附件一、Sniffer 使用指导

目 录

第1章 Sniffer 软件简介

1.1 概述

Sniffer 软件是 NAI 公司推出的功能强大的协议分析软件。本文针对用 Sniffer Pro 网络分析器进行故障解决。利用 Sniffer Pro 网络分析器的强大功能和特征，解决网络问题，将介绍一套合理的故障解决方法。

与 Netxray 比较，Sniffer 支持的协议更丰富，例如 PPPOE 协议等在 Netxray 并不支持，在 Sniffer 上能够进行快速解码分析。Netxray 不能在 Windows 2000 和 Windows XP 上正常运行，Sniffer Pro 4.6 可以运行在各种 Windows 平台上。

Sniffer 软件比较大，运行时需要的计算机内存比较大，否则运行比较慢，这也是它与 Netxray 相比的一个缺点。

1.2 功能简介

下面列出了 Sniffer 软件的一些功能介绍，其功能的详细介绍可以参考 Sniffer 的在线帮助。

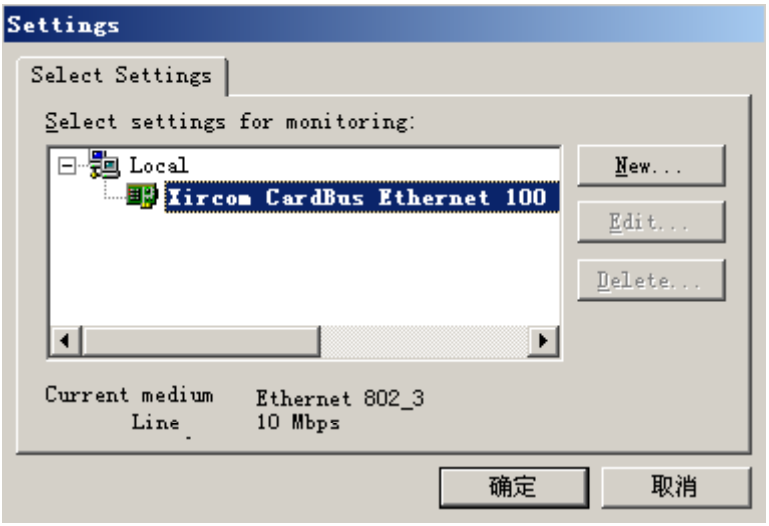
捕获网络流量进行详细分析

利用专家分析系统诊断问题

实时监控网络活动

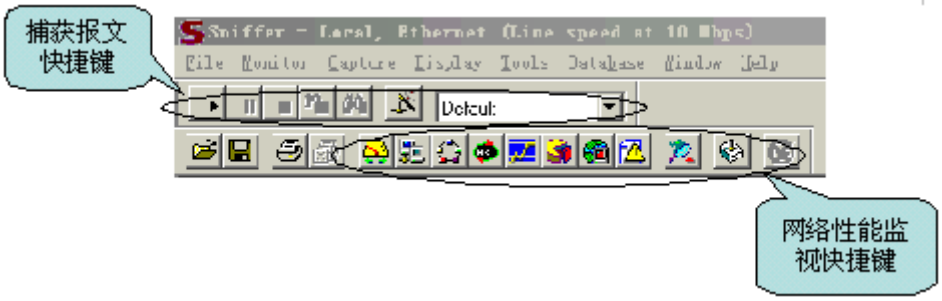
收集网络利用率和错误等

在进行流量捕获之前首先选择网络适配器，确定从计算机的哪个网络适配器上接收数据。位置：File->select settings



选择网络适配器后才能正常工作。该软件安装在 Windows 98 操作系统上，Sniffer 可以选择拨号适配器对窄带拨号进行操作。如果安装了 EnterNet500 等 PPPOE 软件还可以选择虚拟出的 PPPOE 网卡。对于安装在 Windows 2000/XP 上则无上述功能，这和操作系统有关。

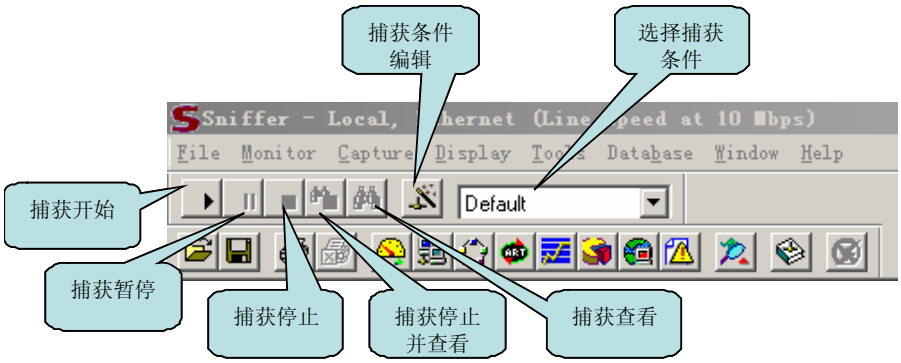
本文将对报文的捕获几网络性能监视等功能进行详细的介绍。下图为在软件中快捷键的位置。



第2章 报文捕获解析

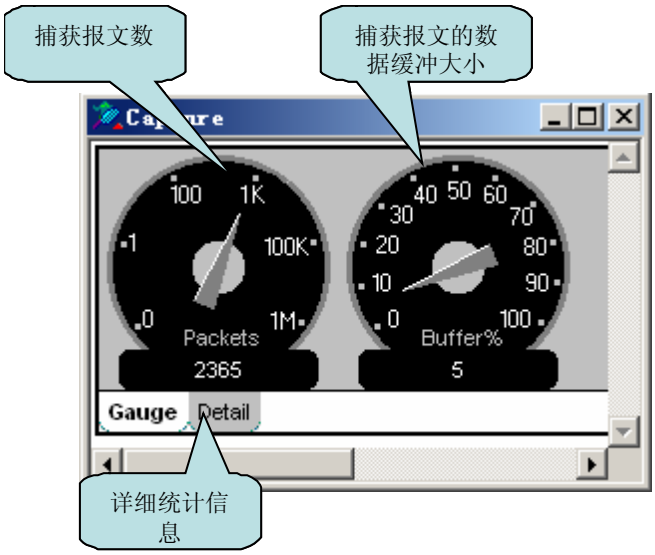
2.1 捕获面板

报文捕获功能可以在报文捕获面板中进行完成，如下是捕获面板的功能图：图中显示的是处于开始状态的面板



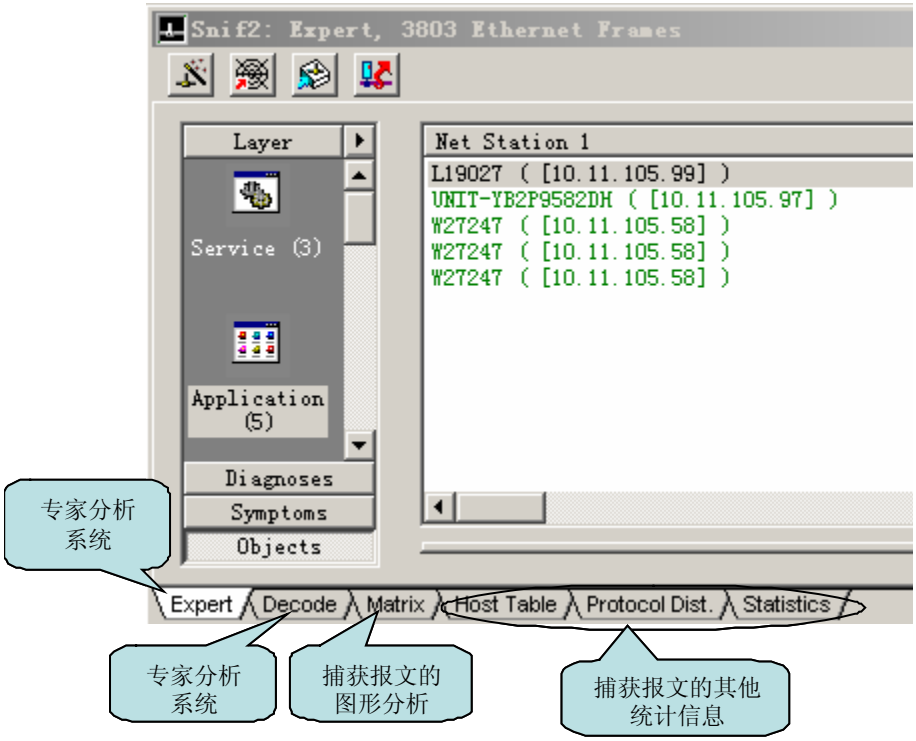
2.2 捕获过程报文统计

在捕获过程中可以通过查看下面面板查看捕获报文的数量和缓冲区的利用率。



2.3 捕获报文查看

Sniffer 软件提供了强大的分析能力和解码功能。如下图所示，对于捕获的报文提供了一个 Expert 专家分析系统进行分析，还有解码选项及图形和表格的统计信息。

