

---

# Wireshark 网络分析实例集锦

(内部资料)

A large, light gray watermark of the Wireshark logo is centered on the page. It features a stylized shark fin above the word "WIRESHARK" in a bold, sans-serif font.

**WIRESHARK**

大学霸

[www.daxueba.net](http://www.daxueba.net)



---

# 前言

由于网络广泛广泛，与网络相关的安全问题也就变的非常重要。为了更好的分析整个网络的情况，人们开始使用各种专业的数据包分析工具。**Wireshark** 是一款最知名的开源网络封包分析软件。它可以抓取网络封包，并以最为详细的方式，显示封包的数据。

**Wireshark** 应用非常广泛。例如，网络管理员使用 **Wireshark** 来检测网络问题；网络安全工程师使用 **Wireshark** 检查数据传输的安全问题；开发者使用它为新的通信协议排错；普通人使用 **Wireshark** 来学习网络协议的相关知识。

由于 **Wireshark** 工具的广泛使用及市场的需求，笔者编写了这本书。本书按照网络分析专业流程，一步步地介绍了 **Wireshark** 各项功能的使用。本书还介绍了命令行下数据捕获的方法，以满足了在命令行下操作的用户。希望各位读者能在本书的带领下熟练地掌握 **Wireshark**，并且成为数据包的分析高手。

## 1.学习所需的系统和设备

本书所讲解的内容基于 Windows 7 和 Red Hat Enterprise Linux 6.4。读者在学习的时候，也可以采用其他操作系统。如果为了方便抓取各种数据，建议读者安装 VM ware，以虚拟各种其他系统或者服务。

## 2.学习建议

大家学习之前，可以致信到 [wireshark@daxueba.net](mailto:wireshark@daxueba.net)，获取相关的资料和软件。如果大家在学习过程遇到问题，也可以将问题发送到该邮箱。我们尽可能给大家解决。

---

# 目 录

|       |                                |    |
|-------|--------------------------------|----|
| 第 1 章 | Wireshark 的基础知识.....           | 1  |
| 1.1   | Wireshark 的功能.....             | 1  |
| 1.1.1 | Wireshark 主窗口界面.....           | 1  |
| 1.1.2 | Wireshark 的作用.....             | 2  |
| 1.2   | 安装 Wireshark.....              | 3  |
| 1.2.1 | 获取 Wireshark.....              | 3  |
| 1.2.2 | 安装 Wireshark.....              | 5  |
| 1.3   | Wireshark 捕获数据.....            | 11 |
| 1.5   | 认识数据包 .....                    | 12 |
| 1.6   | 捕获 HTTP 包 .....                | 14 |
| 1.7   | 访问 Wireshark 资源.....           | 18 |
| 1.8   | Wireshark 快速入门.....            | 21 |
| 1.9   | 分析网络数据 .....                   | 28 |
| 1.9.1 | 分析 Web 浏览数据.....               | 28 |
| 1.9.2 | 分析后台数据 .....                   | 30 |
| 1.10  | 打开其它工具捕获的文件 .....              | 31 |
| 第 2 章 | 设置 Wireshark 视图.....           | 33 |
| 2.1   | 设置 Packet List 面板列 .....       | 33 |
| 2.1.1 | 添加列 .....                      | 33 |
| 2.1.2 | 隐藏、删除、重新排序及编辑列 .....           | 35 |
| 2.2   | Wireshark 分析器及 Profile 设置..... | 41 |
| 2.2.1 | Wireshark 分析器.....             | 41 |
| 2.2.2 | 分析非标准端口号流量 .....               | 43 |
| 2.2.3 | 设置 Wireshark 显示的特定数据类型.....    | 45 |
| 2.2.4 | 使用 Profile 定制 Wireshark.....   | 50 |
| 2.2.5 | 查找关键的 Wireshark Profile.....   | 52 |
| 2.3   | 数据包时间延迟 .....                  | 54 |
| 2.3.1 | 时间延迟 .....                     | 54 |
| 2.3.2 | 检查延迟问题 .....                   | 55 |
| 2.3.3 | 检查时间差延迟问题 .....                | 57 |
| 第 3 章 | 捕获过滤器技巧 .....                  | 61 |
| 3.1   | 捕获过滤器简介 .....                  | 61 |
| 3.2   | 选择捕获位置 .....                   | 62 |
| 3.3   | 选择捕获接口 .....                   | 62 |
| 3.4.1 | 判断那个适配器上的数据 .....              | 63 |
| 3.4.2 | 使用多适配器捕获 .....                 | 63 |
| 3.4   | 捕获以太网数据 .....                  | 64 |
| 3.5   | 捕获无线数据 .....                   | 65 |

---

|       |                                     |     |
|-------|-------------------------------------|-----|
| 3.5.1 | 捕获无线网络数据方式 .....                    | 65  |
| 3.5.2 | 使用 AirPcap 适配器 .....                | 66  |
| 3.6   | 处理大数据 .....                         | 66  |
| 3.6.1 | 捕获过滤器 .....                         | 66  |
| 3.6.2 | 捕获文件集 .....                         | 68  |
| 3.7   | 处理随机发生的问题 .....                     | 70  |
| 3.8   | 捕获基于 MAC/IP 地址数据 .....              | 72  |
| 3.8.1 | 捕获单个 IP 地址数据 .....                  | 72  |
| 3.8.2 | 捕获 IP 地址范围 .....                    | 74  |
| 3.8.3 | 捕获广播或多播地址数据 .....                   | 76  |
| 3.8.4 | 捕获 MAC 地址数据 .....                   | 77  |
| 3.9   | 捕获端口应用程序数据 .....                    | 80  |
| 3.9.1 | 捕获所有端口号的数据 .....                    | 80  |
| 3.9.2 | 结合基于端口的捕获过滤器 .....                  | 81  |
| 3.10  | 捕获特定 ICMP 数据 .....                  | 82  |
| 第 4 章 | 显示技巧 .....                          | 86  |
| 4.1   | 显示过滤器简介 .....                       | 86  |
| 4.2   | 使用显示过滤器 .....                       | 87  |
| 4.2.1 | 显示过滤器语法 .....                       | 87  |
| 4.2.2 | 检查语法错误 .....                        | 89  |
| 4.2.3 | 识别字段名 .....                         | 91  |
| 4.2.4 | 比较运算符 .....                         | 92  |
| 4.2.5 | 表达式过滤器 .....                        | 93  |
| 4.2.6 | 使用自动补全功能 .....                      | 94  |
| 4.2.7 | 手动添加显示列 .....                       | 96  |
| 4.3   | 编辑和使用默认显示过滤器 .....                  | 98  |
| 4.4   | 过滤显示 HTTP .....                     | 100 |
| 4.5   | 过滤显示 DHCP .....                     | 102 |
| 4.6   | 根据地址过滤显示 .....                      | 103 |
| 4.6.1 | 显示单个 IP 地址或主机数据 .....               | 103 |
| 4.6.2 | 显示一个地址范围的数据 .....                   | 106 |
| 4.6.3 | 显示一个子网 IP 的数据 .....                 | 107 |
| 4.7   | 过滤显示单一的 TCP/UDP 会话 .....            | 108 |
| 4.8   | 使用复杂表达式过滤 .....                     | 112 |
| 4.8.1 | 使用逻辑运算符 .....                       | 112 |
| 4.8.2 | 使用括号 .....                          | 114 |
| 4.8.3 | 使用关键字 .....                         | 116 |
| 4.8.4 | 使用通配符 .....                         | 118 |
| 4.9   | 发现通信延迟 .....                        | 119 |
| 4.9.1 | 时间过滤器 (frame.time_delta) .....      | 119 |
| 4.9.2 | 基于 TCP 的时间过滤 (tcp.time_delta) ..... | 120 |

---

|        |                          |     |
|--------|--------------------------|-----|
| 4.10   | 设置显示过滤器按钮 .....          | 123 |
| 4.10.1 | 创建显示过滤器表达式按钮 .....       | 123 |
| 4.10.2 | 编辑、添加、删除显示过滤器按钮 .....    | 124 |
| 4.10.3 | 编辑 preferences 文件 .....  | 125 |
| 第 5 章  | 着色规则和数据包导出 .....         | 128 |
| 5.1    | 认识着色规则 .....             | 128 |
| 5.2    | 禁用着色规则 .....             | 129 |
| 5.2.1  | 禁用指定类型数据包彩色高亮 .....      | 129 |
| 5.2.2  | 禁用所有包彩色高亮 .....          | 131 |
| 5.3    | 创建用户着色规则 .....           | 132 |
| 5.3.1  | 创建时间差着色规则 .....          | 132 |
| 5.3.2  | 快速查看 FTP 用户名密码着色规则 ..... | 133 |
| 5.3.3  | 创建单个会话着色规则 .....         | 136 |
| 5.4    | 导出数据包 .....              | 138 |
| 5.4.1  | 导出显示包 .....              | 138 |
| 5.4.2  | 导出标记包 .....              | 140 |
| 5.4.3  | 导出包的详细信息 .....           | 141 |
| 第 6 章  | 构建图表 .....               | 147 |
| 6.1    | 数据统计表 .....              | 147 |
| 6.1.1  | 端点统计 .....               | 147 |
| 6.1.2  | 网络会话统计 .....             | 149 |
| 6.1.3  | 快速过滤会话 .....             | 150 |
| 6.1.4  | 地图化显示端点统计信息 .....        | 152 |
| 6.2    | 协议分层统计 .....             | 155 |
| 6.3    | 图表化显示带宽使用情况 .....        | 156 |
| 6.3.1  | 认识 IO Graph .....        | 156 |
| 6.3.2  | 应用显示过滤器 .....            | 157 |
| 6.4    | 专家信息 .....               | 161 |
| 6.5    | 构建各种网络错误图表 .....         | 162 |
| 6.5.1  | 构建所有 TCP 标志位包 .....      | 163 |
| 6.5.2  | 构建单个 TCP 标志位包 .....      | 164 |
| 第 7 章  | 重组数据 .....               | 166 |
| 7.1    | 重组 Web 会话 .....          | 166 |
| 7.1.1  | 重组 Web 浏览会话 .....        | 166 |
| 7.1.2  | 导出 HTTP 对象 .....         | 171 |
| 7.2    | 重组 FTP 会话 .....          | 174 |
| 7.2.1  | 重组 FTP 数据 .....          | 174 |
| 7.2.2  | 提取 FTP 传输的文件 .....       | 176 |
| 第 8 章  | 添加注释 .....               | 180 |
| 8.1    | 捕获文件注释 .....             | 180 |
| 8.2    | 包注释 .....                | 180 |

|       |  |     |
|-------|--|-----|
| 8.2.1 | 添加包注释 .....                            | 181 |
| 8.2.2 | 查看包注释 .....                            | 181 |
| 8.3   | 导出包注释 .....                            | 183 |
| 8.3.1 | 使用 Export Packet Dissections 功能导出..... | 183 |
| 8.3.2 | 使用复制功能导出包 .....                        | 185 |
| 第 9 章 | 捕获、分割、合并数据 .....                       | 189 |
| 9.1   | 将大文件分割为文件集 .....                       | 189 |
| 9.1.1 | 添加 Wireshark 程序目录到自己的位置.....           | 189 |
| 9.1.2 | 使用 Capinfos 获取文件大小和包数 .....            | 189 |
| 9.1.3 | 分割文件 .....                             | 190 |
| 9.2   | 合并多个捕获文件 .....                         | 195 |
| 9.3   | 命令行捕获数据 .....                          | 196 |
| 9.3.1 | Dumpcap 和 Tshark 工具.....               | 196 |
| 9.3.2 | 使用捕获过滤器 .....                          | 199 |
| 9.3.3 | 使用显示过滤器 .....                          | 199 |
| 9.4   | 导出字段值和统计信息 .....                       | 200 |
| 9.4.1 | 导出字段值 .....                            | 200 |
| 9.4.2 | 导出数据统计 .....                           | 202 |

# 第 1 章 Wireshark 的基础知识

Wireshark（前称 Ethereal）是一个网络包分析工具。该工具主要是用来捕获网络包，并显示包的详细情况。本章将介绍 Wireshark 的基础知识。

## 1.1 Wireshark 的功能

在学习 Wireshark 之前，首先介绍下它的功能。了解它的功能，可以帮助用户明确可以借助该工具完成哪些工作。本节将介绍 Wireshark 的基本功能。

### 1.1.1 Wireshark 主窗口界面

在学习使用 Wireshark 之前，首先需要了解该工具主窗口界面中每部分的作用。Wireshark 主窗口界面如图 1.1 所示。

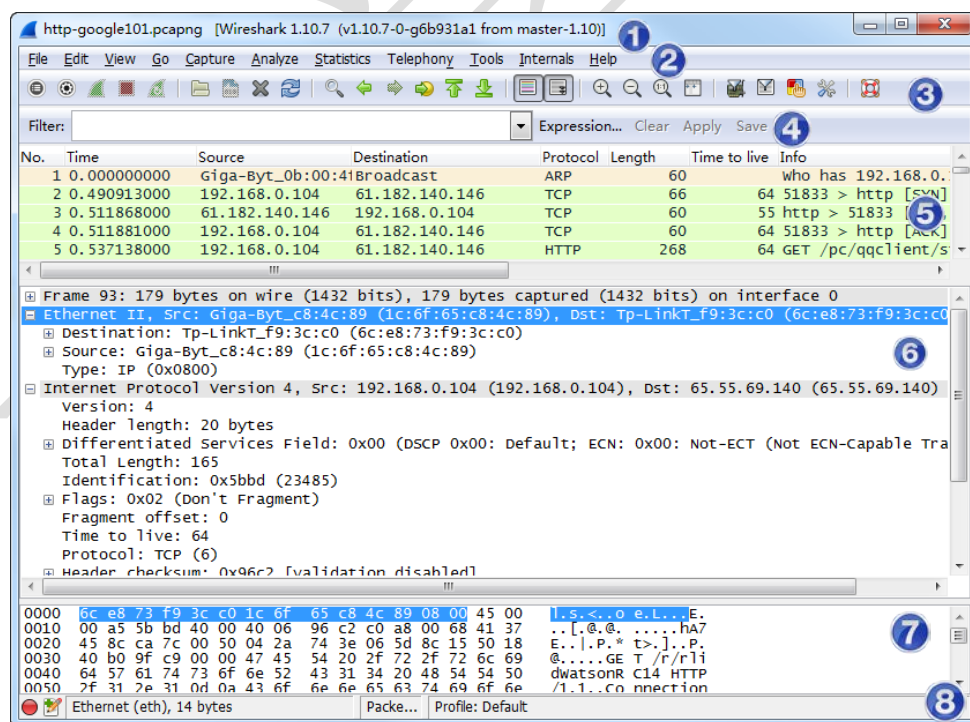


图 1.1 Wireshark 主窗口界面

在图 1.1 中，以编号的形式已将 Wireshark 每部分标出。下面分别介绍每部分的含义，如下所示：

- ①标题栏——用于显示文件名称、捕获的设备名称、Wireshark 版本号。



- ☐ ②菜单栏——Wireshark 的标准菜单栏。
- ☐ ③工具栏——常用功能快捷图标按钮。
- ☐ ④显示过滤区域——减少查看数据的复杂度。
- ☐ ⑤Packet List 面板——显示每个数据帧的摘要。
- ☐ ⑥Packet Details 面板——分析封包的详细信息。
- ☐ ⑦Packet Bytes 面板——以十六进制和 ASCII 格式显示数据包的细节。
- ☐ ⑧状态栏——专家信息、注释、包数和 Profile。

