3节

3.1.2.2 LCP数据报文的分类

在前一小节我们可以看到，一共包括12种LCP数据报文，我们依据各报文的的功能又将其具体细化为以下三类：

- 链路配置报文，主要用来建立和配置一条链路的。
- 链路终止报文，主要用来终止一条链路的。
- 链路维护报文，主要用来维护和调试链路的。

3.1.2.3 LCP的链路配置报文

链路配置报文与其它两类报文有着明显的区别，它主要是用来协商链路的配置参数选项的，因此这种报文的数据域还要携带许多配置参数选项的，而另外两类报文中部分报文的格式会稍有不同（请参见RFC1661），图3-4表明了数据配置选项的报文格式：

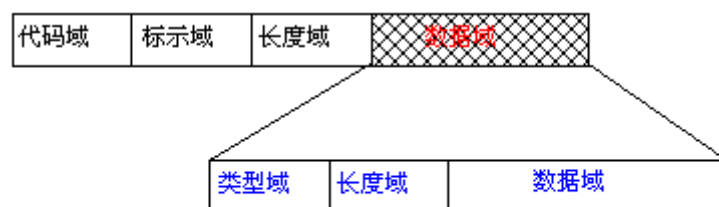


图3-4 配置报文中配置选项的数据格式

链路配置报文主要包括 Config-Request、Config-Ack、Config-Nak 和 Config-Reject 四种报文。

当通信双方需要建立链路时，无论哪一方都需要发送 Config-Request 报文并携带每一端自己所希望协商的配置参数选项，下表为一些可选的配置参数选项：

类型值	参数选项	类型值	参数选项
0x00	Reserved	0x05	Magic-Number
0x01	Maximum-Receive-Unit	0x06	CBCP
0x02	Async-Control-Character-Map	0x07	Protocol-Field-Compress
0x03	Authentication-Protocol	0x08	Address-and-Control-Field-Compress
0x04	Quality-Protocol	0x0D	Multilink-Protocol

这个表格内的内容并非完全，可能还有一新定义的配置选项未列在其中，如有兴趣可参照相关的文档说明。

当接收方收到 Config-Request 报文时，会在剩下的三种类型的报文中选择一种来响应对方的请求报文，到底选择哪种报文来响应对方需依据以下两个条件：

- 不能完全识别配置参数选项的类型域，我们知道一个 Config-Request 报文中会同时携带多个配置参数选项，而对于一个支持 PPP 协议的通信设备也不一定会支持上表中所有列出的配置选项，即使支持，也可能在实际应用中关闭掉某些选项功能。（例如：当使用 PPP 协议通信的一端可能将一些无用的配置选项都关闭了，而仅支持 0x01 和 0x03 两个配置参数选项，因此当对方发送的 Config-Request 报文中含有 0x04 配置选项时，对于本端而言这个配置参数选项就无法识别，也即是不支持这个配置参数选项的协商）。
- 如果能支持完全识别配置参数选项，但接收端也可能不认可 Config-Request 报文中配置参数选项数据域中的内容（例如：当一端发送魔术字配置参数选项中的魔术字为全 0，而对端认为应该为其它值，这种情况就属于不支持配置参数选项中的内容）。

所以依据上面的两个条件，我们就可以明确在回应对方配置请求报文时，采用何种报文回应。

- 当接收 Config-Request 报文的一端能识别发送过来的所有配置参数选项且认可所有配置参数选项数据域的内容时，接收端将会给对端回一个 Config-Ack 报文并将配置请求报文中的配置参数选项原封不动的放置在 Config-Ack 报文的数据域内（根据协议的规定是不可改变配置参数选项

的顺序)。当配置请求报文的发送端收到Config-Ack报后,则会从当前阶段进入到下一个阶段。

例2: 假设点对点通信的一端发送了一个Config-Request报文, 报文内容如下:

7E FF 03 C0 21 01 01 00 17 02 06 00 0A 00 00 05 06 00 0B 42 CB 07 02 08 02 0D 03 06 7E

当对端正确接收到了该报文后, 应该发送一个Config-Ack报文, 报文内容如下:

7E FF 03 C0 21 02 01 00 17 02 06 00 0A 00 00 05 06 00 0B 42 CB 07 02 08 02 0D 03 06 7E

唯一改变的内容就是代码域(02表示是Config-Ack报文), 此例与例1完全一样, 但两都说明的问题有则重点。

- 当接收Config-Request报文的一端能识别发送端所发送过来的所有配置参数选项, 但对部分配置参数选项数据域中的内容不认可时, 接收端将会给对端回应一个Config-Nak报文, 该报文中只携带不认可的配置参数选项, 而这些配置参数选项的数据内容为本端希望的值。然而当接收端收到Config-Nak报文后, 会重新发送Config-Request报文, 而这个Config-Request报文与上一次所发送的Config-Request报文区别在于那些被对端不认可的配置参数选项的内容被填写到刚刚协商完后再次发送的Config-Request报文中(Config-Nak报文发送回来的那些配置参数选项)。

例3: 假设点对点通信的一端发送了一个Config-Request报文, 报文内容如下:

7E FF 03 C0 21 01 01 00 17 02 06 00 0A 00 00 05 06 00 0B 42 CB 07 02 08 02 0D 03 06 7E

该数据报文中下划线的配置参数选项的内容为对端不认可的。

当对端正确接收到了该报文后, 发现类型域为0x02的配置参数选项可识别, 但该配置参数选项数据域的内容不认可, 应发送一个Config-Nak报文且该报文中将携带希望的配置参数选项内容, 报文内容如下:

7E FF 03 C0 21 03 01 00 0A 02 06 00 0E 00 00 7E

该报文中返回的值已经被更改, 且当发端收到该报文后会重新发送一个Config-Request报文, 报文内容如下:

7E FF 03 C0 21 01 04 00 17 02 06 00 0E 00 00 05 06 00 0B 42 CB 07 02 08 02 0D 03 06 7E

仔细观察是不是新的配置请求报文与老的配置请求的报文ID不一样。

- 当接收Config-Request报文的一端不能识别所有的发送端发送过来的配置参数选项时, 此时接收端将会向对端回一个Config-Reject报文, 该报文中的数据域只携带那些不能识别的配置参数选项(当配置参数选项的类型域不识别时)。当对端接收到Config-Reject报文后, 同样会再次发送一个Config-Request报文, 这个配置请求报文与上一次发送的区别在于将不可识别的那些配置参数选项给删除了。

例4: 假设点对点通信的一端发送了一个Config-Request报文, 报文内容如下:

7E FF 03 C0 21 01 01 00 17 02 06 00 0A 00 00 05 06 00 0B 42 CB 07 02 08 02 0D 03 06 7E

下划线所表示的配置参数选项为对端不可识别的。

当对端正确接收到了该报文后，发现类型为0x02的配置参数选项不识别，应该回应一个Config-Reject报文，报文内容如下：

7E FF 03 C0 21 04 01 00 0A 02 06 00 0A 00 00 7E

该报文如果被原发送端接收后，又会重新发送一个Config-Request报文，报文内容如下：

7E FF 03 C0 21 01 04 00 11 05 06 00 0B 42 CB 07 02 08 02 0D 03 06 7E

这时我们能看到，类型为02的配置选项在下一次的请求报文中被删除了。

所以我们能够看出，链路配置阶段也可能要经过几次协商后才能完成，但这与点对点两端的设备有着密切的联系。对于PPP的两个端点而言，两者是独立完成各自的配置参数选项的协商过程的。具体的配置参数选项待后续的章节中讲解。

3.1.2.4 LCP的链路终止报文

链路终止报文分为Terminate-Request和Terminate-Reply两种报文。LCP报文中提供了一种机制来关闭一个点对点的连接，想要关断链路的一端会持续发送Terminate-Request报文，直到收到一个Terminate-Reply为止。接收端一旦收到了一个Terminate-Request报文后，必须回应一个Terminate-Reply报文，同时等待对端先将链路断开后，再完成本端的所有断开的操作。

LCP的链路终止报文的数据域与链路配置报文的数据域不一样，链路终止报文中无需携带各配置参数选项。对于链路终止报文也同样需要ID一致，当接收到Terminate-Reply报文才会做链路终止操作。

3.1.2.5 LCP的链路维护报文

除上述两种报文类型以外，剩余的所有报文类型均归属链路维护报文。

- 当接收端发现LCP报文的代码域是一个不合法的值时，将会向发送端回应一个Code-Reject报文，在回应报文中会将所拒绝报文的内容附上。

例5：假设点对点通信的一端发送了一个Config-Request报文，报文内容如下：

7E FF 03 C0 21 0E 01 00 19 02 06 00 0A 00 00 05 06 00 0B 42 CB 07 02 08 02 0D 03 06 1F 02 7E

有下划线的部分表示这个代码域在标准中未定义，从而无法识别。

当对端正确接收到了该报文后，发现LCP数据报文的代码域为0x0E时，应该发送一个Code-Reject报文，报文内容如下：

7E FF 03 C0 21 07 01 00 1D 0E 01 00 19 02 06 00 0A 00 00 05 06 00 0B 42 CB 07 02 08 02 0D 03

06 1F 02 7E

- 当接收端发现所接收到的数据帧的协议域是一个不合法的值时，将会向发送端回应一个Protocol-Reject报文，发送端收到该拒绝报文后将停止发送该种协议类型的数据报文了。Protocol-Reject报文只能在LCP的状态机处于Opened状态时才可被处理，而在其它状态接收到该报文将被丢弃。在Protocol-Reject报文的数据域内将携带所拒绝报文的协议类型和报文内容。

