实验 网络数据包的监听与分析

一 实验目的

1.掌握使用 **Wireshark** 软件监听和捕获网络数据包。

2.掌握通过实际观察网络数据进行分析而了解网络协议运行情况。

二 实验要求

1.设备要求：计算机若干台（装有 Windows 2000/XP/2003 操作系统、装有网卡），局域网环境，主机装有 **Wireshark** 工具。

2.每组 1 人，独立完成。

三 实验预备知识

1. Wireshark 简介

Wireshark 是一个开放源码的网络分析系统，也是是目前最好的开放源码的网络协议分析软件之一，支持 Linux 和 Windows 平台，支持 500 多种协议分析。

网络分析系统首先依赖于一套捕捉网络数据包的函数库。这套函数库工作在在网络分析系统模块的最底层。作用是从网卡取得数据包或者根据过滤规则取出数据包的子集，再转交给上层分析模块。从协议上说，这套函数库将一个数据包从链路层接收，将其还原至传输层以上，以供上层分析。在 Linux 系统中，1992 年 Lawrence Berkeley Lab 的 Steven McCanne 和 Van Jacobson 提出了包过滤器，称之为 BPF（BSD Packet Filter），设计了基于 BPF 的捕包函数库 Libpcap。在 Window 系统中，意大利人 Fulvio Risso 和 Loris Degioanni 提出并实现了 Winpcap 函数库，其实现思想来源于 BPF。

2. Wireshark 的简单操作方法

安装 **Wireshark** 之前，需要安装 Winpcap，安装过程比较简单。安装完成后，启动 **Wireshark**，如图 2.1 所示。

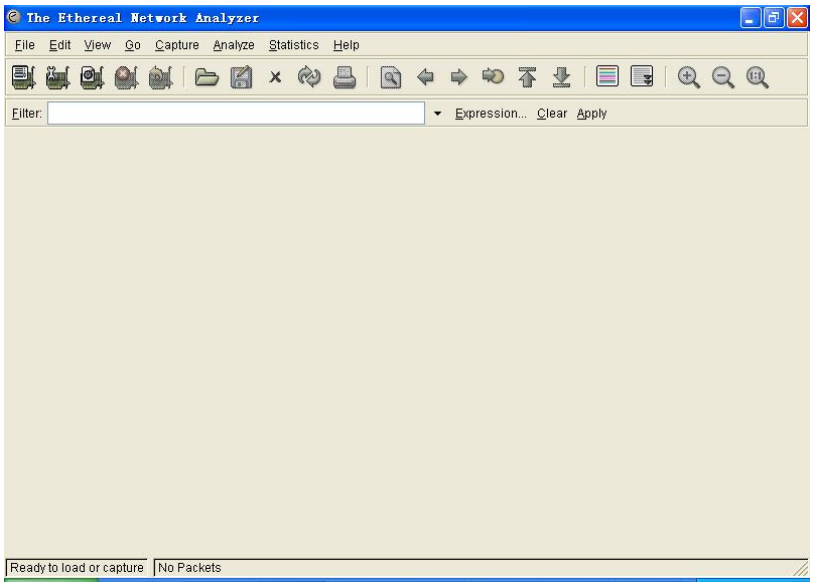


图 2.1 启动 **Wireshark** 后的界面

设置 Capture 选项。选择“Capture”-“Options”，弹出“Capture Options”界面，设置完成后点击“Capture”而开始捕获数据，如图 2.2 所示。

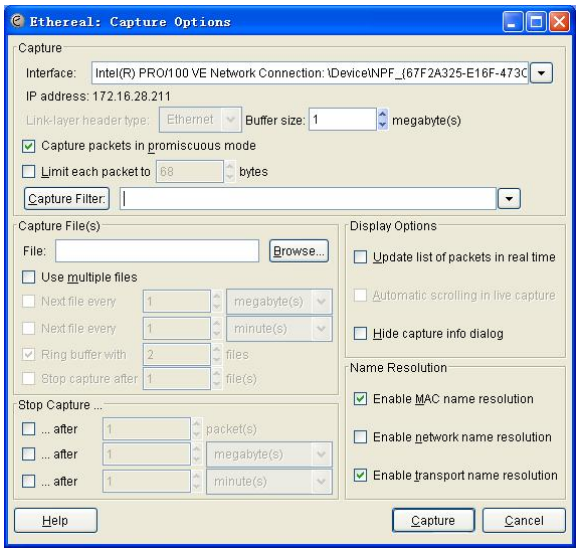


图 2.2 “Capture Options”界面

在“Capture Options”界面中，主要选项如下：

- “Interface”是要求选择在哪个接口（网卡）上抓包。
- “Limit each packet”是限制每个包的大小，缺省情况不限制。
- “Capture packets in promiscuous mode”是否打开混杂模式。如果打开，抓取所有的数据包。一般情况下只需要监听本机收到或者发出的包，因此应该关闭该选项。
- “Capture Filter”是指过滤器，可以过滤掉某些数据包而只抓取满足过滤规则的数据包。
- “File”是指如果需要将抓到的包保存到文件中，在这里输入文件名称。
- “Ring buffer”是指是否使用循环缓冲。缺省情况下不使用，即一直抓包。注意，循环缓冲只有在写文件的时候才有效。