### 1.1.2 Wireshark 的作用

Wireshark（前称 Ethereal）是一个最受欢迎的网络数据包分析软件。网络数据包分析软件的功能是截取网络数据包，并尽可能显示出最为详细的网络数据包数据。它是一个最知名的开源应用程序的安全工具。Wireshark 可以运行在 Windows、MAC OS X、Linux 和 UNIX 操作系统上，它甚至可以作为一个 Portable App 运行。本节将介绍 Wireshark 常用的功能。使用 Wireshark 可以快速分析一些任务。如下所示：

#### 1.一般分析任务

- ☐ 找出在一个网络内的发送数据包最多的主机。
- ☐ 查看网络通信。
- ☐ 查看某个主机使用了哪些程序。
- ☐ 基本正常的网络通信
- ☐ 验证特有的网络操作。
- ☐ 了解尝试连接无线网络的用户。
- ☐ 同时捕获多个网络的数据。
- ☐ 实施无人值守数据捕获。
- ☐ 捕获并分析到/来自一个特定主机或子网的数据。
- ☐ 通过 FTP 或 HTTP 查看和重新配置文件传输。
- ☐ 从其它捕获工具导入跟踪文件。
- ☐ 使用最少的资源捕获数据。

#### 2.故障任务

- ☐ 为故障创建一个自定义的分析环境。
- ☐ 确定路径、客户端和服务延迟。
- ☐ 确定 TCP 问题。
- ☐ 检查 HTTP 代理问题。
- ☐ 检查应用程序错误响应。
- ☐ 通过查看图形显示的结果，找出相关的网络问题。
- ☐ 确定重载的缓冲区。
- ☐ 比较缓慢的通信到正常通信的一个基准。
- ☐ 找出重复的 IP 地址。
- ☐ 确定 DHCP 服务或网络代理问题。
- ☐ 确定 WLAN 信号强度问题。
- ☐ 检测 WLAN 连接的次数。
- ☐ 检查各种网络配置错误。

- ❑ 确定应用程序正在加载一个网络片段。

### 3.安全分析（网络取证）任务

- ❑ 为网络取证创建一个自定义分析环境。
- ❑ 检查使用非标准端口的应用程序。
- ❑ 确定到/来自可疑主机的数据。
- ❑ 查看哪台主机正在尝试获取一个 IP 地址。
- ❑ 确定“phone home”数据。
- ❑ 确定网络侦查过程。
- ❑ 全球定位和映射远程目标地址。
- ❑ 检查可疑数据重定向。
- ❑ 检查单个 TCP 或 UDP 客户端和服务端之间的会话。
- ❑ 检查到恶意畸形的帧。
- ❑ 在网络数据中找出攻击签名的关键因素。

### 4.应用程序分析任务

- ❑ 了解应用程序和协议如何工作。
- ❑ 图形应用程序的带宽使用情况。
- ❑ 确定是否将支持应用程序的链接。
- ❑ 更新/升级后检查应用程序性能。
- ❑ 从一个新安装的应用程序中检查错误响应。
- ❑ 确定哪个用户正在运行一个特定的应用程序。
- ❑ 检查应用程序如何使用传输协议，如 TCP 或 UDP。

## 1.2 安装 Wireshark

在大部分操作系统中，默认是没有安装 Wireshark 工具的。如果要使用该工具，首先需要学习安装 Wireshark。Wireshark 对主流的操作系统都提供了支持，其中包括 Windows、MAC OS X 以及基于 Linux 的系统。本节将介绍在 Windows 和 Linux 下安装 Wireshark 的方法。

### 1.2.1 获取 Wireshark

Wireshark 的所有操作系统版本都可以从官方网站获取到，Wireshark 的官方网站是 <http://www.wireshark.org>，如图 1.2 所示。该工具目前最新的稳定版本是 1.10.7。

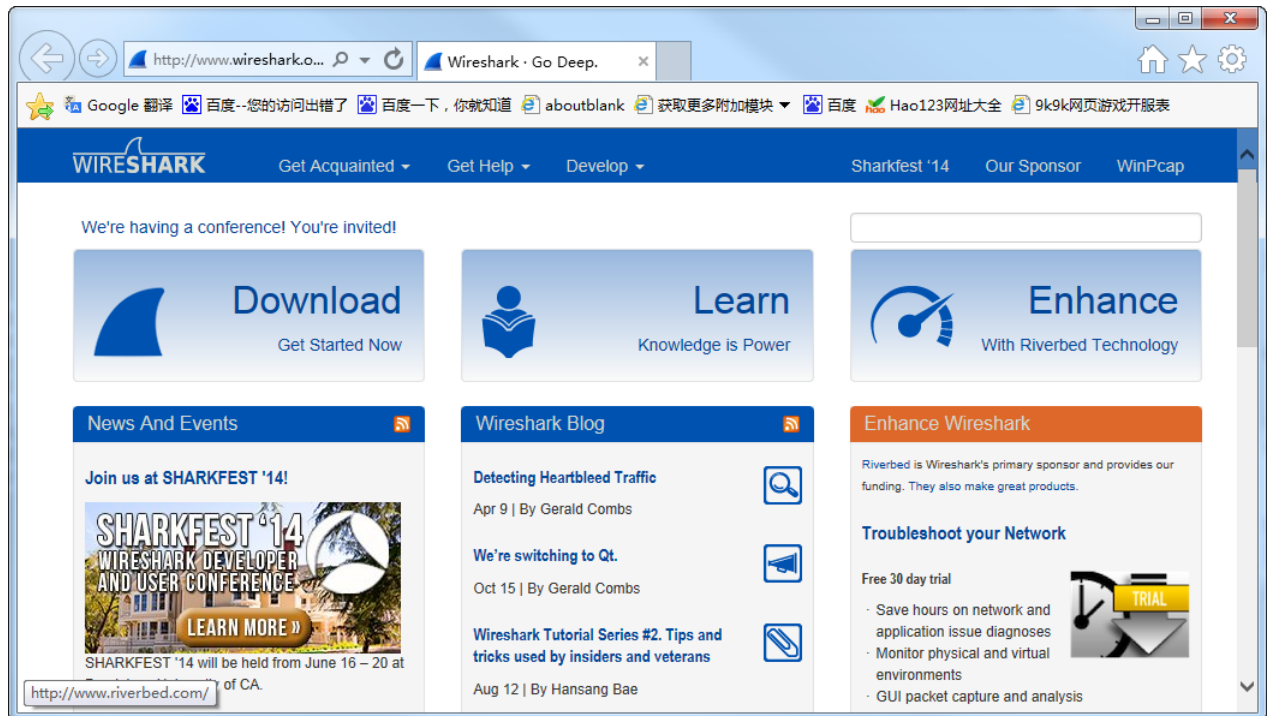


图 1.2 Wireshark 官方网站

在该界面单击 **Download** 按钮，将显示如图 1.3 所示的界面。

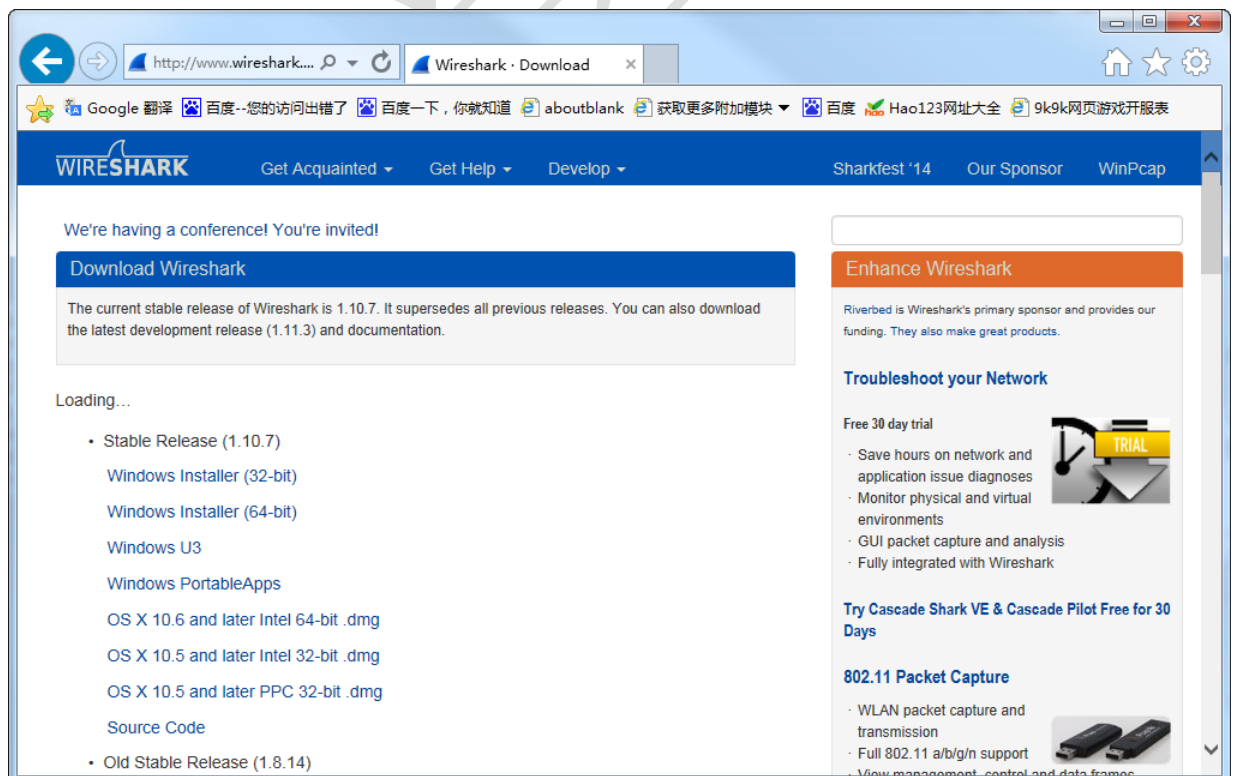


图 1.3 Wireshark 下载界面

从该界面可以看到，Wireshark 目前最新的版本是 1.10.7。该网站提供了 Windows、OS X 和源码包的下载地址。根据自己的操作系统，下载相应的软件包。

## 1.2.2 安装 Wireshark

在 Wireshark 的下载页面，可以看到所有 Wireshark 支持的操作系统列表。用户可以根据自己的操作系统，选择下载对应的软件包。本机将介绍分别在 Windows 和 Linux 上安装 Wireshark。