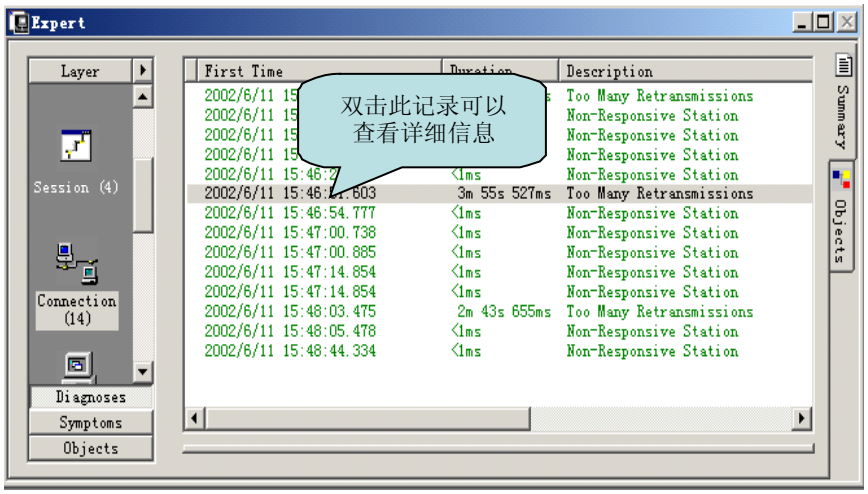


专家分析

专家分分析系统提供了一个只能的分析平台，对网络上的流量进行了一些分析对于分析出的诊断结果可以查看在线帮助获得。

在下图中显示出在网络中 WINS 查询失败的次数及 TCP 重传的次数统计等内容，可以方便了解网络中高层协议出现故障的可能点。

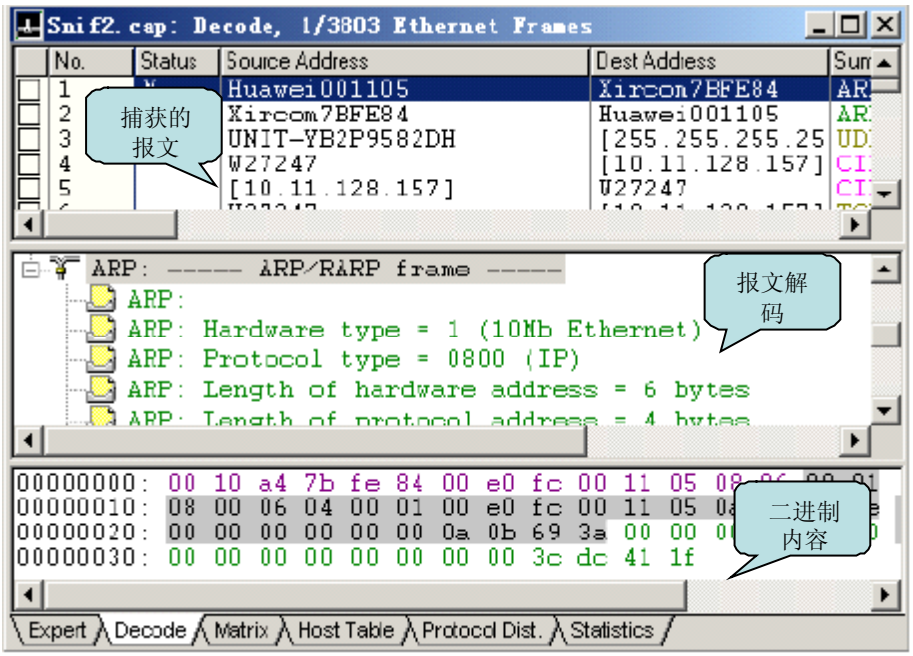
对于某项统计分析可以通过用鼠标双击此条记录可以查看详细统计信息且对于每一项都可以通过查看帮助来了解起产生的原因。



解码分析

下图是对捕获报文进行解码的显示，通常分为三部分，目前大部分此类软件结构都采用这种结构显示。对于解码主要要求分析人员对协议比较熟悉，这样才能看懂解析出来的报文。使用该软件是很简单的事情，要能够利用软件解码分析来解决问题关键是要对各种层次的协议了解的比较透彻。工具软件只是提供一个辅助的手段。因涉及的内容太多，这里不对协议进行过多讲解，请参阅其他相关资料。

对于 MAC 地址，Snffier 软件进行了头部的替换，如 00e0fc 开头的就替换成 Huawei，这样有利于了解网络上各种相关设备的制造厂商信息。



功能是按照过滤器设置的过滤规则进行数据的捕获或显示。在菜单上的位置分别为 Capture->Define Filter 和 Display->Define Filter。

过滤器可以根据物理地址或 IP 地址和协议选择进行组合筛选。

统计分析

对于 Matrix，Host Table，Portocol Dist. Statistics 等提供了丰富的按照地址，协议等内容做了丰富的组合统计，比较简单，可以通过操作很快掌握这里就不再详细介绍了。

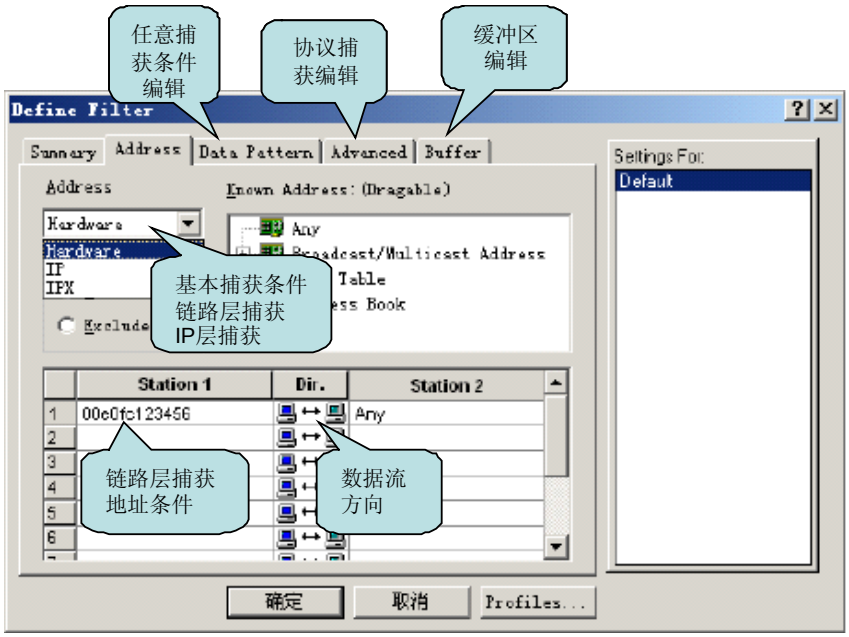
2.4 设置捕获条件

基本捕获条件

基本的捕获条件有两种：

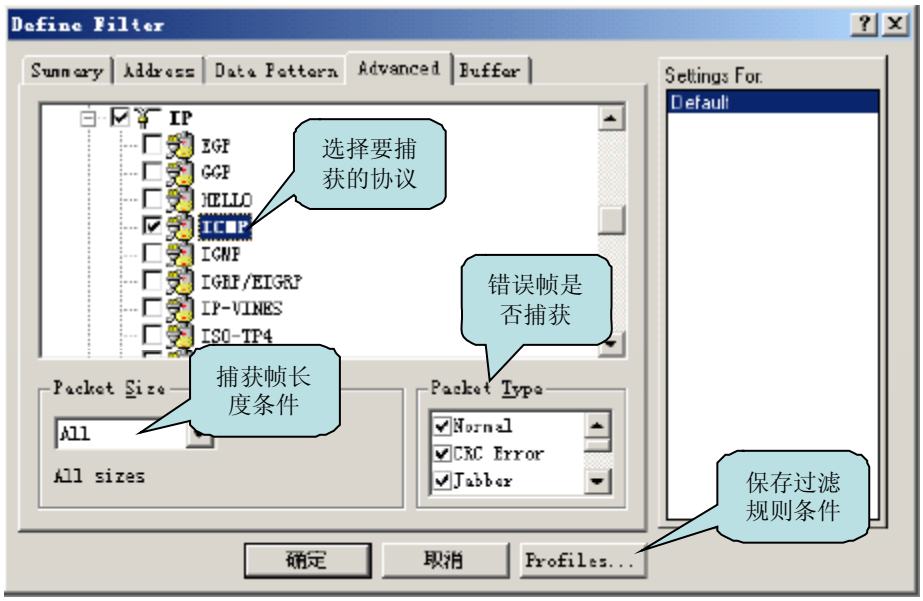
1、链路层捕获，按源 MAC 和目的 MAC 地址进行捕获，输入方式为十六进制连续输入，如：00E0FC123456。