设置完“Capture Options”后，选择“Capture”-“Start”开始捕获数据，如图 2.3 所示。

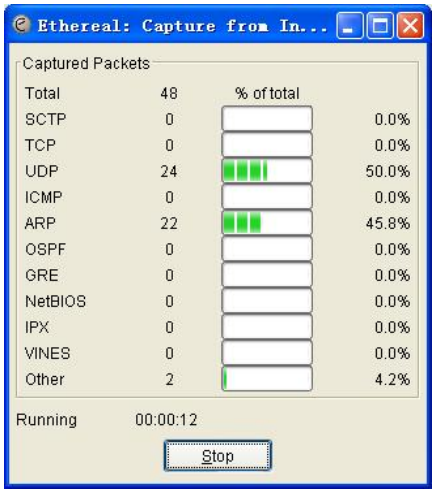


图 2.3 开始捕获数据

点击“Stop”完成数据捕获，如图 2.4 所示。根据捕获的数据对各种协议，如 ARP、ICMP、TCP、UDP 等协议进行分析。

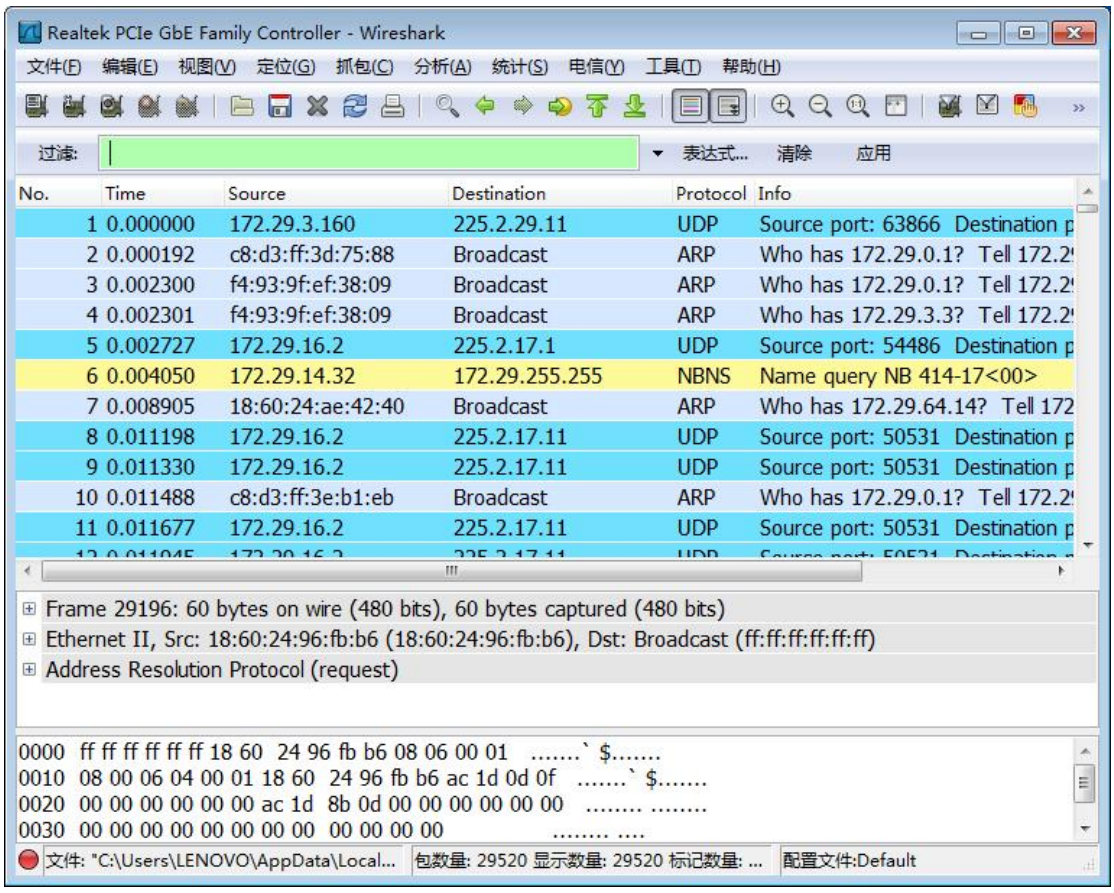


图 2.4 捕获数据包后的协议分析

2. Wireshark 过滤器的简单使用

如果需要抓取某些特定的数据包时，可以有两种方法，一是在捕获数据包之前定义好包过滤器，这样就只能捕获到设定好的那些类型的数据包。包过滤器用来捕获感兴趣的数据包，用在捕获数据包过程中。包过滤器使用的是 Libcap 过滤器语言，在 Tcpdump 的手册中有详细的解释，基本结构是：

[not] primitive [and|or [not] primitive ...]

另外一种方法是捕获本机收到或者发出的全部数据包，然后使用显示过滤器，只让 Ethereal 显示所需要的那些类型的数据包。下面主要介绍这种方法。

在捕获数据包完成后，可以根据“协议”、“是否存在某个域”、“域值”和“域值之间的比较”等四个规则来过滤数据包。

例如，如果只需查看使用 ARP 协议的数据包，在 **Wireshark** 窗口中的“Filter”中输入 arp（注意是小写）， 然后回车或点击“Apply”，**Wireshark** 就会只显示 ARP 协议的数据包，如图 2.5 所示。