### 1.在 Windows 系统中安装 Wireshark

【实例 1-1】在 Windows 中安装 Wireshark。具体操作步骤如下所示：

（1）从 Wireshark 官网下载最新版本的 Windows 安装包，其名称为 Wireshark-win32-1.10.7.exe。

（2）双击下载的软件包，将显示如图 1.4 所示的界面。

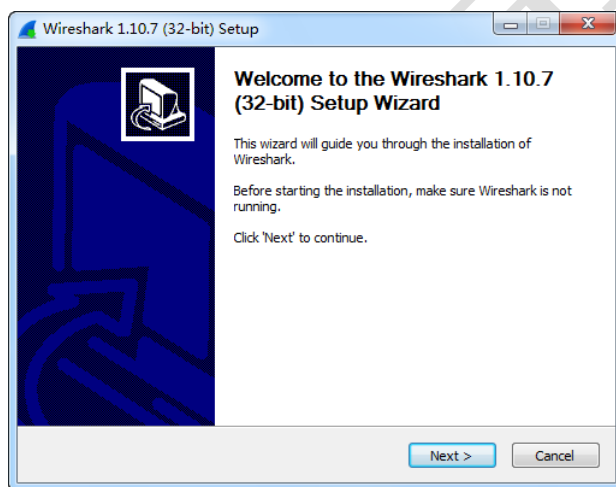


图 1.4 欢迎界面

（2）该界面显示了 Wireshark 的基本信息。此时单击 Next 按钮，将显示如图 1.5 所示的界面。

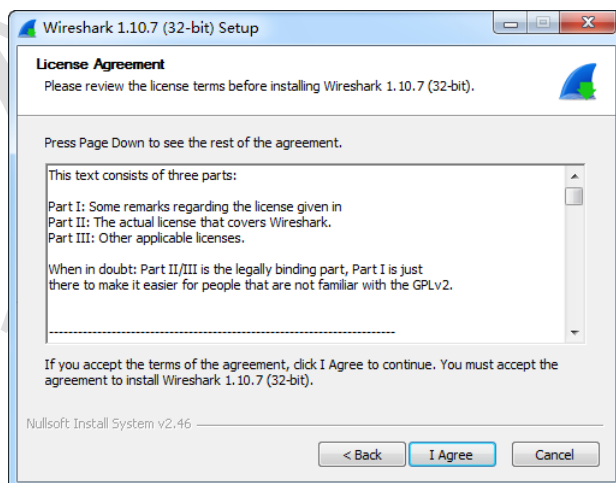


图 1.5 许可协议对话框

（3）该界面显示了使用 Wireshark 的许可证条款信息。此时单击 I Agree 按钮，将显示如图 1.6 所示的界面。

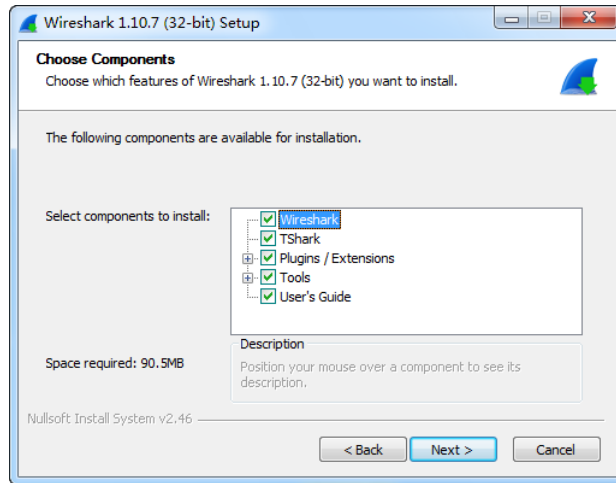


图 1.6 选择组件对话框

(4) 该界面选择希望安装的 Wireshark 组件，这里使用默认的设置。然后单击 Next 按钮，将显示如图 1.7 所示的界面。

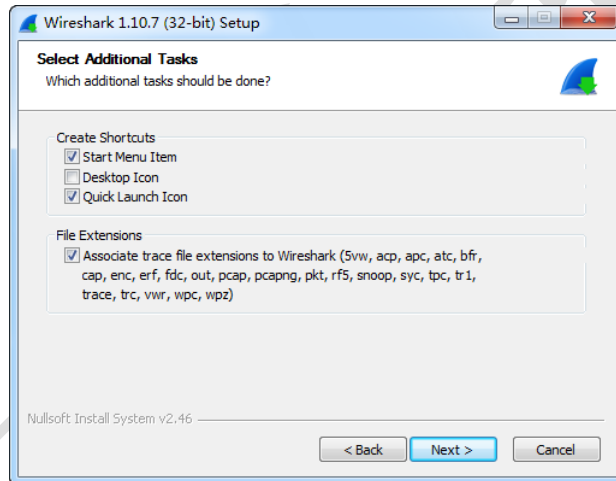


图 1.7 Additional Tasks 对话框

(5) 该界面用来设置创建快捷方式的位置和了解文件扩展名。设置完后单击 Next 按钮，将显示如图 1.8 所示的界面。

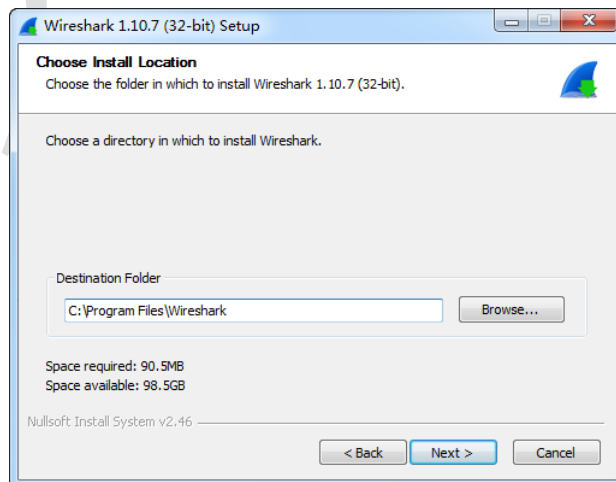


图 1.8 安装位置对话框

(6) 在该界面选择 Wireshark 的安装位置。然后单击 Next 按钮，将显示如图 1.9 所示的界面。

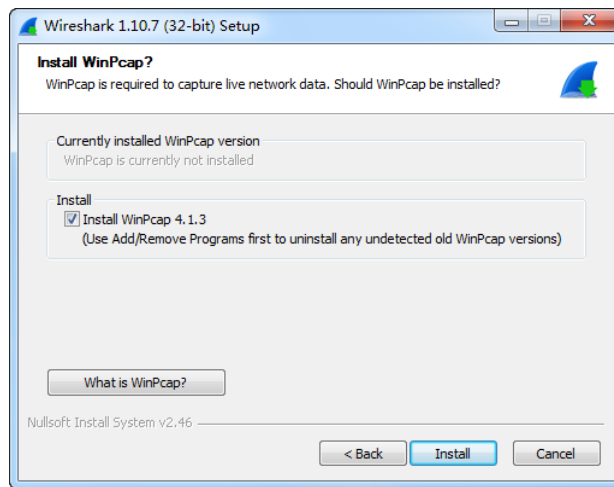


图 1.9 安装 WinPcap 对话框

(7) 该界面提示是否要安装 WinPcap。如果要使用 Wireshark 捕获数据，必须要安装 WinPcap。所以这里必须将 Install WinPcap 4.1.3 复选框勾上，然后单击 Install 按钮，Wireshark 将开始安装。等 Wireshark 安装过程进行了大约一半的时候，将弹出如图 1.10 所示的界面。

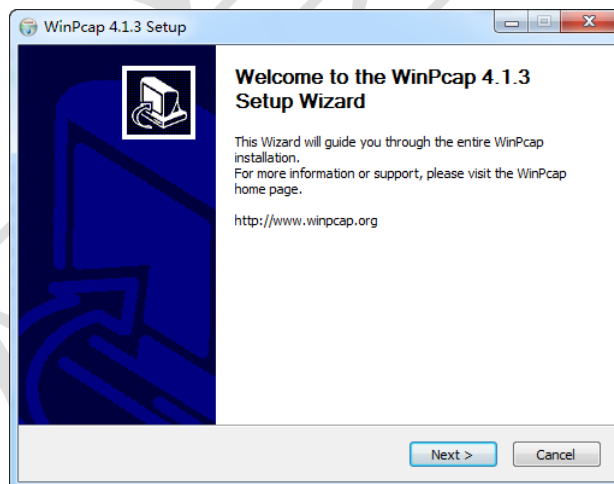


图 1.10 WinPcap 欢迎界面

(8) 该界面显示了 WinPcap 基本信息。此时单击 Next 按钮，将显示如图 1.11 所示的界面。

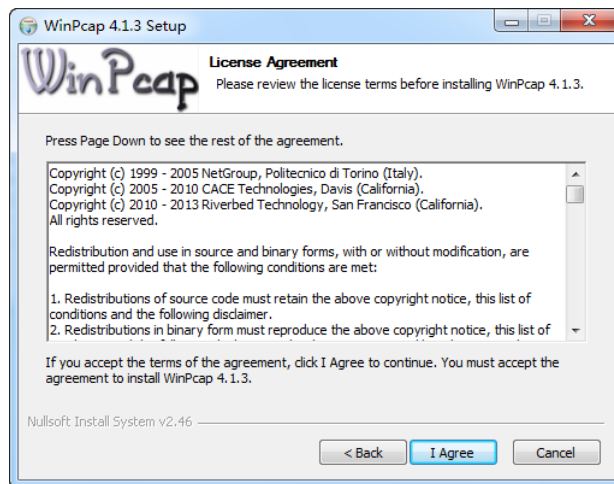


图 1.11 WinPcap 许可证条款对话框

(9) 该界面显示了 WinPcap 许可证条款信息。此时单击 I Agree 按钮，将显示如图 1.12 所示的界面。

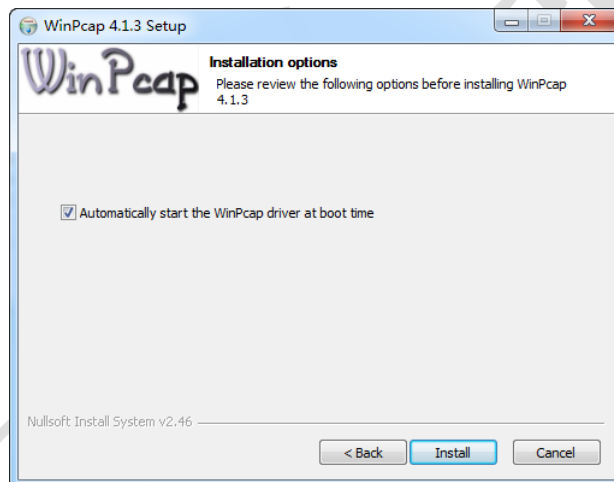


图 1.12 安装选项

(9) 在该界面显示了安装 WinPcap 选项，然后单击 Install 按钮，将显示如图 1.13 所示的界面。

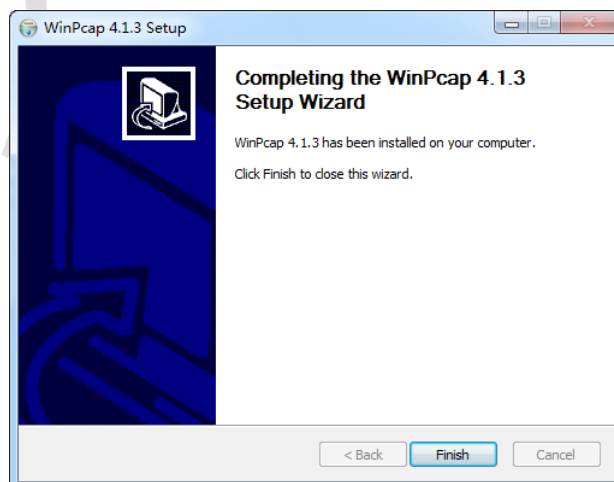


图 1.13 安装 WinPcap 完成

(10) 从该界面可以看到 WinPcap 已安装完成。此时单击 Finish 按钮，将继续安装 Wireshark。安装完成后，将显示如图 1.14 所示的界面。

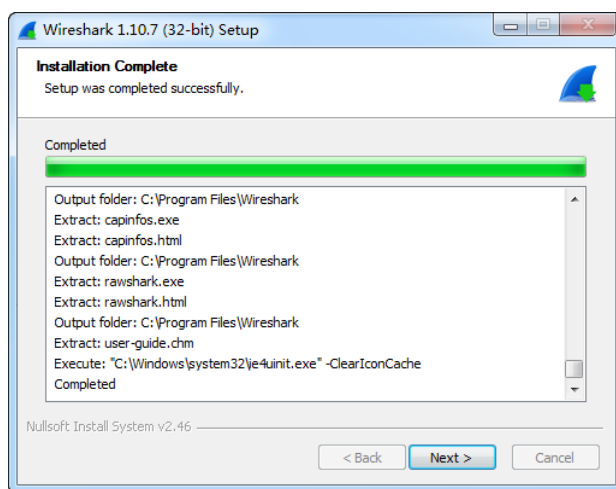


图 1.14 Wireshark 安装完成

(11) 从该界面可以看到 Wireshark 已经安装完成。此时单击 Next 按钮，将显示如图 1.15 所示的界面。

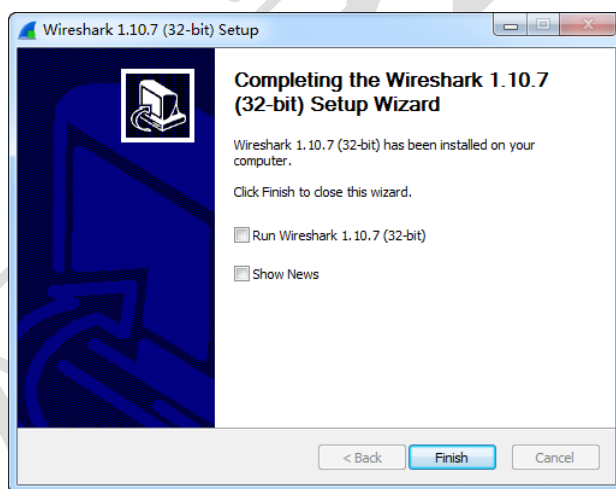


图 1.15 完成页面

(12) 从该界面可以看到 Wireshark 设置向导完成。此时如果想直接启动 Wireshark，则选择 Run Wireshark 1.10.7(32-bit)复选框。然后单击 Finish 按钮，Wireshark 即可启动。

## 2. 在 Linux 系统中安装 Wireshark

**【实例 1-2】**下面演示在 Red Hat Linux 系统中安装 Wireshark。具体操作步骤如下所示：

(1) 从 Wireshark 官网下载 Wireshark 的源码包，其软件名为 wireshark-1.10.7.tar.bz2。

(2) 解压 Wireshark 软件包。执行命令如下所示：

```
[root@localhost ~]# tar jxvf wireshark-1.10.7.tar.bz2 -C /usr/
```

执行以上命令后，Wireshark 将被解压到/usr/目录中。

(3) 配置 Wireshark 软件包。执行命令如下所示：

```
[root@localhost ~]# cd /usr/wireshark-1.10.7/
```



```
[root@localhost wireshark-1.10.7]# ./configure
```

（4）编译 Wireshark 软件包。执行命令如下所示：

```
[root@localhost wireshark-1.10.7]# make
```

（5）安装 Wireshark 软件包。执行命令如下所示：

```
[root@localhost wireshark-1.10.7]# make install
```

以上过程成功执行完后，表示 Wireshark 软件已成功安装。

接下来就可以使用 Wireshark 工具了。在终端输入命令 `wireshark`，启动该工具。如下所示：

```
[root@localhost ~]# wireshark
```

执行以上命令后，将显示如图 1.16 所示的界面。



图 1.16 警告信息

该界面提示当前系统使用 `root` 用户启动了 Wireshark 工具，可能是危险的。如果可以直接单击“确定”按钮启动 Wireshark，如图 1.17 所示。如果不想让该窗口再次弹出，将 `Don't show this message again` 前面的复选框勾上。

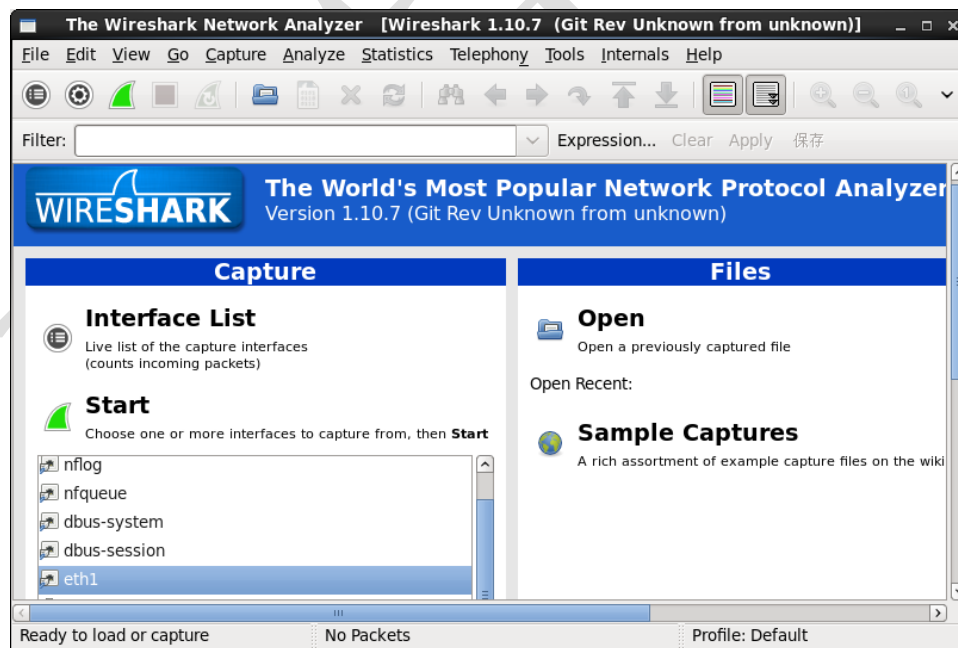


图 1.17 Wireshark 主界面

该界面显示了 Wireshark 的相关信息。该界面显示了 Wireshark 的四部分，由于截图，所以将该界面缩小。每部分内容中的命令，都可以使用鼠标单击打开进行查看。在该界面选择将要捕获数据的接口，单击 `Interface List` 命令将显示或者在 `Start` 命令下的方框中选择接