2、IP 层捕获，按源 IP 和目的 IP 进行捕获。输入方式为点间隔方式，如：10.107.1.1。如果选择 IP 层捕获条件则 ARP 等报文将被过滤掉。



高级捕获条件

在“Advance”页面下，你可以编辑你的协议捕获条件，如图：



高级捕获条件编辑图

在协议选择树中你可以选择你需要捕获的协议条件，如果什么都不选，则表示忽略该条件，捕获所有协议。

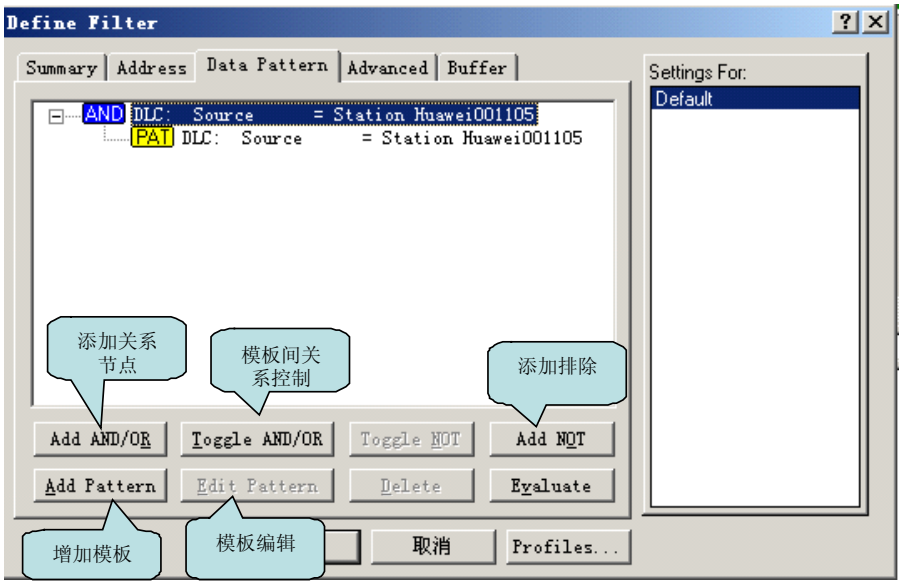
在捕获帧长度条件下，你可以捕获，等于、小于、大于某个值的报文。

在错误帧是否捕获栏，你可以选择当网络上有如下错误时是否捕获。

在保存过滤规则条件按钮“Profiles”，你可以将你当前设置的过滤规则，进行保存，在捕获主面板中，你可以选择你保存的捕获条件。

任意捕获条件

在 Data Pattern 下，你可以编辑任意捕获条件，如下图：

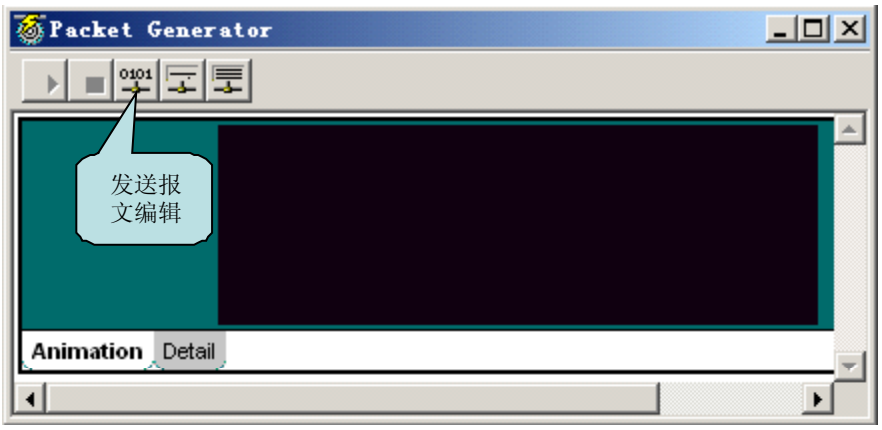


用这种方法可以实现复杂的报文过滤，但很多时候是得不偿失，有时截获的报文本就不多，还不如自己看看来得快。

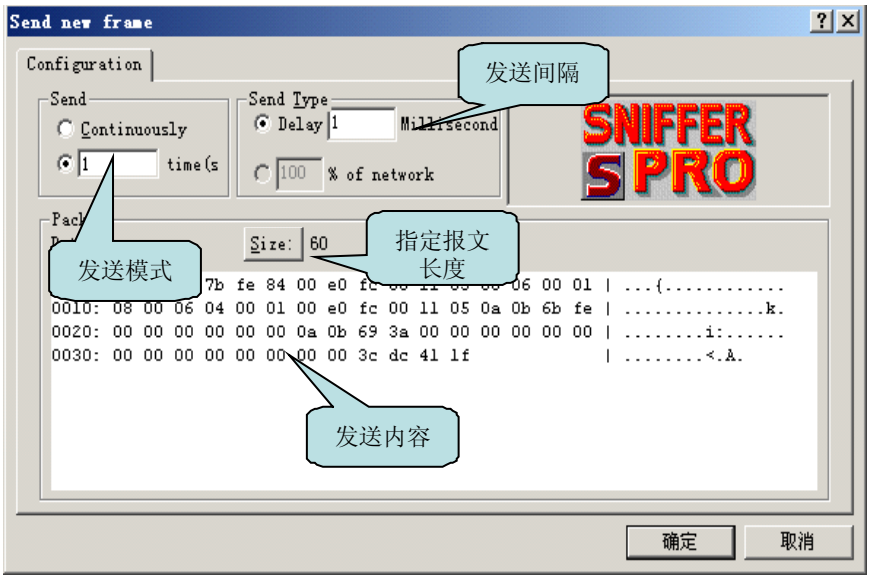
第3章 报文放送

3.1 编辑报文发送

Sniffer 软件报文发送功能就比较弱，如下是发送的主面板图：



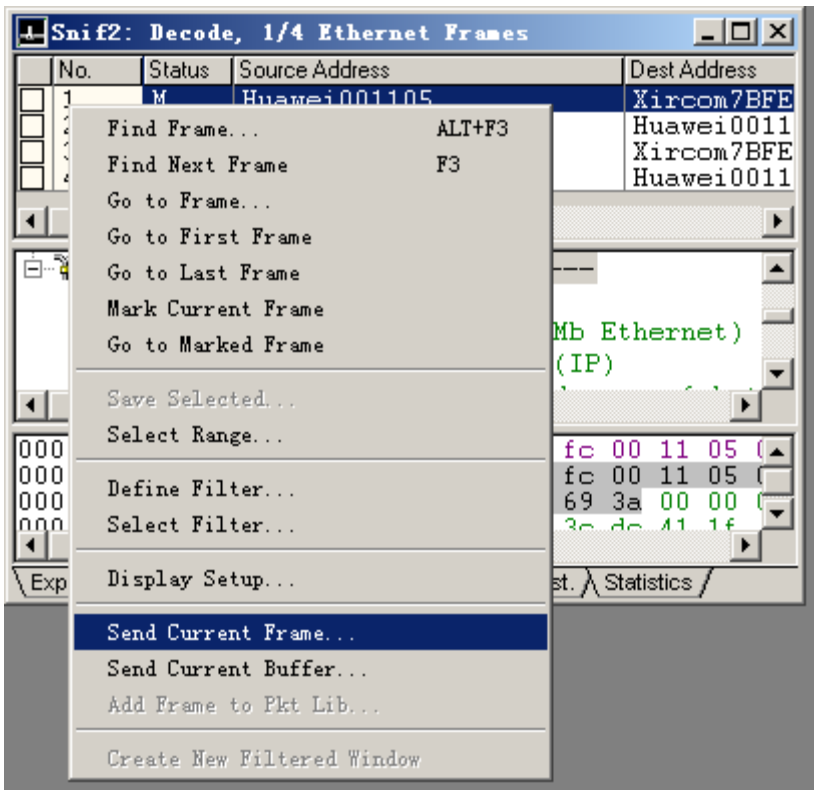
发送前，你需要先编辑报文发送的内容。点击发送报文编辑按钮。可得到如下的报文编辑窗口：



首先要指定数据帧发送的长度，然后从链路层开始，一个一个将报文填充完成，如果是 NetXray 支持可以解析的协议，从“Decode”页面中，可看见解析后的直观表示。

3.2 捕获编辑报文发送

将捕获到的报文直接转换成发送报文，然后修修改改可也。如下是一个捕获报文后的报文查看窗口：



选中某个捕获的报文，用鼠标右键激活菜单，选择“Send Current Packet”，这时你就会发现，该报文的内容已经被原封不动的送到“发送编辑窗口”中了。这时，你在修修改改，就比你全部填充报文省事多了。

