

Wireshark 网络分析实例集锦

（内部资料）

The Wireshark logo, featuring the word "WIRESHARK" in a bold, black, sans-serif font. Above the text is a stylized shark fin icon. The logo is centered on the page and is partially overlaid by a large, light gray watermark that reads "大学霸" (Daxueba) diagonally across the background.

WIRESHARK

大学霸

大学霸——daxueba.net

Wireshark 网络分析实例集锦（大学霸内部资料）

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，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 2 章 设置 Wireshark 视图

Wireshark 默认界面显示了一些基本的信息。如果用户想查看更详细的信息，可以手动设置 Wireshark 的视图。例如，在 Packet List 面板中添加列，设置 Packet Details 面板的显示信息，设置 Profile 等。本章将介绍设置 Wireshark 视图的方法。

2.1 设置 Packet List 面板列

在 Wireshark 的 Packet List 面板中默认包含了几列，如 No、Time、Source、Destination 等。用户也可以手动的添加、删除、隐藏、编辑显示的列。本节将介绍设置 Packet List 面板列。

2.1.1 添加列

Wireshark 默认包含了几列，这些列中显示了基本信息。如果想集中分析一个特定问题，往往需要使用添加列的方法来实现。添加列可以帮助用户快速的查看到所需要的信息。添加列有两种方法。下面分别进行介绍。

1. 第一种方法

第一种添加列的方法展开 Packet Details 面板中的数据包，在展开的数据包中右键单击某个字段，然后选择 Apply as Column 命令即可添加该列。在 Packet Details 面板中，显示了数据帧中包含的字段和值。操作方法如下：

- （1）打开一个捕获文件（名为 http.pcapng）。
- （2）在 Packet Details 面板中，右键单击 Internet Protocol 会话，将显示一个菜单栏。在该菜单栏中，单击 Expand All 命令显示整个帧中的所有字段。
- （3）选择其中一个字段，这里选择 Time to live。右键单击该字段，并选择 Apply as Column 命令，将显示如图 2.1 所示的界面。

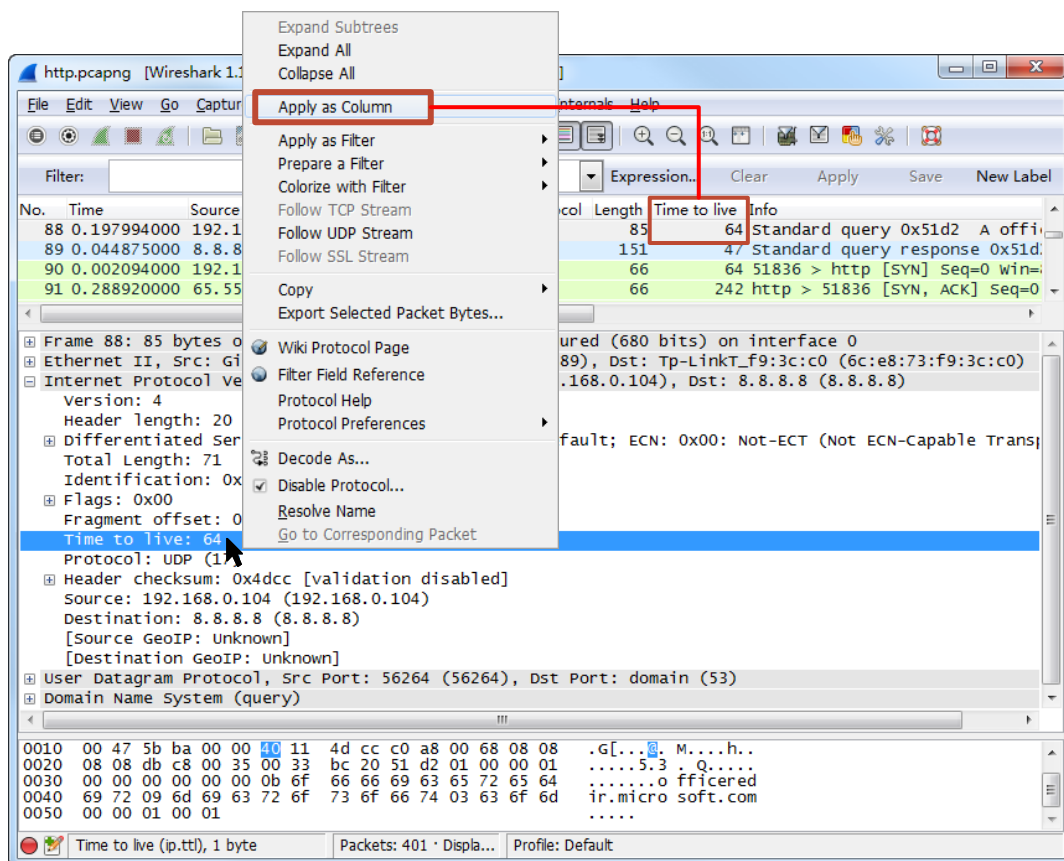


图 2.1 添加列的方法

从该界面可以看到，在 Wireshark 的界面增加了一列 Time to live。

2.第二种方法

如果包中不包含用户想要添加的字段时，使用第一种方法将无法实现。这就需要使用第二种方法来实现。在 Wireshark 的菜单栏中，依次选择 Edit|Preferences|Columns 命令，将显示如图 2.2 所示的界面。

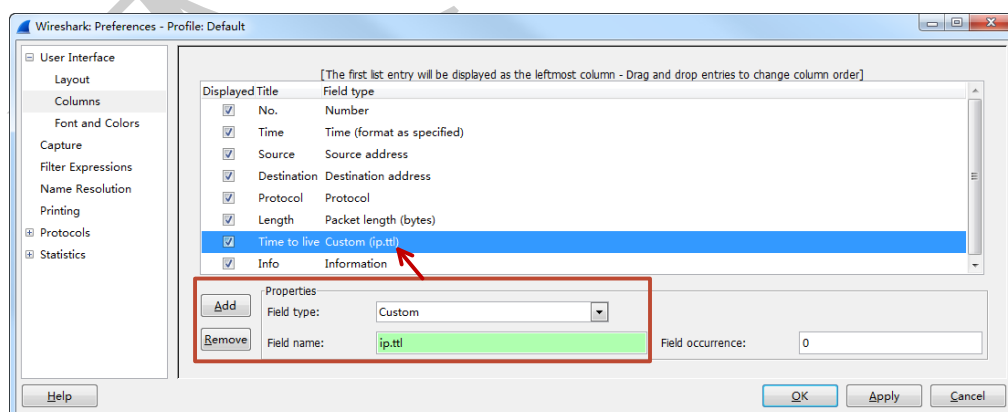


图 2.2 创建列

从该界面中，可以看到 Wireshark 已存在的列。此时可以单击 Field type 区域中的任何字段，通过鼠标拖拽来调整列的顺序，并添加列。在该界面显示的标题也可以重命名，单击鼠标即可修改默认的标题名。如果要添加列，单击 Add 按钮。

2.1.2 隐藏、删除、重新排序及编辑列

用户可以在首选项窗口对列进行各种操作，如隐藏列、删除列、编辑列等。将鼠标靠近 Packet List 面板中的列窗口，右键单击某一列，就可以实现编辑列标题、暂时隐藏（或显示）列或删除列。使用鼠标向左向右拖动窗口，可以对这些列重新排序。本节将介绍对列的操作。

1. 隐藏列

如果当前不需要分析某一列的信息时，就可以将该列隐藏。例如要隐藏 Time to live 列，在 Wireshark 主界面的 Packet List 面板中单击 Time to live 列，将显示如图 2.3 所示的界面。

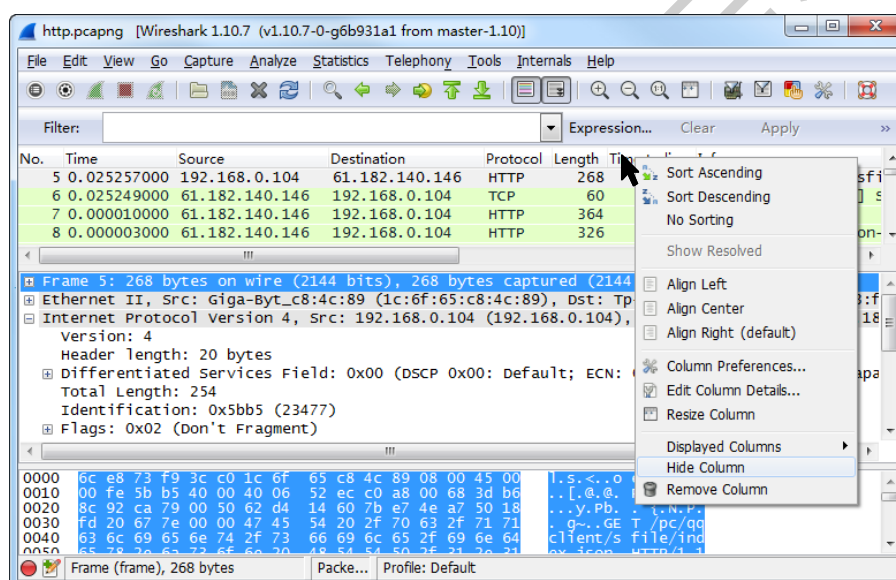


图 2.3 隐藏列

从该界面可以看到显示了该列中所有可用的选项。这里，单击 Hide Columns 命令。此时 Time to live 列就隐藏了，如图 2.4 所示。

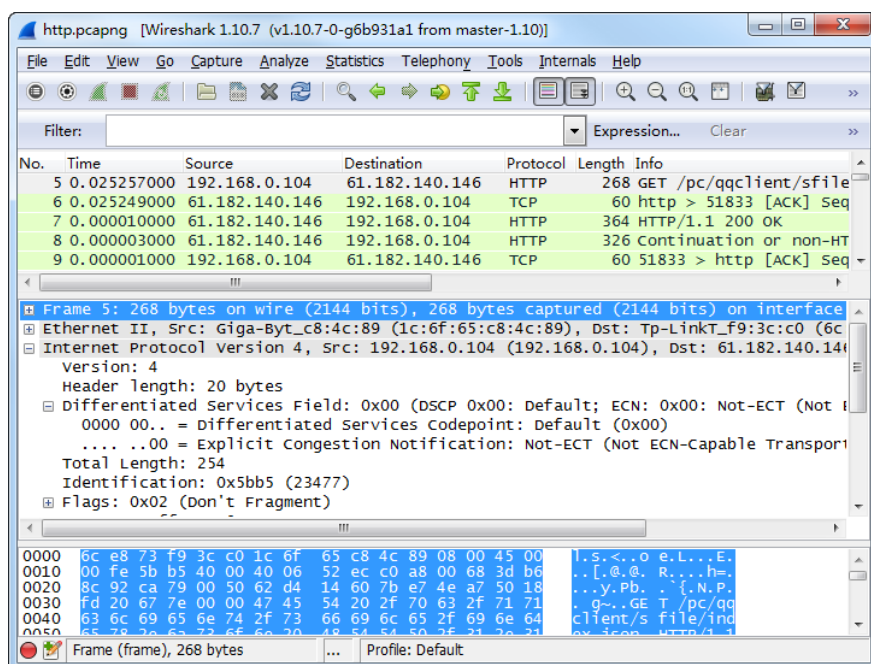


图 2.4 隐藏 Time to live 列

从该界面可以看到 Time to live 列，不存在了。这说明该列已隐藏。当使用该列时，可以单击 Displayed Columns 命令，选择隐藏的列。

2. 删除列

如果不再使用 Time to live 列时，可以将该列删除。在如图 2.3 所示的菜单栏中，选择 Remove Column 命令，将显示如图 2.5 所示的界面。

