

10.4.2 操作方法

DNS 故障主要分为两类：

- 无法解析域名；
- 解析缓慢。

若上述两类故障现象只存在于个别用户主机，需在 LAN 交换机上开启端口镜像功能，并设法将该主机生成的流量重定向至 Wireshark 主机。此时，只需关注由这台主机生成的流量。

若上述两类故障现象普遍存在于网络，需在 LAN 交换机上开启端口镜像功能，并设法将接 DNS 服务器（或连接 Internet 线路）的端口的流量重定向至 Wireshark 主机。

无法解析域名

如何定位问题？

1. 当无法访问 Internet（无法浏览网页、收发 Email 或执行与上网有关的任何一项操作）时，若发现电脑的“本地连接”并未打叉，请 ping 一下有待访问的 Web 站点，看看能否 ping 通^①。
2. 如能 ping 通，则表示网络状况正常。
3. 要是 ping 不通，则可能是因为：
 - Web 站点“禁 ping”；
 - DNS 服务器无法解析有待访问的 Web 站点的域名。
4. 要确定上不了网（无法访问 Internet）是否拜 DNS 故障所赐，需动用 Wireshark，并配置显示过滤器，筛选出 DNS 流量。通过观察筛选出的 DNS 流量，可以发现 DNS 域名无法解析无非以下两种原因：
 - 有待访问的 Web 站点不存在；
 - DNS 服务器不做回应。



注意

也可以使用内置于主机操作系统的 nslookup 工具，来尝试解析有待访问的 Web 站点的域名。

^① 译者注：作者也太心急了，至少也得先 ping 一下本机配置的默认网关的 IP 地址和 DNS 服务器的 IP 地址吧？

5. 当有待访问的 Web 站点不存在时（具体示例详见图 10.9），请仔细观察经过过滤的 Wireshark 抓包文件，将会发现下述情况^①。

- 抓包文件中存在事务 ID 字段值相同的 DNS 查询和响应消息：DNS 查询消息和 DNS 响应消息的事务 ID 字段值相同，便意味着这两种消息是（主机与 DNS 服务器之间的）一问一答。由图 10.9 可知，Wireshark 抓到了事务 ID 字段值同为 0x971e 的一对 DNS 查询/响应消息。
- 一问一答（即 Wireshark 抓到的成对的 DNS 查询和响应消息之间）有一定的延迟：这说明本地 DNS 服务器在执行递归解析操作时经过了长途跋涉。由图 10.9 可知，一问一答的间隔时间为 346ms，之所以会出现这么长的延迟，是因为位于欧洲的本地 DNS 服务器在执行递归解析操作时，一直问到了位于中国台湾的权威名字服务器。
- 抓包文件中存在身背 No such name 字样的 DNS 响应消息。由图 10.9 可知，Wireshark 抓到了本地 DNS 服务器传回的身背 No such name 字样的 DNS 响应消息。仔细观察这条 DNS 响应消息的内容，不难发现，是位于中国台湾的权威名字服务器 a.dns.tw 做出了权威应答“有待访问的 Web 站点的域名不存在”。这充分说明了整个 DNS 系统运转正常，但有待访问的 Web 站点的域名不存在。

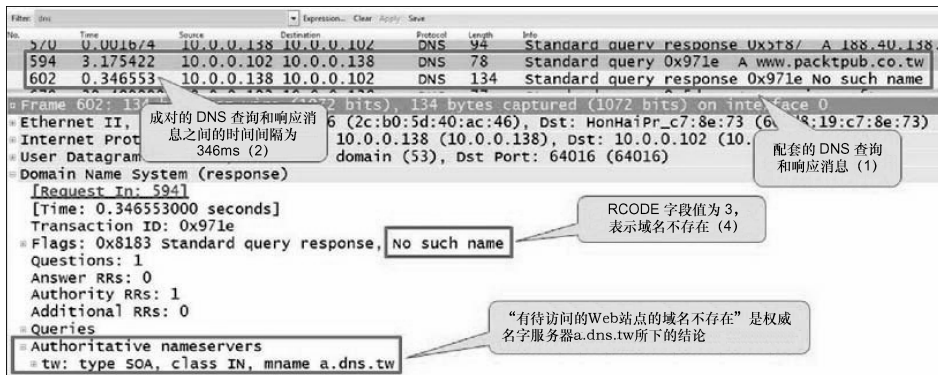


图 10.9

6. 当 DNS 服务器不做回应时，仔细观察 Wireshark 抓包文件中经过筛选的数据包，将会看到以下两图所示场景。

- Wireshark 抓到了身背 refused 字样的 DNS 应答消息：这表示查询请求遭到了 DNS 服务器的拒绝，如图 10.10 所示（为什么会遭到拒绝，下一节详述）。

^① 译者注：原文不成体系，译文酌改。

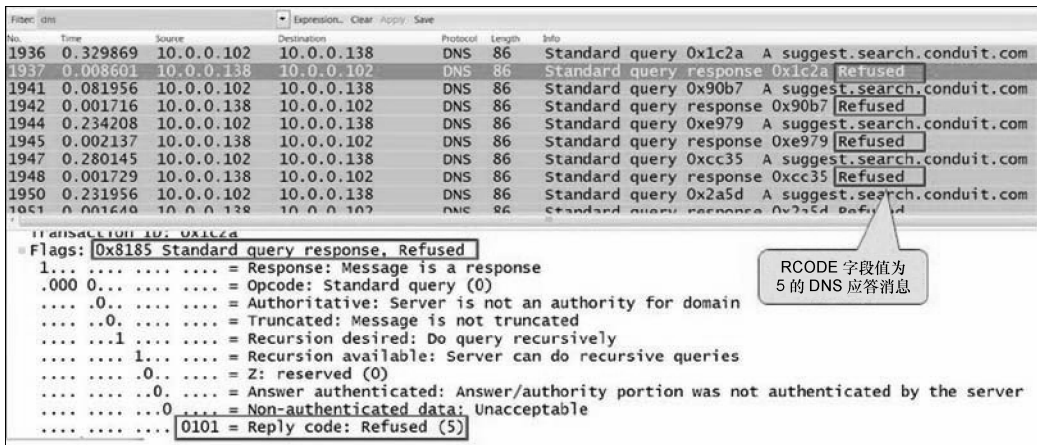


图 10.10

- Wireshark 连续抓到了多条事务 ID 字段值相同的 DNS 查询消息（如图 10.11 所示），但却未抓到一条配套的 DNS 应答消息：这表示 DNS 服务器因某种原因没有回应查询请求。

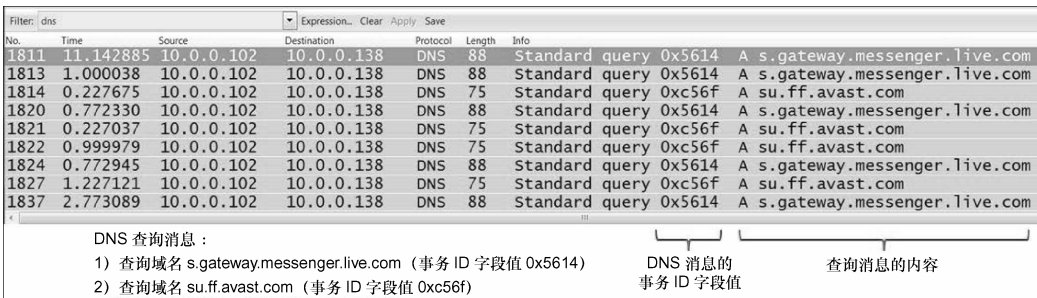


图 10.11

要想看得更仔细一点，请随便选中一条 DNS 查询消息，单击右键，点击右键菜单里的 **Follow UDP Stream** 菜单项，其结果如图 10.12 所示。由图 10.12 可知，主机 10.0.0.102 向本地 DNS 服务器 10.0.0.138 连续发送了 5 条事务 ID 字段值相同的 DNS 查询消息（每条 DNS 查询消息间的时间间隔逐渐由短变长），但未收到任何答复。