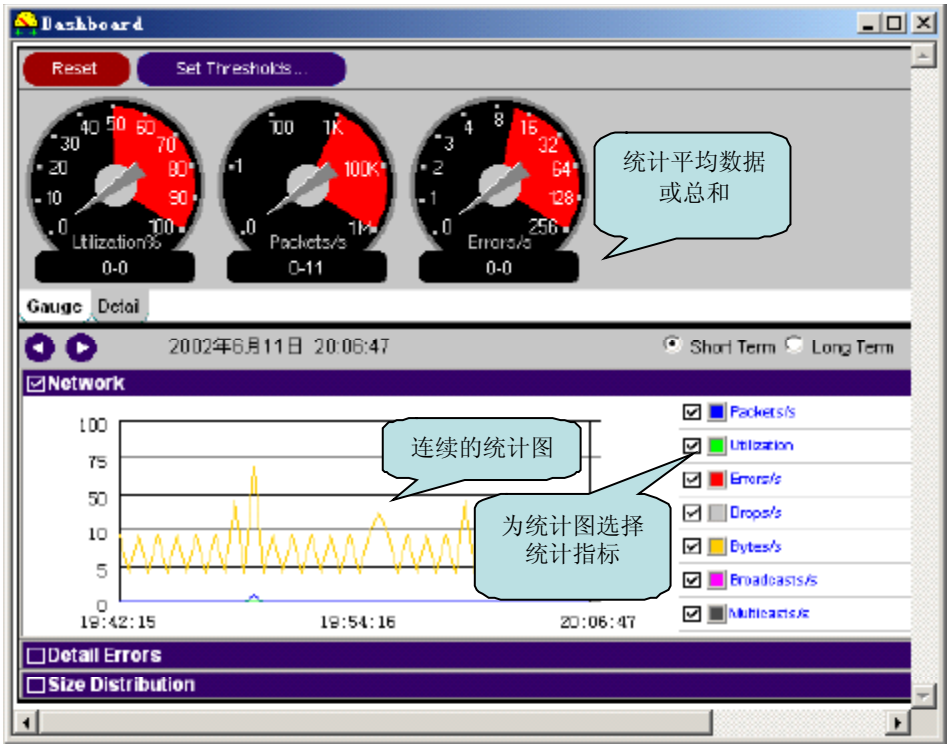
发送模式有两种：连续发送和定量发送。可以设置发送间隔，如果为 0，则以最快的速度进行发送。

第4章 网络监视功能

网络监视功能能够时刻监视网络统计，网络上资源的利用率，并能够监视网络流量的异常状况，这里只介绍一下 Dashbord 和 ART，其他功能可以参看在线帮助，或直接使用即可，比较简单。

4.1 Dashbord

Dashbord 可以监控网络的利用率，流量及错误报文等内容。通过应用软件可以清楚看到此功能。



4.2 Application Response Time (ART)

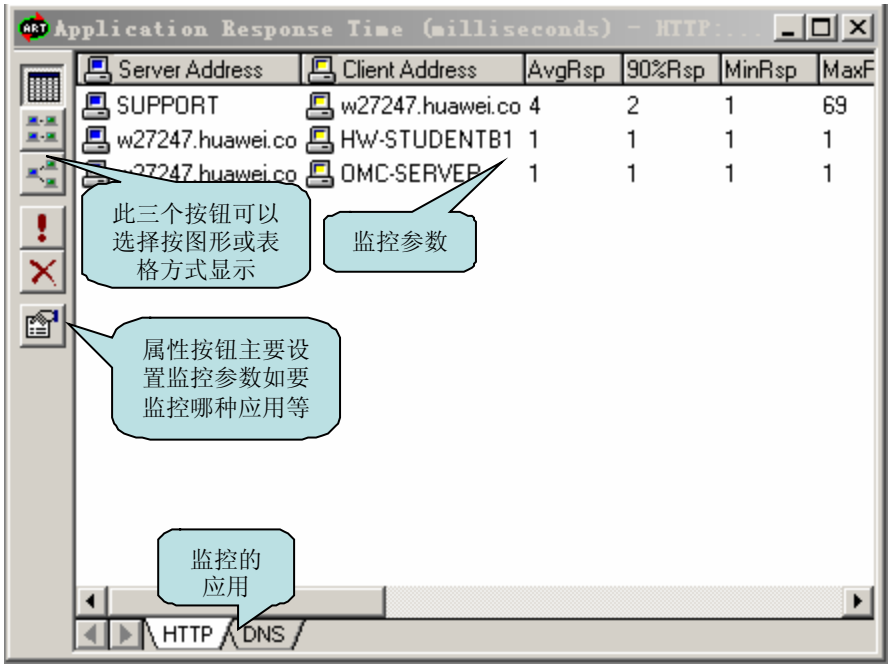
Application Response Time (ART) 是可以监视 TCP/UDP 应用层程序在客户端和服务器响应时间，如 HTTP,FTP,DNS 等应用。

对与 TCP/UDP 响应时间的计算方法如下

TCP For each socket, ART stores the sequence numbers for packets sent by the client and waits for the corresponding ACK packets from the server.

It then measures the time difference between the packet with the stored sequence number and the packet with the ACK to arrive at the response time.

UDP For each socket, ART measures the time between packets going from a client to a server and the next packet going from the server to the client.

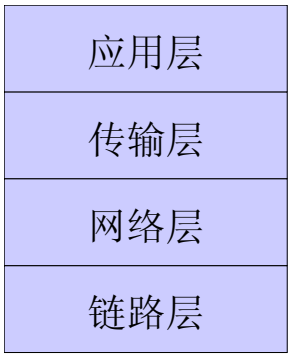


第5章 数据报文解码详解

本章主要对：数据报文分层、以太报文结构、IP 协议、ARP 协议、PPPOE 协议、Radius 协议等的解码分析做了简单的描述，目的在于介绍 Sniffer 软件在协议分析中的功能作用并通过解码分析对协议进一步了解。对其其他协议读者可以通过协议文档和 Sniffer 捕获的报文对比分析。

5.1 数据报文分层

如下图所示，对于四层网络结构，其不同层次完成不通功能。每一层次有众多协议组成。

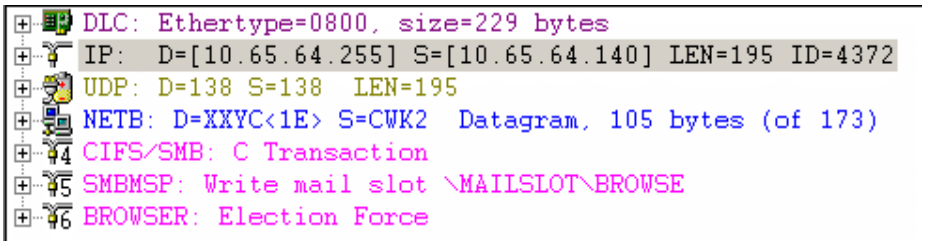


Telnet FTP和e-mail等

TCP 和 UDP

IP ICMP IGMP

设备驱动程序及接口卡



如上图所示在 Sniffer 的解码表中分别对每一个层次协议进行解码分析。链路层对应“DLC”；网络层对应“IP”；传输层对应“UDP”；应用层对对应的是“NETB”等高层协议。Sniffer 可以针对众多协议进行详细结构化解码分析。并利用树形结构良好的表现出来。

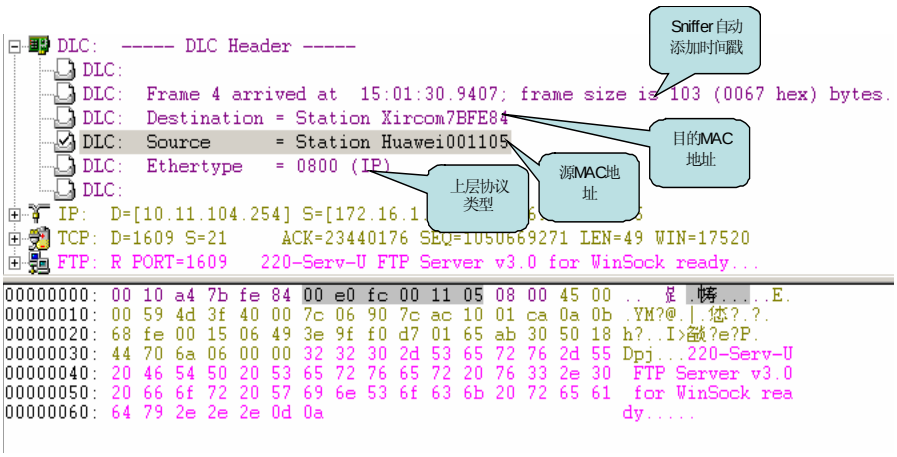
5.2 以太报文结构

EthernetII 以太网帧结构

Ethernet_II

DMAC	SMAC	Type	DATA/PAD	FCS

Ethernet_II 以太网帧类型报文结构为：目的 MAC 地址（6bytes）＋源 MAC 地址＋（6bytes）上层协议类型（2bytes）＋数据字段（46-1500bytes)＋校验（4bytes）。

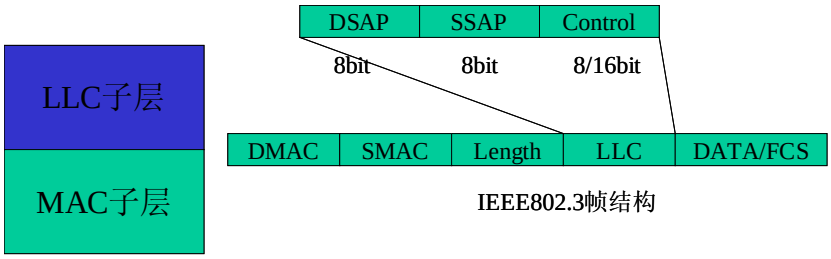


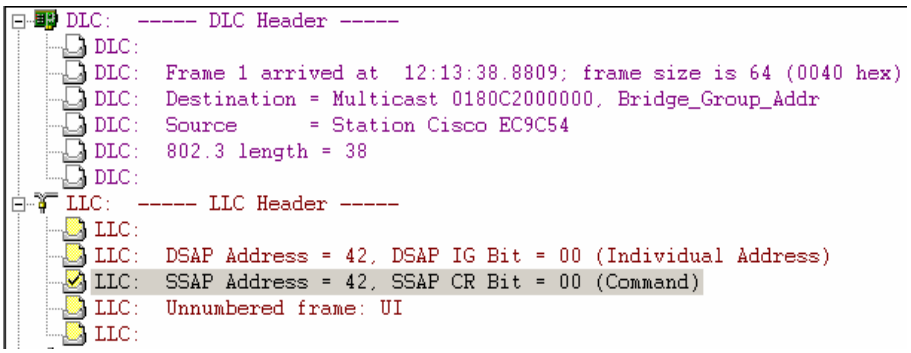
Sniffer 会在捕获报文的时候自动记录捕获的时间，在解码显示时显示出来，在分析问题时提供了很好的时间记录。