例6：假设点对点通信的一端发送了一个协议域未定义（无法识别）的报文，报文内容如下：

7E FF 03 77 77 01 01 00 19 02 06 00 0D 00 00 05 07 03 09 02 0D 04 06 2F 02 7E

其中下划线部分为PPP数据帧的协议域，0x7777表示一个未定义的类型（也即对端无法识别）。

当对端正确接收到了该报文后，发现该报文的协议域为0x7777，该值未在RFC中未有明确定义，应该发送一个Protocol-Reject报文，报文内容如下：

7E FF 03 C0 21 08 01 00 18 77 77 01 00 00 19 02 06 00 0D 00 00 05 07 03 09 02 0D 04 06 2F 02

7E

- Echo-Request报文和Echo-Reply报文主要用来检测双向链路上自环的问题，除此之外还可附带做一些链路质量的测试和其它功能。当LCP状态机处于Opened状态时，如果接收到了Echo-Request，就需向对端回送一个Echo-Reply报文。否则在其它LCP状态下，该类型的报文会被丢弃。这种类型数据报文的长度域后不是直接跟数据域，而是要插入4个字节的Magic-Number（魔术字），该魔术字是在LCP的Config-Request的配置参数选项协商时获得的。

例7：假设点对点通信的一端接收到了一个Echo-Request报文，报文内容如下：

7E FF 03 C0 21 09 01 00 08 02 06 00 0D 7E

有下划线的部分表示魔术字。

当对端正确接收到了该报文后，应该发送一个Echo-Reply报文，报文内容如下：

7E FF 03 C0 21 0A 01 00 08 02 06 00 0D 7E

3.1.3 NCP协议

NCP协议的数据报文是在网络层协议阶段被交换的，在这个阶段所需的一些配置参数选项协商完后，就可以进行网络层的通信，也即是在点对点的链路上可以开始传送网络层的数据报文了。NCP协议主要包括IPCP、IPXCP等，但我们在实际当中最常遇见的也只有IPCP协议了，如果对IPXCP或其它的一些网络控制协议有兴趣，则可参见RFC1552。

3.1.3.1 IPCP

IPCP控制协议主要是负责完成IP网络层协议通信所需配置参数的选项协商的。IPCP在运行的过程当中，主要是完成点对点通信设备的两端动态的协商IP地址。我们依据两端设备的配置选项可将IPCP的协商过程分为“静态”和“动态”。何为静态，何为动态，这是一个相对的概念，也是自己总结出来的，可能不太准确，只作为参考使用。我们在下面会具体描述这两个过程。IPCP的数据的文同LCP的数据报文非常类似，只不过一个是在网络层协议阶段协商配置参数选项，而LCP协议则是在链路建立阶段协商配置参数选项的。除此之外是两者在所使用报文上的相似之处，我们知道LCP共包括十几种报文，而IPCP也包括多种报文，但它的报文类型只是LCP数据报文的一个子集（只有LCP代码域从1到7这七种报文），而且实际的数据报文交换过程中也仅涉及以下几种：Config-Request、Config-Ack、Config-Nak和Config-Reject（代码域从1到4，而链路终止报文一般而言是不在网络协议阶段使用的，而且也是不需要的）。下面就具体介绍一下IPCP控制协议的静态和动态的两个过程，实际上两者的区分是在于互连设备IP地址的获取过程。

- 静态协商，也即是不协商。点对点的通信设备两端在PPP协商之前已配置好了IP地址，所以就无须在网络层协议阶段协商IP地址，而双方唯一要做的就是告诉对方自身的IP地址。在IPCP控制协议完成整个配置的过程时，最理想的情况将会看到如图所示的四种报文，而无其它报文再被发送。如果当配置请求中所携带的网络层配置参数选项类似于LCP配置参数选项协商过程一样，可能会有对方不识别的配置参数选项或不被对方所认可的配置参数选项的内容。这样在这个阶段的协商过程中可能还会看到其它的一些报文。

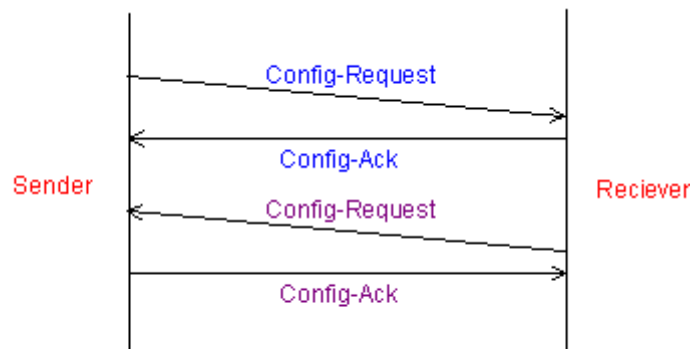


图3-5 静态协商

在静态协商时，如果IPCP的Config-Request报文中只含有地址配置参数选项时（实际中可能还会附带其它配置参数选项，这些配置参数选项的协商与LCP阶段的一样，而我这里只提到了IP地址配置参数选项），无论是发送方还是接收方都同时发送Config-Request报文，其中配置选项中只含有各自的IP地址。当对端收到该报文后，会发送一个Config-Ack报文，这个目的是告诉对端我已经知道了你的IP地址，对路由器而言会增加一条到对端接口的主机路由。刚进入网络层协议阶段时，IPCP的状态机是initial的，但当完成了上述的整个过程后，IPCP的状态机改变为Opened，双方也就可以开始网络层数据网的传送了。

IPCP协议中并未规定，当一端接收到Config-Request报文后，它从报文的配置参数选项中可获知对端的IP地址，但并不与本端的IP地址进行比较。我们也知道，一般而言点对点的两端应该是在一个网段。但如果双方的地址不在一个网段，就不给对方回应Config-Ack报文，而是无条件的回送一个回应报文。因此说点对点通信的两端如果是手动设置每一端的IP地址时，无须双方地址在同一网段。