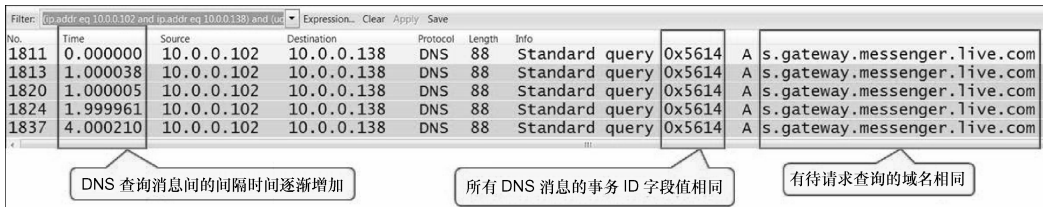


图 10.12

DNS 解析缓慢

如何定位问题？

1. 上网冲浪时，若网页打开缓慢（或使用内网应用程序时，应用程序反应迟钝），请按以下步骤行事。
 - 在交换机上开启端口镜像功能，将其连接 Internet 链路的端口的流量重定向至 Wireshark 主机。使用第 5 章介绍过的 IO Graph 工具，测量过往于该端口的流量的速率，以此来核实 Internet 链路是否过载。
 - 检查 Wireshark 是否抓到了数量惊人的身背 TCP retransmissions 或 TCP duplicate ACK 字样的数据包。若是，则表示存在 TCP 传输问题。
 - 检查 Wireshark 是否抓到了数量惊人的身背 TCP window 字样的数据包。若是，则表示存在 TCP 窗口问题。
2. 若不存在上述问题，则需检查是否存在 DNS 解析问题。
3. 对于有待解决的 DNS 解析问题，要按以下两种情况来处理。
 - 若故障表象为内网应用程序反应迟钝，则需开启端口镜像功能，将交换机上连接内网 DNS 服务器的端口的流量重定向至 Wireshark 主机。
 - 若故障表象为（上网冲浪时）网页打开缓慢（或互联网应用程序反应迟钝），则需将交换机上连接 Internet 链路的端口的流量重定向至 Wireshark 主机。
4. 获悉 DNS 解析所花费的时间，找出解析缓慢的原因。有数种获悉 DNS 解析所耗时间的方法，如下所列。
 - 最简单的方法是，在 Wireshark 数据包列表区域内，选中一个隶属于 DNS 数据流的 DNS 数据包，单击右键，点击 **Follow UDP Stream** 菜单项。然后，观察 Wireshark 数据包列表区域中的 Time 列，就可以获悉相关域名的 DNS 解析时间了。
 - 另一种方法是，借助于内置于 Wireshark 的 IO Graph 工具。在 IO Graphs 窗口中，要先在 Y 轴坐标区域的 Unit 选项的下拉菜单里选择 **Advanced** 选项；再到陡然增大的 IO Graphs 窗口中，选择 **Clac:** 下拉菜单栏里的 **AVG(*)** 菜单项，同时在其右边的 Filter 输入栏内填入显示过滤参数 `dns.time`，选择 **Clac:** 下拉菜单栏里的 **Dot** 菜单项；最后点击相关 **Graph** 按钮，激活 IO Graphs 窗口中与该 graph 相对应的图形，于是便得到了在整个抓包时段内的 DNS 响应时间的汇总图，如图 10.13 所示。

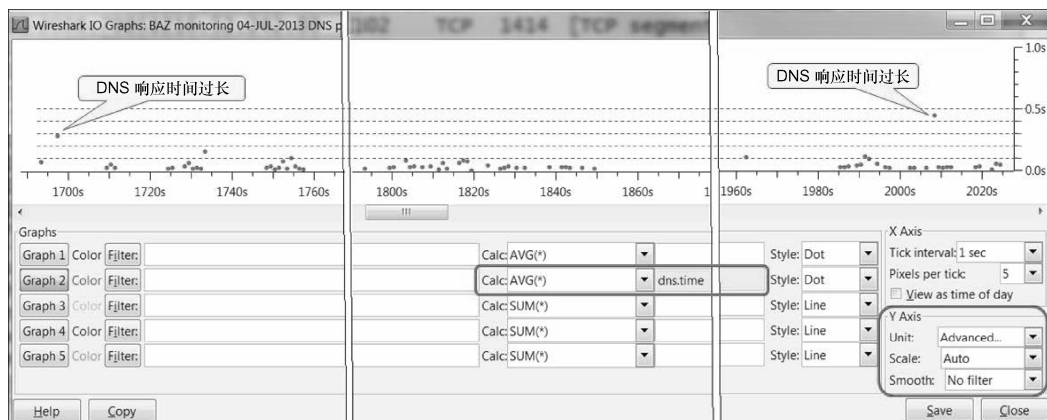


图10.13

由图 10.13 可知，在抓包时段内，绝大多数 DNS 解析的响应时间都在 100 毫秒以内，这完全可以接受；但也可以看到，有两次 DNS 解析响应过慢，一次为 300ms，另一次为 450ms，分别发生在抓包的开始和收尾阶段。



注意

在组织机构的内部网络中，合理的 DNS 解析响应时间应在几十毫秒以内；上网冲浪时，合理的响应时间不应超过 100ms，200ms 之内的响应时间则勉强可以接受。

10.4.3 幕后原理

RFC 1035 定义了 DNS 响应消息中响应代码(RCODE)字段的前 6 个值(0~5)。其他的 RCODE 字段值(6~21)则在后续标准(RFC 2136、RFC 2671、RFC 2845、RFC 2930)中定义。所有已定义的 RCODE 字段值详见 <http://tools.ietf.org/html/rfc2929#section-2.3>。

下表所列为最为常见的 RCODE 字段值及含义：

RCODE 字段值	名 称	含 义	原 因	如何处理
0	NoError	未发生错误	DNS 解析正常	无需采取任何行动
1	FormErr	DNS 服务器无法解析查询请求	当 DNS 服务器收到了格式错误的 DNS 查询消息，或收到了携带某些扩展功能字段的 DNS 查询消息，但无法识别时，便会回复 RCODE 字段值为 1 的 DNS 响应消息	可试着更换一台 DNS 服务器，重新执行 DNS 扩展功能查询请求
2	ServFail	DNS 服务器出现故障，无法处理 DNS 查询请求	当 DNS 服务器回复 RCODE 字段值为 1 的 DNS 响应消息时，便表示其发生了故障	可试着更换一台 DNS 服务器，重新发送查询请求

续表

RCODE 字段值	名 称	含 义	原 因	如何处理
3	NXDomain	请求解析的域名不存在 (界定标准为: 权威名字服务器发出 RCODE 字段值为 1 的 DNS 响应消息)	当 DNS 服务器回复 RCODE 字段值为 1 的 DNS 响应消息时, 便表示请求解析的域名不存在	请检查请求解析的域名是否正确
4	NotImp	DNS 服务器不支持 DNS 查询消息的类型		
5	Refused	DNS 服务器因安全性或策略设置问题, 拒绝处理 DNS 请求	DNS 服务器(因安全性或策略设置问题)不愿意为某些设备提供 DNS 查询服务, 或不愿意回应某些特定的查询请求	

10.4.4 拾遗补缺

DNS 服务器的 IP 地址如何配置(即在本机 TCP/IP 配置的 DNS 服务器配置项里应该填哪一个 IP 地址), 是很多人常问的问题。作者的答案非常简单——应填入离本机最近、解析速度最快的 DNS 服务器的 IP 地址, 这一 IP 地址一般都是本地电信运营商的 DNS 服务器的 IP 地址。

图 10.14 所示为由 Google Namebench 软件(开源软件)测算出的各 DNS 服务器的平均解析时间。

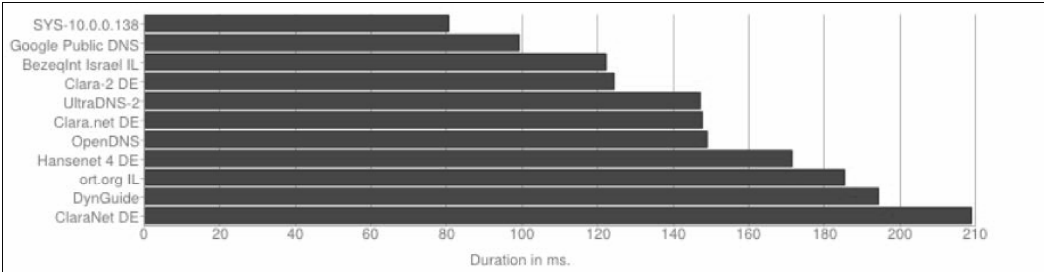


图10.14

- 内网 DNS 服务器的平均解析时间为 80 毫秒, 其 IP 地址为私网地址 10.0.0.138。
- ISP DNS 服务器 (Google public DNS) 的平均解析时间为 100 毫秒。
- 所有位居海外的 DNS 服务器的平均解析时间都高于 120 毫秒。

总而言之, 只要 DNS 解析时间在 100 毫秒之内, 则一切正常; 在绝大多数情况下, 解析时间在 150~200 毫秒之间, 也算正常; 即便是偶而出现几次 DNS 解析缓慢的情况, 也无需担心, 因为这极有可能是本地 DNS 服务器在等待位于地球另一端 (待请求域名所在区域) 的权威名字服务器的应答。

上网冲浪时,若浏览器所要加载的内容过多,则本机 DNS 解析器有可能会执行数十次 DNS 解析操作。图 10.15 所示为作者通过 Web 浏览器访问 www.cisco.com 时 Wireshark 抓取到的 DNS 流量。