源目的 MAC 地址在解码框中可以将前 3 字节代表厂商的字段翻译出来，方便定位问题，例如网络上 2 台设备 IP 地址设置冲突，可以通过解码翻译出厂商信息方便的将故障设备找到，如 00e0fc 为华为，010042 为 Cisco 等等。如果需要查看详细的 MAC 地址用鼠标在解码框中点击此 MAC 地址，在下面的表格中会突出显示该地址的 16 进制编码。

IP 网络来说 Ethertype 字段承载的时上层协议的类型主要包括 0x800 为 IP 协议，0x806 为 ARP 协议。

IEEE802.3 以太网报文结构





上图为 IEEE802.3SNAP 帧结构，与 EthernetII 不通点是目的和源地址后面的字段代表的不是上层协议类型而是报文长度。并多了 LLC 子层。

5.3 IP 协议

IP 报文结构为 IP 协议头＋载荷，其中对 IP 协议头部的分析，时分析 IP 报文的主要内容之一，关于 IP 报文详细信息请参考相关资料。这里给出了 IP 协议头部的一个结构。

版本：4——IPv4

首部长度：单位为 4 字节，最大 60 字节

TOS：IP 优先级字段

总长度：单位字节，最大 65535 字节

标识：IP 报文标识字段

标志：占 3 比特，只用到低位的两个比特

MF（More Fragment）

MF=1，后面还有分片的数据包

MF=0，分片数据包的最后一个

DF（Don't Fragment）

DF=1，不允许分片

DF=0，允许分片

段偏移：分片后的分组在原分组中的相对位置，总共 13 比特，单位为 8 字节

寿命：TTL（Time To Live）丢弃 TTL=0 的报文

协议：携带的是何种协议报文

1 ：ICMP

6 ：TCP

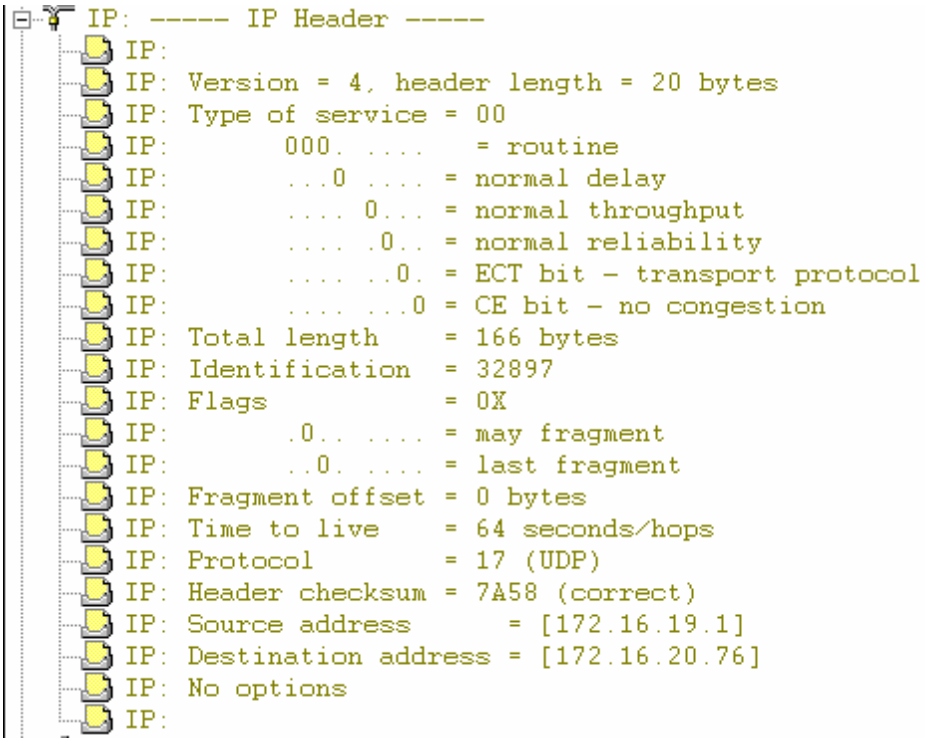
17：UDP

89：OSPF

头部检验和：对 IP 协议首部的校验和

源 IP 地址：IP 报文的源地址

目的 IP 地址：IP 报文的目的地址



上图为 Sniffer 对 IP 协议首部的解码分析结构，和 IP 首部各个字段相对应，并给出了各个字段值所表示含义的英文解释。如上图报文协议（Protocol）字段的编码为 0x11，通过 Sniffer 解码分析转换为十进制的 17，代表 UDP

协议。其他字段的解码含义可以与此类似，只要对协议理解的比较清楚对解码内容的理解将会变的很容易。

5.4 ARP 协议

以下为 ARP 报文结构

ARP 分组具有如下的一些字段：

HTYPE（**硬件类型**）。这是一个 16 比特字段，用来定义运行 ARP 的网络的类型。每一个局域网基于其类型被指派给一个整数。例如，以太网是类型 1。ARP 可使用在任何网络上。

PTYPE（**协议类型**）。这是一个 16 比特字段，用来定义协议的类型。例如，对 IPv4 协议，这个字段的值是 0800。ARP 可用于任何高层协议。

HLEN（**硬件长度**）。这是一个 8 比特字段，用来定义以字节为单位的物理地址的长度。例如，对以太网这个值是 6。

PLEN（**协议长度**）。这是一个 8 比特字段，用来定义以字节为单位的逻辑地址的长度。例如，对 IPv4 协议这个值是 4。

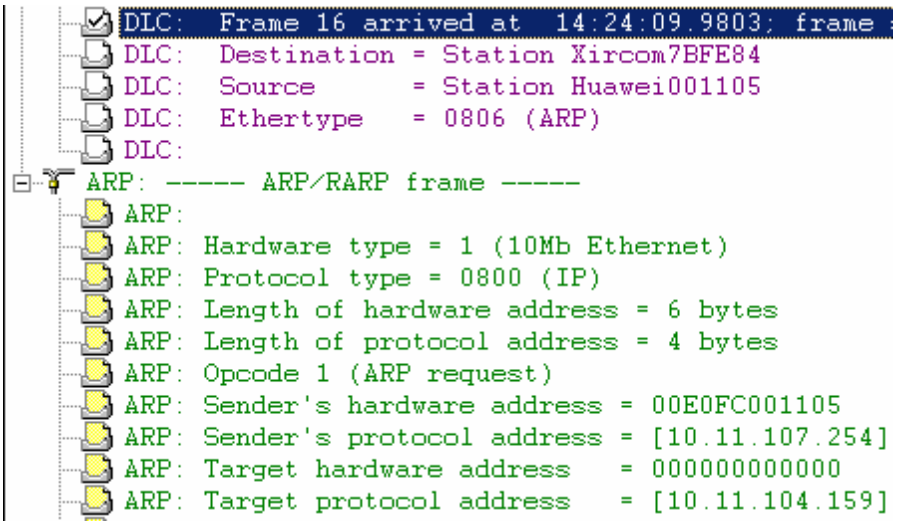
OPER（**操作**）。这是一个 16 比特字段，用来定义分组的类型。已定义了两种类型：ARP 请求（1），ARP 回答（2）。

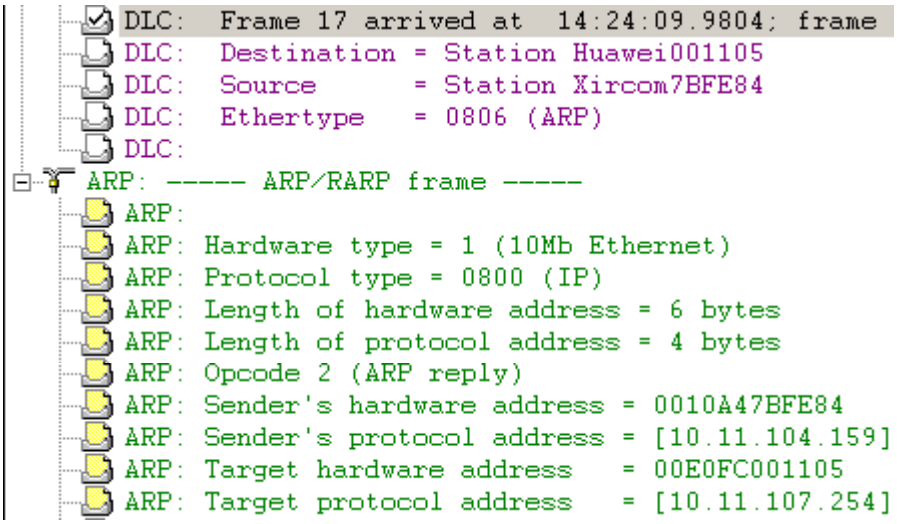
SHA（**发送站硬件地址**）。这是一个可变长度字段，用来定义发送站的物理地址的长度。例如，对以太网这个字段是 6 字节长。