例8：假设IPCP在网络层协议阶段开始协商配置参数选项（这里只举协商IP地址的配置参数选项地的过程），发送方设置IP地址为192.168.0.1，接收方设置IP地址为192.168.0.2，发送方发送给Config-Request报文内容如下：

7E FF 03 80 21 01 01 00 0A 03 06 C0 A8 00 01 7E

在这个例子中我们能看见明显的改变之处再于PPP协议域字段由原先的0xC021改变为0x8021，下划线的部分表示本端的IP地址。

当对端正确接收到了该报文后，应该回应一个Config-Ack报文，报文内容如下：

7E FF 03 80 21 02 01 00 0A 03 06 C0 A8 00 01 7E

同样的接收方给发送方也发送一个Config-Request报文内容如下，但此时报文中IP地址配置参数选项的值为本端的IP地址（192.168.0.2）：

7E FF 03 80 21 01 01 00 0A 03 06 C0 A8 00 02 7E

发送方回应一个Config-Ack报文给接收方，报文内容如下：

7E FF 03 80 21 02 01 00 0A 03 06 C0 A8 00 02 7E

- 动态协商，也即是一端配置为动态获取IP地址，另一端通过手动方式配置IP地址，且允许给对端分配IP地址，这个过程实际上可与窄带拨号上网的过程相一致，如果我们想用计算机上网，均会安装一个拨号适配器，而

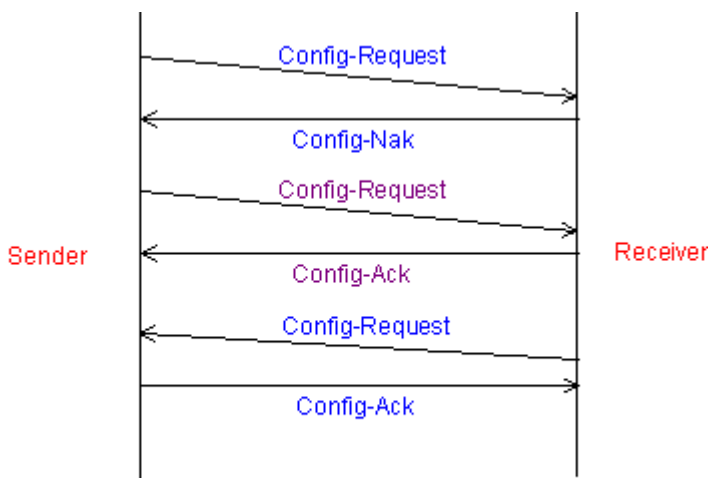


图3-6 动态协商

且计算机中的拨号网络适配器是采用动态获取IP地址的方式。

这个例子与一个例子相似，也就是在IPCP的Config-Request报文中只携带IP地址的配置参数选项。如果是配置参数选项中含有其它配置参数选项，则可能会遇到其它的一些情况（如不识别配置参数选项的代码域或不认可配置参数选项的内容，但对于这些情况的处理方法和LCP配置参数选项的处理方法一致）。图3-6中我们可以看出发送方连续发送了两次Config-Request报文，才能完成发送方的协商过程。而接收方仍然只需要发送一次Config-Request即可完成本端的协商过程。

由于发送方没有配置IP地址（而是动态获取IP地址），所以在IPCP的Config-Request报文的IP地址配置参数配置选项中的IP地址填充全0（也即是0.0.0.0），这样当对端收到这个Config-Request报文时，当接收方收到该配置请求报文后会迅检测IP地址的内容，如果发送为全0，则认为对端的这个IP地址不是我所希望的值，这样就回应一个Config-Nak报文，并将希望分配给对方的IP地址填充到Config-Nak报文内。这时当接收方

收到Config-Nak报文后，就会重新发送一个Config-Request报文，这个报文中的IP地址配置选项为对方在Nak报文中所提供的。

接收方IP地址的配置过程与静态时的一样，只需发送一个Config-Request报文即可，当收到发送方的Config-Ack报文，就表示接收方的IP地址配置完成。

例9：假设IPCP在网络层协议阶段开始协商配置参数选项（这里只举协商IP地址的配置参数选项地的过程），发送方没有配置IP地址，而接收方配置了IP地址为192.168.0.2，接收方可给发送方分配IP地址（192.168.0.1），发送方发送给接收方的Config-Request报文内容如下：