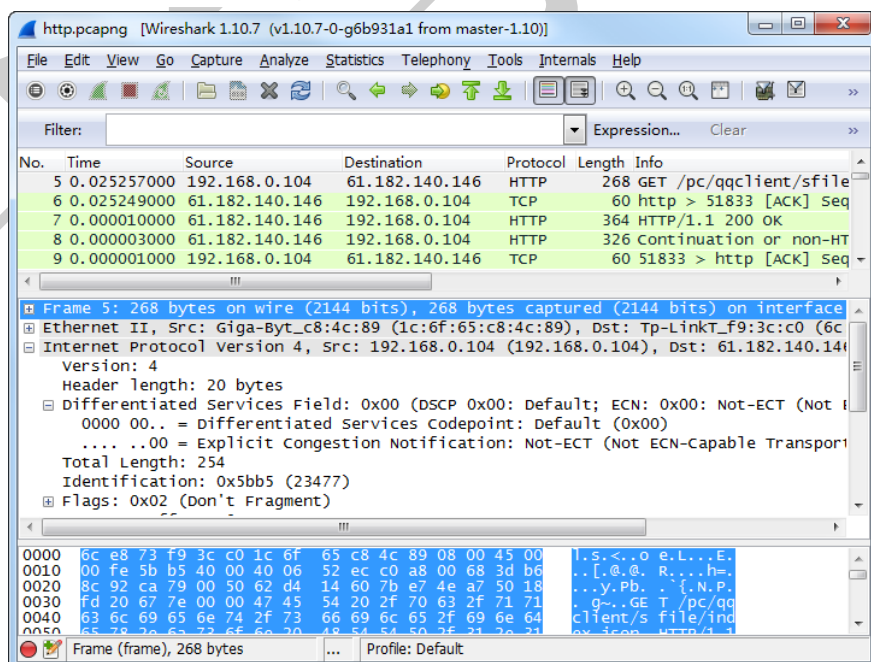


图 2.5 删除 Time to live 列

从该界面可以看到 Time to live 列已经被删除。这里可以看到该界面与图 2.4 的界面一样。但是实际上是不同的，因为删除该列后，在 Displayed Columns 中将不会存在。如果想

再查看该列时，需要重新创建才可以。

3.排序列内容

通过排序列可以使用户更快的分析数据。这里以 http.pcapng 文件为例，对 Time to live 列内容进行排序。打开该文件后，使用鼠标单击 Time to live 列标题就可以了。对该列排序后，该列的数据将从底到高排序，如图 2.6 所示。如果再次单击该列，将从高到底排序。

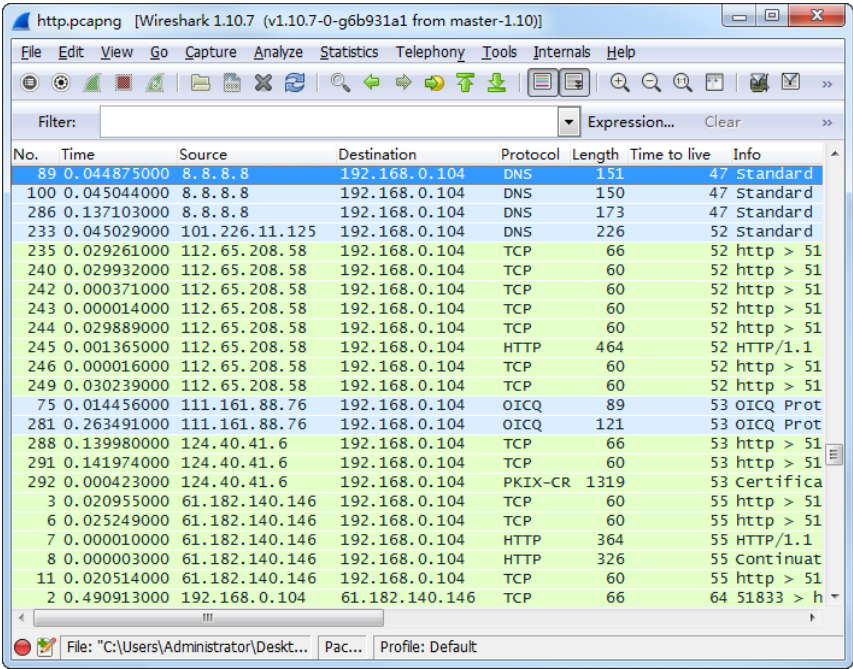


图 2.6 排序 Time to live 列的内容

从该界面可以看到 Time to live 列，是从低到高排序。此时将鼠标滚动到 Wireshark 的顶部，可以看到此时捕获文件中最低的 TTL 值是 47。

4.编辑列

在图 2.3 的菜单栏中选择 Edit Column Details 命令，将显示如图 2.7 所示的界面。

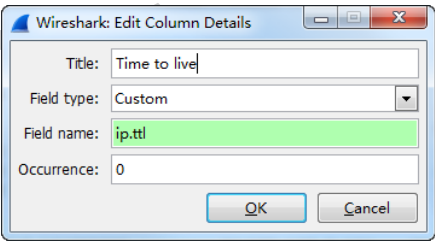


图 2.7 编辑列页面

这里将标题 Time to live 修改为 live，然后单击 OK 按钮，将显示如图 2.8 所示的界面。

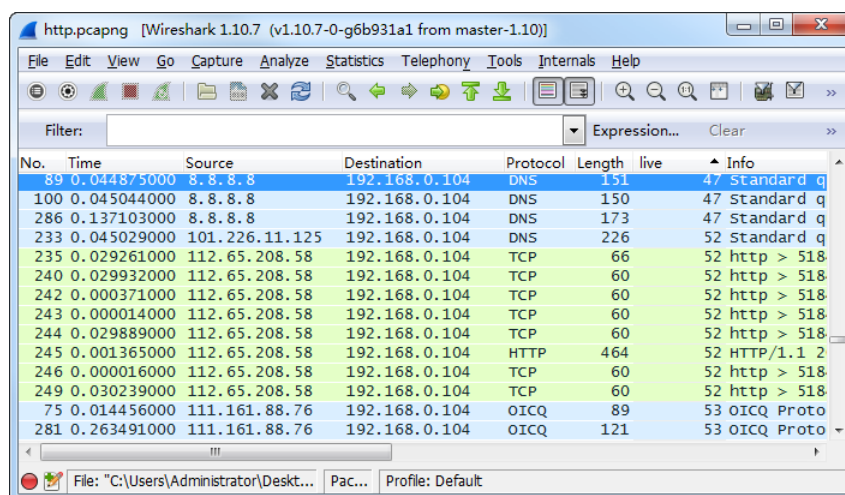


图 2.8 编辑列

从该界面可以看到原来的标题列 Time to live 已经变成 live。

5. 输出列数据

如果要想使用其它工具分析 Wireshark 捕获的数据，可以将 Wireshark 的列数据输出。用户添加列到包列表窗口，然后输出列数据。

例如，如果想要输出 Time to live 列的内容，可以选择 File|Export Packet Dissections|as "CSV"(Comma Separated Values packet summary)file...命令，如图 2.9 所示。然后，将这些数据保存到本地磁盘的一个文件中。

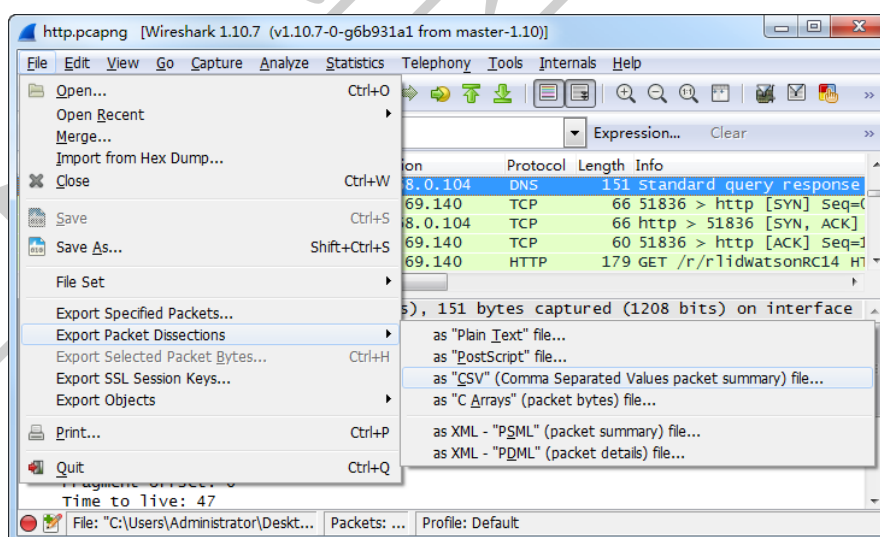


图 2.9 输出列数据

保存的文件中，显示了所有列中的数据。这些列之间使用逗号分隔，此时打开这个 CSV 文件将可以进一步操作数据。例如，使用 Microsoft Excel 打开 CSV 文件，将显示如图 2.10 所示的界面。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.490913	192.168.0.104	61.182.140.146	TCP	60	51833 > http [SYN]
2	0.511868	61.182.140.146	192.168.0.104	TCP	60	http > 51833 [SYN]
3	0.511881	192.168.0.104	61.182.140.146	TCP	60	51833 > http [ACK]
4	0.537138	192.168.0.104	61.182.140.146	HTTP	268	GET /pc/qqclient/s
5	0.562387	61.182.140.146	192.168.0.104	TCP	60	http > 51833 [ACK]
6	0.562397	61.182.140.146	192.168.0.104	TCP	364	[TCP segment of a
7	0.562401	192.168.0.104	61.182.140.146	HTTP	326	HTTP/1.1 200 OK (z
8	0.562401	192.168.0.104	61.182.140.146	TCP	60	51833 > http [ACK]
9	0.562653	192.168.0.104	61.182.140.146	TCP	60	51833 > http [FIN]
10	0.583167	61.182.140.146	192.168.0.104	TCP	60	http > 51833 [FIN]

图 2.10 CSV 文件

从该界面可以看到，在 Wireshark 中的每列信息。

【实例 2-1】下面演示添加 HTTP 协议的 Host 字段作为一列。具体操作步骤如下所示：

- (1) 在 Wireshark 的工具栏中单击 （打开一个捕获的文件）按钮，打开 http.pcapng 文件。
- (2) 在包列表窗口中，向下滚动鼠标选择 59 帧。
- (3) 在包详细窗口中显示了 59 帧中的详细内容。在包详细窗口中单击 Hypertext Transfer Protocol 前面的加号 (+)，展开 59 帧的会话，如图 2.11 所示。

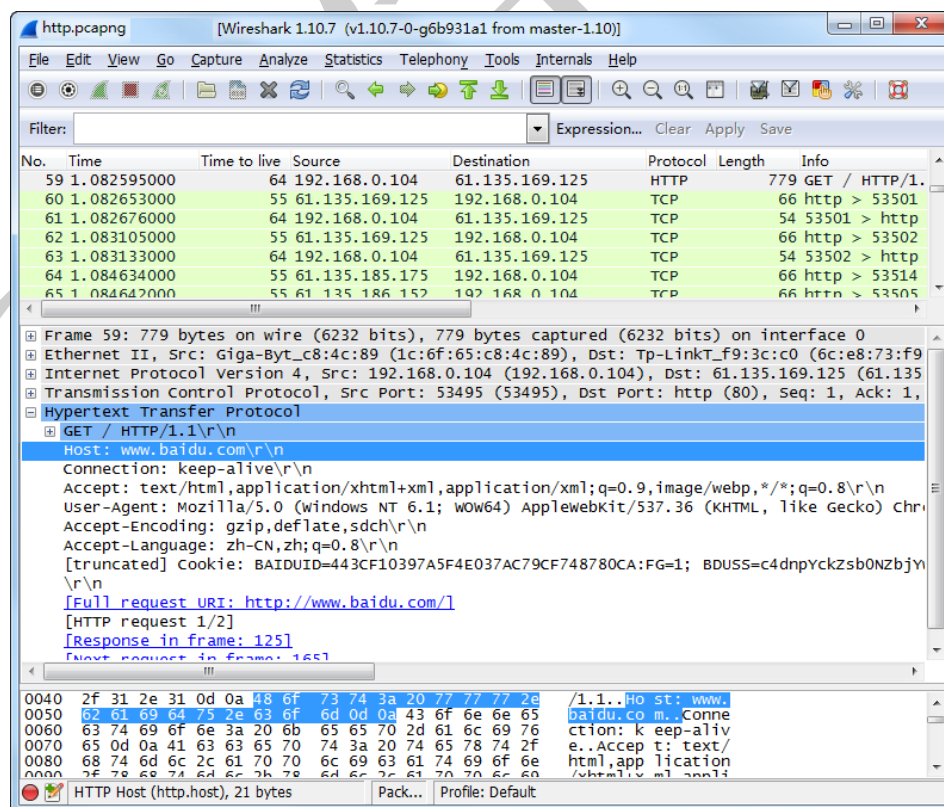


图 2.11 59 帧中的详细内容

(4) 在该界面右键单击 Host 行（包含 www.baidu.com\r\n），并选择 Apply as Column 命令，Host 列将被添加到 Info 列的左边，如图 2.12 所示。此时可以通过单击并拖动列，将列的边缘扩大。