No.	Time	Source	Destination	Protocol	Info
2346	66.906122	10.0.0.2	10.0.0.138	DNS	Standard query 0x7444 A www.cisco.com ①
2347	67.113670	10.0.0.138	10.0.0.2	DNS	Standard query response 0x7444 CNAME www.cisco.com.akadns.net
2361	69.122842	10.0.0.2	10.0.0.138	DNS	Standard query 0x0df4 A api.webrep.avast.com
2362	69.123955	10.0.0.2	10.0.0.138	DNS	Standard query 0xd485 A ap.ff.avast.com ②
2364	69.142771	10.0.0.138	10.0.0.2	DNS	Standard query response 0x0df4 A 77.234.43.95 A 77.234.43.95
2365	69.142819	10.0.0.138	10.0.0.2	DNS	Standard query response 0xd485 A 109.123.117.68 A 77.234.44.10
2369	69.201093	10.0.0.2	10.0.0.138	DNS	Standard query 0x4528 A www.static-cisco.com ③
2397	69.272365	10.0.0.2	10.0.0.138	DNS	Standard query 0xdf38 A www.cisco.com
2398	69.294623	10.0.0.138	10.0.0.2	DNS	Standard query response 0xdf38 CNAME www.cisco.com.akadns.net
2501	69.388773	10.0.0.138	10.0.0.2	DNS	Standard query response 0x4528 CNAME static-cisco.com
2662	69.822286	10.0.0.2	10.0.0.138	DNS	Standard query 0x7cf8 A ciscosystemsinc.tt.omtrdc.net ④
2776	69.911815	10.0.0.138	10.0.0.2	DNS	Standard query response 0x7cf8 A 70.42.13.100 A 66.117.23.1
3056	70.547976	10.0.0.2	10.0.0.138	DNS	Standard query 0x010f A news-tags.cisco.com ⑤
3058	70.566150	10.0.0.2	10.0.0.138	DNS	Standard query 0x80ac A cisco-tags.cisco.com
3059	70.567814	10.0.0.138	10.0.0.2	DNS	Standard query response 0x010f A 72.163.10.14
3060	70.576567	10.0.0.2	10.0.0.138	DNS	Standard query 0x4468 A cisco.112.2o7.net
3066	70.584723	10.0.0.138	10.0.0.2	DNS	Standard query response 0x80ac A 72.163.10.10
3073	70.595827	10.0.0.138	10.0.0.2	DNS	Standard query response 0x4468 A 66.235.139.110 A 66.235.13.1
3122	71.153544	10.0.0.2	10.0.0.138	DNS	Standard query 0xfc41 A tools.cisco.com
3128	71.218031	10.0.0.2	10.0.0.138	DNS	Standard query 0x9d0c A products.mcisco.com ⑥
3129	71.218284	10.0.0.2	10.0.0.138	DNS	Standard query 0x130c A newsroom.cisco.com ⑦
3138	71.348653	10.0.0.138	10.0.0.2	DNS	Standard query response 0xfc41 A 72.163.4.38
3140	71.376504	10.0.0.2	10.0.0.138	DNS	Standard query 0xd077 A ciscocommunities.jive-mobile.com

图 10.15

由图 10.15 可知,作者的计算机首先发出了一条针对域名 www.cisco.com 的 DNS A 记录查询请求(在图中被标记为①)。然后,又依次针对域名 ap.ff.avast.com、www.static-cisco.com、ciscosystemsinc.tt.omtrdc.net、news-tags.cisoc.com、products.mcisco.com、newsroom.cisco.com,各发出了一条 DNS 查询数据包(这些 DNS 查询数据包在图中被标记为②~⑦,其中, ap.ff.avast.com 是 Avast antivirus 的 Web 防护服务器)。

借助于 IO Graph 工具(如图 10.16 所示),可以了解到本次打开网站 www.cisco.com 时,发生的上述 7 次 DNS 解析时间最长为 600 毫秒,这也印证了为了什么每次访问 Cisco 官网主页时,浏览器总要花费好几秒钟才能加载完所有内容。

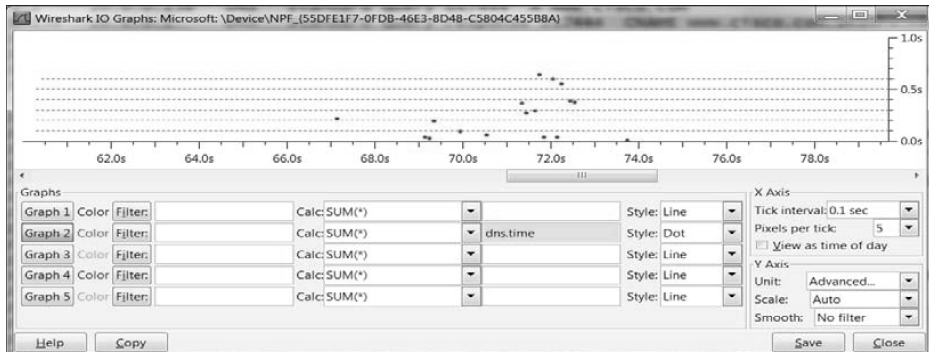


图 10.16

10.5 筛选HTTP流量

本节将介绍如何配置 Wireshark 显示过滤器以筛选 HTTP 流量。

10.5.1 准备工作

将 Wireshark 主机接入 LAN 交换机，激活交换机的端口镜像功能，将有待监控的（连接服务器的）交换机端口流量重定向至 Wireshark 主机；运行 Wireshark 软件，选择正确的网卡，开始抓包。

10.5.2 操作方法

配置 HTTP 显示过滤器的方法包括：直接在抓包主窗口的 Filter 输入栏内输入 HTTP 显示过滤表达式；在抓包主窗口的数据包结构区域内，将 HTTP 数据包的某个 HTTP 属性值指定为显示过滤器的过滤条件；点击抓包主窗口中过滤器工具条上的 **Expression** 按钮，在弹出的 Filter Expression 窗口中构造 HTTP 显示过滤器（具体做法详见本书第 3 章）。

用来筛选 HTTP 流量的显示过滤器分为以下几类。

- 基于名称的 HTTP 显示过滤器。
 - 要让 Wireshark 只显示访问某指定域名（`www.packtpub.com`）的 HTTP 请求数据包，显示过滤器的写法为：`http.host == "www.packtpub.com"`^①。
 - 要让 Wireshark 只显示访问包含了指定字符串的域名（比如，包含 PacktPub 的域名）的 HTTP 请求数据包，显示过滤器的写法为：`http.host contains "packtpub"`。
 - 要让 Wireshark 只显示 Referer 头部内容为“`http://www.packtpub.com/`”的 HTTP 请求数据包，显示过滤器的写法为：`http.referer == "http://www. packtpub.com/"`（亦即让 Wireshark 只显示从 `http://www. packtpub.com/` 链接过来的所有 HTTP 请求数据包）^②。
- 基于 HTTP 请求方法的显示过滤器。
 - 要让 Wireshark 显示包含 GET 请求的所有 HTTP 数据包，显示过滤器的写法为：`http.request.method == GET`。

① 译者注：原文是“Requests to a specific website: `http.host == "www.packtpub.com"`”，译文酌改。

② 译者注：原文是“Requests that were forwarded from PacktPub: `http.referer == "http://www.packtpub.com/"`”。

- 要让 Wireshark 显示所有 HTTP 请求数据包，显示过滤器的写法为：`http.request`。
 - 要让 Wireshark 显示所有 HTTP 响应数据包，显示过滤器的写法为：`http.Response`。
 - 要让 Wireshark 显示包含所有 HTTP 数据包，但包含 GET 方法的 HTTP 请求数据包除外，显示过滤器的写法为：`http.request and not http.request.method == GET`。
- 基于 HTTP 状态码的显示过滤器。
- 要让 Wireshark 显示包含 HTTP 错误状态码的 HTTP 响应数据包，显示过滤器的写法为：`http.response.code >= 400`。
 - 要让 Wireshark 只显示包含 HTTP 客户端错误状态码的 HTTP 响应数据包，显示过滤器的写法为：`http.response.code >= 400 and http.response.code <= 499`。
 - 要让 Wireshark 只显示包含 HTTP 服务器端错误状态码的 HTTP 响应数据包，显示过滤器的写法为：`http.response.code >= 500 and http.response.code <= 599`。
 - 要让 Wireshark 只显示状态码为 404 的 HTTP 响应数据包，显示过滤器的写法为：`http.response.code == 404`。



注意

在配置由非特殊字符作为参数值的 HTTP 显示过滤器（比如，`http.host == packtpub`）时，可以不用为参数值加"`"`。但在配置由特殊字符作为参数值的 HTTP 显示过滤器（比如，`http.host == "packtpub\r\n"`）时，就必须为参数值加"`"`。

10.5.3 幕后原理

本小节将引领读者浏览一遍 HTTP 协议的细节。

HTTP 方法

RFC 2616 定义了几种主要的 HTTP 请求方法。在 RFC 2616 发布之后，又有数种 HTTP 请求方法问世，这些 HTTP 请求方法刊登在了 RFC 2616 的升级版（RFC 2817、RFC 5785、RFC 6266 和 RFC 6585）以及其他的 HTTP 标准（RFC 2518、3252 和 5789）中。

以下所列为定义于 RFC 2616 的几种基本的 HTTP 请求方法。

- **OPTIONS:** HTTP 客户端可使用该方法让 Web 服务器告知其所支持的功能。
- **GET:** HTTP 客户端可使用该方法请求 Web 服务器发送某个资源。
- **HEAD:** 与 GET 方法类似。借助于该方法, HTTP 客户端可在不获取实际资源的情况下, 让 Web 服务器发送资源的概况信息。
- **POST:** HTTP 客户端可利用该方法向 Web 服务器传送数据。比方说, 当使用 webmail 时, 就会调用该方法去传送邮件操作命令。
- **DELETE:** 借助于该方法, HTTP 客户端可请求 Web 服务器删除由 URL 指定的资源。
- **PUT:** HTTP 客户端可利用该方法向 Web 服务器写入数据。有些 Web 服务器允许用户通过 PUT 方法, 在其上创建页面, 但在如此操作之前, 一般要通过用户名/密码认证。
- **TRACE:** 当 HTTP 客户端发起 HTTP 请求时, HTTP 请求报文可能会穿越防火墙、代理服务器或网关等设备, 这些设备可能会修改原始的 HTTP 请求数据包中的内容。借助于 TRACE 方法, HTTP 客户端就能让 Web 服务器弹回 (loopback) 一条 TRACE 响应报文, 其中会携带后者实际收到的 HTTP 请求报文。HTTP 客户端可借此了解原始 HTTP 请求报文是否被损坏或修改过。
- **CONNECT:** 用来连接代理设备。