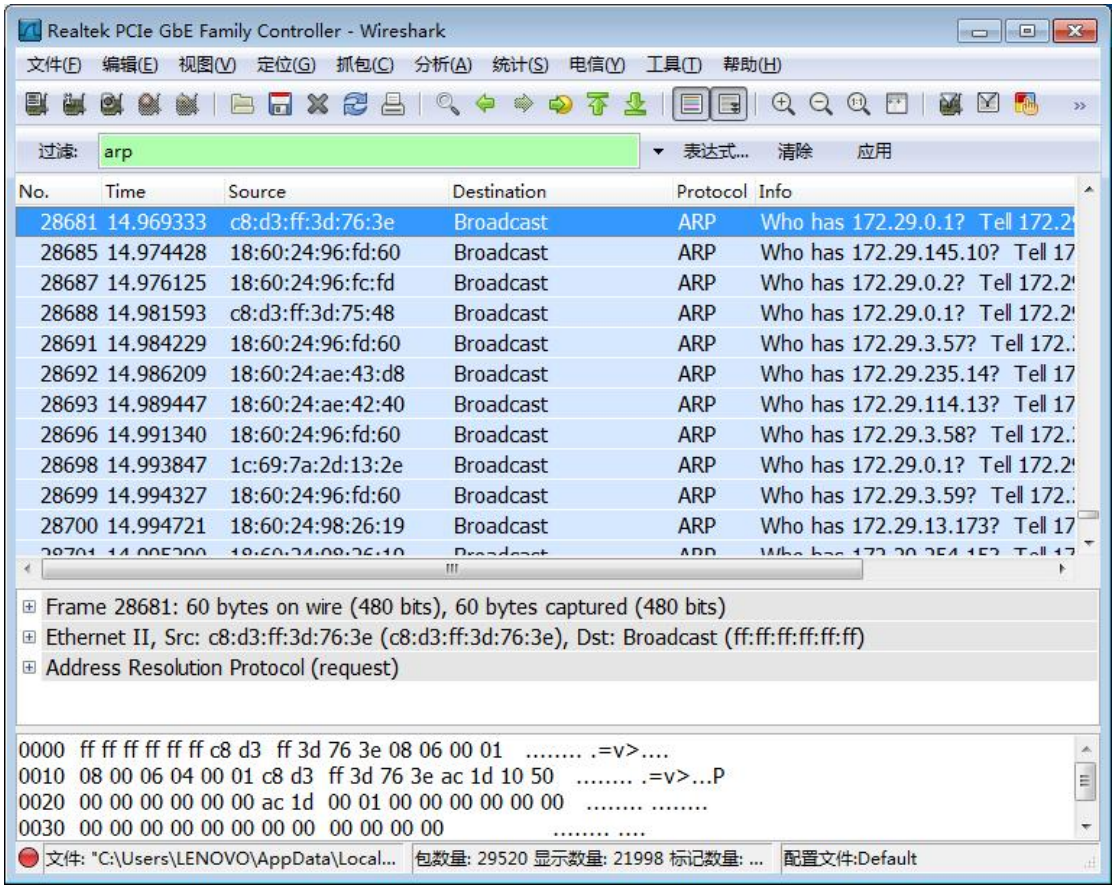


图 2.5 使用协议进行过滤

域值比较表达式可以使用“==”、“>”、“!=”等操作符来构造显示过滤器，例如 ip.addr==10.1.10.20，ip.addr!=10.1.10.20，frame.pkt_len>10 等。域值可以从“Expression”中进行选择，如图 2.6 所示。