7E FF 03 80 21 01 01 00 0A 03 06 00 00 00 00 7E

有下划线的部分表示本端的IP地址。

当对端正确接收到了该报文后，应该回应一个Config-Nak报文，报文内容如下：

7E FF 03 80 21 03 01 00 0A 03 06 C0 A8 00 01 7E

当接收方收到该报文后，重新发送一个Config-Request报文，报文内容如下：

7E FF 03 80 21 01 02 00 0A 03 06 C0 A8 00 01 7E

接收方再次收到发送方的一个Config-Request报文，此时将回应一个Config-Ack报文，报文内容如下：

7E FF 03 80 21 02 02 00 0A 03 06 C0 A8 00 01 7E

请仔细观察一下这些报文在交互过程中，PPP数据帧净载荷内的数据报文的类型域和报文ID。

同样的接收方给发送方也发送一个Config-Request报文，报文内容如下：

7E FF 03 80 21 01 01 00 0A 03 06 C0 A8 00 02 7E

发送方应回送一个Config-Ack给接收方，报文内容如下：

7E FF 03 80 21 02 01 00 0A 03 06 C0 A8 00 02 7E

本章节的一些内容可能没有明确写出，只是将IPCP配置参数选项配置过程中最关键的部分做了一些说明，如果想深入了解解决IPCP或IPXCP，可参见相关的RFC文档。

3.2 总结

- PPP协议的状态转移图包括链路不可用阶段、链路建立阶段、认证阶段、网络层协议阶段和链路终止阶段
- LCP协议依据报文的功能可分为链路配置报文、链路终止报文和链路维护报文

- LCP协议的链路配置报文主要是用来协商一些可选的配置参数选项
- LCP协议的链路终止报文主要是用来终止一条PPP链路
- LCP协议的链路维护报文主要是用来测试和调试PPP链路
- NCP协议主要负责网络层配置参数选项的协商，它包括“静态协商”和“动态协商”

3.3 思考

- 1、当发送端在链路建立阶段开始时，发送一个Config-Request报文，那么当接收端接收到该数据报文后，什么情况下回应Config-Ack报文，什么情况下回应Config-Nak报文，而什么情况下又回应Config-Reject报文？
- 2、按功能分，Echo-Request报文和Echo-Reply报文属于LCP协议哪种类型报文，在这种报文中需要携带链路配置报文中协商何种配置参数选项？
- 3、IPCP协议在进行网络层配置参数选项协商时可能会遇到哪两种协商，当只协商IP地址选项时，请对比一下这两种配置参数选项的区别？当需要进行多种参数配置参数选项时（请回忆一下LCP配置参数选项协商的情况），可能会出现哪几种情况？

第四章 LCP的可选配置参数

4.1 LCP的参数配置选项

LCP协议在对链路配置过程中需要进行一些可选配置参数选项的协商，我们在前面讲述的过程中已列举了许多配置参数选项，但我们只挑选其中较为重要的几个参数做相应的扩展说明（魔术字和认证协议选项）。关于一些更具体的细节和未涉及到的配置参数选项，请参考PPP的RFC文档（RFC1661）。

4.1.1 魔术字（Magic-Number）

魔术字是在LCP的Config-Request报文中被协商的，并且可被其它一些其它类型的LCP数据报文所使用，如前面已经说过的Echo-Request、Echo-Reply报文和Quality-Protocol报文等。对于PPP协议本身它是不要求协商魔术字的，如果在双方不协商魔术字的情况下，某些LCP的数据报文需要使用魔术字时，那么只能是将魔术字的内容填充为全0；反之，则填充为配置参数选项协商后的结果。

魔术字在目前所有的设备当中都是需要进行协商的，它被放在Config-Request的配置选项参数中进行发送，而且需要由自身的通信设备独立产生，协议为了避免双方可能产生同样的魔术字，从而导致通信出现不必要的麻烦，因此要求由设备采用一些随机方法产生一个独一无二的魔术字。一般来说魔术字的选择会采用设备的系列号、网络硬件地址或时钟。双方产生相同魔术字的可能性不能说是没有的，但应尽量避免，通常这种情况是发产在相同厂商的设备进行互连时，因为一个厂商所生产的设备产生魔术字的方法是一样的。

我们知道魔术字产生的作用是用来帮助检测链路是否存在环路，当接收端收到一个Config-Request报文时，会将此报文与上一次所接收到的Config-Request进行比较，如果两个报文中所含的魔术字不一致的话，表明链路不存在环路。但如果一致的话，接收端认为链路可能存在环路，但不一定存在环路，还需进一步确认。此时接收端将发送一个Config-Nak报文，并在该报文中携带一个重新产生的魔术字，而且此时在未接收到任何Config-Request或Config-Nak报文之前，接收端也不会发送任何的Config-Request报文。这时我们假设可能会有以下两种情况发生：

- 链路实际不存在环路，而是由于对方在产生魔术字时与接收端产生的一致，但实际这种情况出现的概率是很小的。当Config-Nak被对端接收到后，应该发送一个Config-Request报文（此报文中的魔术字为Nak报文中

的），当对端接收到后，与上次比较，由于接收端已经在Nak报文中产生了一个不同的魔术字，此时接收端收到的Config-Request报文中的魔术字与上次配置请求报文中不一样，所以接收端可断定链路不存在环路。

- 链路实际上确实存在环路，一段时间后Config-Nak报文会返回到发送该报文的同一端。这时接收端比较这个Config-Nak报文与上一次发出去的一样，因此链路存在环路的可能性又增大了。我们知道当一端收到了一个Config-Nak报文时，又会发送一个Config-Request报文（该报文中的魔术字与Config-Nak中的一致），这样又回到了最初的状态，在这条链路上就会不断的出现Config-Request、Config-Nak报文，因此这样周而复始下去，接收端就会认为PPP链路存在环路的可能性在不断增加，当达到一定数量级时，就可认为此链路存在环路。