口，然后单击 **Start** 命令开始捕获数据。

## 1.3 Wireshark 捕获数据

当用户的计算机连接到一个网络时，它依赖一个网络适配器（如一个以太网卡）和链路层驱动（如 Atheros PCI-E 网卡驱动）来发送和接受数据包。Wireshark 为了捕获和分析数据包，也是依赖网络适配器和网卡驱动来传递数据。本节将介绍 Wireshark 捕获数据工作流程。Wireshark 的系统结构，如图 1.18 所示。

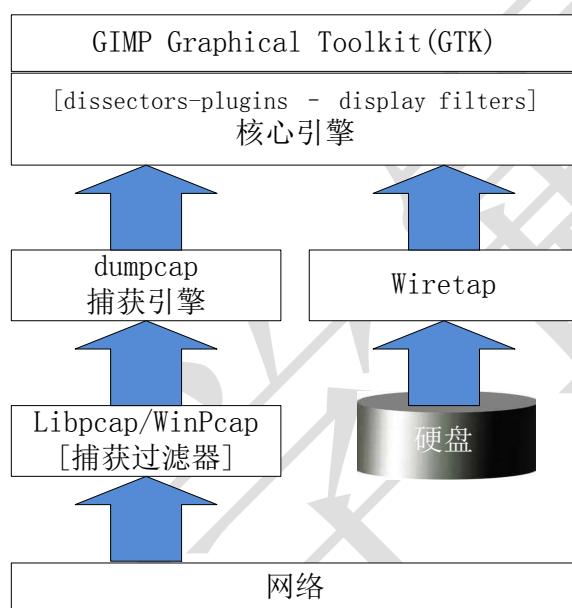


图 1.18 Wireshark 系统结构

在 Wireshark 系统结构中，各模块的功能如下所示：

- (1) **GTK**：图形窗口工具，操控所有的用户输入/输出界面。
- (2) **核心引擎**：将其它模块连接起来，起到综合调度的作用。
- (3) **捕获引擎**：依赖于底层库 Libpcap/WinPcap 库，进行数据捕获。
- (4) **Wiretap** 是用来读取和保存来自于 WinPcap 的捕获文件盒一些其他的文件格式。

在图 1.4 中 Libpcap（WinPcap 是其 Windows 版本）可以提供与平台无关的接口，而且操作简单。它是基于改进的 BPF 开发的。Linux 用户使用 Libpcap，Windows 用户使用 WinPcap。

## 1.5 认识数据包

Wireshark 将从网络中捕获到的二进制数据按照不同的协议包结构规范，显示在 **Packet Details** 面板中。为了帮助用户能够清楚的分析数据，本节将介绍识别数据包的方法。

在 Wireshark 中关于数据包的叫法有三个术语，分别是帧、包、段。下面通过分析一个数据包，来介绍这三个术语。在 Wireshark 中捕获的一个数据包，如图 1.19 所示。每个帧中的内容展开后，与图 1.19 显示的信息类似。



图 1.19 数据包详细信息

从该界面可以看出显示了五行信息，默认这些信息是没有被展开的。各行信息如下所示：

- ❑ Frame：物理层的数据帧概况。
- ❑ Ethernet II：数据链路层以太网帧头部信息。
- ❑ Internet Protocol Version 4：互联网层 IP 包头部信息。
- ❑ Transmission Control Protocol：传输层的数据段头部信息，此处是 TCP 协议。
- ❑ Hypertext Transfer Protocol：应用层的信息，此处是 HTTP 协议。

下面分别介绍下在图 1.19 中，帧、包和段内展开的内容。如下所示：

物理层的数据帧概况

|   |   |
|---|---|
| Frame 5: 268 bytes on wire (2144 bits), 268 bytes captured (2144 bits) on interface 0 | #5 号帧，<br>线路 268<br>字节，实<br>际 捕 获<br>268 字节 |
| Interface id: 0   | #接口 id                                      |
| Encapsulation type: Ethernet (1)  | #封装类型                                       |
| Arrival Time: Jun 11, 2014 09:12:18.469086000 中国标准时间                                  | #捕获日期和时间                                    |
| [Time shift for this packet: 0.000000000 seconds]                                     |   |
| Epoch Time: 1402449138.469086000 seconds  |   |
| [Time delta from previous captured frame: 0.025257000 seconds]                        | #此包与前一包的时间间                                 |
| [Time since reference or first frame: 0.537138000 seconds]                            | #此包与第一帧的时间间                                 |
| Frame Number: 5   | #帧序号  |
| Frame Length: 268 bytes (2144 bits)   | #帧长度  |
| Capture Length: 268 bytes (2144 bits)   | #捕获长度                                       |
| [Frame is marked: False]  | #此帧是否做了标记：否                                 |
| [Frame is ignored: False]   | #此帧是否被忽略：否                                  |
| [Protocols in frame: eth:ip:tcp:http]   | #帧内封装的协议层次结                                 |

|  |            |
|--|------------|
| [Number of per-protocol-data: 2]               | #          |
| [Hypertext Transfer Protocol, key 0]           |            |
| [Transmission Control Protocol, key 0]         |            |
| [Coloring Rule Name: HTTP]                     | #着色标记的协议名称 |
| [Coloring Rule String: http    tcp.port == 80] | #着色规则显示的字  |

字符串

#### 数据链路层以太网帧头部信息

|   |            |
|---|------------|
| Ethernet II, Src: Giga-Byt_c8:4c:89 (1c:6f:65:c8:4c:89), Dst: Tp-LinkT_f9:3c:c0 (6c:e8:73:f9:3c:c0) |            |
| Destination: Tp-LinkT_f9:3c:c0 (6c:e8:73:f9:3c:c0)  | #目标 MAC 地址 |
| Source: Giga-Byt_c8:4c:89 (1c:6f:65:c8:4c:89)   | #源 MAC 地址  |
| Type: IP (0x0800)   |            |

#### 互联网层 IP 包头部信息

|  |             |
|--|-------------|
| Internet Protocol Version 4, Src: 192.168.0.104 (192.168.0.104), Dst: 61.182.140.146 (61.182.140.146)    |             |
| Version: 4   | #互联网协议 IPv4 |
| Header length: 20 bytes  | #IP 包头部长度   |
| Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) | #差          |
| 分服务字段  |             |
| Total Length: 254  | #IP 包的总长度   |
| Identification: 0x5bb5 (23477)   | #标志字段       |
| Flags: 0x02 (Don't Fragment)   | #标记字段       |
| Fragment offset: 0   | #分的偏移量      |
| Time to live: 64   | #生存期 TTL    |
| Protocol: TCP (6)  | #此包内封装的上层协议 |
| 为 TCP  |             |
| Header checksum: 0x52ec [validation disabled]  | #头部数据的校验和   |
| Source: 192.168.0.104 (192.168.0.104)  | #源 IP 地址    |
| Destination: 61.182.140.146 (61.182.140.146)   | #目标 IP 地址   |

#### 传输层 TCP 数据段头部信息

|   |              |
|---|--------------|
| Transmission Control Protocol, Src Port: 51833 (51833), Dst Port: http (80), Seq: 1, Ack: 1, Len: 214 |              |
| Source port: 51833 (51833)  | #源端口号        |
| Destination port: http (80)   | #目标端口号       |
| Sequence number: 1 (relative sequence number)   | #序列号（相对序列号）  |
| [Next sequence number: 215 (relative sequence number)]  | #下一个序列号      |
| Acknowledgment number: 1 (relative ack number)  | #确认序列号       |
| Header length: 20 bytes   | #头部长度        |
| Flags: 0x018 (PSH, ACK)   | #TCP 标记字段    |
| Window size value: 64800  | #流量控制的窗口大小   |
| Checksum: 0x677e [validation disabled]  | #TCP 数据段的校验和 |

## 1.6 捕获 HTTP 包

在网络中，所有数据的通信都是基于 TCP/IP 协议的。HTTP 也是 TCP/IP 协议中的一种，而且该类数据包也是用户通常最关注的。本节将介绍捕获 HTTP 包的一个过程。

捕获 HTTP 包的实验环境如图 1.20 所示。在该环境中，包括一个客户机、两个交换机、

一个标准路由器和一个网络地址转换路由器和服务器。

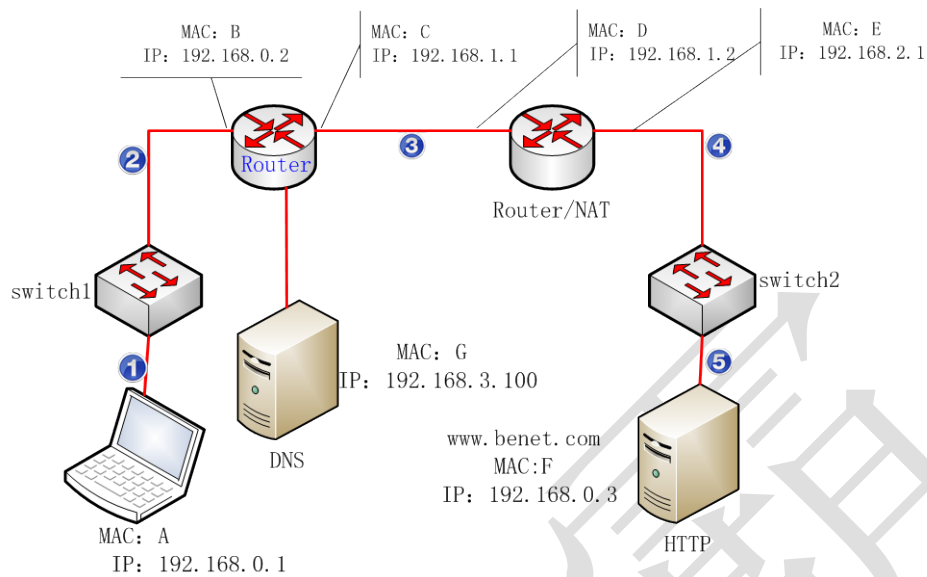


图 1.20 捕获 HTTP 工作流程

(1) 所有设备在 MAC 头部中只能发送本地主机的硬件地址。这个 MAC 头部将沿着第一个路由器的线路剥去，这个 MAC 头部仅临时使用，为了获取包的下一跳。如图 1.21 中，在 IP 头部中，包的地址是从 10.1.0.1（客户端）到 74.125.224.143（服务器）。

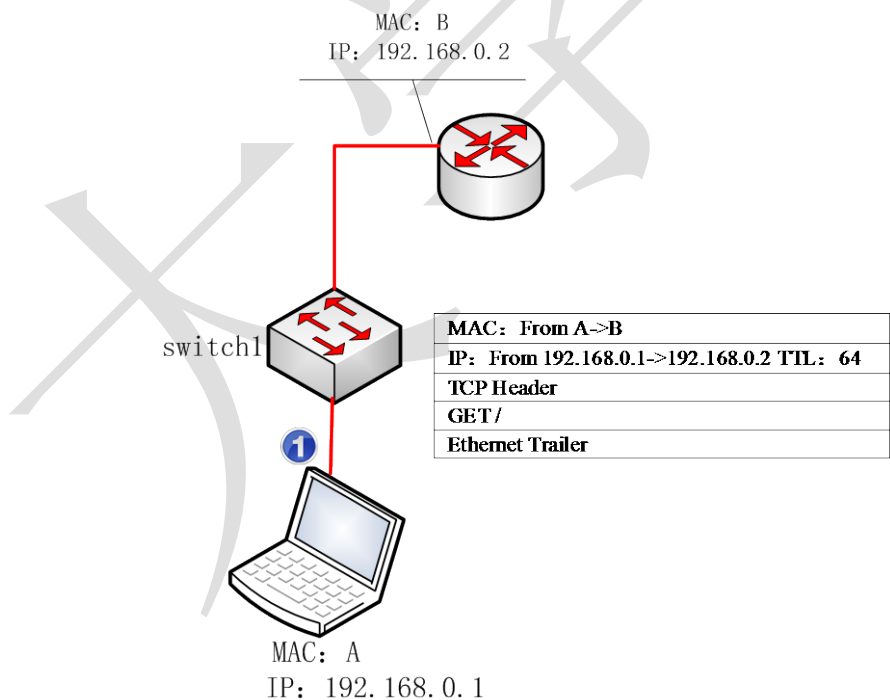


图 1.21 客户端查找本地路由器的 MAC 地址

(2) 真实的交换机不影响数据帧的内容。交换机 1 将简单地查看目标 MAC 地址，为了判断主机是否连接在交换机的其中一个端口上。当交换机找到与 MAC 地址 B 关联的交换端口时，交换机转发数据帧到适当的交换端口，如图 1.22 所示。

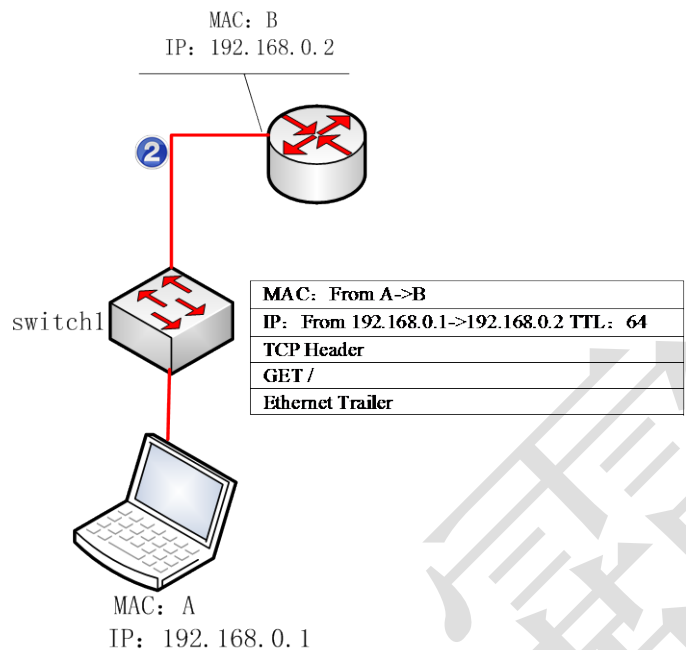


图 1.23 交换机查找关联的端口

(3) 根据数据帧的接收，经过检查确保数据帧不是恶意的，并且数据帧是路由器的 MAC 地址，路由器除去了以太网头部。路由器检查数据包（现在被认为是包，不是帧）的目标 IP 地址，并且查询它的路由表找出如何处理该数据包。如果路由器不知道怎样得到目标 IP 地址（发送的数据包中没有默认网关），该路由器将丢弃该包并发送一个消息返回给发送者。这表明有一个路由问题。用户能使用 Wireshark 捕获这些错误消息，并检查那个路由器不能够将数据转发到目的地。