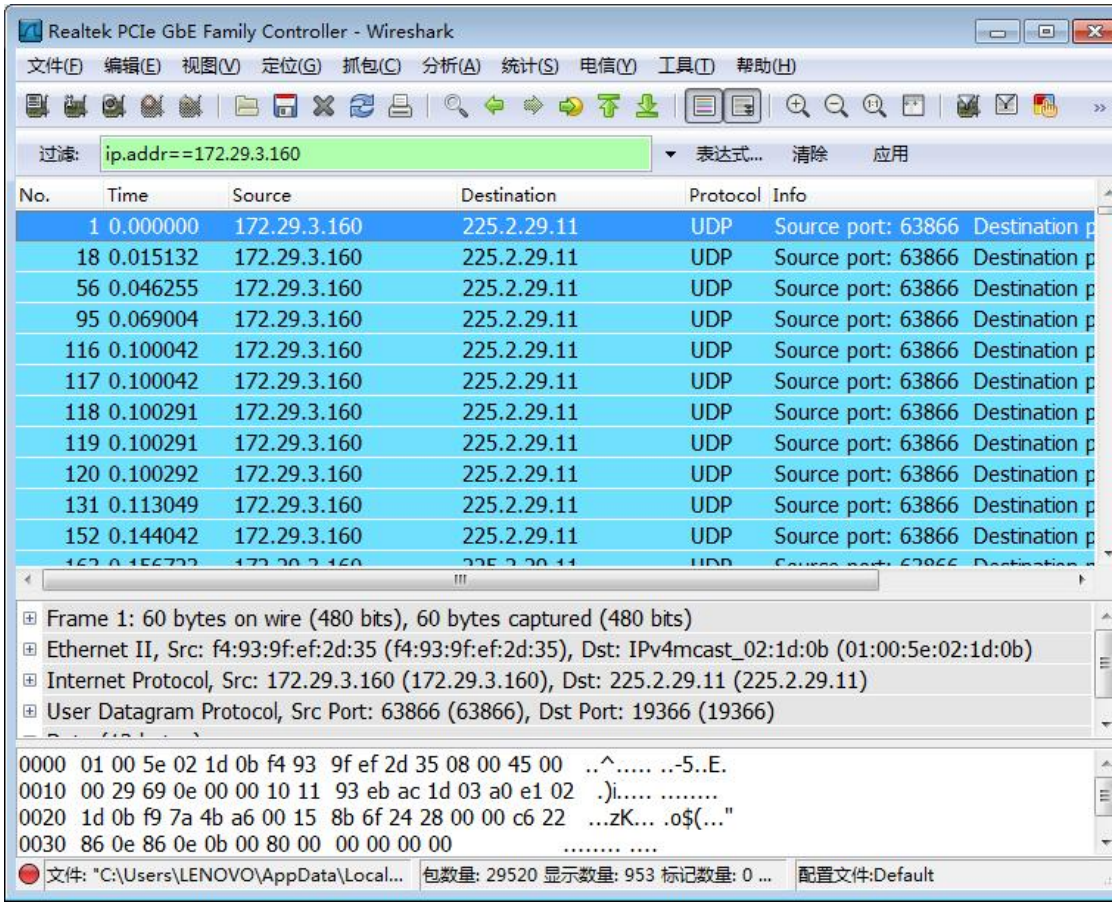


图 2.6 添加域值比较表达式

组合表达式还可以使用“and”、“or”、和“not”等逻辑操作符，其中逻辑与“and”也可用“&&”表示，例如 ip.addr==172.16.28.211&&frame.pkt_len < 100；逻辑或“or”也可用“||”表示，例如 ip.addr==172.16.28.211||ip.addr==172.16.28.254，如图 2.7 所示；逻辑非“not”也可用“!”表示，例如!llc。