HTTP 状态码

HTTP 状态码共分 5 类, 如下所列。

类 别	名 称	含 义
1xx	信息状态码	用来表示一般性信息
2xx	成功状态码	用来表示由 HTTP 客户端所请求执行的动作已被 Web 服务器成功接收、接受或处理
3xx	重定向状态码	用来告知 HTTP 客户端使用其他位置来访问其所感兴趣的资源
4xx	客户端错误状态码	用来表示 HTTP 客户端错误
5xx	服务器端错误状态码	用来表示 HTTP 错误

HTTP 状态码的完整列表详见以下链接: <http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>。

10.5.4 拾遗补缺

使用 Wireshark 分析 HTTP 流量时, 在抓包主窗口的数据包结构区域内, 经常可以看见某

些 HTTP 数据包的 Hypertext Transfer Protocol 树状区域下会多一个名为 **Line-based text data: text/html** 的结构，如图 10.17 所示。

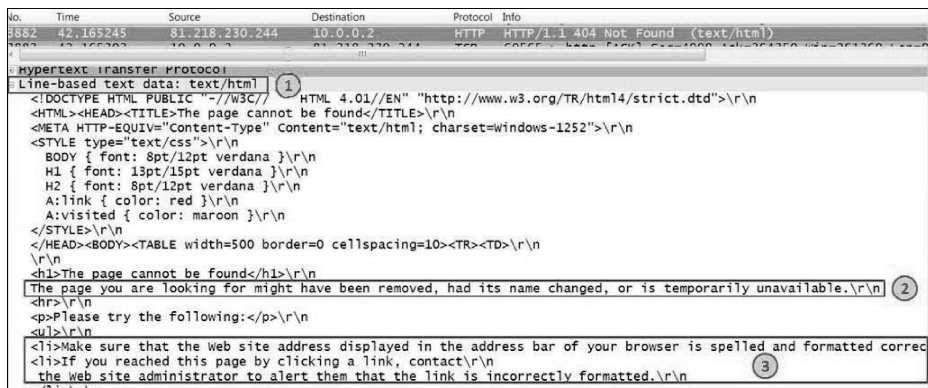


图 10.17

该结构的名称 **Line-based text data: text/html** 在图 10.17 中被标记为①，点击其前面的“+”号，将会暴露出由 Web 服务器返回的错误信息（图 10.17 中的②和③）。

10.6 配置 Preferences 窗口中 protocol 选项下的 HTTP 协议参数

使用 Wireshark 抓取并分析 HTTP 协议流量之前，可调整 Preferences 窗口中 Protocol 选项下 HTTP 协议的某些参数。对这些参数的调整，会影响到 Wireshark 对 HTTP 协议流量的解析，本节将讲解如何调整这些参数。

10.6.1 准备工作

启动 Wireshark 软件，按下面内容的指示行事。

10.6.2 操作方法

1. 点击 **Edit** 菜单下的 **Preferences** 菜单项。
2. 在弹出的 Preferences 窗口中，点击 Protocol 配置选项前的+号，选择 HTTP 协议，如图 10.18 所示。
 - 在图 10.18 所示的 Preference 窗口中，那 4 个复选框是默认勾选的。只要勾选了那个 4 个复选框，即便 HTTP 协议数据包在 IP 层被分片传送，Wireshark

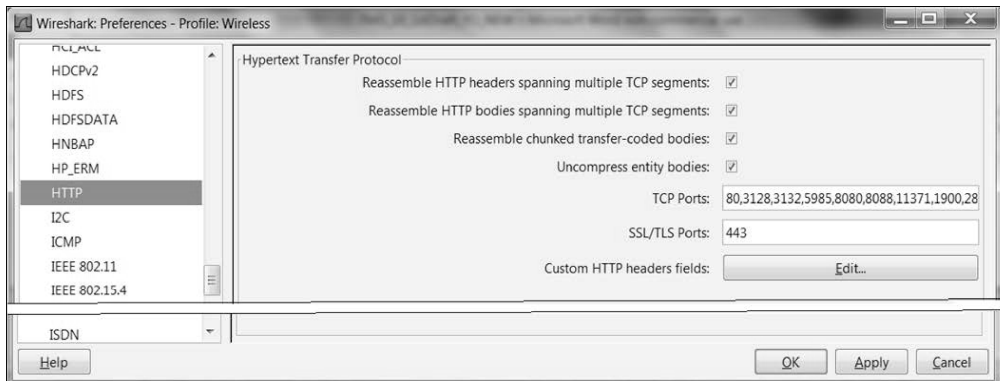


图10.18

也会重组 HTTP 头部和主体。

- 图 10.18 所示的 Preference 窗口中 **TCP Ports** 字样旁，有一个输入栏，其中默认列出了若干 TCP 端口号，Wireshark 会把发往/源于这些端口号的 TCP 流量当做 HTTP 流量来解码。在某些情况下，要想用其他 TCP 端口来跑 HTTP 流量，且希望 Wireshark 将发往/源于此端口号的 TCP 流量视为 HTTP 流量，请将该 TCP 端口号添加进 **TCP Ports** 之后的输入栏。
- 在 **SSL/TLS Ports** 字样旁的输入栏内，默认只有 TCP 443 端口，该输入栏的作用等同于 **TCP Ports** 字样旁的输入栏，要想用其他 TCP 端口来跑 HTTPS 流量，且希望 Wireshark 将发往/源于此端口号的 TCP 流量视为 HTTPS 流量，需将该 TCP 端口号填入此输入栏。

Custom HTTP headers fields 身旁的 Edit 按钮

在图 10.18 所示的 Preference 窗口中 **Custom HTTP headers fields** 字样旁，有一个 **Edit** 按钮，可借助于该按钮功能来创建 http.header 旗下的 HTTP 显示过滤参数。

试举一例，请看图 10.19。

要想让 Wireshark 直接将 HTTP 响应数据包中的 Age 头部作为 HTTP 显示过滤参数来引用，请按以下步骤行事。

1. 点击 HTTP preference 窗口中 **Custom HTTP headers fields** 之后的 **Edit** 按钮（图 10.20 中被标记为①的区域）。
2. 在弹出的 Custom HTTP headers fields Table 窗口中，点击 **New** 按钮（图 10.20 中被标记为②的区域）。



图 10.19

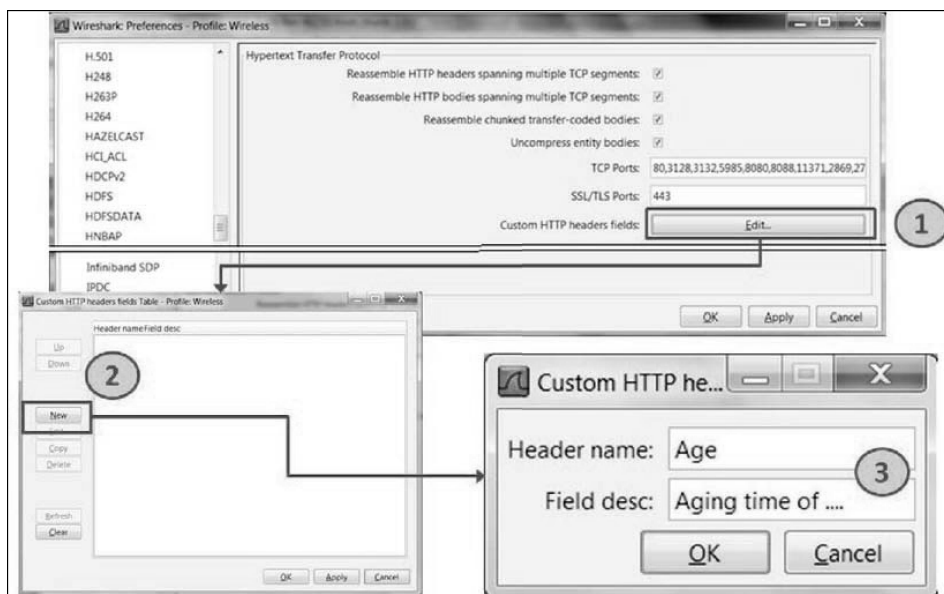


图 10.20

3. 在弹出的小窗口的 **Header name** 输入栏内（图中被标记为③的区域）输入 HTTP 头部的名称 Age(区分大小写); 在 **Field desc** 字段内输入相关注释信息, 作者输入的是 Aging time of ...。
4. 先点击小窗口中的 **OK** 按钮, 再点击 Custom HTTP header fields Table 窗口中的 **OK** 按钮。
5. 现在, 在 Wireshark 抓包主窗口的 Filter 输入栏内, 就可以用显示过滤参数 `http.header.Age` 来直接引用 Age 头部了。



注意

若 HTTP 数据包中包含有自定义的头部，且需要根据此类 HTTP 头部来筛选 HTTP 流量时，便可以使用上述方法来构造 HTTP 显示过滤参数。

10.6.3 幕后原理

在 HTTP preference 窗口中，设立了几个含 reassemble（重组）字样的复选框。设立的原因是，IP 分片的情况时有发生，一旦发生，承载 HTTP 流量的 TCP 报文段也会分布在隶属于同一个 IP 数据包的不同分片内。只要勾选了那几个复选框，Wireshark（在解码时）会重组分布在各 IP 数据包的不同分片内的 HTTP 头部和主体。

10.6.4 拾遗补缺

通常，Wireshark 只会把目的端口号为 80，且包含有效 HTTP 头部的 TCP 数据包识别为 HTTP 数据包^①。要想让 Wireshark 把所有目的端口号为 80 的 TCP 流量都识别为 HTTP 数据包，请按以下步骤行事。