如果路由器有请求转发数据包的消息时，IP 头部的 TTL（跳数）字段值将减一，如图 1.24 所示。并且应用新的以太网报头的包才将其发送给路由器/NAT 设备。

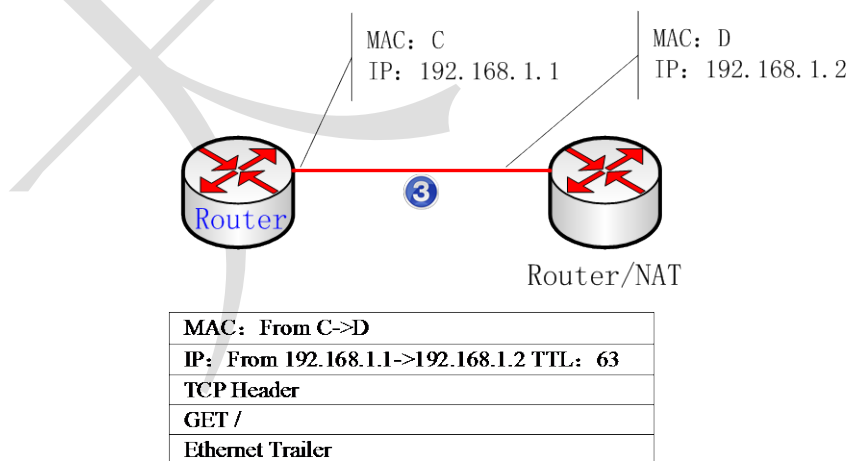


图 1.24 路由器转发数据

(4) 如图 1.25 所示，这个路由器/NAT 设备使用与之前的路由相同的过程转发该数据包。此外，路由器/NAT 设备改变源 IP 地址（网络地址转换）和源端口号，同时注意原始的 IP 地址和源端口号。这个路由器/NAT 设备将这些信息及最近分配出去的 IP 地址和端口号

结合，如图 1.26 所示。

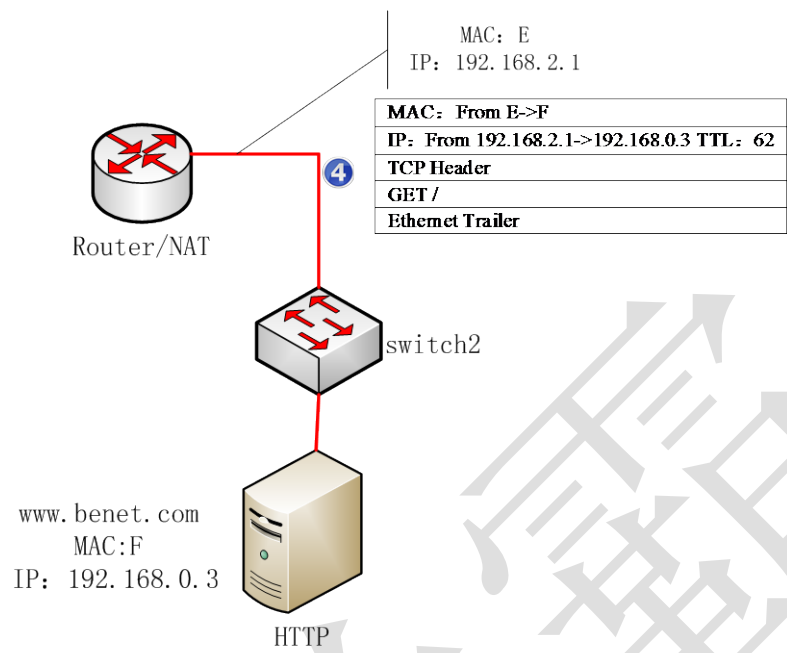


图 1.25 路由器/NAT 设备转发数据

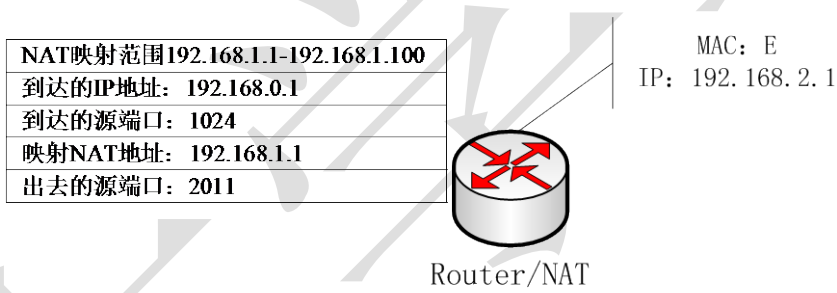


图 1.26 路由器/NAT 结合 IP 和端口信息

(5) 如图 1.27 所示，从该图中可以看到与第 4 步的数据帧相同。因为交换机不能改变数据帧的内容。

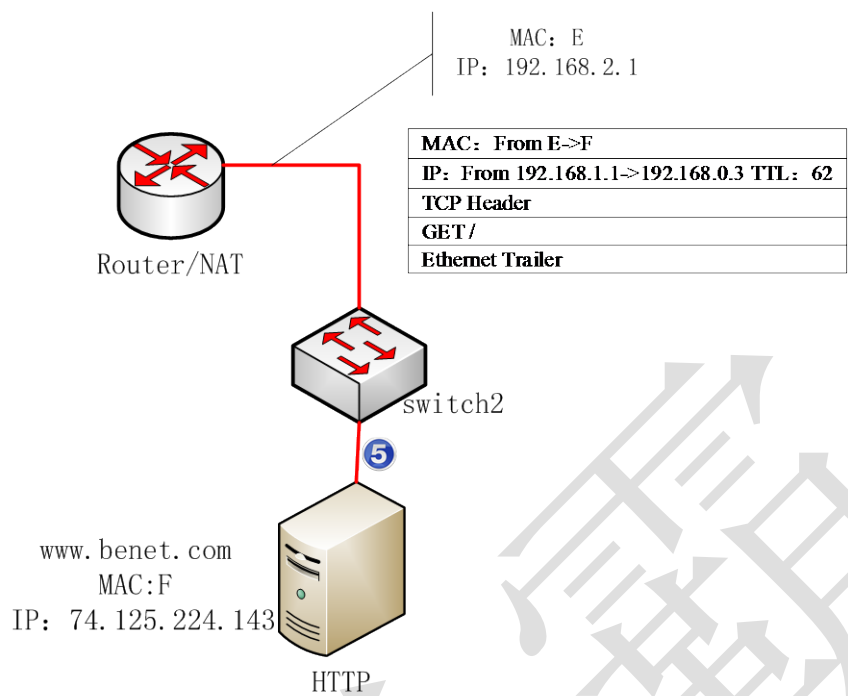


图 1.27 交换机 2 查找匹配的交换端口

## 1.7 访问 Wireshark 资源

在 Wireshark 中可以通过选择 Wiki Protocol Page 命令，访问 Wireshark 相关的信息。用户也可以添加协议或程序名到 URL 中，访问相关连的协议信息。本节将介绍在访问 Wireshark 资源。

启动 Wiki Protocol Page 页面，在 Packet Details 面板中右键单击任何协议即可启动。如图 1.28 所示。



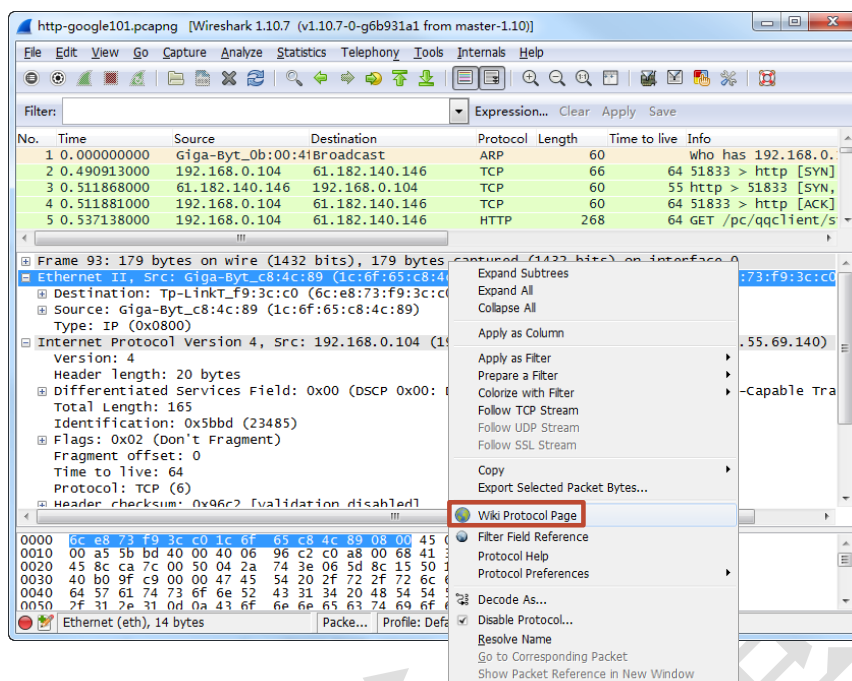


图 1.28 启动 Wiki Protocol Page

在该界面单击 Wiki Protocol Page 命令，将显示如图 1.29 所示的界面。

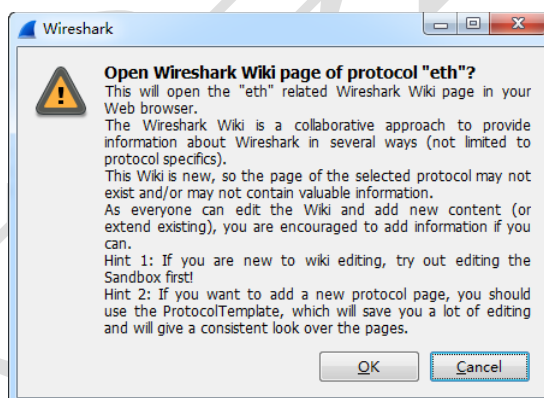


图 1.29 打开“eth”协议页面

该界面提示是否要打开“eth”协议页面。这里单击 OK 按钮，将显示如图 1.30 所示的界面。

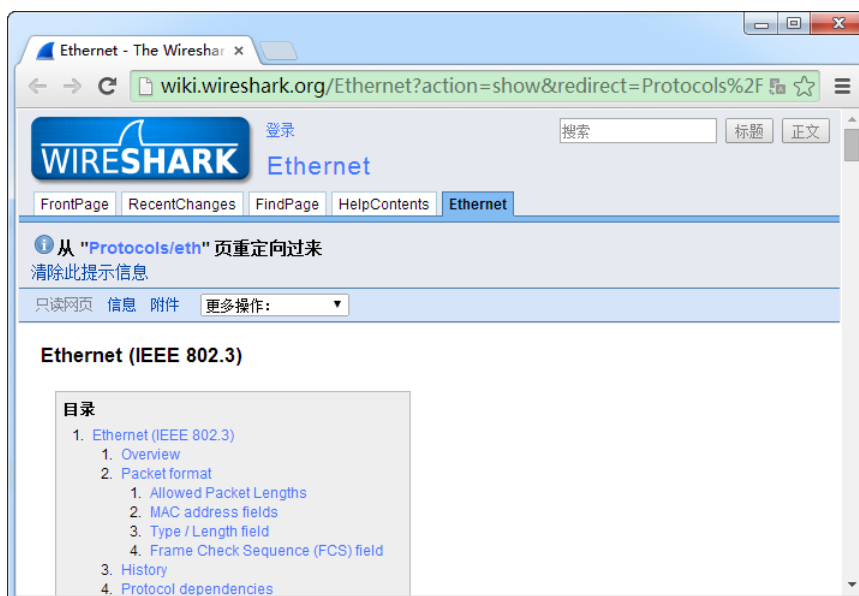


图 1.30 eth 协议页面

从该界面可以看到，此时访问的是 Ethernet 协议页面。

Wireshark 的创始人 Gerald Combs 为 Wireshark 用户开启了一个 Q&A 的论坛，在该论坛上可以提问或回答与 Wireshark 相关的问题。用户可以在 ask.wireshark.org 网站上，讨论与 Wireshark 相同的问题。但是在该网站提问题时，必须要注册一个免费用户。下面将介绍下该论坛中每个区域的作用。

打开 ask.wireshark.org 网站，显示界面如图 1.31 所示。

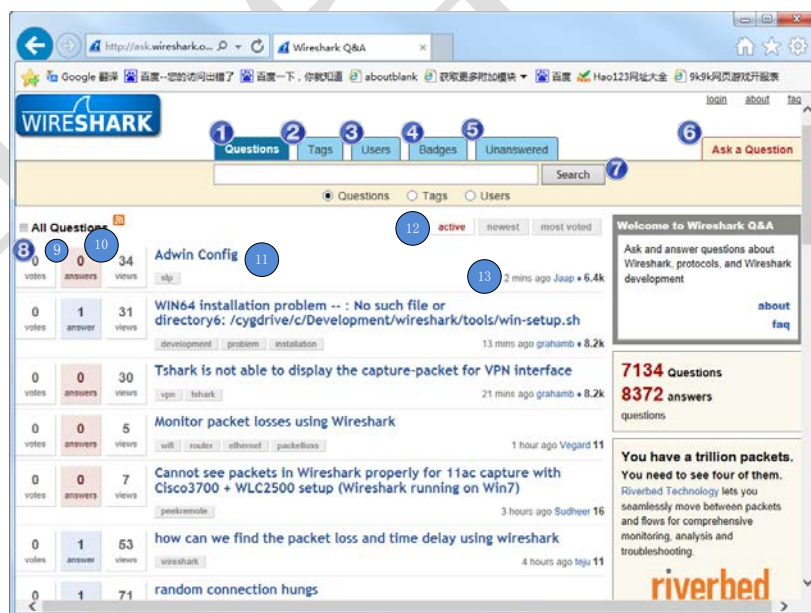


图 1.31 Wireshark 论坛

在该界面使用不同的序号，将每个区域分开。下面分别进行介绍：


- questions 选项卡：单击 questions 后返回所有问题，如图 1.30 所示。
- Tags 选项卡：单击 Tags 选项卡查看 Tags 相关问题的列表——点击 Tags 有关感兴趣的话题，看是否有有帮助的信息。

- ❑ **Users 选项卡**: 单击 Users 选项卡查看参与问答论坛的用户——这个区域还包括他们的地位在徽章中的颜色、数量和行政地位（砖石）。
- ❑ **Badges 选项卡**: 查看在问题解答论坛有多少个参与者参加。
- ❑ **Unanswered 选项卡**: 查看仍然认为是悬而未解决的问题。不幸的是，许多问答参与者不标记问题，即使他们已经“回答”。
- ❑ **Ask a Question 选项卡**: 提问用户的问题。如果在这里没有一个免费的账户，问题将被保存为你创建一个账户，登录新凭证。
- ❑ **Search 按钮**: 搜索用户感兴趣的话题。
- ❑ **Vote 账户**: 论坛用户可以投票表决的问题。
- ❑ **Answer 账户**: 这个数字表明有多少人已经回答了一个问题。
- ❑ **View 账户**: 这个数字表明一个问题已经被浏览的次数。这个可以用来确定最热门的主题。
- ❑ **问题标题和标签**: 单击问题标题跳转到问题页面。该标签包括有问题的主题。
- ❑ **跳转按钮**: 单击任何按钮跳转到活动问题的列表、最新的问题或投票最多的问题。
- ❑ **最后活跃时间**: 这个区域显示一个问题存在多长时间，最近回答问题的用户和最后回答问题的用户。回答问题的用户信息包括业力级别和它们的管理层次。