SPA（**发送站协议地址**）。这是一个可变长度字段，用来定义发送站的逻辑（例如，IP）地址的长度。对于 IP 协议，这个字段是 4 字节长。

THA（**目标硬件地址**）。这是一个可变长度字段，用来定义目标的物理地址的长度。例如，对以太网这个字段是 6 字节长。对于 ARP 请求报文，这个字段是全 0，因为发送站不知道目标的物理地址。

TPA（**目标协议地址**）。这是一个可变长度字段，用来定义目标的逻辑地址（例如，IP 地址）的长度。对于 IPv4 协议，这个字段是 4 字节长。



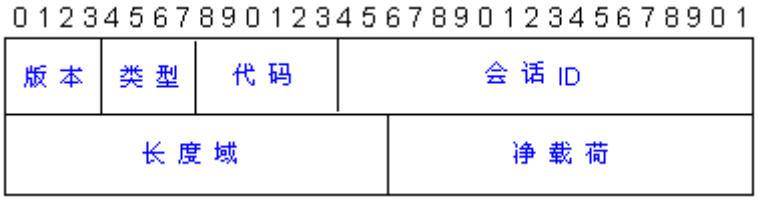


上面为通过 Sniffer 解码的 ARP 请求和应答报文的结构。

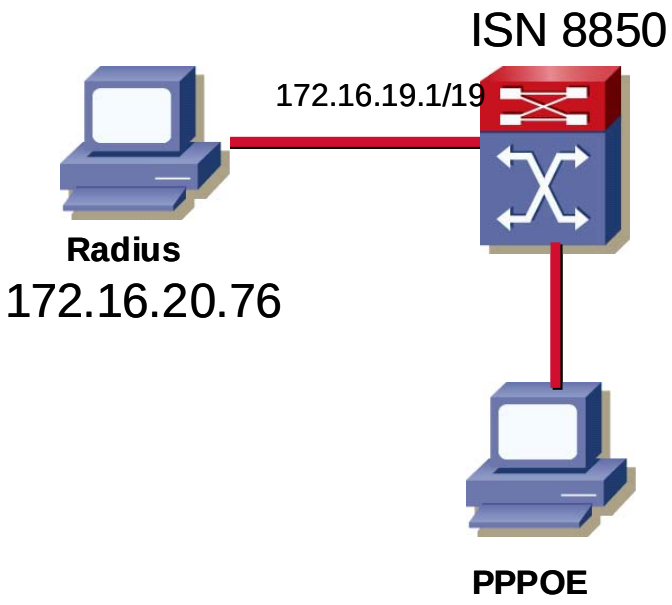
5.5 PPPOE 协议

PPPOE 简介

简单来说我们可能把 PPPOE 报文分成两大块，一大块是 PPPOE 的数据报头，另一块则是 PPPOE 的净载荷（数据域），对于 PPPOE 报文数据域中的内容会随着会话过程的进行而不断改变。下图为 PPPOE 的报文的格式：

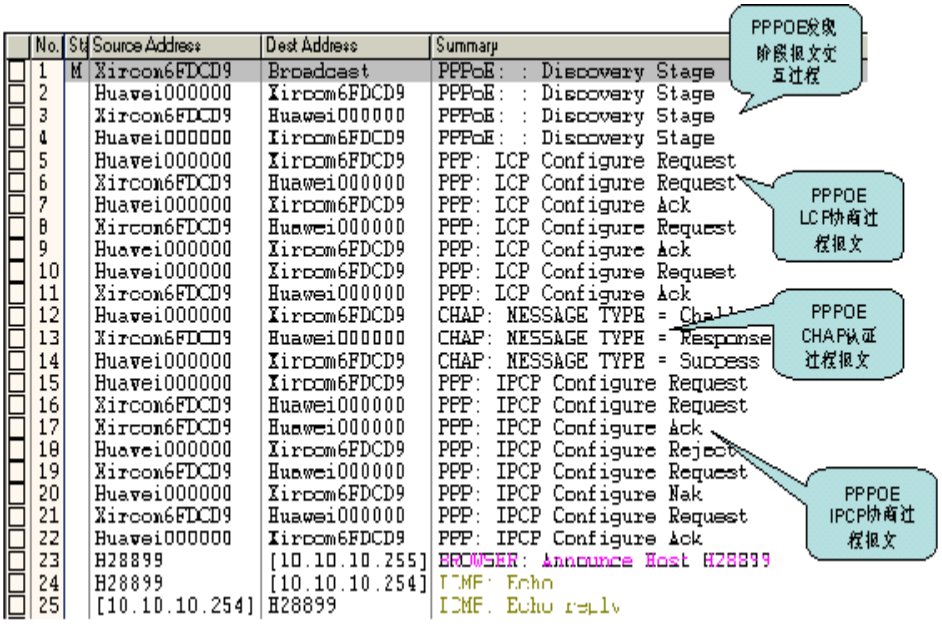


- 数据报文最开始的 4 位为版本域，协议中给出了明确的规定，这个域的内容填充 0x01。
- 紧接在版本域后的 4 位是类型域，协议中同样规定，这个域的内容填充为 0x01。
- 代码域占用 1 个字节，对于 PPPOE 的不同阶段这个域内的内容也是不一样的。
- 会话 ID 点用 2 个字节，当访问集中器还未分配唯一的会话 ID 给用户主机的话，则该域内的内容必须填充为 0x0000，一旦主机获取了会话 ID 后，那么在后续的所有报文中该域必须填充那个唯一的会话 ID 值。
- 长度域为 2 个字节，用来指示 PPPOE 数据报文中净载荷的长度。
- 数据域，有时也称之为净载荷域，在 PPPOE 的不同阶段该域内的数据内容会有很大的不同。在 PPPOE 的发现阶段时，该域内会填充一些 Tag（标记）；而在 PPPOE 的会话阶段，该域则携带的是 PPP 的报文。



捕获报文测试用例图

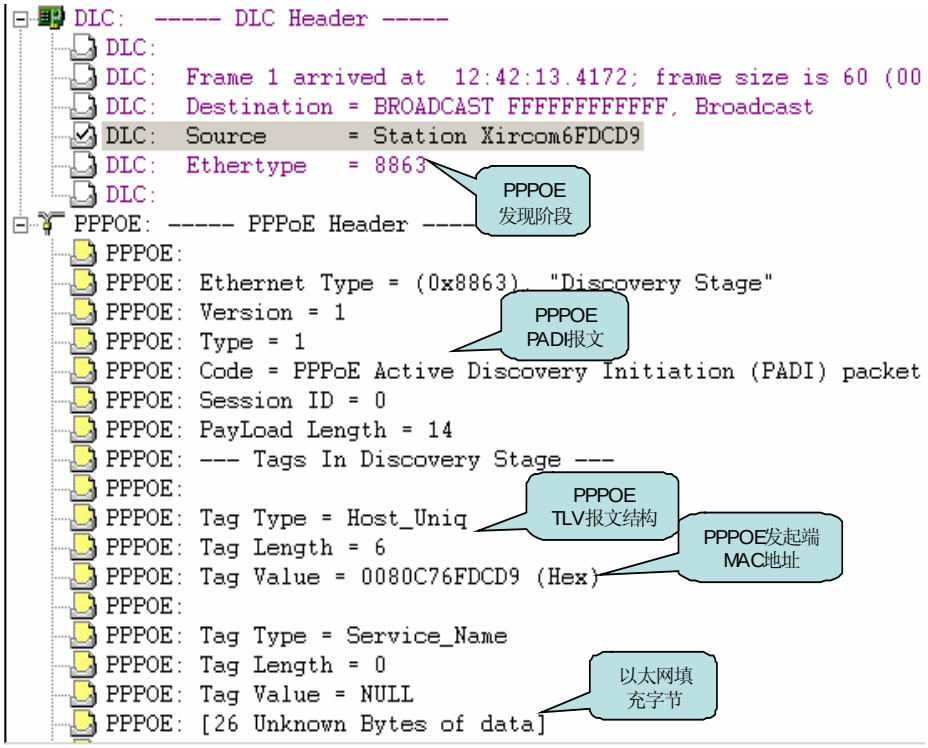
如图所示，Radius Server IP 地址为 172.16.20.76。PPPOE 用户 Radius 报文交互过程分析如下。