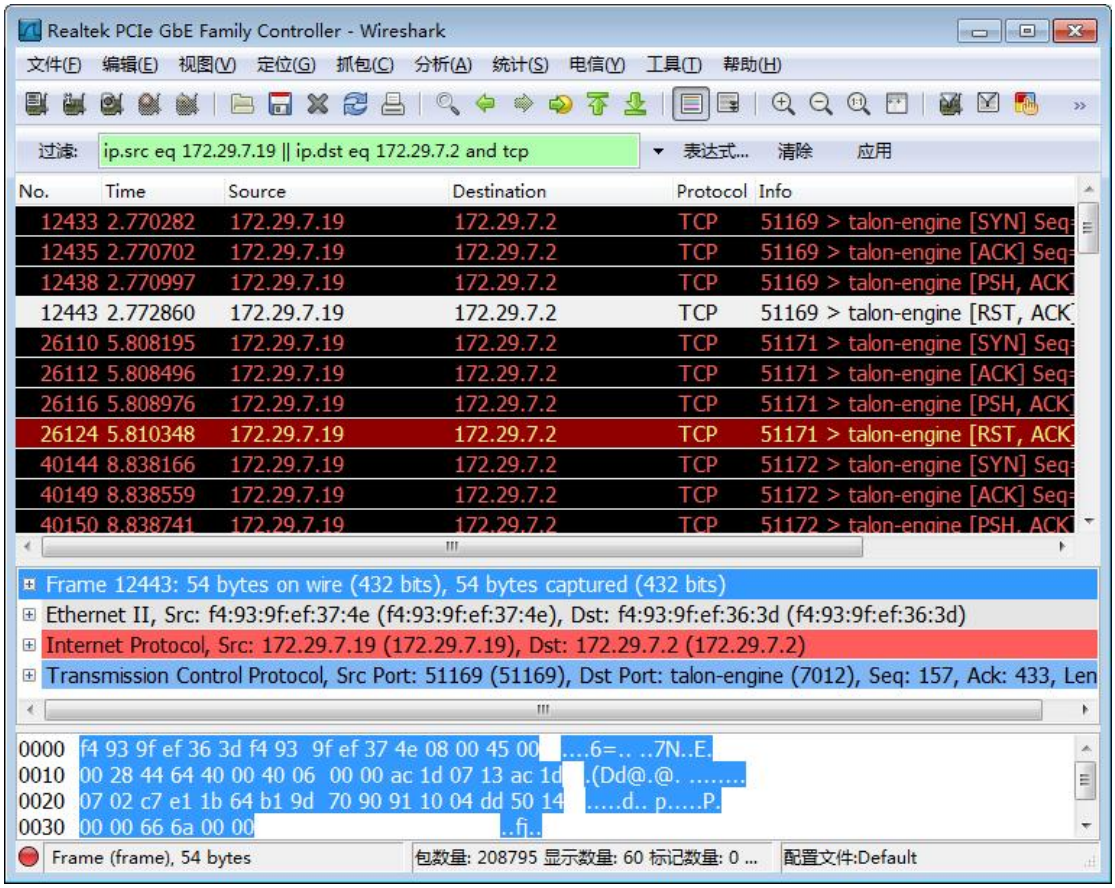


图 2.7 组合表达式的应用

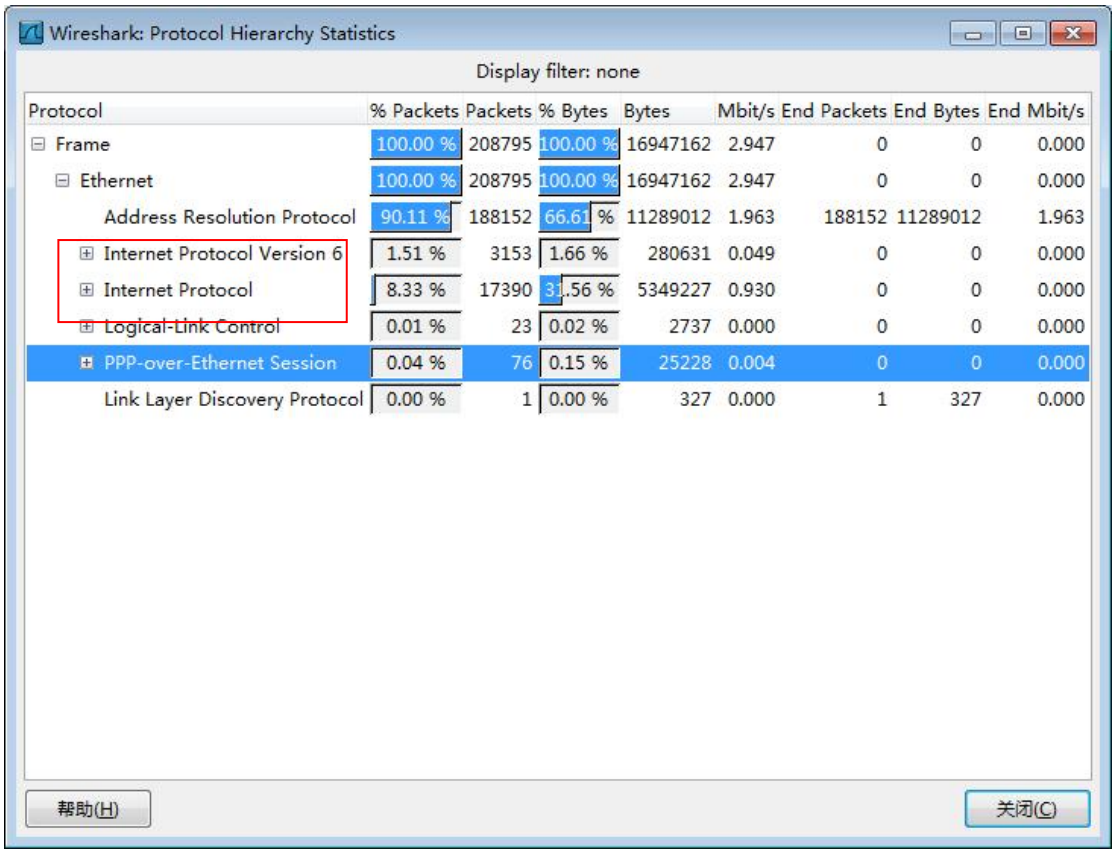
四 实验内容与步骤

本实验指导可利用实验室网络进行。

1.使用 Ethereal（Wireshark）捕获数据包，完成如下操作：

（1）当前网络中主要的网络协议是什么？请记录实验数据。

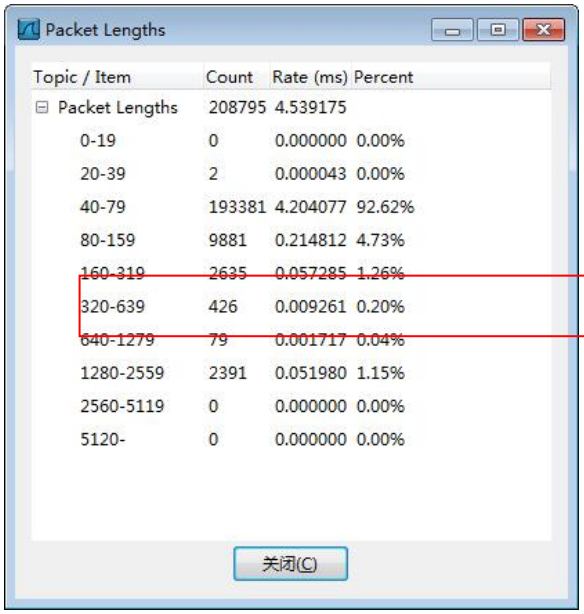
可根据“协议分级统计”（“统计”菜单下）来查看统计结果。如下图所示。



主要的网络协议
IP、UDP、TCP、IGMP、ARP、IPv6、DNS、HTP、ICMPv6

（2）当前网络中主要数据包大小在什么范围？请记录实验数据。

可根据“分组长度”（“统计”菜单下）查看统计结果。如下图所示。



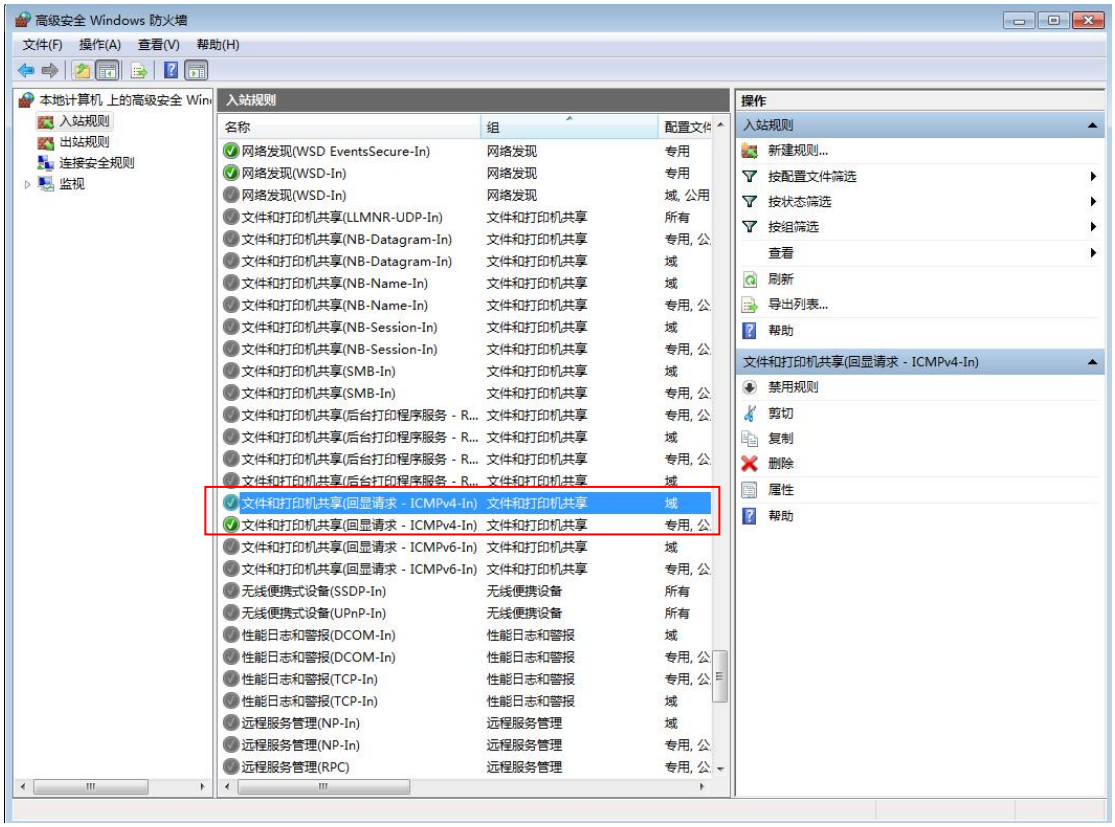
数据包大小范围、个数
20-39B，占 0.00%; 40-79B，占 92.62%; 80-159B，占 4.73%; 160-319B，占 1.26%; 320-639B，占 0.20%; 640-1279B,占比 0.04%; 1280-2559B,占 1.15%