1. 点击 **Edit** 菜单下的 **Preferences** 菜单项，在弹出的 Preferences 窗口中，点击 **Protocol** 选项前的+号，在其中选择 **TCP** 协议。
2. 在 TCP preference 窗口中，取消勾选 **Allow dissector to reassemble TCP streams** 之后的复选框。

10.7 HTTP故障分析

本章将重点关注如何分析 HTTP 故障。HTTP 故障的原因包括：Web 服务器或 HTTP 客户端主机反应迟钝、TCP 性能问题以及本章将要呈现的某些其他原因。

10.7.1 准备工作

若有个别用户反映网页打开过慢，请借助于端口镜像功能，将交换机上连接该用户主机的端口的流量重定向给 Wireshark 主机；若用户普遍反映网页打开过慢，则需要将交换机上连接 Internet 链路的端口的流量重定向给 Wireshark 主机。

10.7.2 操作方法

网页打开过慢的原因有很多，请按以下思路来查找原因。

^① 译者注：“Usually Wireshark shows dissected packets with port 80 as HTTP only if it sees a valid HTTP header”。对于这种文字，译者直接无语。

1. 首先，要确保 Internet 链路没有过载。其次，应观察连接 Internet 链路的交换机端口的统计信息，确保未发生任何与通信链路有关的问题（比如，收到了错包或存在丢包）。
2. 要判断网页打开过慢是否与 TCP 性能问题有关，请按以下步骤行。
 - 检查 Wireshark 抓包文件，看看是否抓到了数量众多的身背 duplicate ACK 或 retransmissions 等字样的 TCP 报文段（在正常情况下，身负上述字样的 TCP 报文段的数量不应超过 TCP 报文段总量的 0.5%~0.8%）。
 - 检查 Wireshark 抓包文件，看看是否抓到了重置 HTTP 连接的 TCP 报文段（RST 标记位置 1 的 TCP 报文段）。若有，则有可能是防火墙在捣鬼或是目标 Web 站点有访问限制。
3. 要判断网页打开过慢是否归因于 DNS 故障，可能存在的 DNS 故障如下所列。
 - DNS 解析缓慢。
 - DNS 服务器无法解析域名，或有待查询的域名不正确。
4. 经过一翻排查之后，若发现网页打开过慢与上述问题无关，请将排障目标对准 HTTP 协议。

**注意**

排除网络故障时，应将视野放宽到整个网络环境甚至是整个 IT 环境。HTTP 与 TCP 息息相关，而 DNS 解析速度则与用户对互联网应用的体验密不可分。因 Web 服务器反应迟钝而导致的 TCP 重传，会使得网页打开过慢；而 DNS 服务器反应迟钝，也会使得网页打开看似变慢。因此，需遵循上述思路，按部就班地解决问题。

若用户普遍反应网页打开过慢，应检查如下事宜。

1. 检查通信链路（Internet 链路）是否过载。
2. 检查通信链路上的传输延迟（在用户主机上 ping Internet 上的 Web 站点）。
3. 设法查明 Web 站点传回的 HTTP（响应数据包中包含的）信息状态码。当使用浏览器打开网页时若发生了 HTTP 访问错误，浏览器一般都会显示出由 Web 服务器传回的 HTTP 错误状态码，但并非总是如此。
4. 在 Wireshark 抓包主窗口的 Filter 输入栏内输入显示过滤表达式 `http.response >= 400`，看看 Wireshark 抓到了多少含有 HTTP 错误状态码的数据包。在本章后续内容中，会举几个与此有关的重要示例。

HTTP 信息状态码

状态码	原因短语	解 释	应对措施
100	Continue	表示 HTTP 请求成功执行完毕，会话可以继续	
101	Switching protocols	表示 Web 服务器根据 HTTP 客户端的指定，将协议切换为 Upgrade 头部中列出的协议	

HTTP 成功状态码

状态码	原因短语	解 释	应对措施
200	OK	HTTP 请求正常	
201	Created	表示 HTTP 请求已经执行，新的资源已经创建	
202	Accepted	表示 HTTP 请求已被接受，仍在处理	
203	Non-Authoritative information	表示实体首部所含信息并非来自于源端服务器，而是来自资源的一份副本	
204	No content	表示 HTTP 响应数据包中只包含若干 HTTP 头部和一状态行，不含实体的主体部分	
205	Reset content	Web 服务器会用该状态码告知浏览器清除当前页面中所有表单元素	
206	Partial content	表明 HTTP 请求部分成功执行	

HTTP 重定向状态码

状态码	原因短语	解 释	应对措施
300	Multiple choices	当 Web 服务器返回该状态码时，便表示(HTTP 客户端所请求的) URL 实际指向多个资源(譬如，所请求的 HTML 文件有英语和法语两个版本)。Web 服务器在返回该状态码的同时，还会提供一个选项列表，让用户选择	
301	Moved permanently	表示 HTTP 客户端所请求的资源被永久性转移，Web 服务器应在 HTTP 响应数据包的 Location 头部中包含资源目前所处的位置(URL)	
302	Moved temporarily (found)	表示 HTTP 客户端所请求的资源被临时转移，Web 服务器应在 HTTP 响应数据包的 Location 头部中包含资源目前所处的位置(URL)，但将来对资源的请求应使用原先的 URL	在遇到这种情况时，在 Web 浏览器中通常只会显示原因短语 Found。然后，Web 浏览器会另发一个 HTTP 请求数据包，其 GET 方法所引用的 URL 将指向资源目前所处的位置
303	See other	Web 服务器会用此代码告知 HTTP 客户端：请用另一 URL 来获取资源。新的 URL 位于 HTTP 响应数据包中的 Location 头部	
304	Not modified	若 HTTP 客户端发起的是条件 GET 请求，且所请求的资源最近未被修改，Web 服务器便会返回此状态码	
305	Use proxy	Web 服务器用此代码来表示必须通过某一代理设备来访问所请求的资源，代理的位置位于 HTTP 响应数据包中的 Location 头部	请检查相关代理设备的情况

HTTP 客户端错误状态码

状态码	原因短语	解释	应对措施
400	Bad request	表示 Web 服务器无法识别 HTTP 请求数据包中的语法。HTTP 客户端在重发 HTTP 请求之前，应做修改	请检查所要访问的 Web 站点的域名或 IP 地址是否正确
401	Authorization required	HTTP 客户端对资源的访问遭 Web 服务器拒绝，因为前者未能通过认证	请检查用户名/密码
402	Payment required	预留供将来使用	
403	Forbidden	表明 HTTP 请求遭 Web 服务器拒绝，这可能是因为 Web 服务器有访问限制	请检查访问凭证。另外，也有可能是因为 Web 服务器负载过重
404	Not found	表示所请求的资源不存在（于 Web 服务器）	这既有可能是因为资源遭到了删除，或原本就不存在，也有可能是因为 URL 拼写错误
405	Method not allowed	表示 Web 服务器不支持或不允许 HTTP 客户端用来请求资源的 HTTP 方法	
406	Not acceptable	HTTP 客户端可通过参数来说明自己愿意接受何种类型的实体。当 Web 服务器上的资源与 HTTP 客户端可接受的 URL 不匹配时，便会返回此状态码	请检查/升级 Web 浏览器
407	Proxy authentication	类似于状态码 401，但为客户端代理服务器所用，该服务器的作用是对 HTTP 客户端所要访问的资源进行认证	HTTP 客户端在访问资源之前需先通过代理服务器的认证
408	Request timed out	Web 服务器在处理 HTTP 请求时若超出了所允许的时间，便会返回此状态码	检查响应时间及网络负载状态
409	Conflict	表示由 HTTP 客户端所提交的 HTTP 请求因与某些既定的规则相冲突，而未能完成	可能是通过 HTTP 上传的文件旧于 Web 服务器所存，或存在与之类似的问题。请检查 HTTP 客户端的所作所为
410	Gone	类似于状态码 404，只是 HTTP 客户端通过 URL 所请求的资源以前确实存在	一般都是 Web 服务器问题，既有可能是文件被删除，也有可能是文件存放位置发生了改变
411	Content length required	表示 HTTP 客户端发出的 HTTP 请求数据包缺少 Content-Length 头部，但 Web 服务器要求在 HTTP 请求数据包中包含这一头部	Web 浏览器与 Web 服务器不兼容，请升级 Web 浏览器
412	Precondition failed	若 HTTP 客户端发起的是条件 GET 请求，且启用某一条件无法满足时，Web 服务器会返回此状态码	WEB 浏览器与 Web 服务器不兼容，请升级 Web 浏览器
413	Request entity too long	当 HTTP 客户端所发送的 HTTP 请求数据包中的实体的主体部分过长，Web 服务器无法处理时，便会返回此状态码	Web 服务器限制

续表

状态码	原因短语	解释	应对措施
414	Request URI too long	当 HTTP 客户端所发送的 HTTP 请求数据包中的 URL 过长，Web 服务器无法处理时，便会返回此状态码	Web 服务器限制
415	Unsupported media type	若 Web 服务器不支持或无法识别 HTTP 客户端所发（HTTP 请求数据包中的）实体内容的类型，则会返回此状态码	Web 服务器限制

图 10.21 呈现了 Wireshark 识别出的 HTTP 客户端错误的简单示例。要想让 Wireshark 生成图 10.21 所示内容，请按以下步骤行事。