No.	St	Source Address	Dest Address	Summary
1	M	Xircom6FDCD9	Broadcast	PPPoE: : Discovery Stage
2		Huawei000000	Xircom6FDCD9	PPPoE: : Discovery Stage
3		Xircom6FDCD9	Huawei000000	PPPoE: : Discovery Stage
4		Huawei000000	Xircom6FDCD9	PPPoE: : Discovery Stage
5		Huawei000000	Xircom6FDCD9	PPP: LCP Configure Request
6		Xircom6FDCD9	Huawei000000	PPP: LCP Configure Request
7		Huawei000000	Xircom6FDCD9	PPP: LCP Configure Ack
8		Xircom6FDCD9	Huawei000000	PPP: LCP Configure Request
9		Huawei000000	Xircom6FDCD9	PPP: LCP Configure Ack
10		Huawei000000	Xircom6FDCD9	PPP: LCP Configure Request
11		Xircom6FDCD9	Huawei000000	PPP: LCP Configure Ack
12		Huawei000000	Xircom6FDCD9	CHAP: MESSAGE TYPE = Chal
13		Xircom6FDCD9	Huawei000000	CHAP: MESSAGE TYPE = Response
14		Huawei000000	Xircom6FDCD9	CHAP: MESSAGE TYPE = Success
15		Huawei000000	Xircom6FDCD9	PPP: IPCP Configure Request
16		Xircom6FDCD9	Huawei000000	PPP: IPCP Configure Request
17		Xircom6FDCD9	Huawei000000	PPP: IPCP Configure Ack
18		Huawei000000	Xircom6FDCD9	PPP: IPCP Configure Reject
19		Xircom6FDCD9	Huawei000000	PPP: IPCP Configure Request
20		Huawei000000	Xircom6FDCD9	PPP: IPCP Configure Nak
21		Xircom6FDCD9	Huawei000000	PPP: IPCP Configure Request
22		Huawei000000	Xircom6FDCD9	PPP: IPCP Configure Ack
23		H28899	[10.10.10.255]	BROWSER: Announce Host H28899
24		H28899	[10.10.10.254]	ICMP: Echo
25		[10.10.10.254]	H28899	ICMP: Echo reply

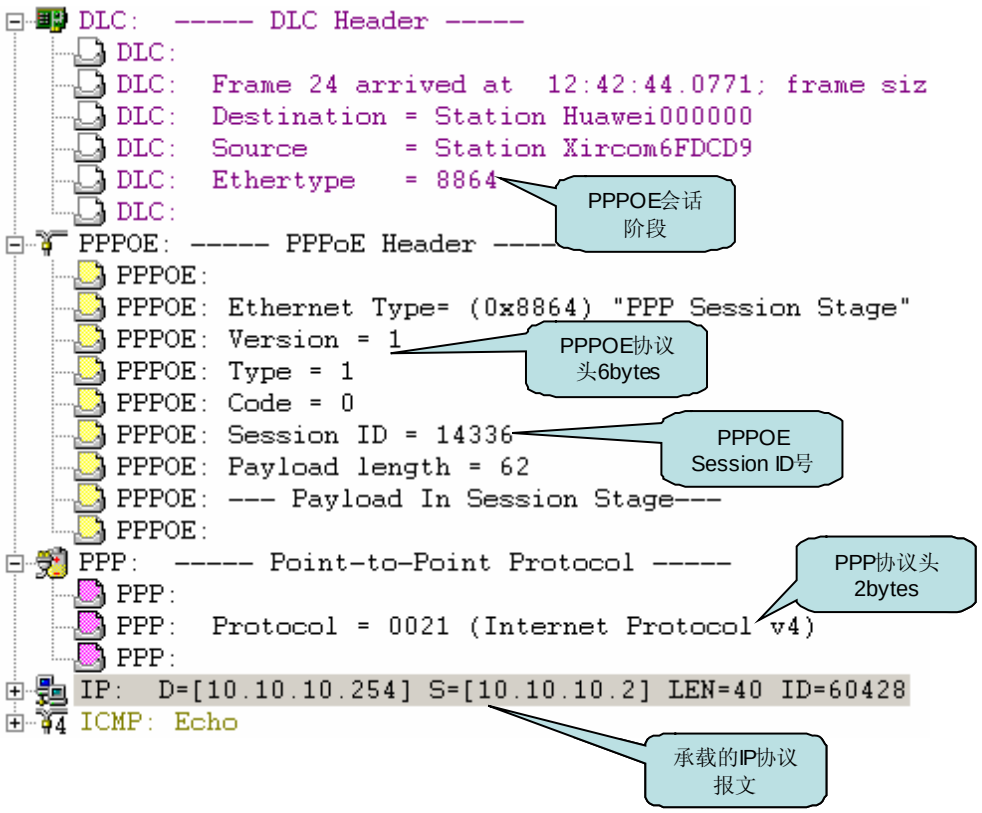
上图为 PPPOE 从发现阶段到 PPP LCP 协商，认证 IPCP 协商阶段和 PPPOE 会话阶段交互过程。

PPPOE 发现阶段，PADI 报文，Sniffer 解码结构如下所示。




```
DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 1 arrived at  12:42:13.4172; frame size is 60 (00
DLC:  Destination = BROADCAST FFFFFFFFFFFF, Broadcast
DLC:  Source      = Station Xircom6FDCD9
DLC:  Ethertype   = 8863
DLC:
PPPOE: ----- PPPoE Header -----
PPPOE:
PPPOE: Ethernet Type = (0x8863), "Discovery Stage"
PPPOE: Version = 1
PPPOE: Type = 1
PPPOE: Code = PPPoE Active Discovery Initiation (PADI) packet
PPPOE: Session ID = 0
PPPOE: PayLoad Length = 14
PPPOE: --- Tags In Discovery Stage ---
PPPOE:
PPPOE: Tag Type = Host_Uniq
PPPOE: Tag Length = 6
PPPOE: Tag Value = 0080C76FDCD9 (Hex)
PPPOE:
PPPOE: Tag Type = Service_Name
PPPOE: Tag Length = 0
PPPOE: Tag Value = NULL
PPPOE: [26 Unknown Bytes of data]
```

PPPOE 会话阶段，Sniffer 解码结构如下所示。



5.6 Radius 协议

Radius 报文简介

标准 Radius 协议包结构

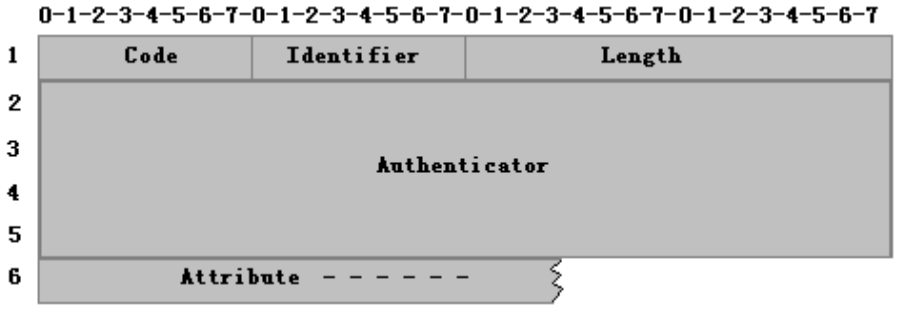


图 9 Radius 包格式

Code：包类型；1 字节；指示 RADIUS 包的类型。

1	Access- request	认证请求
2	Access- accept	认证响应
3	Access- reject	认证拒绝

4 Accounting-request 计费请求

5 Accounting-response 计费响应

*11 Access-challenge 认证挑战

Identifier： 包标识；1 字节，取值范围为 0 ～255；用于匹配请求包和响应包，同一组请求包和响应包的 Identifier 应相同。

Length： 包长度；2 字节；整个包中所有域的长度。

Authenticator：16 字节长；用于验证 RADIUS 服务器传回来的请求以及密码隐藏算法上。

该验证字分为两种：

1、请求验证字---Request Authenticator

用在请求报文中,必须为全局唯一的随机值。

2、响应验证字---Response Authenticator

用在响应报文中，用于鉴别响应报文的合法性。

响应验证字＝MD5(Code+ID+Length+请求验证字+Attributes+Key)

Attributes：属性

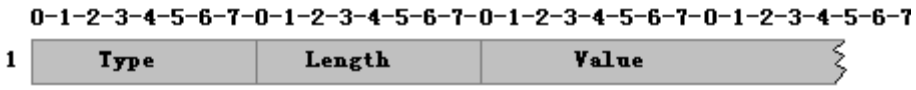
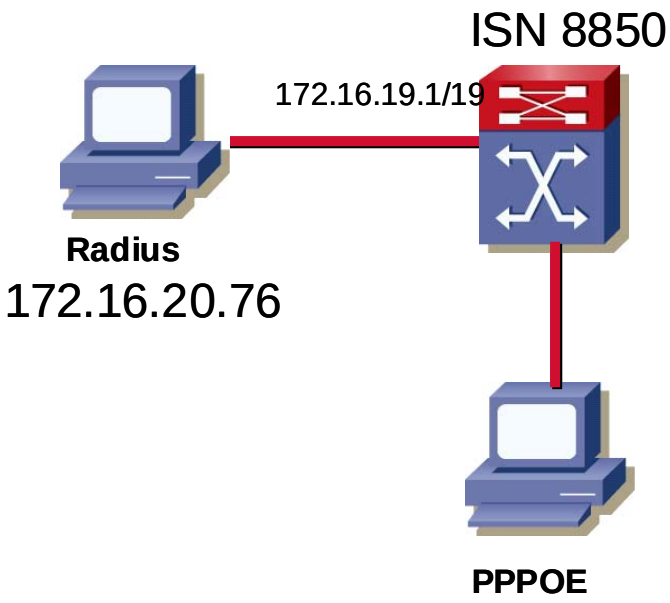