2.协议分析

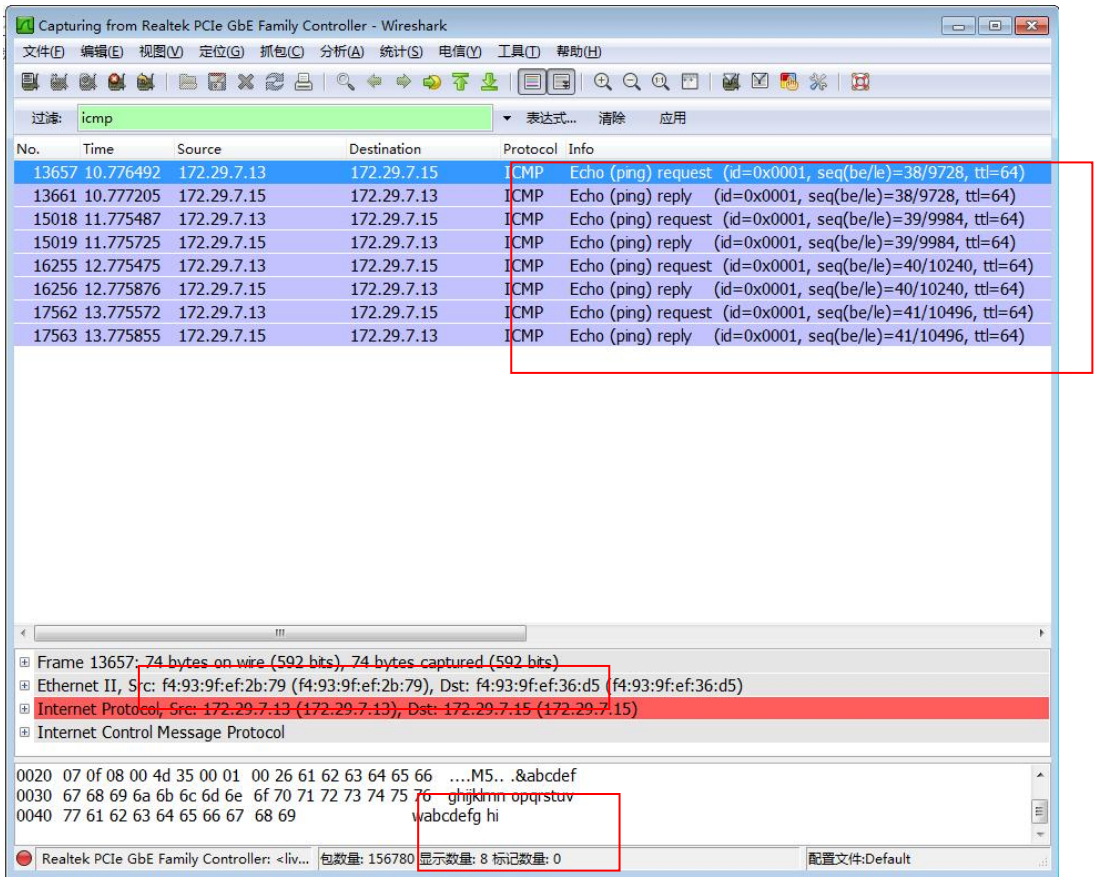
（1）设置源主机的 TCP/IP 属性，并记录其 IP；设置目的主机的 TCP/IP 属性，使其与源主机在同一个网络内，记录 IP。

源主机 IP	目的主机 IP
172.29.7.13	172.29.7.15

（2）启用防火墙中，入站规则中的文件和打印机共享（回显请求-IPv4-In）。



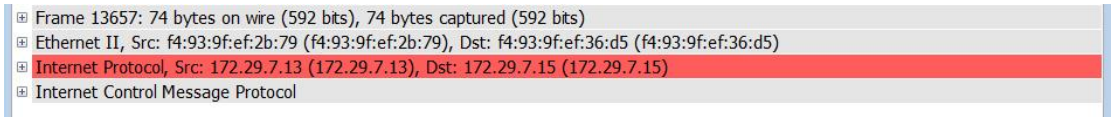
（3）使用显示过滤器 Filter，在 Filter 中输入“icmp”，只显示 ICMP 协议的相关数据包，完成如下分析：