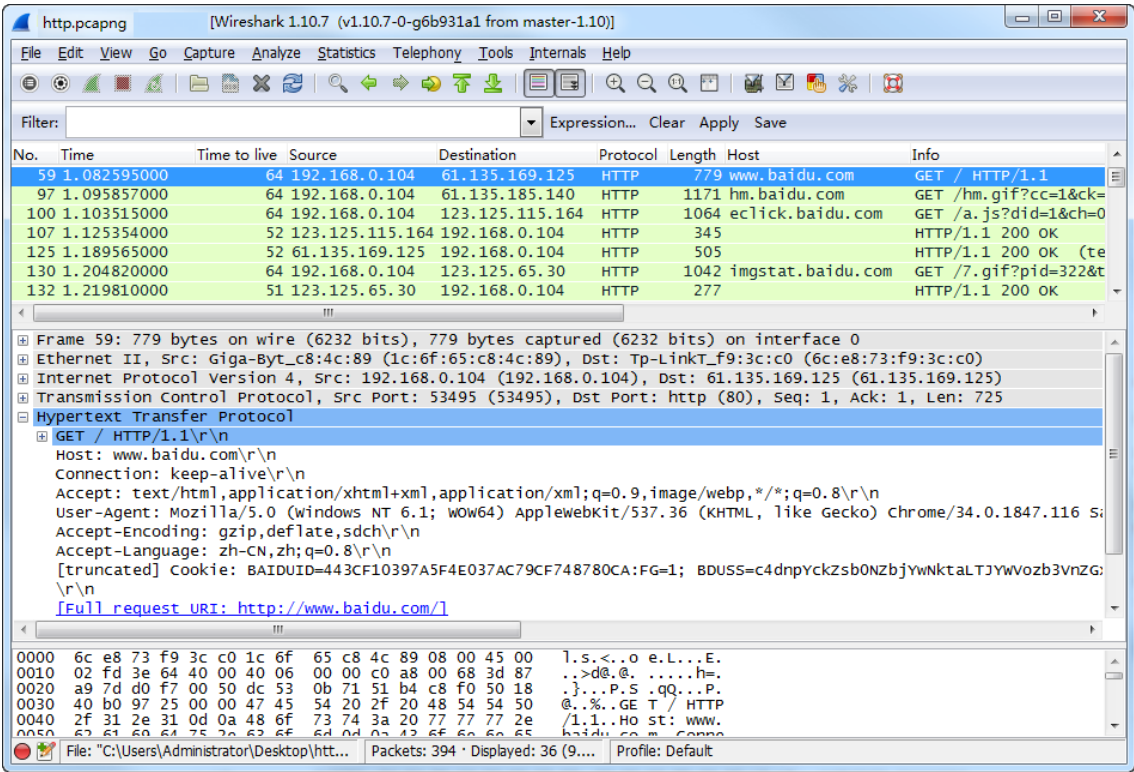


图 2.12 添加的 Host 列

(5) 从该界面可以看到 Host 列被成功添加。此时，可以单击 Host 列进行排序。如果想要查看所有主机客户端发送的请求，可以单击工具栏中的 （跳转到第一个包）按钮，将显示如图 2.13 所示的界面。

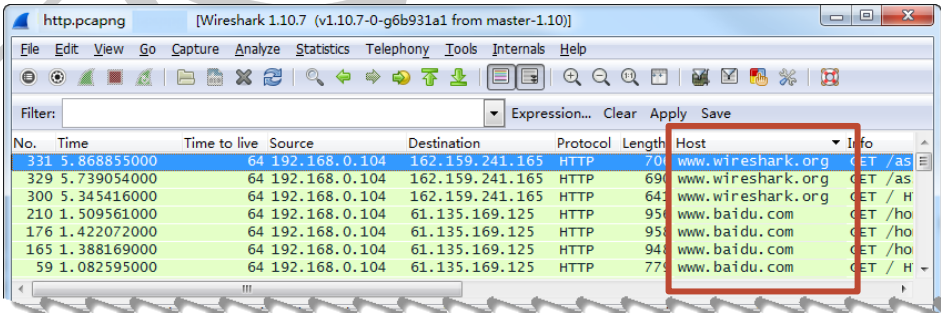


图 2.13 所有发送请求的客户端

(6) 该界面显示所有发送 HTTP 请求的客户端，这样就不用滚动鼠标一个个查找发送 HTTP 请求的客户端。如果想将 Host 列隐藏的话，右键单击 Host 列并选择 Hide Column 即可实现。当再次查看时，单击右键 Host 列并选择 Displayed Columns |Host (http.host) 命令，Host 列将再次被添加到 Wireshark 的窗口列表中。

2.2 Wireshark 分析器及 Profile 设置

Wireshark 分析数据的过程中，通常会经过五个分析器。在 Wireshark 中的所有配置都保存在 Profile 中，Profile 实际是一个目录。用户可以手动创建和切换 Profile。本节将介绍 Wireshark 的分析器及 Profile 的相关设置。

2.2.1 Wireshark 分析器

分析包是 Wireshark 最强大的功能之一。分析数据流过程就是将数据转换为可以理解的请求、应答、拒绝和重发等。帧包括了从捕获引擎或监听库到核心引擎的信息。Wireshark 中的格式由成千上万的协议和应用程序使用，它可以调用各种各样的分析器，以可读的格式将字段分开并显示它们的含义。下面将介绍详细分析 Wireshark 的包信息。

例如，一个以太网网络中的主机向 Web 网站发送 HTTP GET 请求时，这个包将由五个处理器进行处理。分别如下所示：

1. 帧分析器

帧分析器用来检测和显示捕获文件的基本信息，如每个帧的时间戳，如图 2.14 所示。然后帧分析器传递帧给以太网分析器。

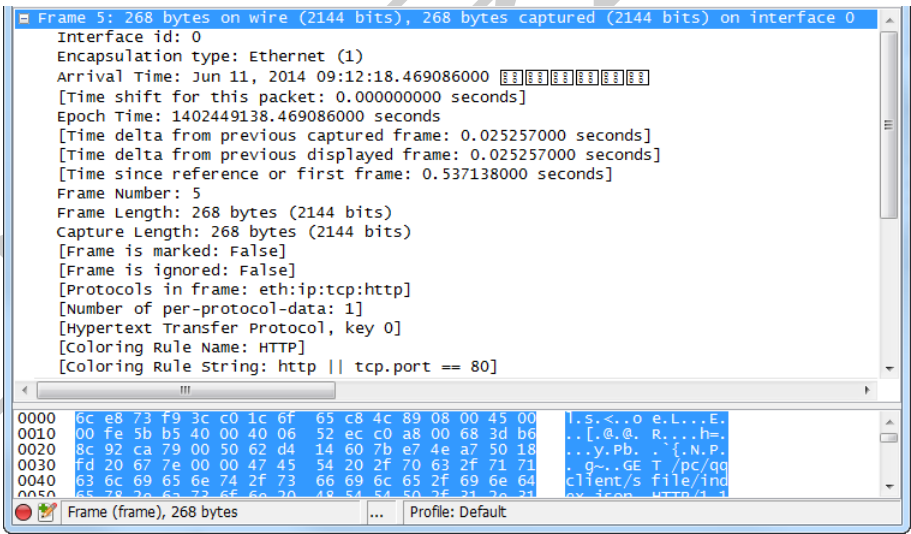


图 2.14 帧分析器

从该界面可以看到第 5 帧中的一些基本信息。例如，帧的编号为 5（捕获时的编号），帧的大小为 268 个字节，帧被捕获的日期和时间，该帧和前一个帧的捕获时间差以及和第一个帧的捕获时间差等。

2. 以太网分析器

以太网分析器用来解码、显示以太网帧（Ethernet II）头部的字段、字段类型的内容等。然后传递给下一个分析器，也就是 IPv4 分析器。如图 2.15 所示，该字段类型值为 0x0806，0x0806 表示是一个 IP 头部。

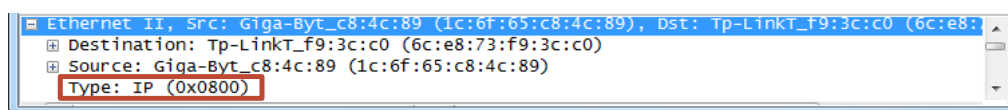


图 2.15 以太网分析器

从该界面可以看到在以太网帧头部中封装的信息，包括发送方的源 MAC 地址和目标 MAC 地址。

3.IPv4 分析器

IPv4 分析器用来解码 IPv4 头部的字段，并基于协议字段的内容传递包到下一个分析器。如图 2.16 所示，该界面显示了 IPv4 分析器中的内容。

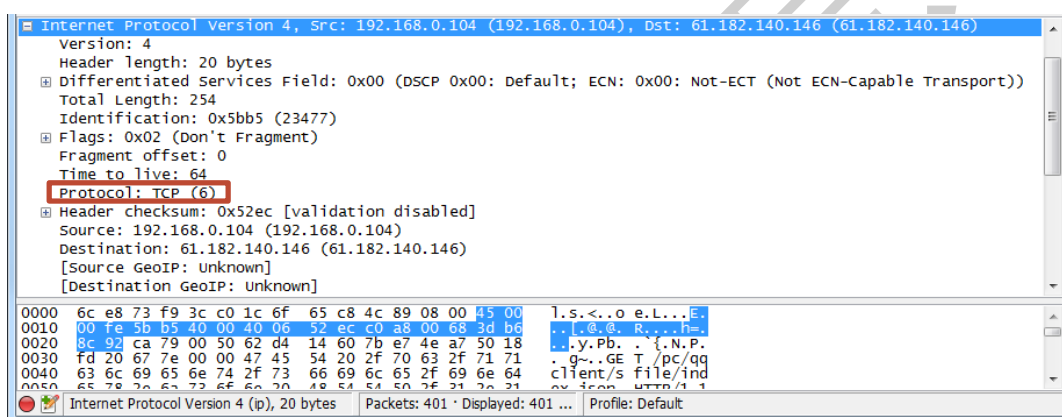


图 2.16 IPv4 分析器

从该界面可以看到 TCP 协议字段的值为 6。

4.TCP 分析器接管

TCP 分析器用于解码 TCP 头部的字段，并基于端口字段的内容，将帧传递给下一个分析器。如图 2.17 所示，该界面显示了 TCP 分析器中的内容。

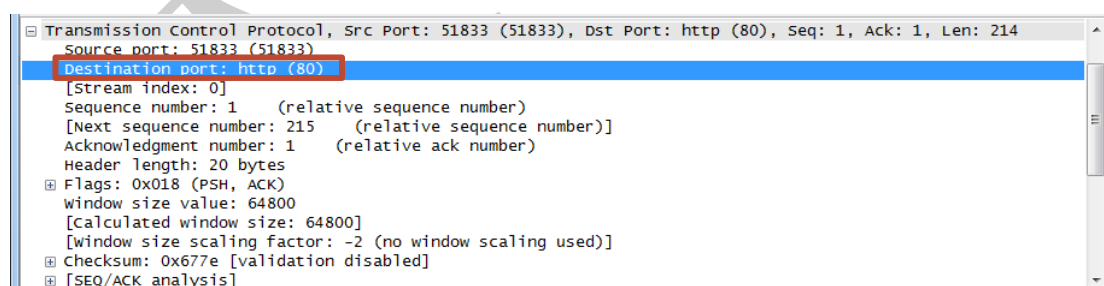


图 2.17 TCP 分析器

从该界面可以看到，目标端口为 HTTP 协议的 80 端口。在下一节，将介绍 Wireshark 如何处理运行在非标准端口上的流量。

5.HTTP 分析器接管

在本例中，HTTP 分析器解码 HTTP 包的字段。在该包中没有嵌入式的协议或应用程序，所以这是帧中应用的最后一个分析器，如图 2.18 所示。



图 2.18 HTTP 分析器

从该界面可以看到，客户端请求了 `www.baidu.com` 网站。

2.2.2 分析非标准端口号流量

应用程序运行使用非标准端口号总是网络分析专家最关注的。关注该应用程序是否有意涉及使用非标准端口，或暗中想要尝试通过防火墙。

1.分配给另一个程序的端口号

当某数据包使用非标准端口上，如果被 Wireshark 识别出是使用另一个程序，则说明 Wireshark 可能使用了错误的分析器，如图 2.19 所示。

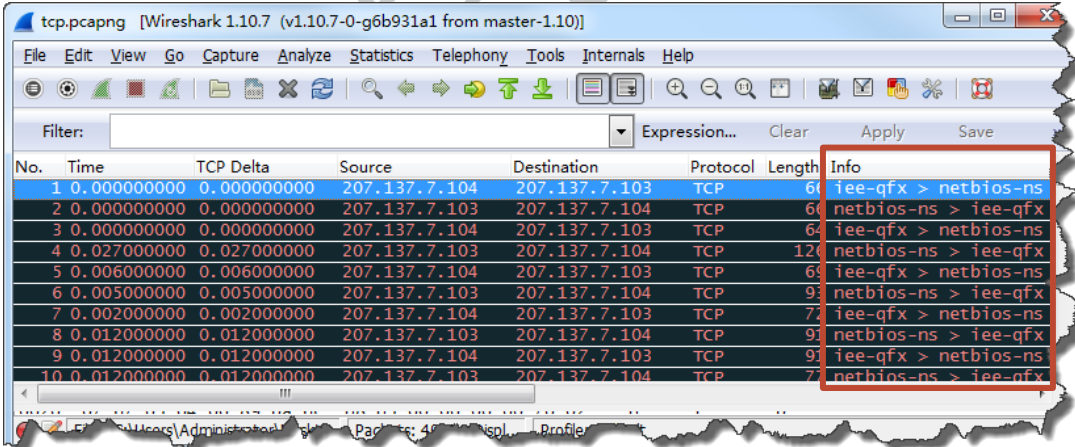


图 2.19 使用非标准端口

从该界面 Packet List 面板中的 Info 列，可以看到显示了 NetBIOS 的信息。但正常的 NetBIOS 流量看起来不是这样的。当 Info 列的端口区域显示 `netbios-ns` 时，Protocol 列显示的都使用的是 TCP 协议。此时查看该文件，发现 Info 列不包含正常的 NetBIOS 名称服务细节。