## 1.8 Wireshark 快速入门

当用户成功地在系统中安装好 Wireshark 后，就可以开始熟悉使用它了。为了帮助用户轻松掌握 Wireshark 的使用，本节将详细介绍 Wireshark 的入门知识。

【实例 1-3】Wireshark 的使用。具体操作步骤如下所示：

(1) 本例中以 Windows 操作系统为例，介绍 Wireshark 的使用。在启动菜单栏中单击 Wireshark 图标，启动该工具。启动界面，如图 1.32 所示。如果已经有捕获好的文件，单击图中的 （打开文件）按钮，选择要打开的捕获文件。

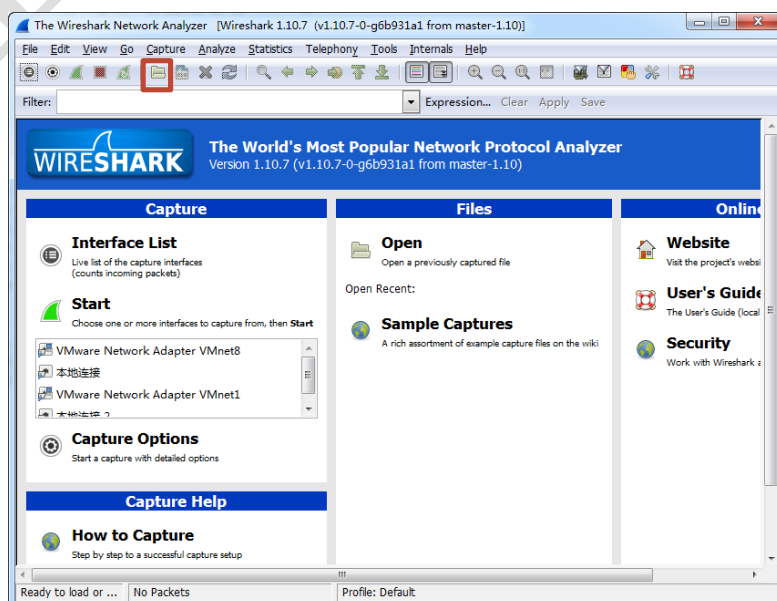


图 1.32 Wireshark 主界面

(2) 在该界面单击 **Interface List** 命令选择接口，如图 1.33 所示。用户也可以在该界面 **Start** 按钮下的方框中，选择接口。然后单击 **Start** 按钮，将开始捕获数据。

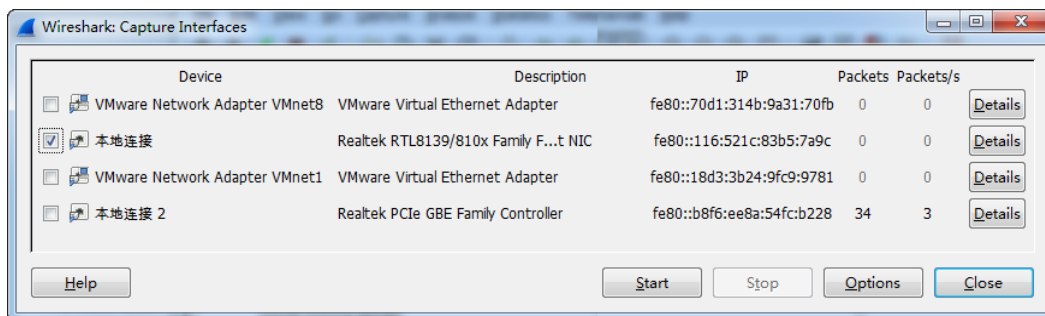


图 1.33 捕获接口列表

(3) 从该界面可以看到，共有四个接口可以捕获数据。这里选择本地连接，然后单击 **Start** 按钮，将显示如图 1.34 所示的界面。

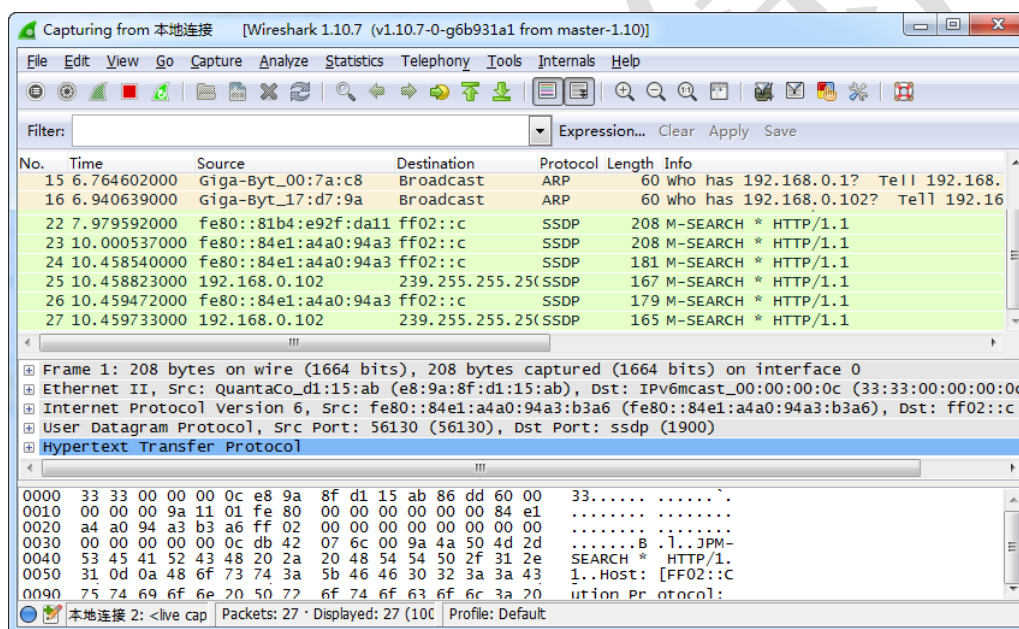



图 1.34 捕获数据过程

(4) 该界面显示了捕获数据的过程。如果要停止捕获，单击 （停止捕获）按钮。该界面就是 **Wireshark** 的主窗口界面，在该界面可以对数据进行各种的操作。如过滤、统计、着色、构建图表等。关于 **Wireshark** 主窗口界面每部分的含义，在第 1.1.1 节已经介绍。下面将分别依次介绍每部分中的作用。

## 1.菜单栏

**Wireshark** 的菜单栏界面如图 1.35 所示。在该界面中被涂掉的两个菜单，在工具栏中进行介绍。

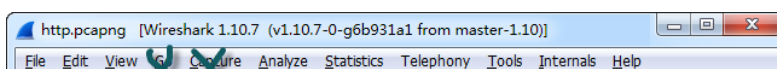


图 1.35 菜单栏

该菜单栏中每个按钮的作用如下所示：

- File: 打开文件集、保存包、导出 HTTP 对象。
- Edit: 清楚所有标记的包、忽略包和时间属性。
- View: 查看/隐藏工具栏和面板、编辑 Time 列、重设颜色。
- Analyze: 创建显示过滤器宏、查看启用协议、保存关注解码。
- Statistics: 构建图表并打开各种协议统计窗口。
- Telephony: 执行所有语音功能（图表、图形、回放）
- Tools: 根据包内容构建防火墙规则、访问 Lua 脚本工具。
- Internals: 查看解析器表和支持协议的列表。
- Help: 学习 Wireshark 全球存储和个人配置文件

## 2.工具栏

当用户详细了解工具栏中每个按钮的作用后，用户就可以快速的进行各种操作。在工具栏中，每个按钮的作用如图 1.36 所示。

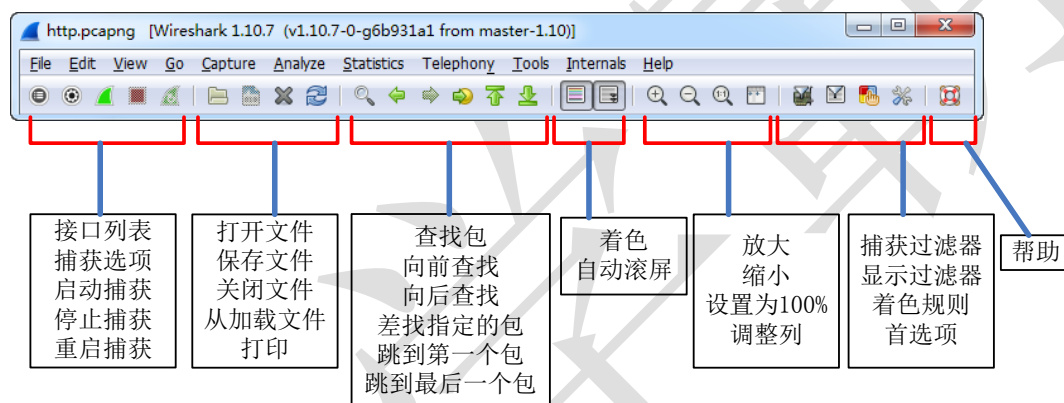


图 1.36 工具栏

## 3.显示过滤器区域

当用户面对大量需要处理的数据时，可以通过使用显示过滤器快速的过滤自己需要的数据。在显示过滤器区域中的每部分作用如图 1.37 所示。

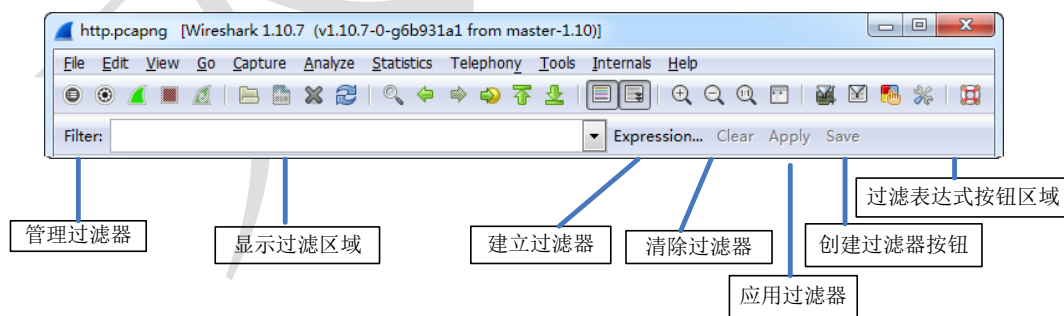


图 1.37 显示过滤器区域

## 4.Wireshark 面板

Wireshark 有三个面板，分别是 Packet List 面板、Packet Details 面板、Packet Bytes 面板。这三个面板的位置，如图 1.38 所示。



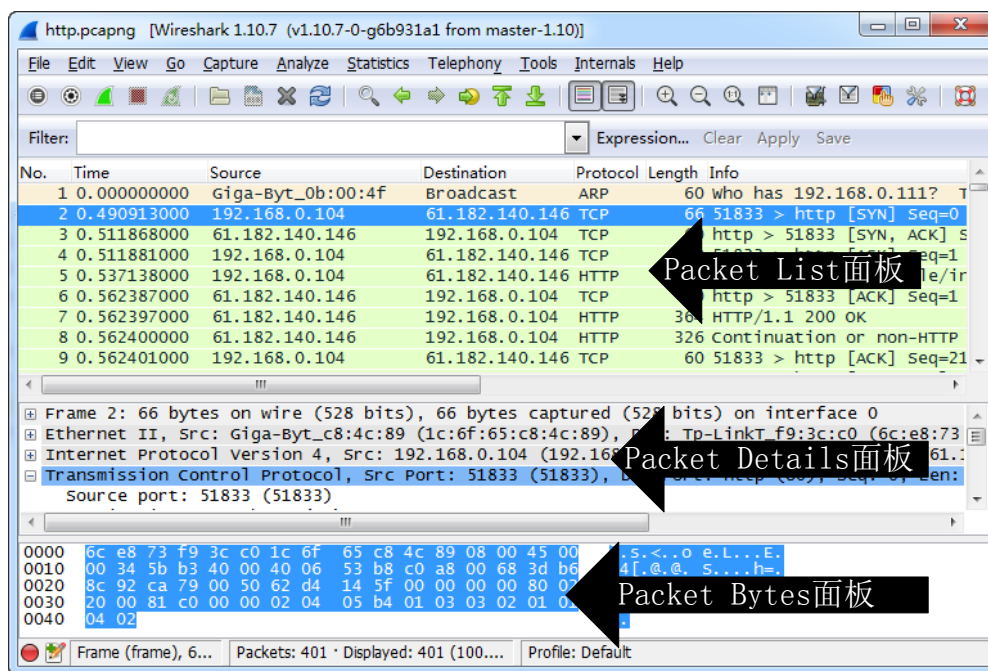


图 1.38 Wireshark 面板

在该界面将三个面板已经标出。这三个面板之间是互相关联的，如果希望在 Packet Details 面板中查看一个单独的数据包的具体内容，必须在 Packet List 面板中单击选中那个数据包。选中该数据包之后，才可以通过在 Packet Details 面板中选择数据包的某个字段进行分析，从而在 Packet Bytes 面板中查看相应字段的字节信息。

下面介绍每个面板的内容。

(1) Packet List 面板：该面板用表格的形式显示了当前捕获文件中的所有数据包。从图 1.38 中，可以看到该面板中共有七列，每列内容如下所示：

- No (Number) 列：包的编号。该编号不会发生改变，即使使用了过滤也同样如此。
- Time 列：包的时间戳。时间格式可以自己设置。
- Source 和 Destination 列：显示包的源地址和目标地址。
- Protocol 列：显示包的协议类型。
- Length 列：显示包的长度。
- Info 列：显示包的附加信息。

在该面板中，可以对面板中的列进行排序、调整列位置、隐藏列、显示列、重命名或删除列等操作。下面以例子的形式将分别介绍在该面板中可操作的功能。

【实例 1-4】演示 Packet List 面板中可实现的功能。如下所示：

### 1. 列排序

打开一个捕获文件 http.pcapng，如图 1.39 所示。

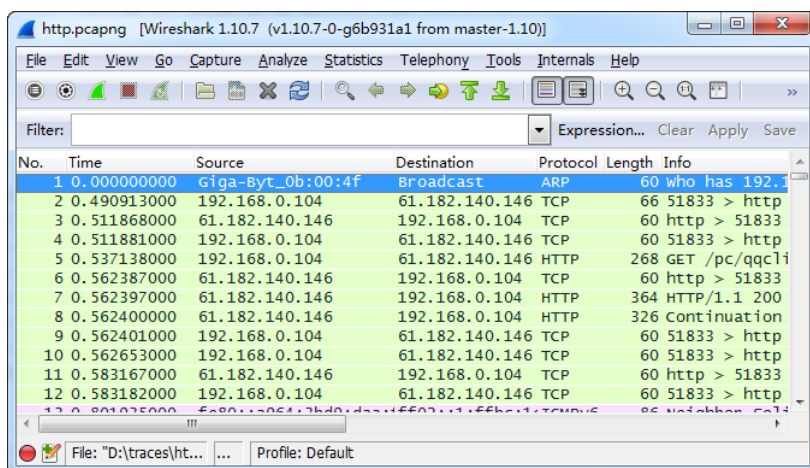


图 1.39 http.pcapng 捕获文件

该界面显示了 http.pcapng 捕获文件中的数据包。默认 Wireshark 是以数据包编号由低到高排序。例如，要对 Protocol 列排序，单击 Protocol 列标题，将显示如图 1.40 所示的界面。