问题	回答
ICMP 请求包和应答包个数	共 8 个
ICMP 数据包的大小	74bytes
TTL（Time to live）	64
Data (32 bytes)	abcdefghijklmnopqrstuvwabcdefghi

（5）给出 icmp 数据包按照网络层次封装的数据包名称。

可结合 icmp 分组数据包的结构图说明。如图所示。



各行信息和 TCP/IP 或 OSI 模型一一对应。Frame 是物理层的数据帧概况；Ethernet II 是数据链路层以太网帧头部信息；Internet Protocol 是网络层 IP 包头信息；Transmission Control Protocol 是传输层的数据段头部信息，此处是 TCP。ICMP 数据包封装到 IP 数据包

中，IP 数据包封装到 Ethernet II 包中，然后形成帧 Frame 在网络中传输。

ICMP 数据包封装到 IP 数据包中，IP 数据包封装到 Ethernet II 包中，然后形成帧 Frame 在网络中传输。

五 练习与思考

（1）设置捕获过滤器，捕捉并分析局域网上的所有 Ethernet Broadcast 帧。在 Ethereal（**Wireshark**）中选择菜单“Capture”- “Capture Filters”，弹出“Capture Filters”界面。“Filter name”选项为 Fiter 名称，“Filter string”选项设置为：broadcast，如图 2.8 所示。

（2）通过选择菜单“Capture”- “Options”- “Capture Filter:”，选择之前建立的过滤条件，然后点击下面的“start”，开始捕捉数据包。如图 2.9 所示。