- 1. 在 Wireshark 抓包主窗口的数据包列表区域中，选中一个身背 HTTP 4xx 状态码的数据包，点击右键。
- 2. 在弹出的右键菜单中点击 **Follow TCP Stream** 菜单项，将会看到图 10.21 所示的窗口。

```
GET /poker-client/broadcast.htm HTTP/1.1 ①
Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, application/x-ms-application, application/x-ms-xbap, application/vnd.ms-xpsdocument, application/xaml+xml, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Referer: http://www.888poker.com/poker-client/promotions.htm ②
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; GTB7.1; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; .NET CLR 1.1.4322; .NET CLR 2.0.50727; OfficeLiveConnector.1.3; OfficeLivePatch.0.0; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; InfoPath 1)
Host: www.888poker.com ③
HTTP/1.1 404 Not Found ④
Date: Sun, 16 Oct 2011 09:11:58 GMT
Server: Microsoft-IIS/6.0
srv: 2344432
```

图10.21

- 3. 现在，来解读一下图中所示内容。
 - HTTP 客户端所要访问的网址为 `www.888poker.com//poker-client/broadcast.htm`（在图 10.21 中分别被标记为①和③）。
 - 访问此网址的 HTTP 请求经由网址 `http://www.888poker.com/poker-client/promotions.htm` 链接而来（在图 10.21 中被标记为②）。
 - Web 服务器返回的 HTTP 状态码为 **404 Not Found**（在图 10.21 中被标记为④）。

需要澄清的是，作者从不赌博，只是在用 Wireshark 执行排障任务时顺手截了个屏（`www.888poker.com` 是一个赌博网站）。

HTTP 服务器错误状态码

状态码	原因短语	解 释	应对措施
500	Internal server error	Web 服务器遭遇意外情况时，便会返回此状态码。所谓意外情况，是指服务器因出错而无法完成 HTTP 客户端对某一 URL 的访问请求	当 Web 服务器上运行有 CGI 程序时，Perl 代码中的错误会导致其返回此状态码
501	Not implemented	表示 Web 服务器未能执行 HTTP 客户端的访问请求	Web 服务器故障
502	Bad gateway	充当代理或网关的服务器从请求响应链中的下一个环节收到无效响应（比如，该服务器无法连接到其父网关）时，便会返回此代码	服务器故障
503	Service unavailable	表示有待访问的服务或请求访问的资源当前处于失效状态	Web 服务器故障
504	Gateway timeout	类似于状态码 408，但返回此状态码的是网关或代理设备	服务器宕机或停止响应
505	HTTP version not supported	表示 Web 服务器不支持（HTTP 客户端用来）与其通信的 HTTP 协议的版本	Web 服务器不支持（HTTP 客户端使用的）HTTP 协议的版本

浏览网页时，有诸多原因会导致 Web 浏览器提示“service unavailable (code 503)”错误。现在来举一个用 Wireshark 排除此类故障的示例。有一小型办公网络，里面总有人反映：用 Web 浏览器打开 Facebook 的主页，一点问题都没有，但打开之后，只要点主页里任何一个链接，就会弹出一个提示“service unavailable”的新页面^①。由图 10.22 所示的 Wireshark 截屏可知，是网络中的一台防火墙在捣鬼。

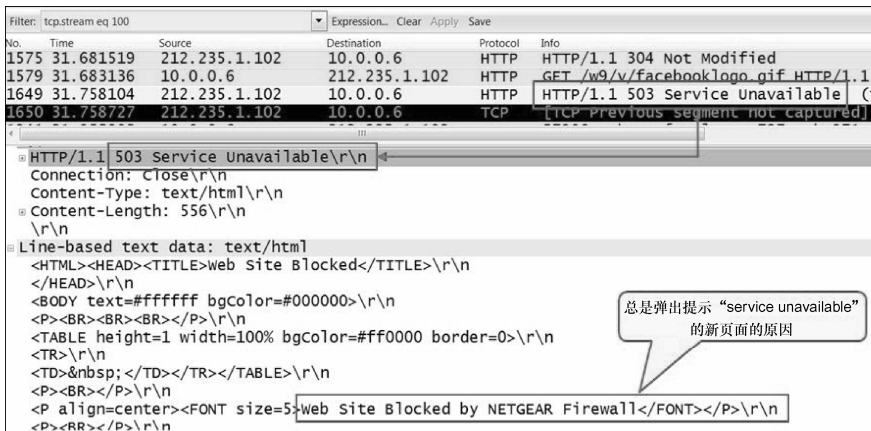


图10.22

① 译者注：原文是“In the following example there is a small office that has the following complaint: they can browse Facebook, but the moment they click on a link on this site, they get the new page as blocked”。真不敢想象，连故障现象都描述不清的人是怎么处理故障的？

10.7.3 幕后原理

通过 Web 浏览器，以 HTTP 方式打开网页的同时，用 Wireshark 抓包，能观察到下述情况。

1. TCP 连接的建立过程（TCP 三次握手过程）。
2. 身负 GET 字样的 HTTP 请求数据包。
3. 用来承载（下载）数据的 TCP 数据包。



注意

在大多数情况下，用 Web 浏览器访问某个网站的主页时，可能会建立 10 条以上的 HTTP 连接。比方说，在访问某些新闻站点的主页（比如，www.cnn.com、www.foxnews.com、www.bbc.co.uk）时，Web 浏览器会同时打开多条 HTTP 连接，分别用来加载其经济、体育、生活、天气等各个“频道”的数据。甚至会出现访问某个页面时，Web 浏览器打开上百条 HTTP 连接加载数据的情况。

对一个需要通过多条 HTTP 连接才能加载完毕的 Web 页面来说，在打开每条 HTTP 连接时，都会衍生出 DNS 查询/响应数据包、TCP “三次握手”报文段、包含 HTTP GET 方法的 HTTP 请求数据包，但 Web 浏览器中显示的内容都是由后续的 TCP 数据包承载。

10.7.4 拾遗补缺

用 Wireshark 分析抓到的 HTTP 数据包时，若在抓包主窗口的数据包内容区域看不到有用的信息，请在数据包列表区域选中一个隶属于某条 HTTP 连接的数据包，点击右键，选择右键菜单中的 **Follow TCP Stream** 菜单项^①。Wireshark 会立刻弹出一个 Follow TCP Stream 窗口，其中会显示出通过该 HTTP 连接传递的全部数据。

还有一款常用来分析 HTTP 流量的流量分析攻击，名叫 **Fiddler**，下载链接为 <http://fiddler2.com/>。Fiddler 属于自由软件，其主要用途为 HTTP debugging，如何使用该工具不在本书探讨范围之内。

10.8 导出 HTTP 对象

在 Wireshark 抓包主窗口的 **File** 菜单的 **Export Objects** 菜单项下，有一个 **HTTP** 子菜单项，可利用该子菜单功能从抓包文件中导出有关 HTTP 的统计信息（即通过 HTTP 访问的 Web 站点信息和资源信息）^②。

^① 译者注：原文是“When you don't see anything in the packet details pane, right-click on a packet and choose Follow TCP stream。”作者的表达能力真让人着急。

^② 译者注：原文是“Exporting HTTP objects is a simple feature for exporting HTTP statistics—websites and files accessed by HTTP”。

10.8.1 操作方法

要想从 Wireshark 抓包文件中导出 HTTP 对象，请按以下步骤行事。

1. 请在使用 Wireshark 抓包的同时（或打开一个之前保存的抓包文件），选择抓包主窗口的 **File** 菜单，点击其 **Export Objects** 菜单项下的 **HTTP** 子菜单项。**HTTP Object List** 窗口会立刻弹出，如图 10.23 所示。

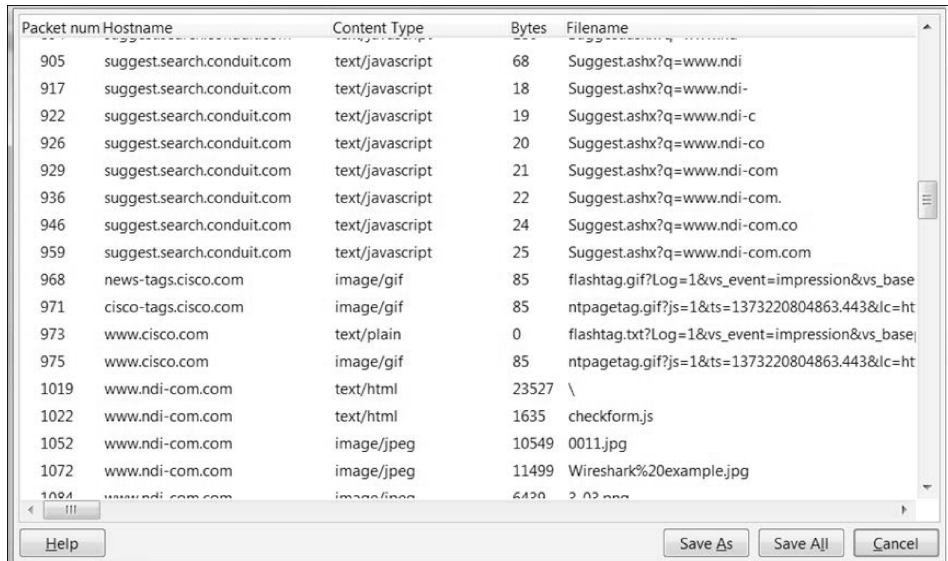


图 10.23

2. 在 HTTP Object List 窗口中，会列出被访问过的 Web 站点的名称，以及各 Web 站点上被访问过的文件信息（包括每个文件的类型、大小、名称等）。
3. 可点击 HTTP Object List 窗口底部的 **Save As** 或 **Save All** 按钮，来保存文件数据。
4. 在 HTTP Object List 窗口的 **Content Type** 一栏中，可能会出现的内容如下所示。
 - 文件类型 **text/plain**、**text/html**、**text/javascript**：表示通过 HTTP 访问的文件类型为文本文件，若为 **text/javascript**，则需仔细检查，因为可能存在安全隐患。
 - 文件类型 **image/jpeg**、**image/gif**：表示通过 HTTP 访问的文件类型为图像，可用图片浏览工具来打开此类文件。
 - 文件类型 **application/json**、**application/javascript** 等：表示通过 HTTP 访问的文件类型为应用程序。