2.手动强制解析数据

手动强制解析数据有两个原因，分别如下：

- ☐ Wireshark 使用了错误的解析器，因为非标准端口已经关联了一个分析器。
- ☐ Wireshark 不能为数据类型启动解析器。

强制解析器解析数据，右键单击在 Packet List 面板中的不能解析的/解析错误的包，并选择 Decode AS。如图 2.19 所示，通常 TCP 建立连接使用三次握手。客户端与服务端之间共三个 TCP 包，建立成功后应该是 HTTP 协议。但是该界面都是 TCP 协议，说明有未正确解析的数据。这里选择第 4 个包，右键单击选择 Decode AS，将弹出如图 2.20 所示的界面。

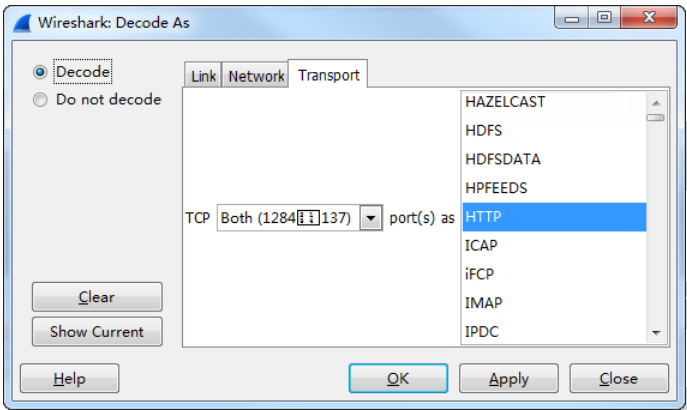


图 2.20 选择解码器

在该界面选择正确的解码协议（这里选择 HTTP），然后单击 OK 按钮。这时，正确解码后显示界面如图 2.21 所示。

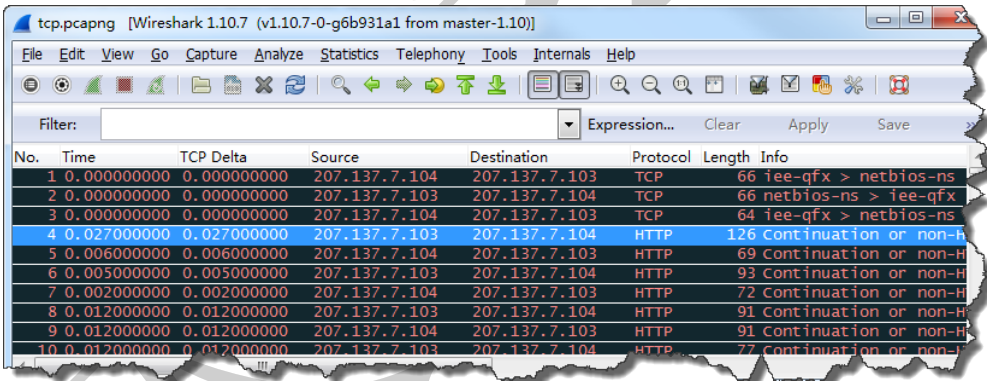


图 2.21 使用 HTTP 解码器

从该界面可以看到 Protocol 和 Info 列的信息都发生了变化。

3.怎样启动解析器

启动解析器的过程如图 2.22 所示。

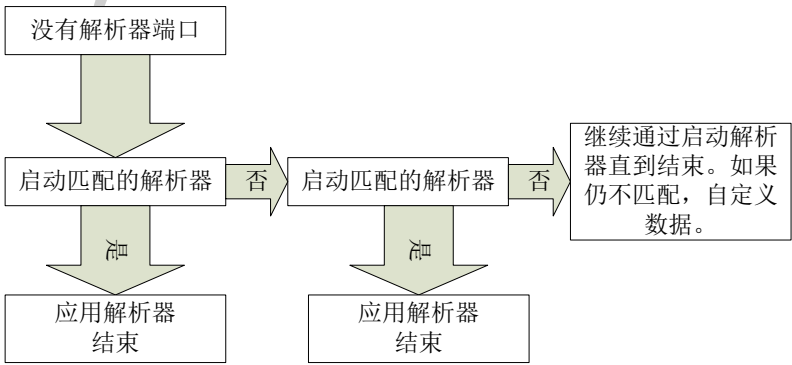


图 2.22 启动解析器过程

启动解析器过程如下所示：

（1）Wireshark 将数据传递给第一个可用的启动器。如果该解析器中没有解析器端口，则传递给下一个匹配的解析器。

（2）如果该解析器能解析发生来数据的端口，则使用该解析器。如果不能解析，则再传递给下一个匹配的解析器。

（3）如果该解析器匹配，则使用并结束解析。如果仍然不能解析，再次将数据传递。依次类推，指定结束。

（4）如果直到结束仍不匹配，则需要自定义数据。

4.调整解析器

如果确定在网络中运行了非标准端口的数据，此时可以在 HTTP 协议的首选项设置中添加该端口。例如，用户想要 Wireshark 解析来自 81 端口号的 HTTP 数据。添加过程如下：

（1）在工具栏中依次选择 Edit|Preferences|Protocols|HTTP，将显示如图 2.23 所示的界面。

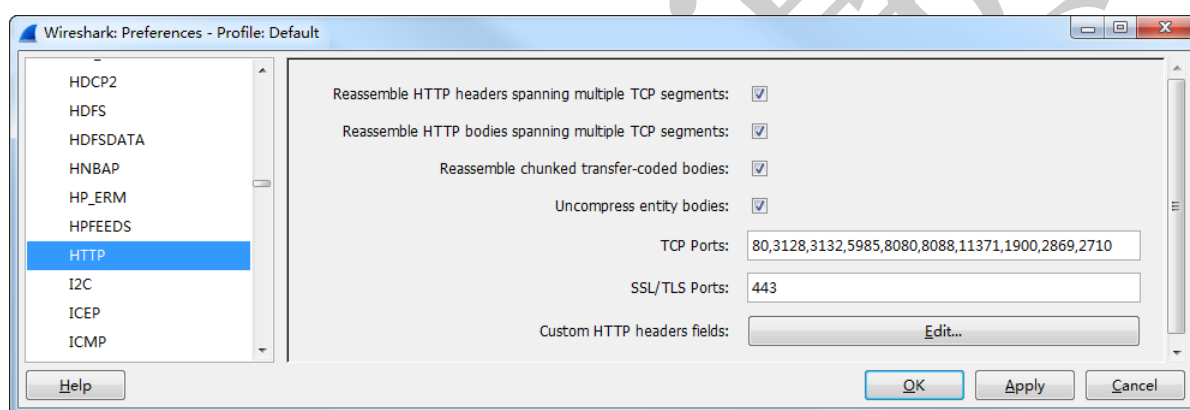


图 2.23 HTTP 协议首选项

（2）在该界面右侧，可以看到默认设置的端口号。在 TCP Ports 对应的文本框中，添加 81 端口号。添加完后，单击 OK 按钮。

2.2.3 设置 Wireshark 显示的特定数据类型

Wireshark 提供了首选项设置，用户可以根据需要进行设置，如设置用户接口、名称解析、过滤器表达式。但是只设置首选项中的配置，仍然有些数据包不能显示。如果想更详细的了解一个数据包，可以设置 Wireshark 特定的数据类型。

1.用户接口设置

在 Wireshark 的工具栏中依次选择 Edit|Preference，将显示如图 2.24 所示。

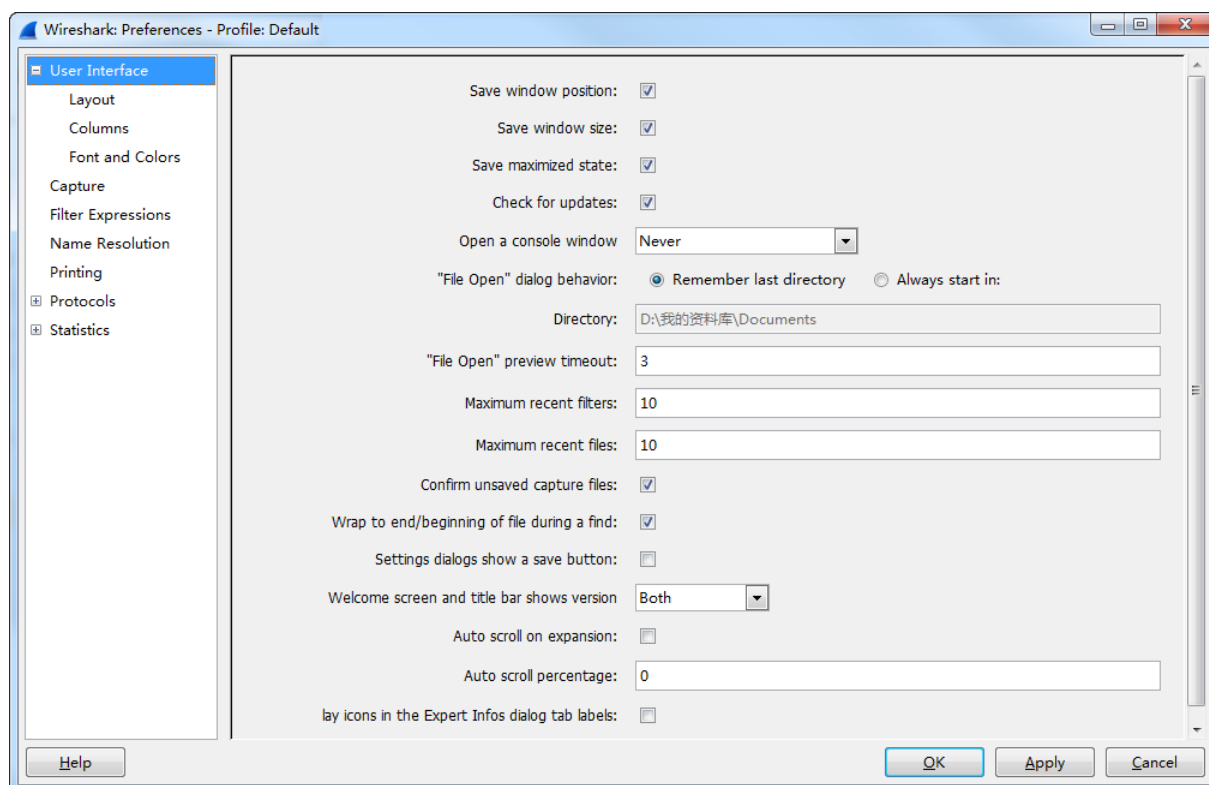


图 2.24 首选项设置

该界面显示了三部分设置，分别是 User Interface、Protocols 和 Statistics。在该界面的右侧框中，就可以对用户接口进行设置了。

2. 开启名称解析

在 Wireshark 工具栏中依次选择 Edit|Preferences|Name Resolution，将显示如图 2.25 所示的界面。用户也可以通过在 View|Name Resolution 的下拉菜单中开启相应的名称解析，但这个设置是临时的。如果想永久生效，需要在首选项中设置。设置后，该信息将被保存在当前配置文件中。