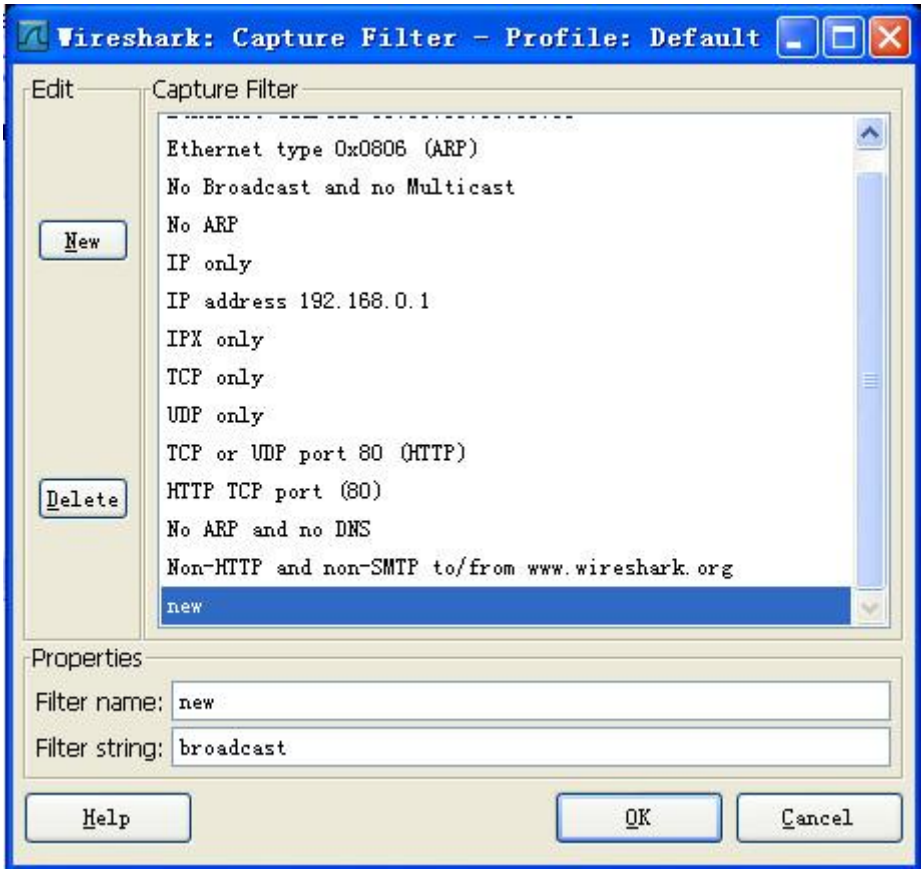
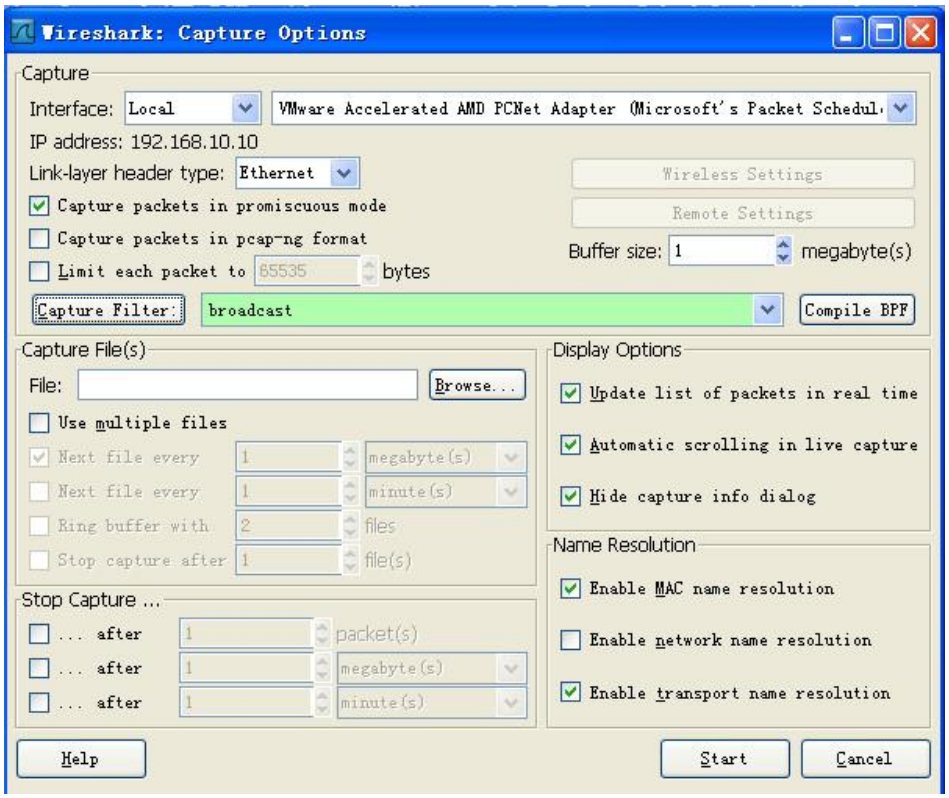
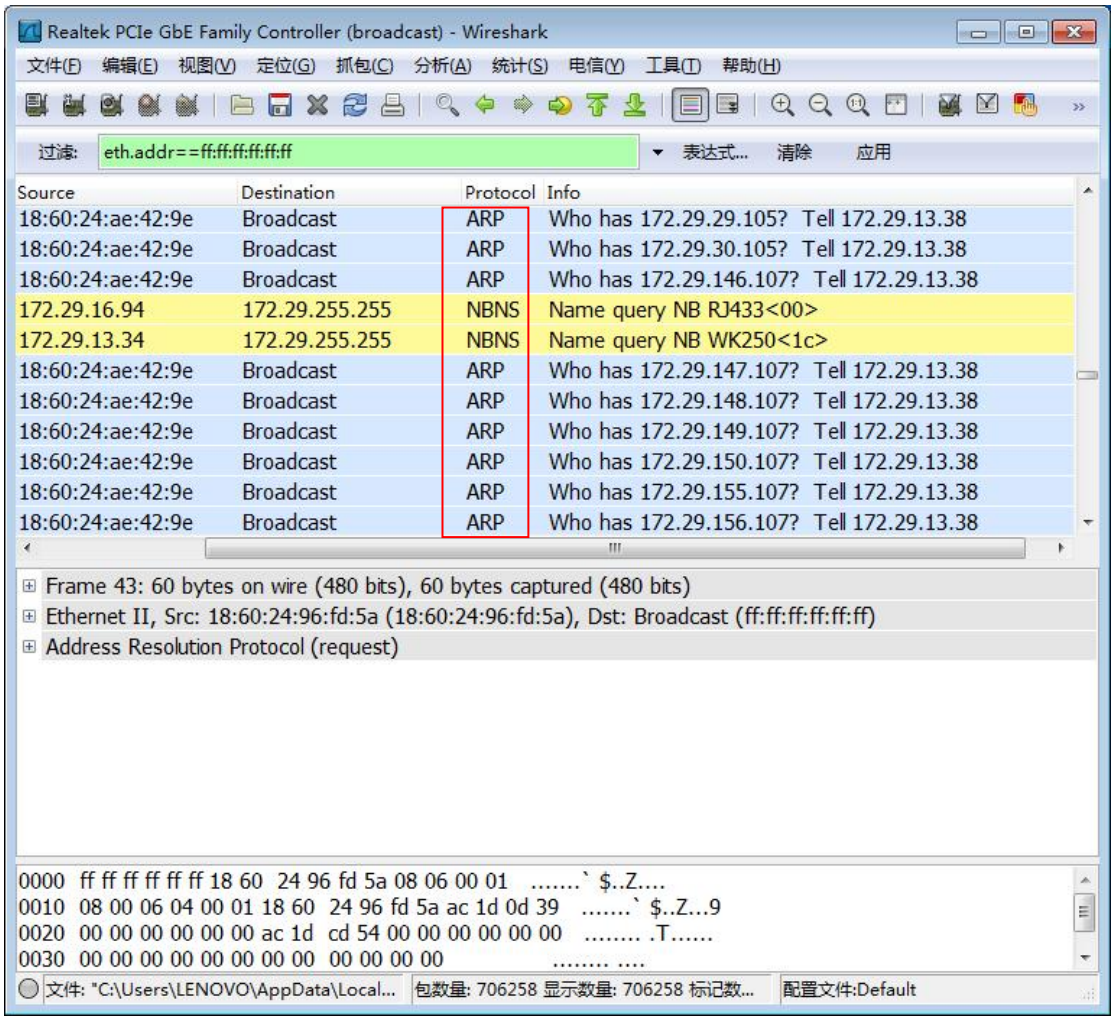


图 2.8 设置“Capture Filters”



① 观察并分析哪些主机在发广播帧（广播帧是指目的地址为 broadcast 的数据包），这些帧的高层协议是什么？

高层协议可以在“Protocol”字段查看，如图所示。发广播帧的主机主要有 172.29.16.94，172.29.7.61，172.29.13.51，172.29.13.15，172.29.13.57，172.29.7.22，172.29.13.21。这些帧的高层协议是 ARP、NBNS。



② 该 LAN 的共享网段上连接了多少台计算机？

可以通过“Statistics”下的“Conversations”对话框查看有多少 IP 地址是活跃的。如图所示，显示当前有两个 IP 地址是活跃的。