图 10 属性格式

属性域是 TLV 结构编码。



测试用例图

下图为用户端 PPPOE，Radius Server 和 BAS 交互的认证上线和下线的过程。

No.	St	Source Address	Dest Address	Summary
1	M	[172.16.19.1]	[172.16.20.76]	RADIUS: Access-Request Id = 25
2		[172.16.20.76]	[172.16.19.1]	RADIUS: Access-Accept Id = 25
3		[172.16.19.1]	[172.16.20.76]	RADIUS: Accounting-Request Id = 26
4		[172.16.20.76]	[172.16.19.1]	RADIUS: Accounting-Response Id = 26
5		[172.16.19.1]	[172.16.20.76]	RADIUS: Accounting-Request Id = 27
6		[172.16.20.76]	[172.16.19.1]	RADIUS: Accounting-Response Id = 27

报文 1：BAS 请求 Radius Server 认证报文。

报文 2：Radius Server 回应 BAS 认证通过报文。

报文 3：BAS 计费请求报文。

报文 4：Radius Server 计费响应报文。

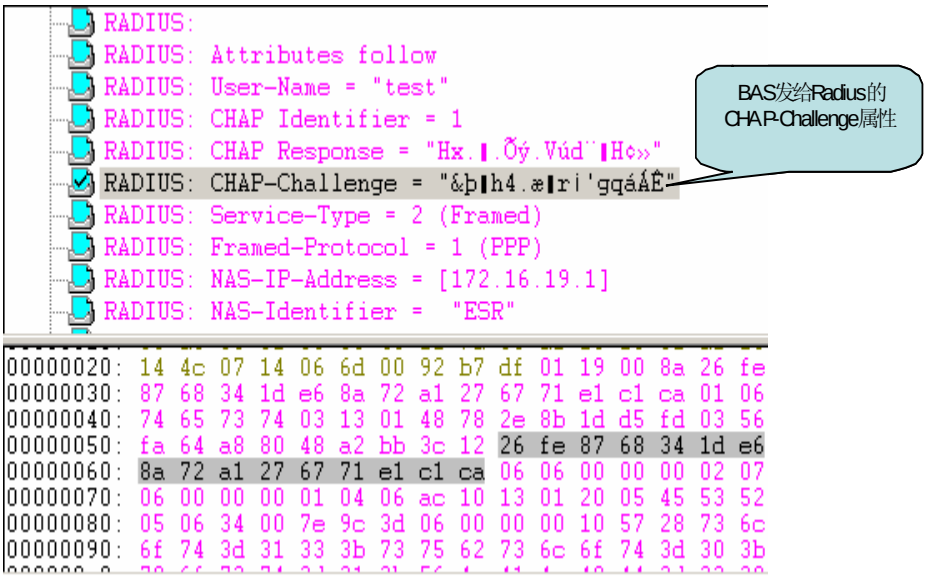
报文 5：BAS 计费结束报文。

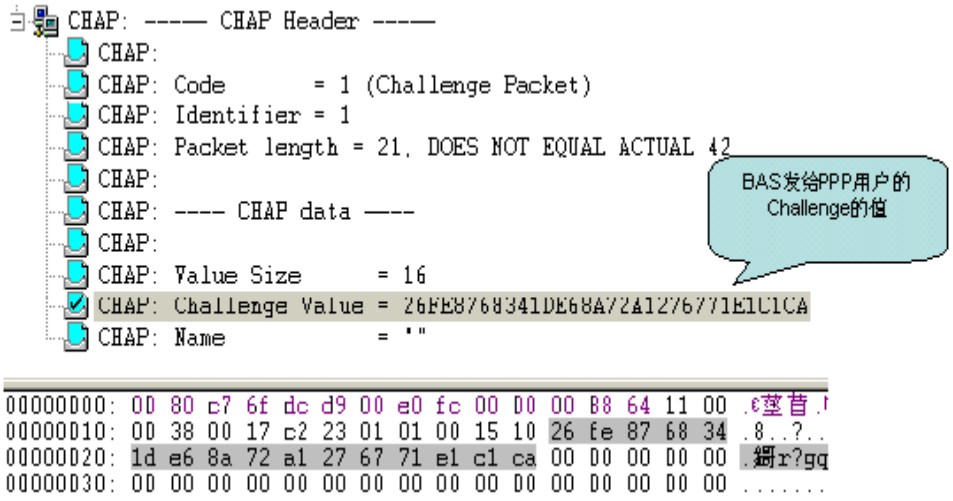
报文 6：Radius Server 计费结束响应报文。

从中可以看出对于报文请求和响应是通过 IP 地址+Radius 协议域中 ID 号进行配对识别的。

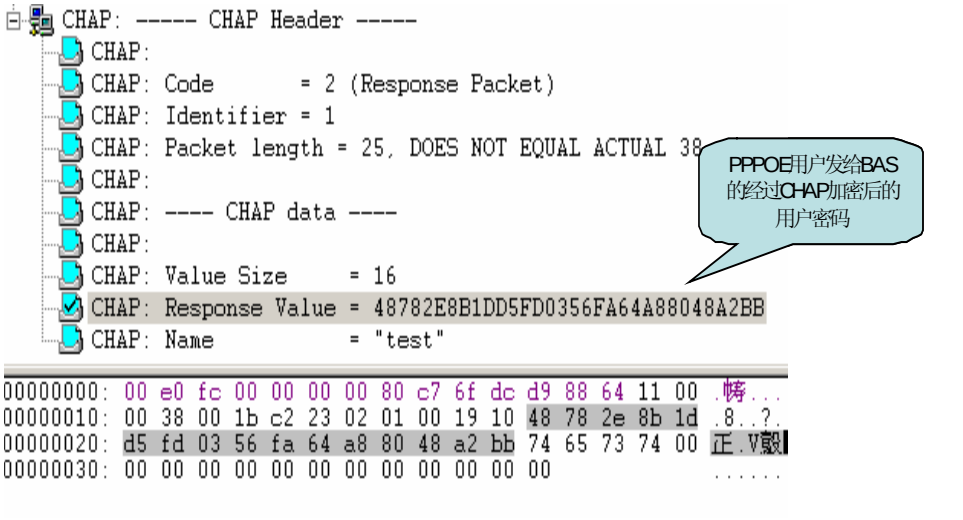
上图显示了 BAS 发起的 Radius 认证请求（Code＝1）报文的结构。Radius 报文是承载在 UPD 协议之上的，这里我没不关注上层报文的结构。

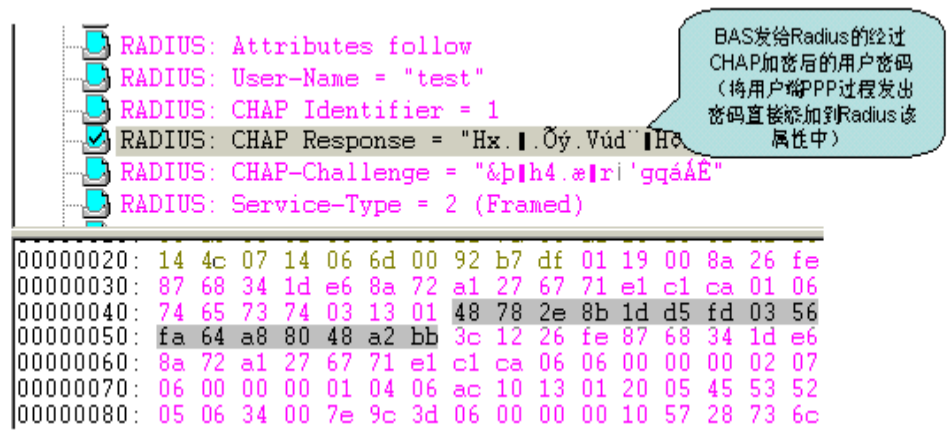
下图为 PPPOE CHAP 认证过程的 Radius 认证请求报文和 PPPOE 中 CHAP 认证的 Challenge 报文。通过比较可以方便看出 BAS 发出的 Challenge 值为“26fe8768341de68a72a1276771e1c1ca”与 PPPOE 中 CHAP 认证过程中 BAS 发给 PPPOE 用户的 Challenge 值是一致的。





下图为 PPPOE 用户发为 BAS 的经过 CHAP 加密后的用户密码和 BAS 发给 Radius Server 中认证请求报文用户秘密属性域的比较。可以看出在 Radius 认证过程中，BAS 设备将 Challenge 属性和用户加密后的密码发给 Radius 进行验证。





通过比较可以清楚了解协议各字段含义相互关系，为问题处理提供有效的手段。

下面为 PPPOE 用户 Radius 认证的 Sniffer 捕获的报文。


chap-client.cap


RadiusServer.cap