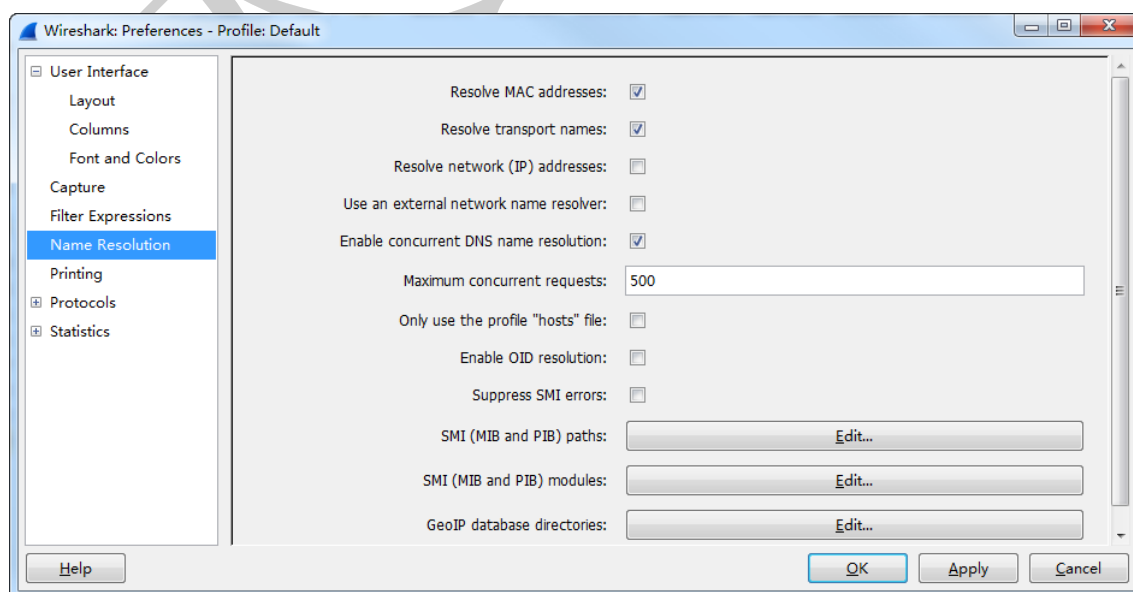


图 2.25 名称解析设置

从该界面可以看到有三种类型的名字解析可用。这三种名字解析含义如下所示：

- ❑ **MAC 地址解析（Resolve MAC address）**：这种类型的名字解析使用 ARP 协议，试图将数据链路层 MAC 地址（如 00:09:5B:01:02:03）转换为网络层地址（如 10.100.12.1）。如果这种转换尝试失败，Wireshark 会使用程序目录中的 `ethers` 文件尝试进行转换。Wireshark 最后的方法就是将 MAC 地址的前三个字节转换到设备的 IEEE 指定制造商名称，例如 `Netgear_01:02:03`。
- ❑ **传输名称解析（Resolve transport names）**：这种类型的名称解析尝试将一个端口转换成一个与其相关的名称。例如，可以将端口 80 显示为 `http`。
- ❑ **网络地址解析（Resolve network(IP)address）**：这种类型的名称解析试图将一个网络层地址（如 192.168.1.50 这个 IP 地址），转换为一个易读的 DNS 名称（如 `MarketingPC1.domain.com`）。

如果要开启名称解析，只需要将相应名字解析前的复选框勾上即可。

3.过滤器表达式

在 Wireshark 工具栏中依次选择 `Edit|Preferences|Filter Expressions` 命令，将显示如图 2.26 所示的界面。

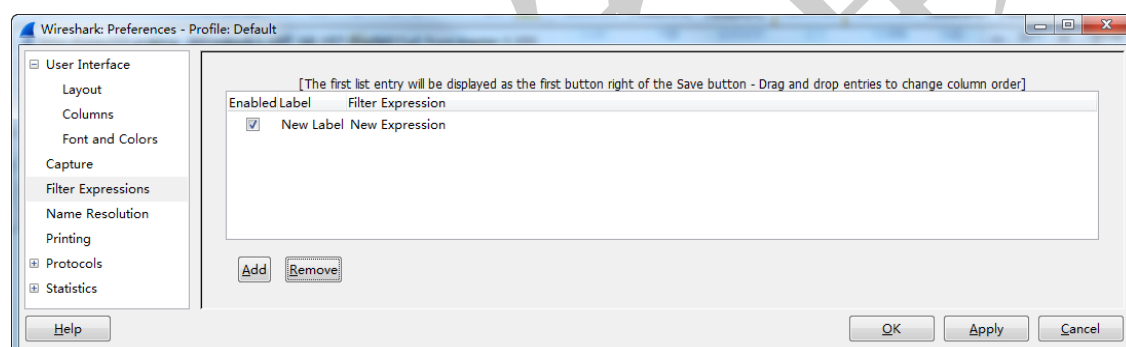



图 2.26 设置过滤器表达式

在该界面可以通过单击 `Add`、`Remove` 按钮，来添加、删除过滤器。添加用户最想使用的显示过滤器，这样在捕获文件中可以很好的应用。

【实例 2-2】设置 Wireshark 首选项。具体操作步骤如下所示：

- (1) 打开 `http.pcapng` 文件。
- (2) 在 Wireshark 工具栏中单击 （编辑首选项）图标，将显示如图 2.27 所示的界面。

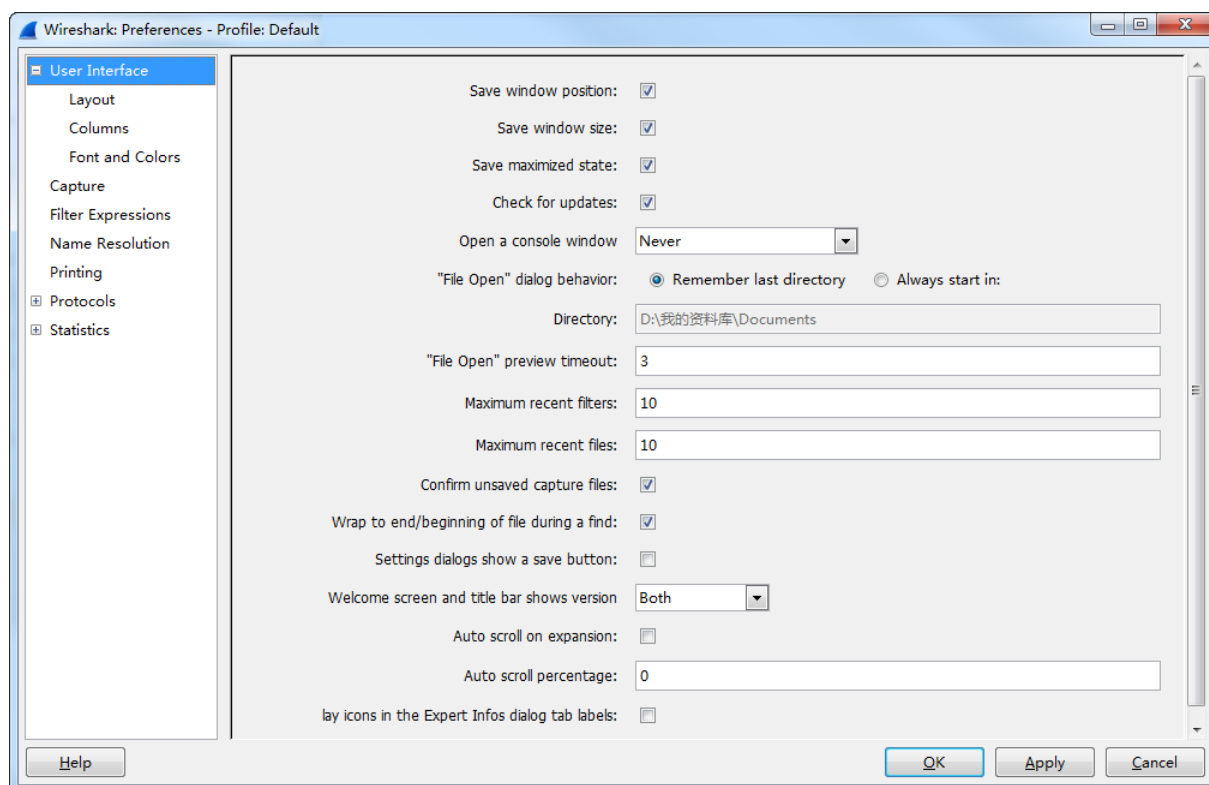


图 2.27 编辑首选项

(3) 在该界面修改 **Maximum recent filters** 和 **Maximum recent files** 的默认设置，修改为 30。修改这两个设置后，用户能够快速的调用更多最近使用过的过滤器和文件。

(4) 单击 **OK** 按钮，将返回 Wireshark 主界面。

4. 设置协议和应用程序

用户可以在首选项设置中，查看 Wireshark 包含的所有可编辑设置的协议和应用程序。在首选项设置中，单击 **Protocols** 前面的加号 (+)，即可查看所有应用程序和协议。对于定义协议，最快的方法就是右键单击 **Packet Details** 面板中的数据进行设置。

【实例 2-3】在 Wireshark 主界面可以使用单击右键的方法，检查和修改 IP、UDP、TCP 设置。下面介绍禁用 IP、UDP 和 TCP 校验验证。具体操作步骤如下所示：

(1) 这里选择在 **Packet List** 面板中的 19 帧，右键单击 **Packet Details** 面板中的 **Internet Protocol**，将显示如图 2.28 所示的界面。

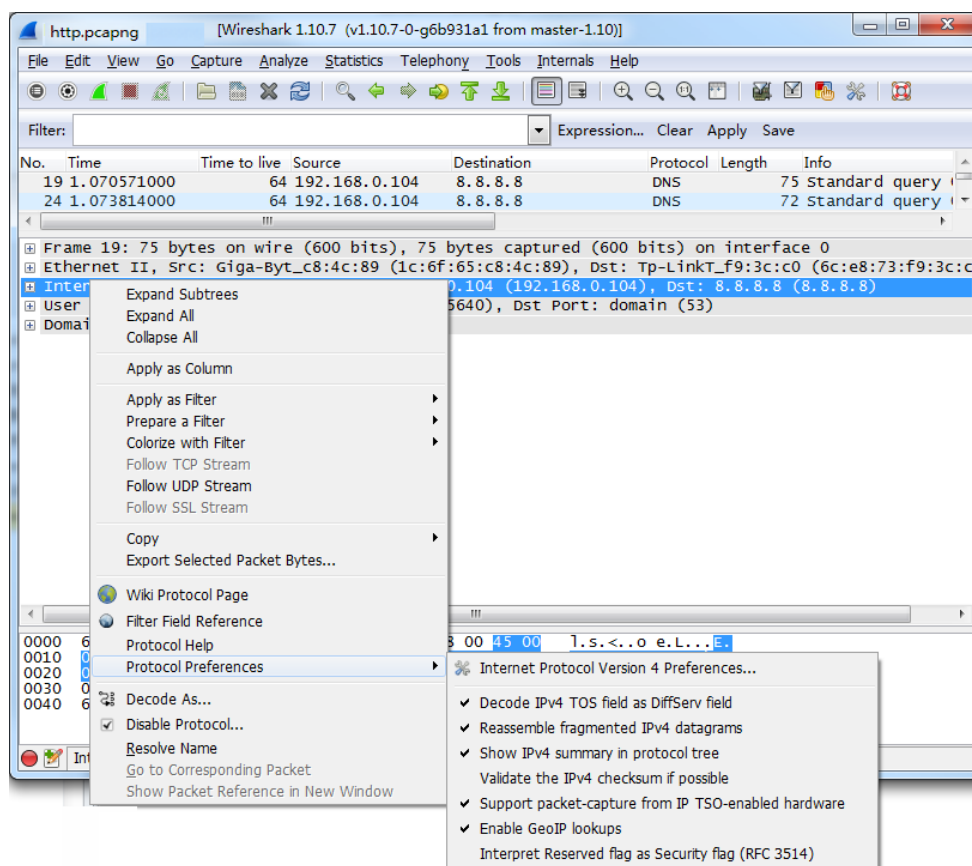


图 2.28 Internet Protocol 菜单

(2) 将鼠标停留在在该界面 Internet Protocol 菜单的 Protocol Preference 选项上，将出现该选项的子菜单。在该子菜单中 Validate the IPv4 checksum if possible 是启用的，则单击该选项将其禁用。

