

[百度首页](#) | [登录](#)[新闻](#) [网页](#) [贴吧](#) [知道](#) [MP3](#) [图片](#) [百科](#)

PPPOE协议

[帮助](#)[进入词条](#)[搜索词条](#)[百度百科](#) > [浏览词条](#)[编辑词条](#) [发表评论](#) [历史版本](#) [打印](#) [添加到收藏](#)

PPPOE协议

1. 前言

PPPOE(PPP over Ethernet, RFC2516, 值得注意的是此RFC不是Standard而是Information类型的)定义了如何在以太网上传输PPP数据包的方法, 目前流行的宽带类型ADSL就是通过PPPoE实现的。

2. 通信过程概述

建立PPPOE通道(ADSL拨号)分两个阶段:[发现阶段](#)和[PPP会话阶段](#)。

在发现阶段, 以太网上的客户机要[找到一个访问集中器\(AC, Access Concentrator\)](#), 就是ADSL MODEM, 一般家用时一般就只有一个AC; 但如果是一个以太网内可能会有多条ADSL, 就会有多个AC, 这时客户机就从中选择一个。[发现阶段完成后](#), 客户机和AC都得到要在以太网上建立PPP通道的相关信息。

发现阶段是无状态的, 也就是两边都不用保存以前的状态信息; 只有PPP会话开始后, 双方就要建立一个虚拟的PPP通信接口, 具体在Linux下会有ppp0网卡, 在windows下网络连接中增加ADSL的接口。

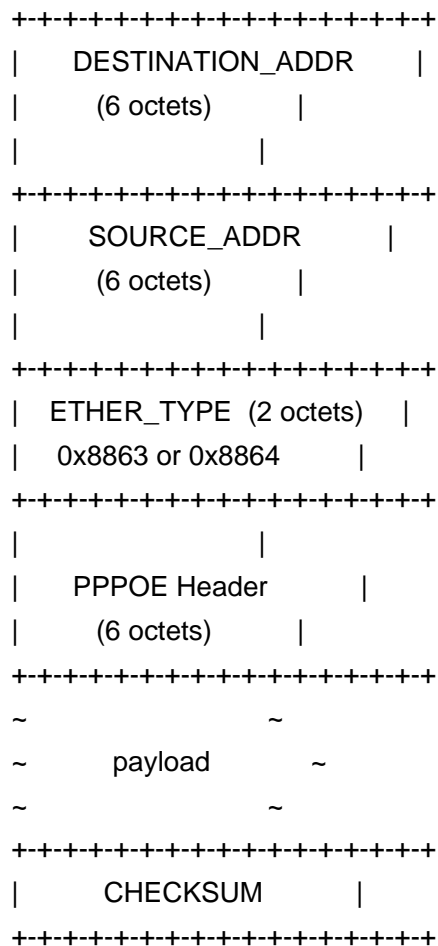
3. 协议头格式

3.1 协议值

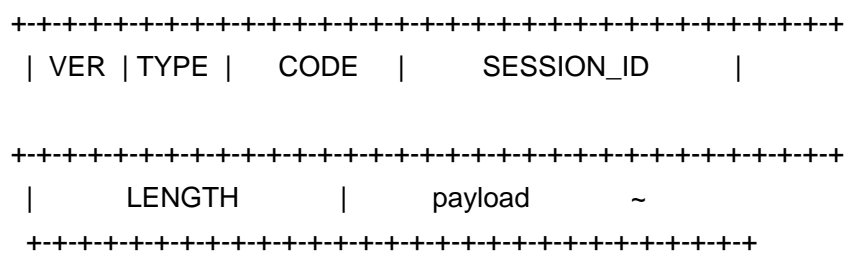
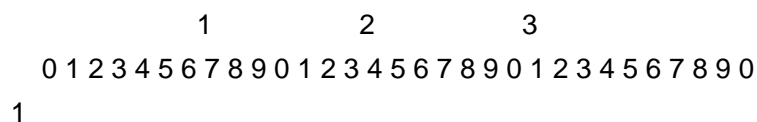
PPPOE数据是直接放在以太头数据之上的, 其等级和ARP、IP等是相同的, 在以太头的类型字段中, [用0x8863表示是PPPOE发现阶段数据](#), [用0x8864表示PPP会话阶段数据](#), 如下所示。(类比: 0x0800表示IP数据, 0x0806表示ARP数据)