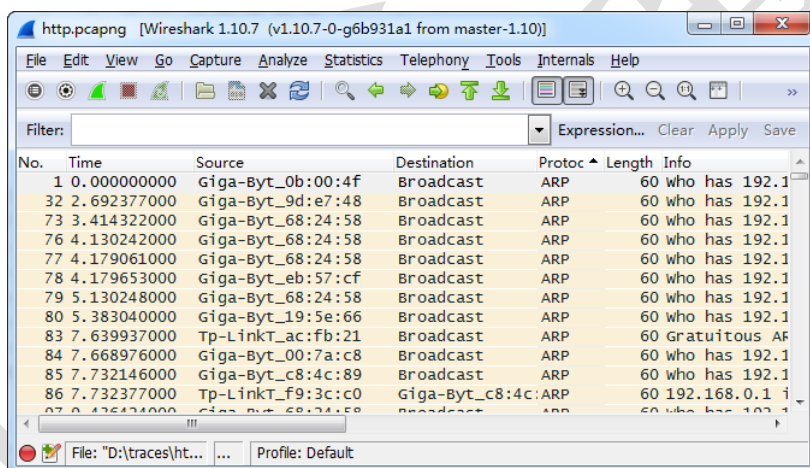


图 1.40 排序 Protocol 列

将该界面与图 1.39 进行比较，可以发现有很大变化。从该界面可以看到 No 列的顺序发生了变化，协议列开始都为 ARP。

## 2. 移动列位置

如移动 http.pcapng 捕获文件中的 Protocol 列，到 Time 后面。使用鼠标选择 Protocol 列，然后拖拽该列到 Time 后面，将显示如图 1.41 所示的界面。

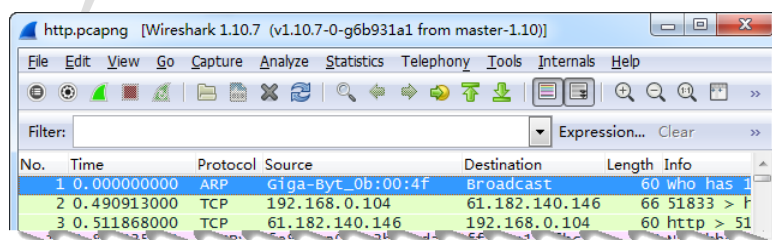


图 1.41 移动 Protocol 列

## 3. 隐藏、显示、重命名、删除列

在捕获文件 http.pcapng 中, 右键单击操作的列标题 (如隐藏 Length 列), 将弹出一个下拉菜单, 如图 1.42 所示。

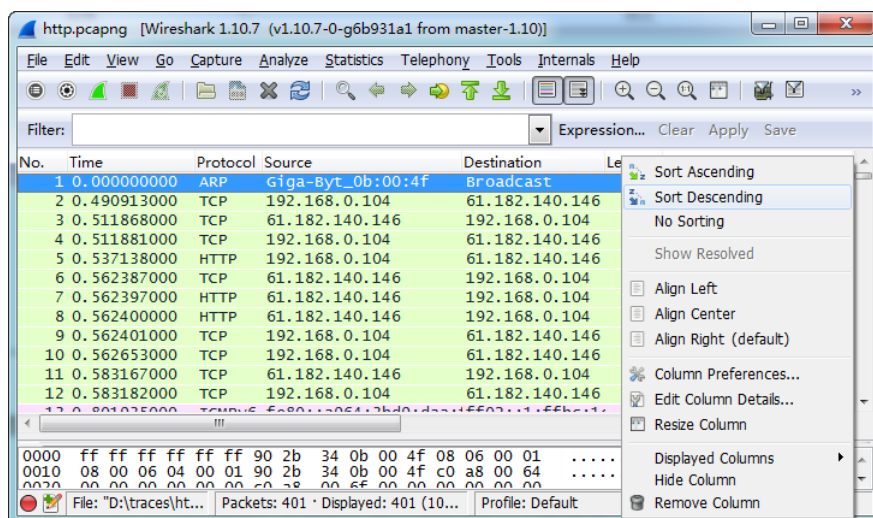


图 1.42 列操作选项

在弹出的菜单中选择 Hide Column 选项。在该菜单中可以选择 Edit Column Details、Displayed Columns 和 Remove Column 选项, 分别做重命名、显示列和删除列操作。

在 Wireshark 中, 还可以对 Packet List 面板中所有数据包进行许多操作。如应用过滤器、着色、重发数据等。用户可以通过右键单击任何一个数据包, 查看可用的选项, 如图 1.43 所示。

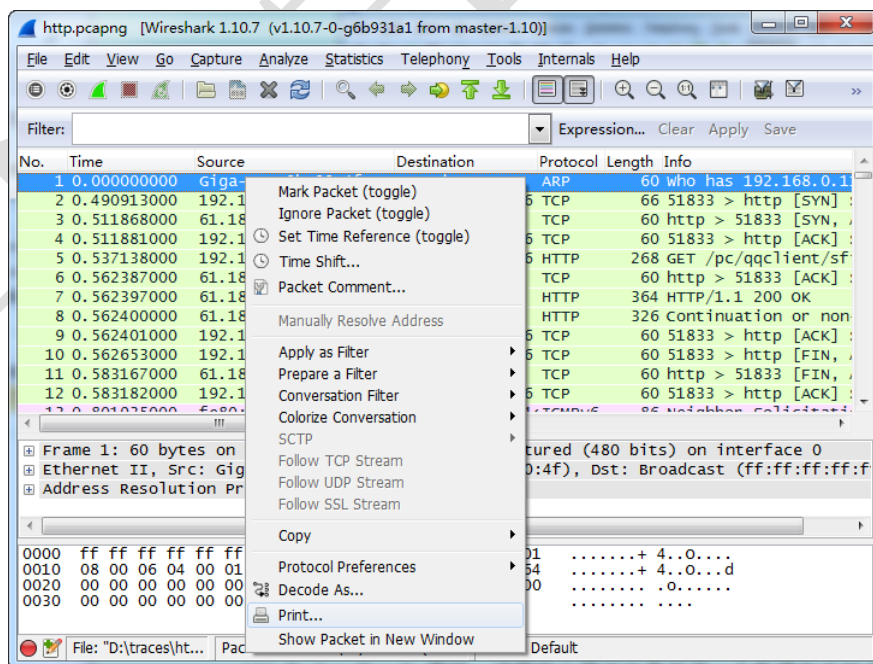


图 1.43 可用选项

在该界面显示了在 Packet List 面板中, 数据包的可用选项。在该选项中, 使用着色功能可以快速的找出有问题的数据包。用户可以改变或创建额外的着色规则, 提醒出现不正常的数据。

(2) Packet Details 面板: 该面板分层次地显示了一个数据包中的内容, 并且可以通过



展开或收缩来显示这个数据包中所捕获到的全部内容。

在 **Packet Details** 面板中，默认显示的数据的详细信息都是合并的。如果要查看，可以点击每行前面的+号展开帧的会话。用户也可以选择其中一行并单击右键，在弹出的菜单中选择 **Expand All** 或 **Expand Subtrees** 展开所有会话或单个会话。展开帧会话，如图 1.44 所示。

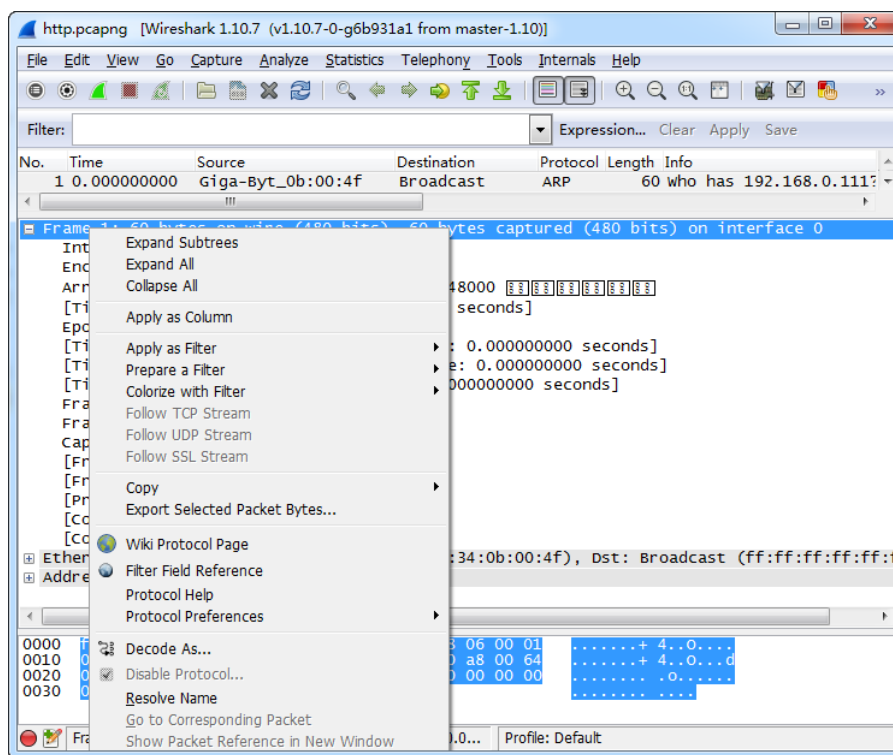


图 1.44 展开单个会话

从该界面可以看到帧会话被展开了。如果要展开所有，则在该菜单中选择 **Expand All** 选项。

(3) **Packet Bytes** 面板：该面板中的内容可能是最令人困惑的。因为它显示了一个数据包未经处理的原始样子，也就是其在链路上传播时的样子。

在该面板中的数据是以十六进制和 ASCII 格式显示了帧的内容。当在 **Packet Details** 面板中选择任意一个字段后，在 **Packet Bytes** 面板中包含该字段的字节也高亮显示。如果不想看到 **Packet Bytes** 面板的话，可以在菜单栏中依次选择 **View|Packet Bytes** 命令将其关闭。当查看的时候，使用同样的方法将其打开。



## 5. 状态栏

状态栏是由两个按钮和三列组成的。其中，这三列的大小在必要时可以调整。状态栏中每部分含义如图 1.45 所示。



图 1.45 状态栏

下面分别详细介绍下状态栏中每部分的作用。如下所示：

- ❑ ：该按钮是专家信息按钮。该按钮的颜色是为了显示包含在专家信息窗口中最高水平的信息。专家信息窗口可以提醒用户，在捕获文件中的网络问题和数据包的注释
- ❑ ：该按钮是捕获文件注释按钮。单击该按钮，可以添加、编辑或查看一个捕获文件的注释。该功能只可以在以.pcapng 格式保存的捕获文件使用。
- ❑ 第一列（获取字段、捕获或捕获文件信息）：当在捕获文件中选择某个字段时，在状态栏中将可以看到文件名和列大小。如果点击 Packet Bytes 面板中的一个字段，将在状态栏中会显示其字段名，并且 Packet Details 面板也在发生着变化。
- ❑ 第二列（包数）：当打开一个捕获文件时，在状态栏中的第二列将显示该文件的总包数。在图 1.29 中，显示了捕获的数据包数量、显示包数和加载时间。如果当前捕获文件中有包被标记，则状态栏中将会出现标记包数。
- ❑ 第三列（当前使用的 Profile）：表示当前使用的 Profile。在图 1.45 中，表示正在使用 Default Profile。Profiles 可以创建，这样就可以自己定制 Wireshark 的环境。

## 1.9 分析网络数据

在 Wireshark 中的数据包，都可以称为是网络数据。每个网络都有许多不同的应用程序和不同的网络涉及。但是一些常见的包中，通常都会包括一些登录程序和网络浏览会话。本节将介绍分析网络数据的方法。

### 1.9.1 分析 Web 浏览数据

通常在访问 Web 服务器过程中，会涉及到 DNS、TCP、HTTP 三种协议。由于此过程中来回发送的数据包较为复杂，所以下面将介绍分析 Web 浏览数据。

【实例 1-5】分析访问 Web 浏览数据。具体操作步骤如下所示：

(1) 捕获访问 www.baidu.com 网站的数据包，并保存该文件名为 http-wireshar.pcapng。本例中捕获的文件如图 1.46 所示。

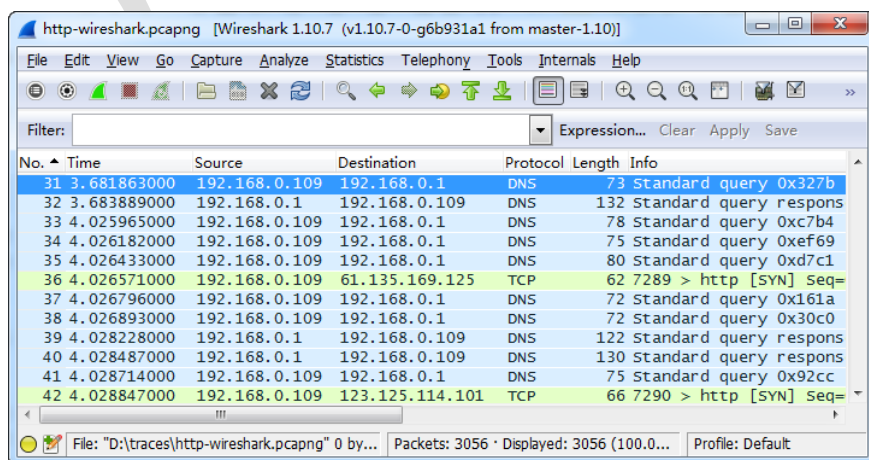


图 1.46 http-wireshar.pcapng 捕获文件

(2) 接下来通过该捕获文件中的数据，分析访问 Web 的整个过程。在该捕获过程中，

将包含 DNS 请求、响应、TCP 三次握手等数据。如图 1.47 所示,在该界面显示了在访问网站之间 DNS 解析过程。

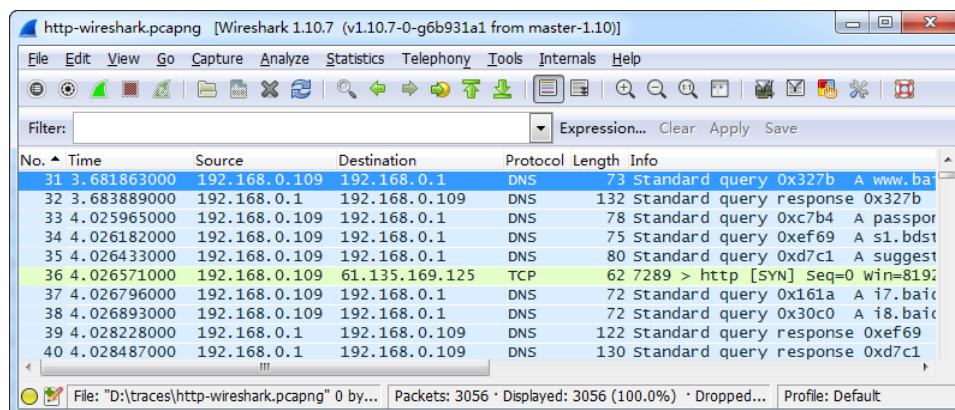


图 1.47 DNS 解析

(3) 在该界面 31 帧,是 DNS 将 www.baidu.com 解析为一个 IP 地址的数据包(被称为一个“A”记录)。32 帧表示返回一个与主机名相关的 IP 地址的 DNS 响应包。如果客户端支持 IPv4 和 IPv6,在该界面将会看到查找一个 IPv6 地址(被称为“AAAA”记录)。此时,DNS 服务器将响应一个 IPv6 地址或混杂的信息。