(3) 再次选择 19 帧右键单击 Packet Details 面板中的 User Datagram Protocol，在弹出的菜单中依次选择 Protocol Preference|Validate the UDP checksum if possible 命令，将该选项禁用。

(4) 在 Packet List 面板中选择 59 帧。右键单击 Packet Details 面板中的 Transmission Control Protocol 会话，在弹出的菜单中依次选择|Protocol Preference|Validate the TCP checksum if possible 命令，将该选项禁用。此外，还需要设置额外的几个选项。如下所示：

- ☐ 禁用 Allow subdissector to reassemble TCP streams 选项
- ☐ 启用 Track number of bytes in flight 选项
- ☐ 启用 Calculate conversation timestamps 选项

以上三个协议的含义如下所示：

- ☐ Allow subdissector to reassemble TCP streams: 该选项默认设置是启用的。但是当分析 HTTP 数据时，它可能带来一定的困扰。例如，一个 HTTP 服务使用响应码（如 200 OK）响应客户端请求，并且它包括一些请求的文件包。这时，Wireshark 就不能显示响应码。相反，Wireshark 将显示[TCP Segment of a Reassembled PDU]（协议数据单元）。启用该选项和不启用该选项，显示的结果如图 2.29 和 2.30 所示。

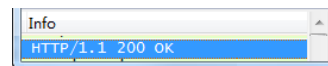
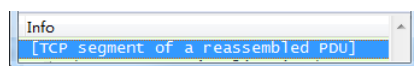


图 2.29 启用 TCP 重组

图 2.30 禁用 TCP 重组

- ❑ **Track number of bytes in flight:** 该选项用来设置通过 TCP 连接发送数据字节数，但还是被认为“未确认的字节”。用户可以配置 Wireshark，查看目前在 TCP 通信中有多少未确认的数据。当启用该选项后，在 Packet Details 面板中一个新的会话将被附加到 TCP 头部。在 TCP 建立连接之前，这个新字段将不会显示。
- ❑ **Calculate conversation timestamps:** 该选项用来在单个 TCP 会话内，设置 TCP 时间戳。这使用户可以获得基于单个 TCP 会话中的第一帧或单独 TCP 会话前一帧的时间戳。当启用该选项后，在 Packet Details 面板中一个新的会话将被附加到 TCP 头部，如图 2.31 所示。

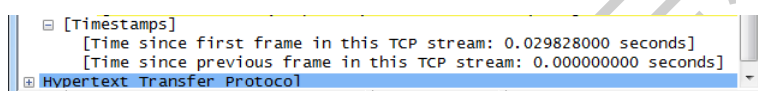


图 2.31 添加的时间戳

(7) 根据以上要求设置完后，包显示的信息将发生改变。这里单击 108 帧，展开在 Packet Details 面板中的 Transmission Control Protocol Line、SEQ/ACK analysis 和 Timestamps 会话，将显示如图 2.32 所示的界面。

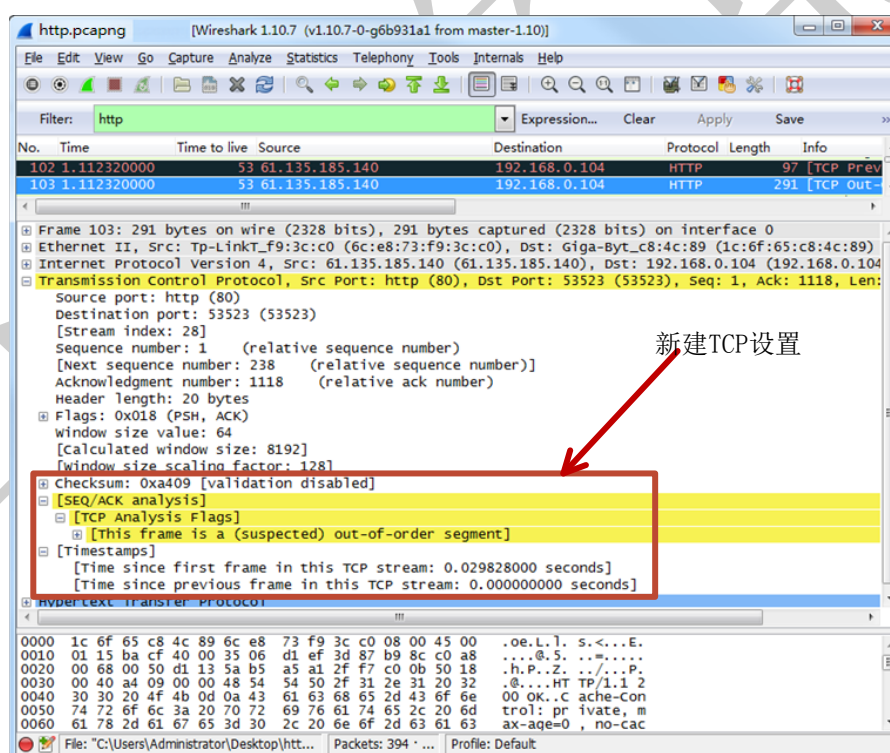


图 2.32 改变的 TCP 设置

2.2.4 使用 Profile 定制 Wireshark

Wireshark 有些特定的特征适合故障诊断任务。而某些定义设置的特征，可能适合网络取证任务。Profile 允许用户为不同的分析过程，定制单独的 Wireshark 配置。下面将介绍使

用 Profile 定制 Wireshark。

1.基本的 Profiles

Profiles 是一个目录。该目录中包含了多个 Profile 目录。每个 Profile 目录包含了 Wireshark 运行时加载的配置和支持文件。例如，用户可以创建一个 Profile，用来关注安全问题。这个安全的“Profile”可能包含有显示过滤器、首选项设置以及着色规则。

2.创建新的 Profile

创建 Profile 有两种方法。如下所示：

（1）右击 Profile 列进行创建。

在 Wireshark 界面的底部，右击 Profile 列，将弹出如图 2.33 所示的菜单。在该界面单击 New 命令，将显示如图 2.34 所示的对话框。

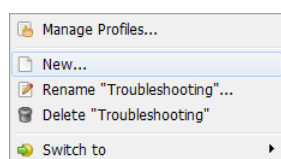


图 2.33 Profile 菜单

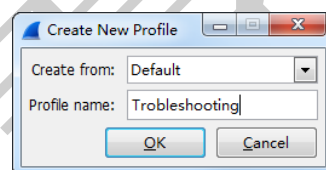


图 2.34 创建新的 Profile

在该对话框的 Create from 的下拉列表中选择可用的 Profile，并设置新建的 Profile 名称，这里分别设置为 Default 和 Troubleshooting。然后单击 OK 按钮，将看到当前的 Profile 变为 Troubleshooting，如图 2.35 所示。这样，我们就基于 Default Profile 创建了一个新的 Profile，并命名为 Troubleshooting。

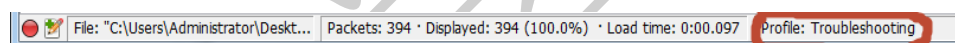


图 2.35 新建的 Profile

现在所有的捕获过滤器设置、显示过滤器设置、颜色规则和首选项设置，都将被保存到 Troubleshooting Profile 中。

（2）通过工具栏中的选项创建。

在工具栏中依次选择 Edit|Configuration Profiles 命令，将显示如图 2.36 所示的界面。

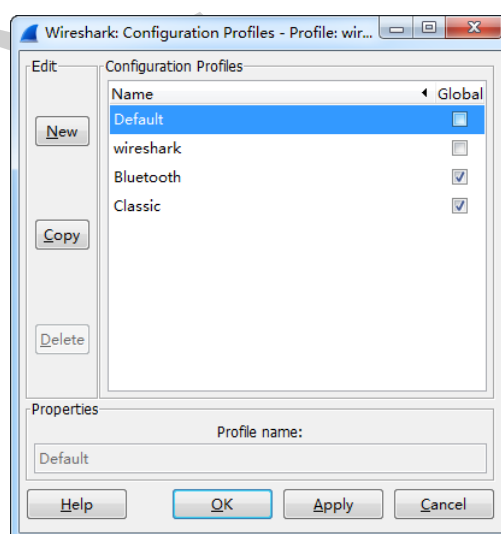


图 2.36 配置 Profiles

在该界面可以看到默认已经添加的 Profile。此时单击 New 按钮，将显示如图 2.37 所示

的界面。

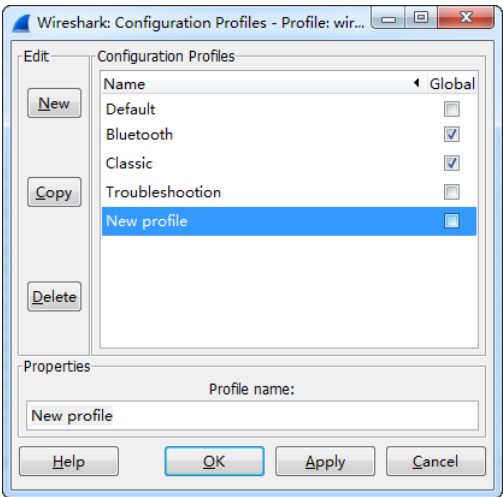


图 2.37 新建 Profile

从该界面可以看到新建的 Profile，其默认的名称是 New profile。该名称可以进行修改，设置成自己想使用的名称。设置完后，单击 OK 按钮。

2.2.5 查找关键的 Wireshark Profile

Wireshark 设置存储在两个位置，分别是 global configuration 目录和 personal configuration 目录。了解 Wireshark 的存储设置，能使用户快速更改设置，或和其它人、Wireshark 系统共享配置。

对于不同的操作系统上安装的 Wireshark，这些目录的位置可能不同。用户可以在菜单栏中依次选择 Help|About Wireshark|Folders 命令，查看这些目录的位置，如图 2.38 所示。

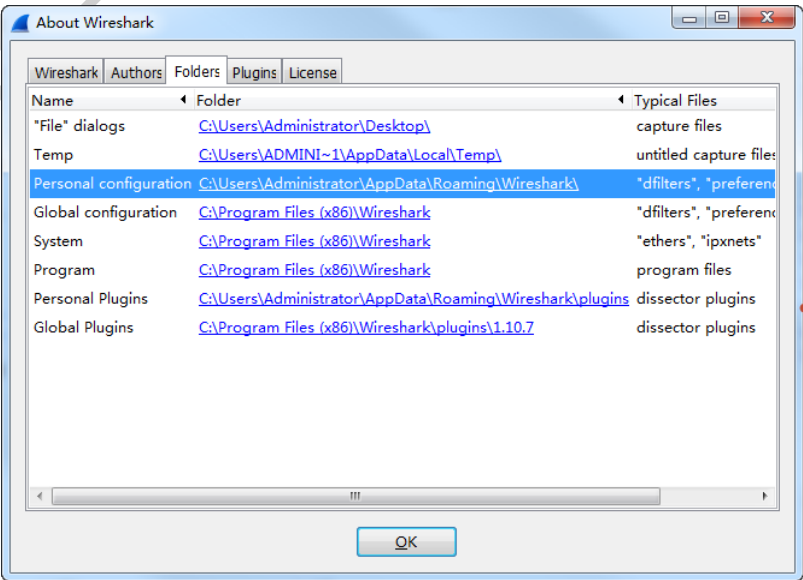


图 2.38 Wireshark 的配置文件位置

从该界面可以看到 global configuration 和 personal configuration 的目录位置。

【实例 2-4】下面导入一个 DNS/HTTP 错误 Profile。具体操作步骤如下所示：

- (1) 从前言中提供的网址中下载资源文件。该 Profile 目录被压缩在为一个单个文件。
- (2) 通过选择 Help|About Wireshark|Folders 命令，双击查看 personal configuration folder 的目录结构。
- (3) 将 Wireshark 创建的 Profile 目录，作为第一个定义的 Profile。如果没有看见 Profiles 目录，用户可以手动创建一个。打开 Profiles 目录，如图 2.39 所示。