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5

©2007 Baidu



3.2 PPPOE协议头



PPPOE协议头有6个字节，正好和14字节的以太头实现了4字节对齐，包括以下字段：

VER: 版本号，4位，必须为0x01
TYPE: 类型，4位，必须是0x01
CODE: 8位，在发现阶段和PPP会话阶段有不同的定义，表示PPPOE数据类型
SESSION_ID: 16位，用来定义一个PPP会话，在发现过程中定义。
LENGTH: 16位，表示负载长度，不包括以太头和PPPOE头。

4. 发现阶段

4.1 TAG

Diagram illustrating the structure of a 32-bit tag:

- Bit positions 0 to 31 are shown above the tag structure.
- The tag is divided into three main fields:
 - TAG_TYPE**: Bits 1 to 7.
 - TAG_LENGTH**: Bits 8 to 14.
 - TAG_VALUE**: Bits 15 to 31, followed by an ellipsis indicating further bits.

0x0000	End-Of-List
0x0101	Service-Name
0x0102	AC-Name
0x0103	Host-Uniq
0x0104	AC-Cookie
0x0105	Vendor-Specific
0x0110	Relay-Session-Id
0x0201	Service-Name-Error
0x0202	AC-System-Error
0x0203	Generic-Error

4.2 PPPOE主动发现初始包

一个PADI包的例子为:

```

          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|               0xffffffff               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|    0xffff    |   Host_mac_addr    |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Host_mac_addr (cont)   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  ETHER_TYPE = 0x8863    | v = 1 | t = 1 | CODE =
0x09 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  SESSION_ID = 0x0000    |   LENGTH = 0x0004
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|  TAG_TYPE = 0x0101      |   TAG_LENGTH = 0x0000
|
+-----+-----+-----+-----+-----+-----+-----+-----+

```

4.3 PPPOE主动发现提议包

PPPOE主动发现提议包(PPPoE Active Discovery Offer, PADO)由AC发出，用来回应客户机的PADI包，以太头中的目的地址是客户机的MAC地址，PPPOE头中的CODE为0x07，SESSION_ID值必须为0，负载部分必须包含一个AC-Name类型的TAG，用来指示本AC的名称，一个在PADI包中指定的Service-Name的TAG，另外可以包含其他Service-Name的TAG。如果AC不对该客户机提供服务，AC就不回应PADO包。一个PADO包的例子为：

```

          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Host_mac_addr               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Host_mac_addr (cont)   |
Access_Concentrator_mac_addr |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Access_Concentrator_mac_addr (cont)   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  ETHER_TYPE = 0x8863    | v = 1 | t = 1 | CODE =
0x07 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  SESSION_ID = 0x0000    |   LENGTH = 0x0020
|
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

| TAG_TYPE = 0x0101 | TAG_LENGTH = 0x0000
|
+-----+
| TAG_TYPE = 0x0102 | TAG_LENGTH = 0x0018
|
+-----+
| 0x47 | 0x6f | 0x20 | 0x52 |
+-----+
| 0x65 | 0x64 | 0x42 | 0x61 |
+-----+
| 0x63 | 0x6b | 0x20 | 0x2d |
+-----+
| 0x20 | 0x65 | 0x73 | 0x68 |
+-----+
| 0x73 | 0x68 | 0x65 | 0x73 |
+-----+
| 0x68 | 0x6f | 0x6f | 0x74 |
+-----+