(4) 如图 1.48 所示,在该界面看客户端和服务器之间 TCP 三次握手(36、63、64 帧)和客户端请求的 GET 主页面(65 帧)。然后服务器收到请求(74 帧)并发送响应包(75 帧)。此时,服务器将发送主页面给客户端(77 帧)。

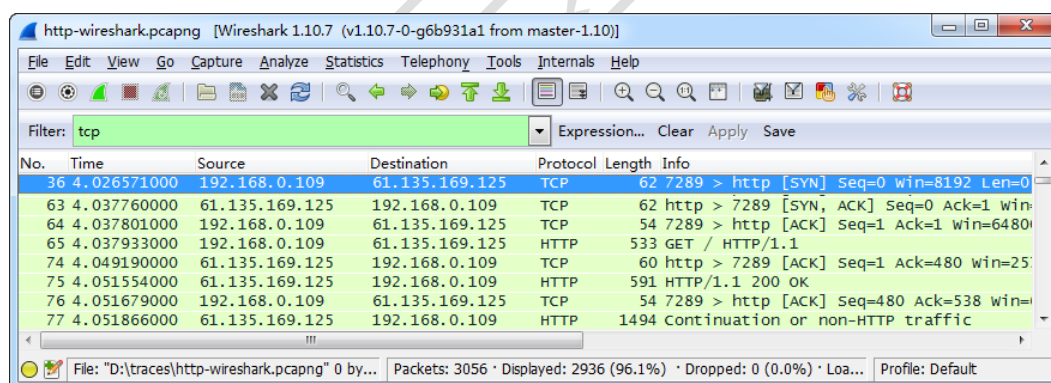


图 1.48 TCP 三次握手

(5) 当客户端从相同的服务器上再次请求访问另一个链接时,将会再次看到一个 GET 数据包(100 帧),如图 1.49 所示。

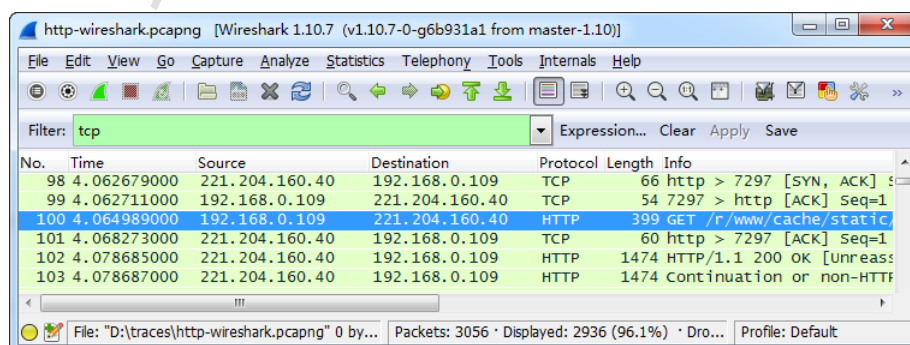


图 1.49 请求另一个元素

此外，如果链接另一个 Web 站点时，客户端将再次对下一个站点进行 DNS 查询（166、167、168、169、170 帧），如图 1.50 所示。

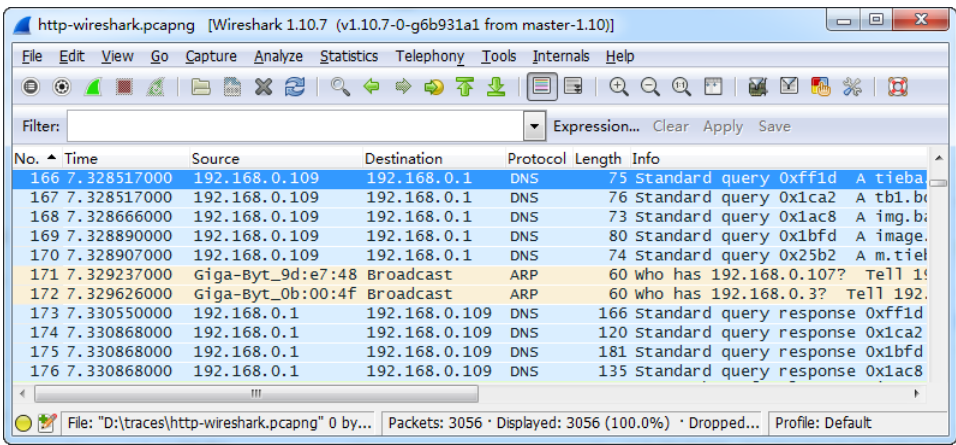



图 1.50 请求下一个站点

### 1.9.2 分析后台数据

后台数据是由操作系统运行自动产生的。在后台数据中可以查看到，Java 查找更新、病毒检测工具查找更新或 Dropbox 检查等。如果对后台数据很熟悉的话，诊断网络问题时就不必要浪费时间在后台程序上了。下面将介绍分析后台数据。

**【实例 1-6】** 捕获并分析 Windows 7 中的后台数据。具体操作步骤如下所示：

- (1) 除 Wireshark 之外，将 Windows 7 中运行的所有程序都关闭。
- (2) 在 Wireshark 的工具栏中单击 （选择捕获接口）按钮，将显示如图 151 所示。

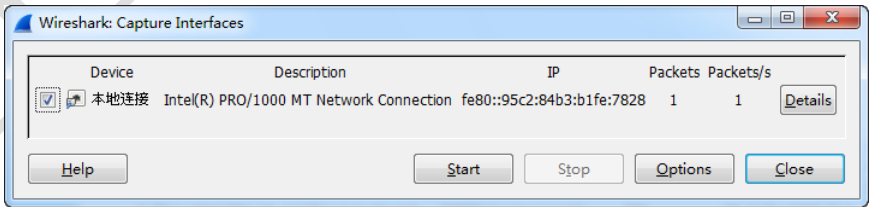


图 1.51 选择捕获接口

- (3) 在该界面选择捕获接口，然后单击 Start 按钮。将显示如图 1.52 所示的界面。

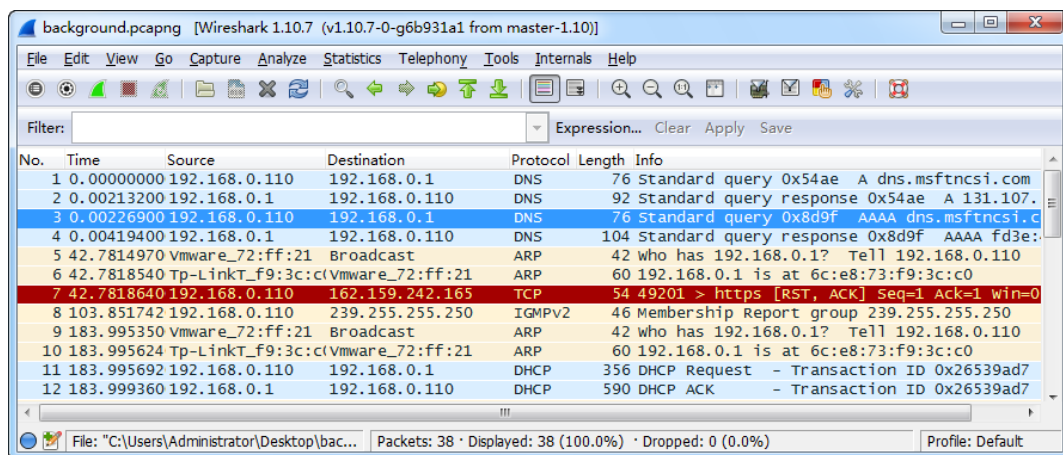


图 1.3652 捕获的后台程序

(4) 等待 Wireshark 捕获几分钟后，单击 Stop Capture 按钮停止捕获。

(5) 如果要保存捕获的文件，在工具栏中单击 Save 按钮，指定保存的位置和文件名。这里将该文件保存为 background.pcapng。

(6) 现在分析 background.pcapng 捕获文件中的后台数据。如下所示：

在该捕获文件中可以看到第 1-4 帧是 DNS 数据包。其中，1、3 帧是 DNS 查询包、2、4 帧是 DNS 响应包。从 1.36 中可以看到，该数据包中第 1 帧是请求查询的 DNS 地址 dns.msftncsi.com，2 帧是响应 DNS 的数据包。3、4 帧和 1、2 帧作用是一样的，唯一不同的是这两个数据包是请求解析为 IPv6 地址，1、2 帧请求解析的地址是 IPv4。该 DNS 查询数据包是由 Windows 系统发送的，用来判断网络是否连通，也就是说是否能够访问网站。如果 dns.msftncsi.com 能解析成功，则说明网络连通正常，也就是捕获文件中的看到的第 2 帧。还有一种判断网络是否连通，可以访问 <http://www.msftncsi.com/ncsi.txt>。如果访问成功，这网络连通正常。

5、6 帧是 ARP 广播包和响应包。之所以捕获到该包，是因为 ARP 表是通过内建的 SNMP 管理的，不管 SNMP 服务是否开启，都会产生 ARP 包。所以，就会有第 6 帧中的 ARP 响应包。

7 帧表示访问 162.159.242.165 网站错误，产生的数据包。

8 帧是 IGMPv2 包。IGMPv2 表示是 IGMP 协议的第二个版本。该协议包含了离开信息，允许迅速向路由协议报告组成员终止情况，这对高带宽组播组或易变型组播成员是非常重要的。从该包的目的地地址可以看到是发送给 239.255.255.250，该地址是一个多播地址。该数据包是由于电脑中可能装了某个播放器插件产生的。

11、12 帧是 DHCP 请求和确认包。之所以有 DHCP 包是因为，本机使用了 DHCP 自动获取的方法。DHCP 是有一定的租期，当租期一到，主机会自动向服务器请求 IP 地址。

## 1.10 打开其它工具捕获的文件

Wireshark 被认为是一个标准的数据包捕获和分析工具，使用它还可以打开其它工具捕获的文件。所以用户需要知道哪些工具可以与 Wireshark 互操作。本节将介绍在 Wireshark 下打开其它工具捕获的文件。

在 Wireshark 下打开一个捕获文件，通常是在菜单栏中依次选项 File|Open 命令，选择



要打开的捕获文件。Wireshark 可以使用 Wiretap 库转化文件的格式。例如，使用 Sun Snoop 工具捕获的文件（该文件后缀名是.snoop），如果要使用 Wireshark 打开时，Wireshark 的 Wiretap 库会执行输入/输出功能处理该数据包。

打开一个捕获文件。在菜单栏中依次选择 File|Open 命令（或单击工具栏中的 File Open 按钮），将显示如图 1.53 所示的界面。

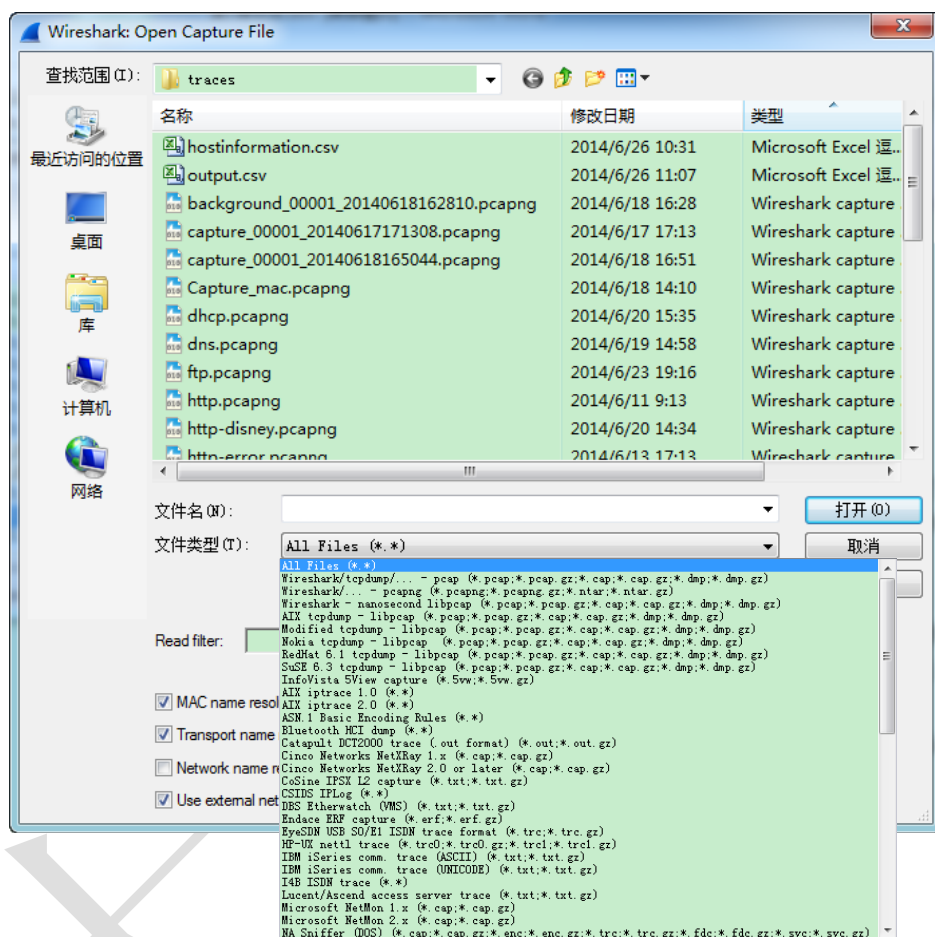


图 1.53 选择文件类型

在该界面可以看到 Wireshark 中支持的所有文件类型。选择相应的文件类型，然后单击“打开”按钮，即可打开捕获文件。