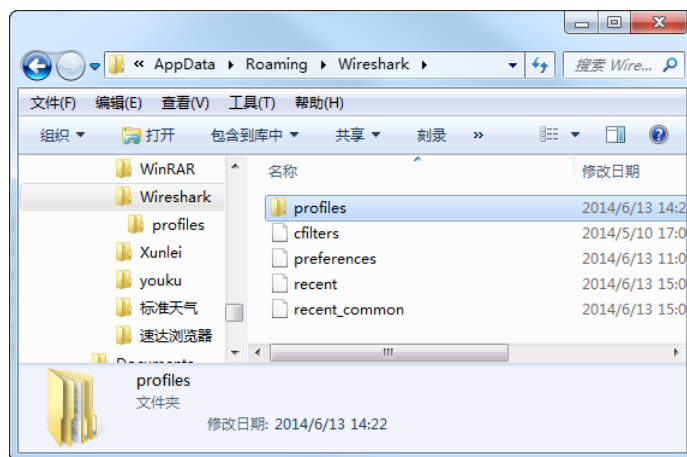


图 2.39 Profiles 目录

- (4) 解压 httpdnsprofile.zip 文件，进入该文件将看到有一个名为 HTTP-DNS_Errors 目录。将该目录移动到 Profiles 目录中。
- (5) 返回到 Wireshark，单击 Profile 列，将显示如图 2.40 所示的界面。

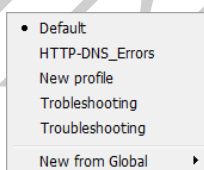


图 2.40 Profile 列

- (6) 在该菜单栏中，选择 HTTP-DNS_Errors profile 测试新建的 Profile。
- (7) 此时打开一个捕获的文件，运行在 HTTP-DNS_Errors 配置文件，效果如图 2.41 所示。

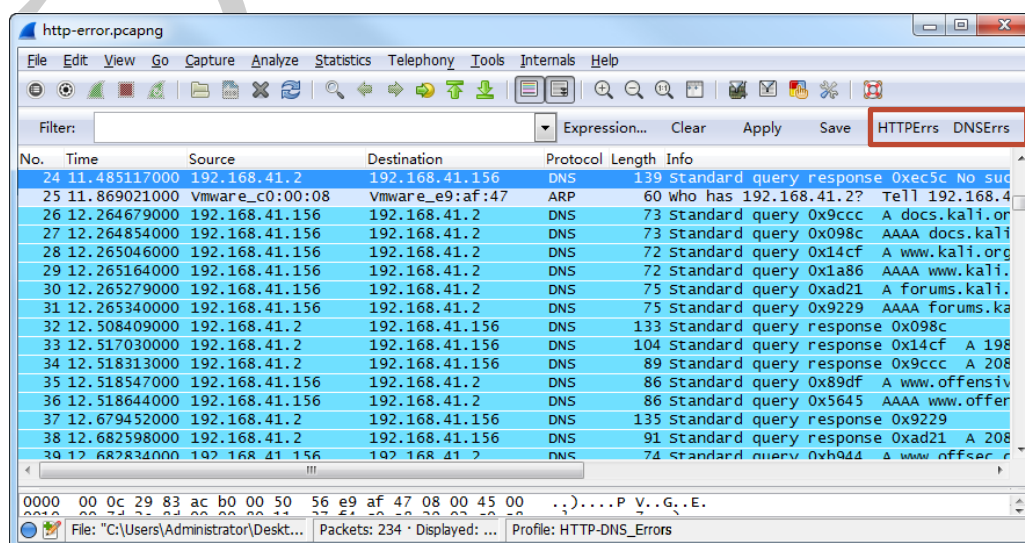


图 2.41 HTTP-DNS_Errors Profile

(8) 从该界面可以看到在显示过滤区域，增加了 HTTPErrors 和 DNSErrs 两个按钮。

2.3 数据包时间延迟

数据包时间延迟是指一个数据包从用户的计算机发送到网站服务器，然后再立即从网站服务器返回用户计算机的来回时间。存在这种时间延迟是不可避免的，本节将介绍数据包时间延迟的原因及类型。

2.3.1 时间延迟

延迟是用于定义时间延迟的一种衡量。当一个主机发送一个请求并等待回复时，总有一些延迟。如果延迟的时间过长，可能是由路径或端点导致的问题。Time 和 Info 列可以查看延迟的三种类型——线路延迟、客户端延迟和服务端延迟。下面分别介绍延迟的三种类型。

1. 线路延迟的显示和原因

线路延迟通常被称为往返时间（RTT）延迟。根据数据包传输的特性，可以判断是否是线路延迟。当服务器收到一个 SYN 数据包时，由于不涉及任何传输层以上的处理，发送一个响应只需要非常小的处理量。即使服务器正承受巨大的负载，通常也会迅速地向 SYN 数据包响应一个 SYN/ACK，这样就排除了服务器导致高延迟的可能性。同时也可以排除客户端高延迟的可能性。因为它在此时除了接受 SYN/ACK 数据包以外什么也没做。这样，就可以确定是属于线路延迟。

路径延迟可能是由一些基础设施设备造成的，如企业路由器。当在一个网络上存在瓶颈的时候，也可能造成路径延迟和数据包丢失。如图 2.42 所示，当用户发送 SYN 包时，开始建立 TCP 连接。如果在收到服务器返回的 SYN/ACK 数据包之前有很长的延迟，则说明是受到线路延迟影响数据通信。

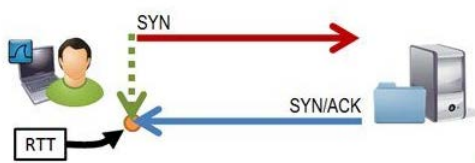


图 2.42 路径延迟

在 Wireshark 中，通过查看 TCP 三次握手可以看到路径延迟。在 SYN-ACK 包发送之前，查看客户端并观察客户端向服务器发送 SYN 数据包。

2. 客户端延迟的显示和原因

客户端延迟通常是由用户、应用程序或缺乏足够的资源造成的。有自然的“人为”延迟（当等待用户点击屏幕上的东西时），但并不是所有都是用户人为造成的。用户应该查找行动迟缓的应用程序造成的客户端延迟。

前面提到的三种延迟类型，客户端延迟是最常见的一种。大多数应用程序的加载在服务器端通信。然而，如果碰巧有一个应用程序加载在客户端和服务器之间，那么必须考虑客户

端的响应时间。如图 2.43 所示，这是一个客户端延迟的例子。

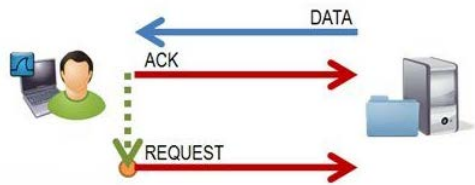


图 2.43 客户端延迟

当客户端发往 TCP 握手过程中最好一个 ACK 包给服务器后，

3.服务器延迟的显示和原因

服务器延迟发生在服务器缓慢响应发送来的请求时。这可能是因为在服务器无法处理一个错误的应用程序或其它类型干扰，需要请求另一个服务器来处理该问题。这样将会导致服务器延迟响应。如图 2.44 所示，下面是服务器延迟的例子。

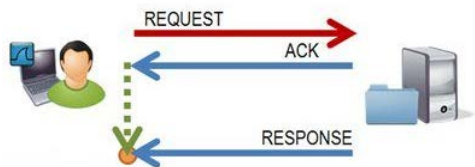


图 2.44 服务器延迟

4.延迟定位框架

上面根据数据包成功地定位了从客户端到服务器的网络高延迟的原因。为了更快速的解决延迟问题，下面使用图解的形式介绍，如图 2.45 所示。

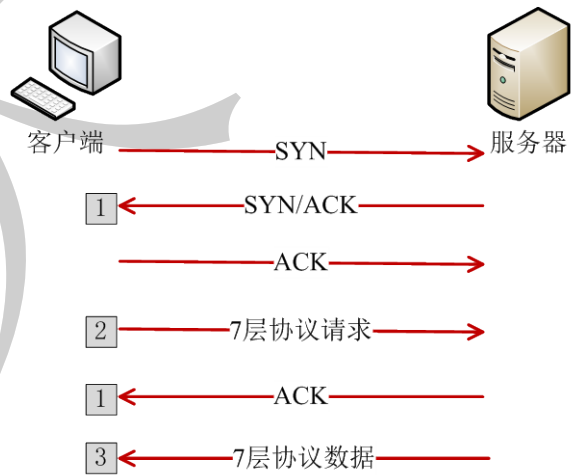


图 2.45 判断延迟问题

在该图中，数字 1 表示线路延迟；数据值 2 表示客户端延迟；数字 3 表示服务器延迟。结合前面介绍的数据包，可以使用这张图快速的解决自己的延迟问题。

2.3.2 检查延迟问题

默认时间列的设置类型为 Seconds Since Beginning of Capture。其实，Wireshark 标志的第一个数据包的捕获时间是 0.000000000。对于第一个包后的每个包的时间列值，都显示的是捕获过程经过多长时间捕获到的。用户可以依次选择 View|Time Display Format|Seconds Since Previous Displayed Packet 命令，查看到增量时间最高的值（从一个数据包到下一个包）。设置完成后，该设置信息将被保存在正在使用的 Profile 中。

以上设置完成后，单击 Time 列两次，将从高到底对时间进行排序。如图 2.46 所示，该界面显示设置 Time 列后的个排序形式。

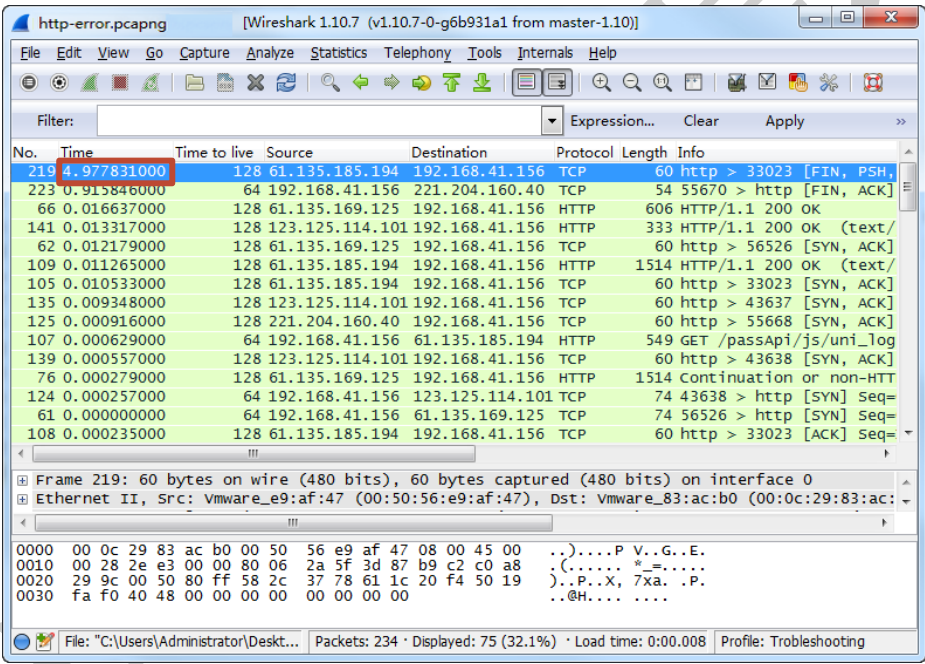


图 2.46 时间排序

从该界面可以看到 http-error.pcapng 文件的时间列，是从高到底排序的。下面以 http-error.pcapng 捕获文件为例，通过过滤查看 TCP 协议的数据包，来判断时间延迟的问题。如下所示：