但在实际应用中根据不同设备实现PPP协议的方法，我们在链路环路检测时可采用两种方法。第一种机制就是如上面所述的，这个过程不断地重复，最终可能会给LCP状态机发一个Down事件，这时可能会使LCP的状态机又回到初始化阶段，又开始新一轮的协商。当然对于某些设备还会采用第二种机制，就是不产生任何事件去影响当前LCP的状态机，而是停留在请求发送状态。但这时认为链路有环路的一端设备需要不断的向链路上发送Echo-Request报文来检测链路环路是否被解除，当接收端收到Echo-Reply报文时，就认为链路环路被解除，从而就可能进行后续的PPP的过程。

4.1.2 认证协议

PPP协议也提供了可选的认证配置参数选项，缺省情况下点对点通信的两端是不进行认证的。在LCP的Config-Request报文中不可一次携带多种认证配置选项，必须二者择其一（PAP/CHAP），选择最希望的那一种，一般是在PPP设备互连的设备上进行配置的，但一般设备会默认支持一个缺省的认证方式（PAP是大部分设备所默认的认证方式）。当对端收到该配置请求报文后，如果支持配置参数选项中的认证方式，则回应一个Config-Ack报文；否则回应一个Config-Nak报文，并附上自希望双方采用的认证方式。当对方接收到Config-Ack报文后就可以开始进行认证了，而如果收到的是Config-Nak报文，则根据自身是否支持Config-Nak报文中的认证方式来回应对方，如果支持则回应一个新的Config-Request（并携带上Config-Nak报文中所希望使用的认证协议），否则将回应一个Config-Reject报文，那么双方就无法通过认证，从而不可能建立起PPP链路。

PPP支持两种授权协议：PAP（Password Authentication Protocol）和CHAP（Challenge Hand Authentication Protocol）。

4.1.2.1 PAP认证

我们所知两个设备在使用PAP进行认证之前，应该确认那一方是验证方，那一方是被验证方。实际上对于使用PPP协议互连的两端来说，既可作为认证方，也可作为被认证方。但通常情况下，PAP只使用一个方向上的认证。一般在两端设备使用PAP协议之前，均会设备上进行一些相应的配置，对于宽带工程师而言MA5200可谓是大家最熟悉的产品了，它默认就作为验证方，但可通过使用命令PAP Authentication PAP/CHAP来更改认证方式，而对于被验证方而言只需设置用户名和密码即可。

PAP认证是两次握手，在链路建立阶段，依据设备上的配置情况，如果是使用PAP认证，则验证方在发送Config-Request报文时会携带认证配置参数选项，而对于被验证方而言则是不需要，它只需要收到该配置请求报文后根据自身的情况给对端返回相应的报文。如果点对点的两端设备采用的是PAP双向认证时，也即是它同时也作为验证方，则此时需要在配置请求报文中携带认证配置参数选项。因此，我们可以总结一下，如果对于点对点的两个设备在PPP链路建立的过程中使用的认证方式为PAP的话，那么验证方在其Config-Request报文中必须含有认证配置参数选项，且该认证配置参数选项的数据域为0xC023，下图为PAP认证的过程：

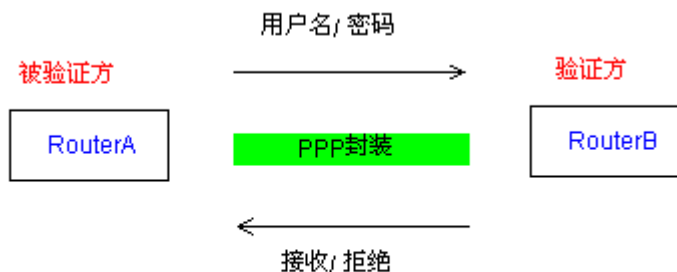


图4-1 PAP认证过程

当通信设备的两端在收到对方返回的Config-Ack报文时，就从各自的链路建立阶段进入到认证阶段，那么作为被验证方此时需要向验证方发送PAP认证的请求报文，该请求报文携带了用户名和密码，当验证方收到该认证请求报文后，则会根据报文中的实际内容查找本地的数据库，如果该数据库中有与用户名和密码一致的选项时，则回向对方返回一个认证请求响应，告诉对方认证已通过。反之，如果用户名与密码不符，则向对方返回验证不通过的响应报文。如果双方都配置为验证方，则需要双方的两个单向验证过程都完成后，方可进入到网络层协议阶段，否则在一定次的认证失败后，则会从当前状态返回链路不可用状态。

例10: 如图4-1所示, 当路由器A (被验证方) 收到了路由器B 的Config-Ack报文后, 因为使用PAP认证, 所以作为被验证方的路由器A应主动向验证方 (路由器B) 发送认证请求报文 (PAP Authenticate), 用户名和密码均为163, 报文的内容如下:

7E FF 03 C0 23 01 01 00 0C 03 31 36 33 03 31 36 33 7E

下划线的前四个字节是用户名, 后四个字节是密码。

当路由器B收到了该报文后, 会向路由器A回应一个PAP Authenticate Ack报文, 报文内容如下:

7E FF 03 80 21 02 01 00 05 00 7E

此时所回应的报文中, 并未携带任何数据, 如果是认证不通过, 则会在返回的报文中指是因何原因无法认证通过, 可能是无此用户名或密码不匹配。

4.1.2.2 CHAP认证

与PAP认证比起来, CHAP认证更具有安全性, 从前面认证过程的数据包交换过程中不难发现, 采用PAP认证时, 被验证是采用明文的方式直接将用户名和密码发送给验证方的, 而对于PAP认证则不一样。

CHAP为三次握手协议, 它只在网络上传送用户名而不传送口令, 因此安全性比PAP高。在验证一开始, 不像PAP一样是由被验证方发送认证请求报文了, 而是由验证方向被验证方发送一段随机的报文, 并加上自己的主机名, 我们通称这个过程叫做挑战。当被验证方收到验证方的验证请求, 从中提取出验证方所发送过来的主机名, 然后根据该主机名在被验证方设备的后台数据库中去查找相同的用户名的记录, 当查找到后就使用该用户名所对应的密钥, 然后根据这个密钥、报文ID和验证方发送的随机报文用Md5加密算法生成应答, 随后将应答和自己的主机名送回, 同样验证方收到被验证方发送回应后, 提取被验证方的用户名, 然后去查找本地的数据库, 当找到与被验证方一致用户名后, 根据该用户名所对应的密钥、保留报文ID和随机报文用Md5加密算法生成结果, 和刚刚被验证方所返回的应答进行比较, 相同则返回Ack, 否则返回Nak, 下图为CHAP的认证过程:

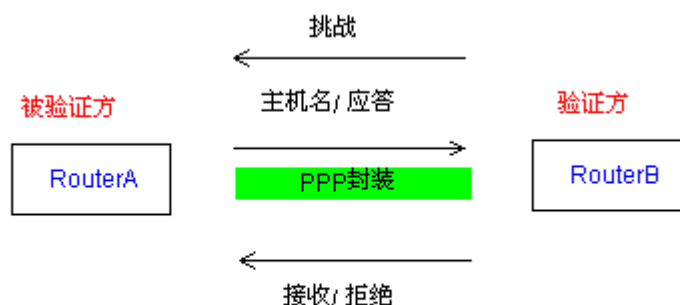


图4-2 CHAP认证过程

例11：如图4-2所示，当路由器A（被验证方）收到了路由器B 的Challenge报文后，报文内容如下：

```
7E FF 03 C2 23 01 01 00 1C 10 FF 41 CF 22 AA 8E F1 B9 99 9A 79 A7 56 78 C4 A7
4d
```

```
41 35 32 30 30 41 7E
```

下划线的前16个字节是验证方随机产生的一段报文，后7个字节是验证方的主机名（MA5200A），而且单个字节10表示随机报文的长度。

而此时路由器A会根据用户名所对应的密钥使用报文的ID和该报文的内容生成一个回应报文，报文内容如下：

```
7E FF 03 C2 23 02 01 00 1F 10 18 86 22 FF CE 81 D0 68 FF 80 85 00 A7 E3 85 35 70
70
```

```
6B 69 73 73 40 68 75 61 7E
```

我们将这个回应报文与验证方发送的挑战报文进行比较，报文的代码域已由原01改为02，总报文的长度有变化，主要后而一个下划线的内容是被验证方的主机名（ppkiss@hua），而且此时回应的16个字节的报文已经是经过MD5算法加密过的。

当验证方收到了这个回应报文后，会根据报文中被验证方的主机名（ppkiss@hua）在本地数据库中去找密钥，然后再对原先发送的那段挑战报文进行MD5的算法加密，如果所得的结果与对方刚发过来的16个字节的加密值一样的话，则就会发送一个报文通知被验证方，你的认证已通过，我们可以进入到下一个阶段了。在实际应用当中，我们很多都是使用PC机来进行拨号这个过程，实际中当验证方发送挑战后，PC机只接收而并不去查本地数据库，而直接使用在拨号对话框中所输入的密码和报文的ID及报报文的内容进行MD5算法加密（这个在PC机采用PPPOE软件拨入到MA5200时就是这样的）。

下面来看一下验证通过时，验证方给被验证方所发送的一段报文内容：

```
7E FF 03 C2 23 03 01 00 17 57 65 6C 63 6F 6D 65 20 74 6F 20 4D 41 35 32 30 30 41
2E
```

```
7E
```

此时所回应的报文的代码域为03，且报文的实际内容是，Welcom to MA5200A。

4.1.3 MRU（Maxium Receive Unit）

这个配置参数选项主要是Config-Request报文的发送端告诉接收端，本端接收到的PPP数据帧的数据域的最大值。通常情况下这个参数选项使用默认值（1500字节），因此在Config-Request报文中双方都不会携带这个配置参数