```

4.4 PPPOE主动发现请求包

PPPOE主动发现请求包(PPPoE Active Discovery Request, PADR)由客户机发出，因为可能会有多个AC对客户机发出的PADI包回应了PADO包，客户机从回应的PADO包中选择一个AC发送PADR包，以太头中的目的地址是该AC的MAC地址，PPPOE头中的CODE为0x19，SESSION_ID值必须为0，负载部分必须只包含一个Service-Name类型的TAG表示请求的服务类型，另外可以包含其他TAG。

4.5 PPPOE主动发现会话确认包

PPPOE主动发现会话确认包(PPPoE Active Discovery Session-confirmation, PADS)由AC发出，收到客户机的PADR包后，AC将产生一个SESSION_ID值用来标志本次PPP会话，以PADR包方式发送给客户机。以太头中的目的地址是客户机的MAC地址，PPPOE头中的CODE为0x65，SESSION_ID值必须为所生成的那个SESSION_ID，负载部分必须只包含一个Service-Name类型的TAG，表示该服务类型被AC接受，另外可以包含其他TAG。如果AC不接受PADR中的Server-Name，PADS中则包含一个Service-Name-Error类型的TAG,这时SESSION_ID设置为0。

4.6 PPPOE主动发现停止包

PPPOE主动发现停止包(PPPoE Active Discovery Terminate,

PADT)表示PPPOE会话过程的结束，AC和客户机都可以主动发出。以太头中的目的地址是对方的MAC地址，PPPOE头中的CODE为0xa7，SESSION_ID值必须为PPPOE会话过程的SESSION_ID，不需要TAG。

5. PPP会话阶段

在PPP会话阶段，PPP包被封装在PPPOE以太帧中，以太包目的地址都是单一的，以太协议为0x8864，PPPOE头的CODE必须为0，SESSION_ID必须一直为发现阶段协商出的SESSION_ID值，PPPOE的负载是整个PPP包，PPP包前是两字节的PPP协议ID值。

一个PPPOE会话过程包的例子为：

```

          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Access_Concentrator_mac_addr          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|Access_Concentrator_mac_addr(c)|   Host_mac_addr
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Host_mac_addr (cont)          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ETHER_TYPE = 0x8864    | v = 1 | t = 1 | CODE =
0x00 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| SESSION_ID = 0x1234    |   LENGTH = 0x????
|
+-----+-----+-----+-----+-----+-----+-----+-----+
| PPP PROTOCOL = 0xc021   |   PPP payload
~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

注意：由于PPPOE头是6字节，PPP协议ID号两个字节，一共要占用8个字节，而以太网的MTU值为1500，所以上层PPP负载数据不能超过1492字节，所以PPP协商时协商的最大接收单元值不能超过1492字节，也就是相当于在PPPOE环境下的MTU是1492字节。

6. ADSL拨号过程简述

客户机启动拨号程序，发送PADI包，ADSL MODEM回应PADO包，客户机再发送PADR包，ADSL MODEM回应PADS包后建立PPPOE通道，随后客户机进行普通的PPP协议拨号过程，不过PPP数据包都是包装进以太帧中的，拨号成功后客

户机和服务器之间建立了PPP通道，ADSL MODEM起到将以太帧转换为PPP包的作用。ADSL虽然是用电话线，但所用频率不是通话用的频率，所以ADSL拨号不影响打电话。通信结束后，会发送PADT断开PPPOE通道。

7. 结论

由于ADSL的大面积使用，PPPOE也随之应用，了解其通信协议和数据格式对于底层驱动的开发、协议分析、访问控制等是必要的，现在不论在windows还是linux下都可以很好地支持PPPOE。

 [编辑词条](#)

开放分类：

[网络](#)、[技术](#)、[协议](#)、[ADSL](#)

贡献者：

[踏月色之回魂](#)、[xxyzzxh](#)

关于本词条的评论（共0条）：

对本词条发表评论：

最多不超过1000字

发表评论

[返回页首](#)