- (1) 打开 http-error.pcapng 捕获文件。
- (2) 查看 TCP 协议的数据包，如图 2.47 所示。

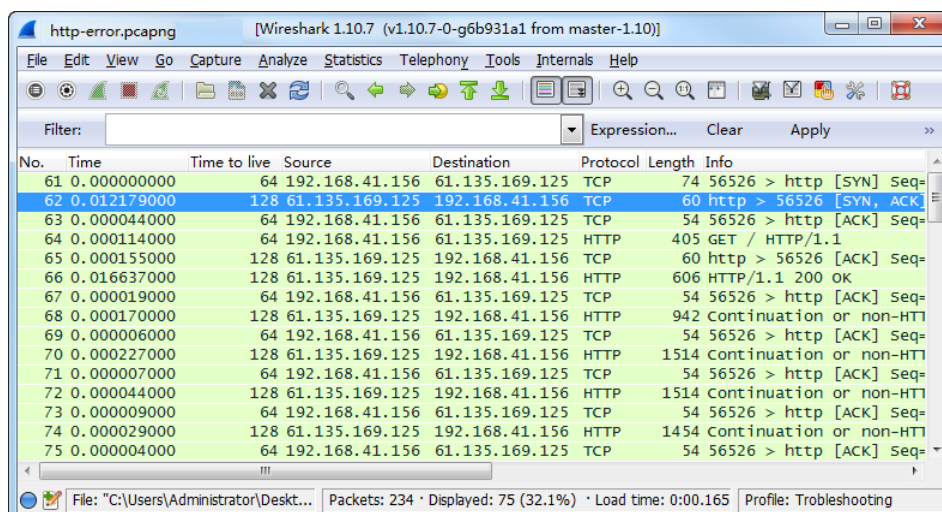


图 2.47 捕获文件的数据包

(3) 从该界面可以看到 TCP 的三次握手过程。其中 61、62、63 为 TCP 的三次握手，这三个数据包中时间延迟最长的是 62。62 包为 TCP 的第二次握手，也是服务器发送的确认包。该捕获文件是在客户端上的，由此可以判断出这是一个路径延迟。

2.3.3 检查时间差延迟问题

在首选项中设置 Protocols|TCP 协议，启用 Calculate conversation timestamps 选项，并设置使用默认的 Profile。设置完成后，现在来学习如何创建一个时间差列。添加 TCP 时间差列。具体操作步骤如下所示：

(1) 展开 TCP 头部，右击 Time since previous frame in the TCP stream 部分，并选择 Apply as Column 命令，如图 2.48 所示。

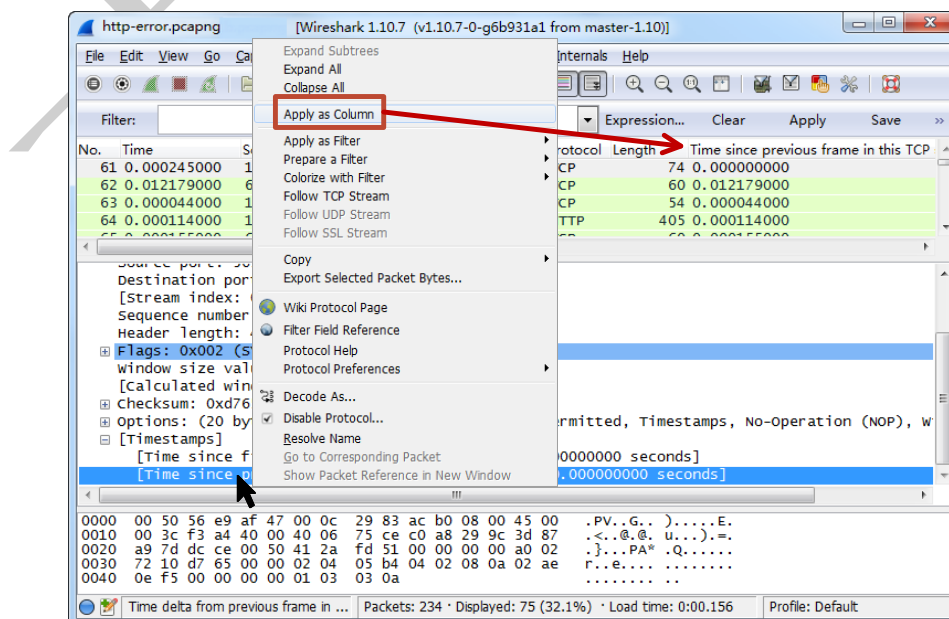


图 2.48 添加列

(2) 从该界面可以看到，在 Packet List 面板中新添加了一列。新添加的列名太长，将该名重命名。右击新添加的列名，并选择 Edit Column Details 命令，如图 2.49 所示。单击 Edit Column Details 命令后，将显示如图 2.50 所示的界面。

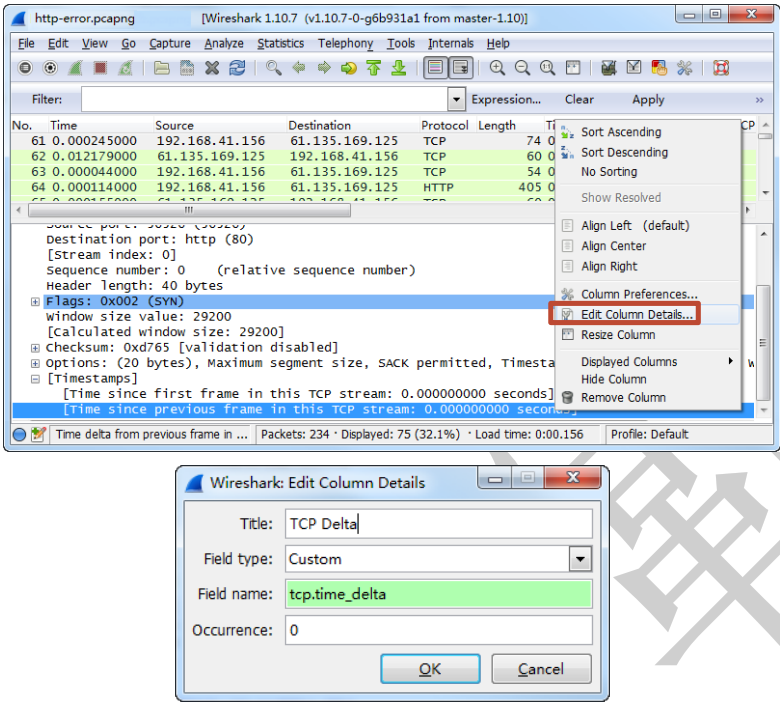


图 2.49 编辑列

图 2.50 重命名

(3) 这里将原来的名修改为 TCP Delta，如图 2.50 所示。然后单击 OK 按钮，将显示如图 2.51 所示的界面。

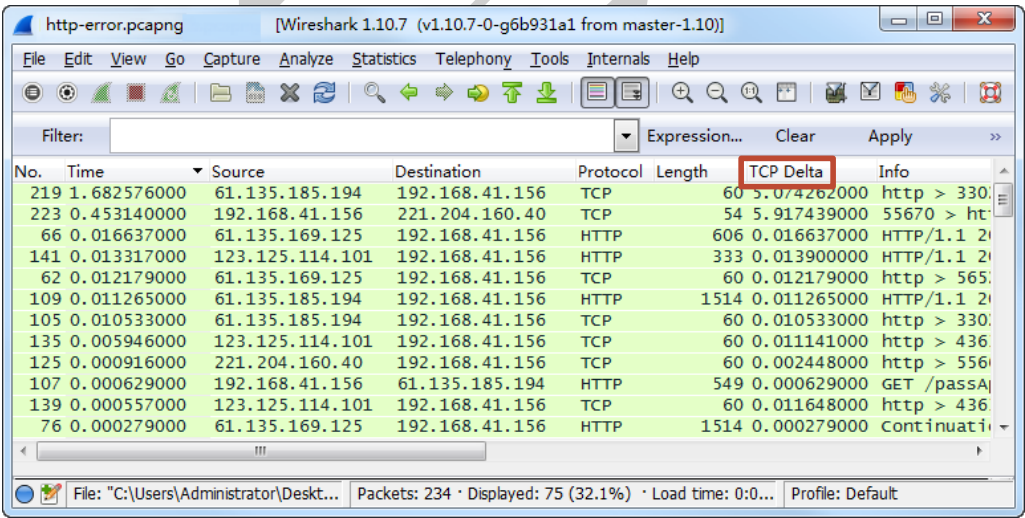


图 2.51 TCP Delta 列

(4) 从该界面可以看到列名已经被成功的修改。
现在来观察一下，Time 列和 TCP Delta 列之间的差异。如图 2.52 所示，将 http-error.pcapng 文件中新建的 TCP Delta 列移动到已存在的 Time 列右边。将 Time 列从高到底排序，查看 Time 列和 TCP Delta 列之间的差异。

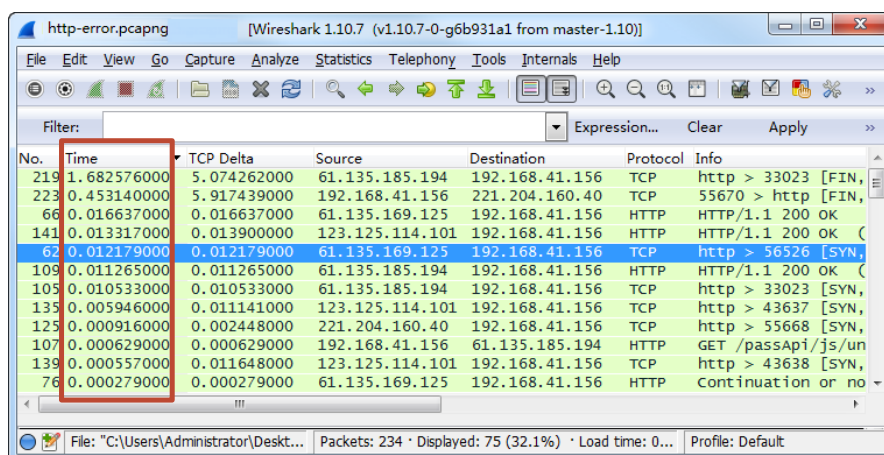


图 4.52 比较 Time 和 TCP Delta 列

从该界面可以看到，Time 列进行了排序。接下来，需要对 TCP Delta 列进行排序。在对 TCP Delta 列排序之前，先找出每个 TCP 会话的延迟。然后，考虑下为什么有些延迟被认为是正常的。

不要专注于某些分组类型，它们延迟是正常的。如下所示：

- ☐ .ico file requests: 该包表示打开浏览器时，在浏览标签上的一个图标。
- ☐ SYN packets: 该包是用来建立 TCP 连接的。该包第一次捕获到后，将会让用户连接到一个 Web 服务器。TCP 连接的第一个包前（SYN 包），会有一个延迟。
- ☐ FIN, FIN/ACK, RST, or RST/ACK packets: 该包是用来终止 TCP 连接的。当用户单击另一个标签或最近没有访问一个网站，或浏览会话配置加载页面后自动关闭时，浏览器发送这些包。
- ☐ GET requests: 当用户点击请求下一个页面的连接时，生成该包。其他时候，一些 GET requests 可能由后台进程启动（如.ico 文件 GET 请求）。
- ☐ DNS queries: 该包在一个 Web 浏览会话期间，在不同时间发送的，如许多个超链接在客户端上加载同一个页面时。
- ☐ TLSv1 encrypted alerts: 该包通常是在关闭连接过程前看到的（TCP 重置）。即使加密，但警惕可能是一个 TLS 关闭请求。

此时将 TCP Delta 列从高到低排序，如图 4.53 所示。

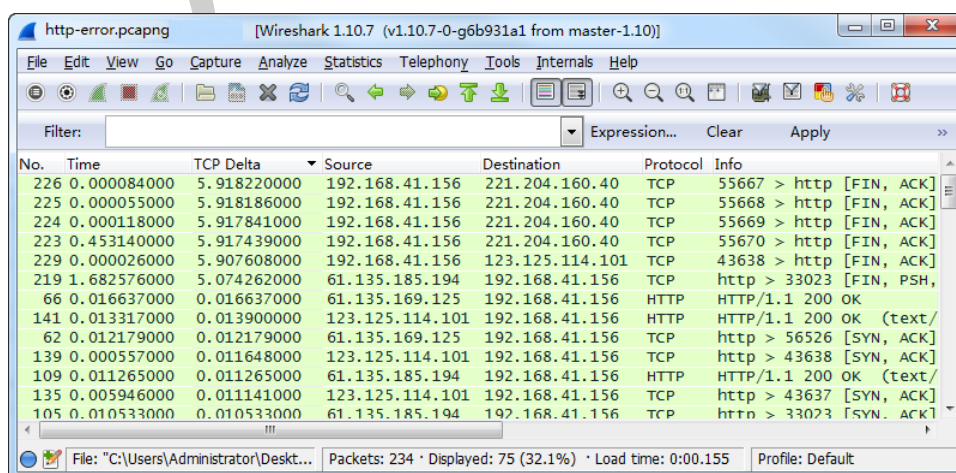


图 4.53 TCP Delta 排序

从该界面可以看到前六个包都是 FIN/ACK 包，这几个包出现延迟是正常的。而且，它们的延迟时间较长。