- Wireshark 可识别的其他文件类型。



注意

要想让 **File** 菜单中 **Export Objects** 菜单项下的 HTTP 子菜单项功能生效，需先在 **TCP** 首选项设置窗口（点击 **Edit** 菜单中的 **Preferences** 菜单项，在弹出的 **Preferences** 窗口中，点击左侧 **Protocol** 之前的+号，选中 **TCP** 协议）中点选 **allow subdissector to reassemble TCP streams** 之后的复选框，激活 Wireshark 的 TCP 数据包重组功能。

还可点击图 10.23 中的 **Save All** 按钮，选择一个目录来存储从抓包文件中导出的所有 HTTP 对象。HTTP 对象既可以是图片文件（图中所示 1052 和 1072 号数据包所承载的内容），也可以是文本文件（1019 和 1022 号数据包所承载的内容），还可以是其他格式的文件。

10.8.2 幕后原理

只要点击 **File** 菜单中 **Export Objects** 菜单项下的 **HTTP** 子菜单项，Wireshark 就会扫描当前所抓数据包（或打开的抓包文件）中的 HTTP 数据流，对各种 HTML 对象（比如，HTML 文档、图片文件、可执行文件以及其他可读文件格式）进行重组，好让用户将这些对象存盘。稍后，可以使用适当的程序来读取这些 HTTP 对象（若为可执行文件，则可以通过双击的方式来执行）。Wireshark 所具备的这一功能是一把双刃剑，既能用来窃取机密，也可以起到备份的作用（比如，可用来备份通过 Email 发出的附件）。

10.8.3 拾遗补缺

下列软件也可以以图形化的方式来执行 Wireshark 所具备的上述功能：

- **Xplico** (<http://www.xplico.org/>);
- **NetworkMiner** (<http://www.netresec.com/?page=NetworkMiner>)。



注意

用 **File** 菜单的 **Export Objects** 菜单项下的 **HTTP** 子菜单项功能监控 HTTP 流量时，要是发现有人从路数不正的网站下载了可疑的应用程序，且应用程序的文件名也十分可疑时，请“百度（Google）”一下，这可能预示着危险（第 14 章会对此展开深入探讨）。

10.9 HTTP数据流分析及Follow TCP Stream窗口

本书前文已经简单介绍了 Wireshark 软件的 **Follow TCP Stream** 特性，该特性对网络监控非常有用，能让网管人员窥探到所抓 TCP 数据流的内在。本节会讲解如何借助于 **Follow TCP Stream** 特性分析 HTTP 流量。

10.9.1 准备工作

将 Wireshark 主机接入 LAN 交换机，激活交换机的端口镜像功能，把受监控端口的流量重定向至 Wireshark 主机；启动 Wireshark 软件，选择正确的网卡，开始抓包。

10.9.2 操作方法

要想在 Wireshark 软件中打开 Follow TCP Stream 窗口，请按以下步骤行事。

1. 在 Wireshark 抓包主窗口的数据包列表区域中，从有待分析的那股 HTTP 数据流中选择一个数据包，单击右键。
2. 在弹出的右键菜单中点击 **Follow TCP Stream** 菜单项，即可让 Wireshark 只显示隶属于这股 HTTP 数据流的数据包了。在 Wireshark 抓包主窗口的 Filter 输入栏内，不但可以看见这股 HTTP 数据流在抓包文件中的编号，还会弹出图 10.24 所示的窗口。

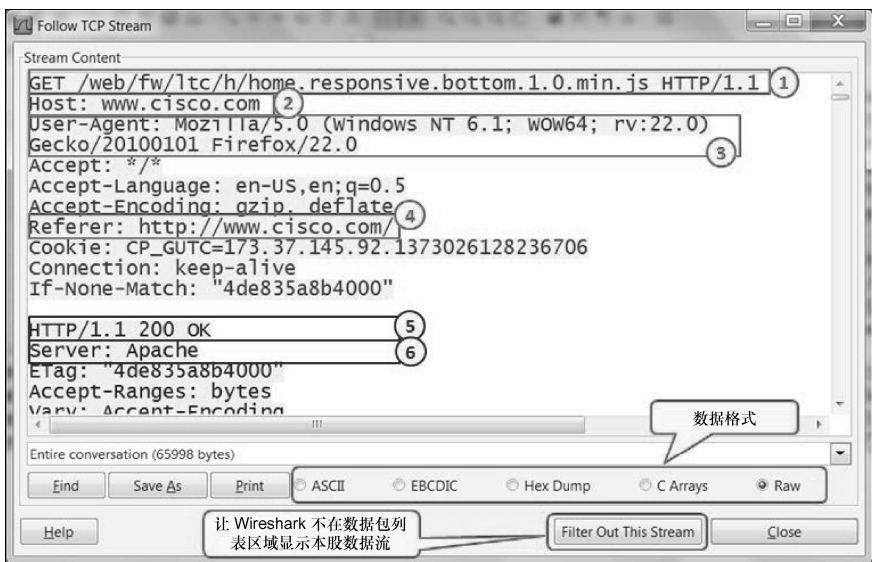


图 10.24

3. 借助于 Follow TCP Stream 窗口中的内容，可窥探到 HTTP 数据流的内在。由图 10.24 可知，这股 HTTP 数据流包含了以下内容。

- HTTP 客户端在执行 HTTP 请求操作时使用了 GET 方法（图 10.24 中的①）。
- HTTP 客户端所访问的主机名为 **www.cisco.com**（图 10.24 中的②）。
- 执行 HTTP 请求操作的浏览器为 Mozilla Firefox（图 10.24 中的③）。

- HTTP 请求数据包中 `referrer` 头部字段值为 `http://www.cisco.com` (图 10.24 中的④)。
- 在 HTTP 响应数据包中返回的信息状态码为 `220 OK` (图 10.24 中的⑤)。
- Web 服务器为 Apache Web 服务器 (图 10.24 中的⑥)。

4. 借助于 Follow TCP Stream 窗口，定位网络故障非常简单，再举两个例子。

- 可发现偷偷用 Kazaa (类似于 BT) 客户端“做种”的内网用户，如图 10.25 所示。

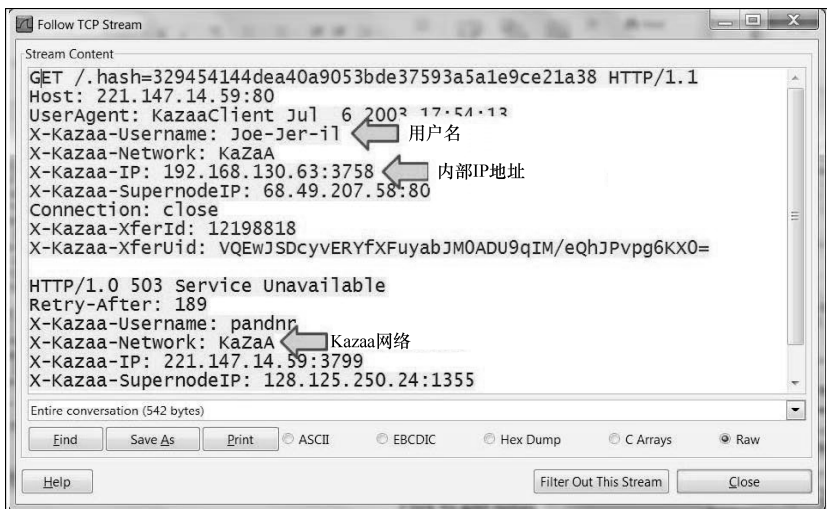


图10.25

- 可发现软件 bug，请看图 10.26。

只需把图 10.26 中圈里的文字放到百度 (Google) 里一搜，马上就可以知道这属于历史遗留性 bug。

5. 借助于 Follow TCP Stream 窗口，还可以发现：

- 错误或 bug 消息^①；
- 病毒或蠕虫——若在 Follow TCP Stream 窗口中看见了 `blast`、`probe` 或 `xprobe` 之类的单词，则十有八九与病毒或蠕虫有关。要是看见以上述单词命名的可执行文件 (扩展名为 `.exe`)，则更要保持十二分地警惕 (与此有关的内容详见第 14 章)。

① 译者注：原文是 “Error and bugs messages”，直译。



图10.26

10.9.3 幕后原理

通过 Follow TCP Stream 功能筛选出的 HTTP 数据流囊括了 HTTP 客户端与 Web 服务器之间从建立 HTTP TCP 连接的三次握手，到关闭 HTTP TCP 连接的四次握手之间的所有 TCP 报文段。该特性能够将 HTTP 数据流一股股的相互隔开，对排除网络故障大有裨益。