选项。当在某些特殊应用中，可能会使用到小于1500字节或大于1500字节的情况，这时在Config-Request报文就会携带要协商的MRU配置参数选项值。

4.2 总结

- 魔术字可以在链路配置阶段被协商，数据报文可借助魔术字来检测PPP链路是否存在环路
- PAP（密码认证协议）认证是二次握手，它是直接在网络上传送明文的用户名和密码，因此这种协议安全性不高
- CHAP（挑战性握手认证协议）认证是三次握手，它只在网络上传送验证方和被验证方的主机名，而并不传送密码，因此相比之处CHAP比PAP更安全
- PPP协议缺省的MRU是1500，而对于通信的双方可根据实际需要MRU进行协商

4.3 思考

- 1、PPP协议可通过几种方式来检测PPP链路是否存在环路？
- 2、请比较PAP认证和CHAP认证？

第五章 PPP扩展协议

5.1 PPP扩展协议概述

5.1.1 MP出现的背景

我们知道ISDN可以在两个系统之间提供2B+D和30B+D多通道捆绑能力，从而为用户能够提供更多可用的带宽。诸如上述的许多链路捆绑功能需要软件和硬件的协同工作，而且更多的基于硬件来实现的。然而我们是否考虑过仅仅通过软件的实现来完成链路捆绑的功能，同时还考虑到很多实际链路的情况，对于软件在实现过程中还要能对不同速率的链路进行捆绑。我们可以通过在发送数据之前增加一定数据的字节头，其中含有为重组数据而所需的一些字段。

随着PPP的广泛应用，MP作为PPP功能扩展协议应运而生。它可为用户提供更大的带宽，实现数据的快速转发。同时，还可实现对链路资源进行动态分配，有效的利用宝贵的资源。但随着网络技术的发展，网络的带宽已不再是瓶颈，所以对于使用PPP扩展协议已没有实际意义，只在本章中简单做一下介绍，如果想进一步了解该协议，可参考相应的RFC1717或提供的参考书目。

5.1.2 MP（Multilink Protocol）协议

MP的协商较为特殊。MP配置参数选项的协商是在LCP协商过程中完成的，协商MP配置参数选项的目的完成以下几个过程：

- 表明系统是否支持将多个物理链路捆绑成一个逻辑链路
- 系统在多链路上接收到了对端发送的数据单元后，能够通过附加在这些数据之前的重组字段对这些分段的数据单元进行重组
- 逻辑链路为了能够提高传输的效率，可以不使用单一PPP物理链路上的最大接收单元，可以重新协商新的逻辑链路上使用的最大接收单元进行数据报文的发送和接收。

MP协议可以用来灵活的调整点对点系统之间的多条独立物理链路，它可为整个系统提供一个虚拟链路，虚拟链路的带宽是N个链路的捆绑之和（ $N \geq 1$ ）。而对于被捆绑的链路并未做出特殊要求，可以将同步链路和异步链路进行捆绑，同样也可将低速链路和高速链路进行捆绑。使用该协商可将多个PPP的链路捆绑成一条使用，而决定不同通道是否需进行多链路捆绑有两个条件：只有两个链路的Discriminator和验证方式、用户完全相符时，才能对两个链路进行捆绑。这就意味着只有当验证完成后，才能真正完成MP的协

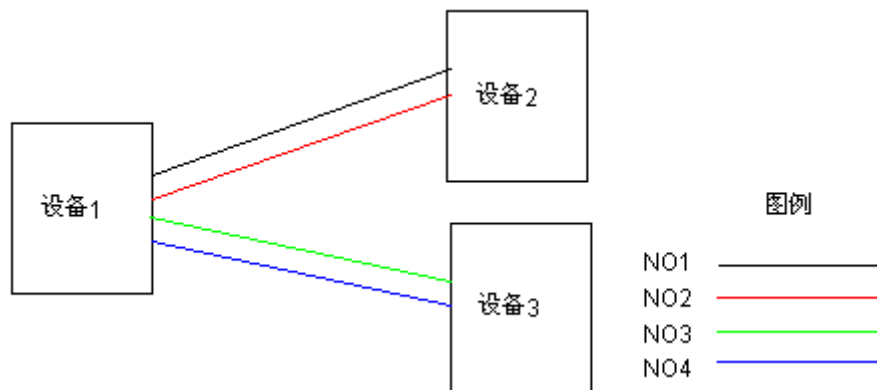
商过程。MP不会导致链路的拆除。如果配置了MP，两个链路不符合MP条件，则会建立一条新的MP通道，这同时也表明允许MP为单链路。MP的捆绑是完全依照用户进行的，只有相同用户才能进行捆绑。如一端配置了MP，另一端不支持或未配MP，则建立起来的链路为非MP链路。

5.2 总结

- MP协议属于PPP协议的扩展协议
- MP协议可依据终端指示符和验证方式对不同的物理链路进行捆绑

5.3 思考

1、如下图所示，设备1与设备2通过两条物理链路互连（使用链路1和链路2），而设备1与设备3也通过两条物理链路互连（使用链路3和链路4），我们知道



设备链路的捆绑是依据终端指示符和验证方式的组合进行的，试想一下使用这两项的4种组合设备1在进行链路捆绑中会出现哪些情况。

第六章 PPP的状态机

6.1 PPP扩展协议概述

由于PPP的状态机，对于我们实际使用当中没有太大的意义，如果想进一步深入了解PPP协议的话，请参见PPP的RFC文档。