10.10 HTTPS协议流量分析——SSL/TLS基础

HTTPS 协议是 HTTP 协议的安全版本，HTTPS 中的那个 S 表示：HTTP 协议传输的安全性由 SSL/TLS（安全套接字层/传输层安全）来保证。在使用网银、Web 邮箱或基于 HTTP 协议但需安全性保障的应用时，就会用到 HTTPS 协议。

本节将介绍 HTTPS 协议的运作方式及其排障方法。

10.10.1 准备工作

将 Wireshark 主机接入 LAN 交换机，激活交换机的端口镜像功能，将受监控端口的流量重定向至 Wireshark 主机；运行 Wireshark 软件，选择正确的网卡，开始抓包。HTTPS 协议通过 TCP 443 端口来通信，应关注 Wireshark 抓到的源或目的 TCP 端号为 443 的所有流量。

10.10.2 操作方法

要使用 Wireshark 分析 HTTPS 协议流量，请按以下步骤行事。

1. 需掌握 HTTPS 握手阶段的步骤，与此有关的内容详见 10.10.3 节。

2. 应关注 Wireshark 抓到的 HTTPS 握手消息的顺序，应确保此类数据包按图 10.27 所示顺序出现在 Wireshark 抓包主窗口的数据包列表区域，这些数据包应身背图 10.27 中括号里的文字。

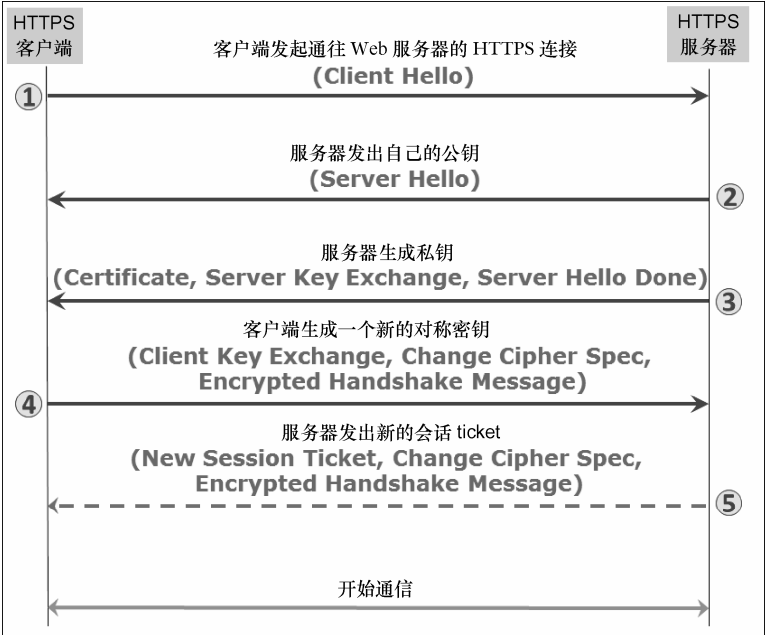


图10.27

3. 以下所列为 RFC 2246 中定义的各种常见的 HTTPS 告警（alert）消息。

- **close_notify**: 用来通知接收方，发送方将不会在该 HTTPS 会话上传输任何消息。在此之后，接收方将忽略收到的任何数据。
- **unexpected_message**: 当 HTTPS 客户端或服务器收到不应收到的消息时，便会返回此消息。这属于关键性错误告警消息，表示 HTTPS 客户端或服务器在实现上存在重大缺陷。
- **bad_record_mac**: 当 HTTPS 客户端或服务器收到了包含错误 MAC（消息认证码）的记录时，便会返回此消息^①。这属于关键性错误告警消息，表示 HTTPS 客户端或服务器在实现上存在重大缺陷。
- **decryption_failed**: 当 HTTPS 客户端或服务器收到了以错误的方法加密的

^① 译者注：所谓“记录”（record），是指通过 HTTPS 会话传输的数据，每一条记录都是由记录头部+经过加密的数据+MAC 构成。

TLS 密文时，便会返回此消息。这属于关键性错误告警消息，表示 HTTPS 客户端或服务器在实现上存在重大缺陷。

- **record_overflow:** 当 HTTPS 客户端或服务器收到的 TLS 密文记录的长度超出所允许的范围时，便会返回此消息。这属于致命性错误告警消息，表示 HTTPS 客户端或服务器在实现上存在重大缺陷。
- **decompression_failure:** 当 HTTPS 客户端或服务器无法对收到的记录进行解压缩时，便会返回此消息。这属于关键性错误告警消息，表示 HTTPS 客户端或服务器在实现上存在重大缺陷。
- **handshake_failure:** 表示 HTTPS 客户端和服务器在握手时无法就使用的安全参数达成一致意见。这属于关键性错误告警消息，表示 HTTPS 客户端或服务器在实现上存在重大缺陷。
- **bad_certificate:** 表示证书损坏、签名不正确或相关类似错误。
- **unsupported_certificate:** 表示接收方不支持发送方发出的证书类型。
- **certificate_revoked:** 表示证书被其签发者撤销。
- **certificate_expired:** 表示证书过期。
- **certificate_unknown:** 表示证书不被接受，但原因不明。
- **illegal_parameter:** 表示收到的握手消息中有字段越界或与别的字段不一致。这属于关键性错误告警消息，表示 HTTPS 客户端或服务器在实现上存在重大缺陷。
- **unknown_ca:** 表示收到了有效证书，但因与已知的受信 CA 不匹配而不被接受。这属于致命性错误告警消息，表示 HTTPS 客户端或服务器在实现上存在重大缺陷。
- **access_denied:** 表示收到了有效证书，但证书中的身份通不过访问控制检查。
- **decode_error:** 当 HTTPS 客户端或服务器收到的 SSL/TLS 消息过长或其中有字段越界，以至于无法解码时，便会返回此消息。这属于关键性错误告警消息，表示 HTTPS 客户端或服务器在实现上存在重大缺陷。
- **decrypt_error:** 表示握手加密操作失败，可能的情况有：签名验证失败、密钥交换失败、finished 消息验证失败等。

- **export_restriction:** 表示 SSL/TLS 实现违反出口限制。
- **protocol_version:** 此告警消息由 HTTPS 服务器发送，表示客户端使用了无法识别的协议版本。
- **insufficient_security:** 此告警消息由 HTTPS 服务器发送，表示客户端使用的加密套件的比服务器所要求的要弱。
- **internal_error:** 表示此告警消息的发送方遭遇内存分配或硬件故障等内部错误。
- **user_canceled:** 表示了 HTTPS 握手操作被用户取消，并非协议故障。
- **no_renegotiation:** HTTPS 客户端或服务器都可以发出此消息，来响应初始握手之后的 hello 请求消息。

只要遭遇上面提及的任何一种 HTTPS 告警信息，HTTPS 连接都无法建立。

10.10.3 幕后原理

SSL 和 TLS 协议都能用来保证某些特殊应用程序（比如，HTTP、SNMP、Telnet 等）的安全性。SSL 版本 1、2、3 是 Netscape 公司于 20 世纪 90 年代中期为其 Navigator 浏览器开发的，而 TLS 协议则是 IETF 的标准协议，先后定义于 RFC 2246、RFC 4492、RFC 5246 和 RFC 6176。TLS1.0 于 1999 年 1 月以 SSL 3.0 升级版的形式首次在 RFC 2246 中露面。

TLS 握手协议用来建立 TLS 连接，其规程如下所列。

1. 服务器与客户端相互交换 Hello 消息，就用来保护数据的加密算法的选定达成一致意见，同时交换用来生成密钥的随机数。
2. 服务器与客户端相互交换加密参数，就预主密钥的选用达成一致意见。
3. 服务器与客户端相互交换证书及加密信息，彼此认证。
4. 服务器与客户端根据预主密钥生成主密钥，同时交换随机数。
5. 服务器与客户端验证对方是否计算出了相同的安全参数，同时验证 TLS 握手本身有未被攻击者所乘。

上述规程按以下次序执行。

1. 服务器与客户端选择加密算法。
 - 客户端发出 **Client Hello** 消息（图 10.28 中用①来标记的数据包）。

- 服务器发出 **Server Hello** 消息（图 10.28 中用②来标记的数据包）。
- 2. 服务器与客户端相互认证，交换密钥（图 10.28 中用③和④来标记的数据包）。
- 3. 握手完成（图 10.28 中用⑤来标记的数据包）。

No.	Time	Source	Destination	Protocol	Info
157	16.866912	10.0.0.3	173.194.34.86	TCP	62900 > https [SYN] Seq=0 win=8192 Len=0 MSS=14
158	16.953453	173.194.34.86	10.0.0.3	TCP	https > 62900 [SYN, ACK] Seq=0 Ack=1 win=62920
159	16.953528	10.0.0.3	173.194.34.86	TCP	62900 > https [ACK] Seq=1 Ack=1 win=66792 Len=0
160	16.954763	10.0.0.3	173.194.34.86①	TLSv1	Client Hello
161	17.040545	173.194.34.86	10.0.0.3	TCP	https > 62900 [ACK] Seq=1 Ack=173 win=64000 Len
162	17.043587	173.194.34.86	10.0.0.3②	TLSv1	Server Hello
163	17.043715	173.194.34.86	10.0.0.3③	TLSv1	Certificate, Server Key Exchange, Server Hello
164	17.043790	10.0.0.3	173.194.34.86	TCP	62900 > https [ACK] Seq=173 Ack=1936 win=66792
165	17.066539	10.0.0.3	173.194.34.86④	TLSv1	Client Key Exchange, Change Cipher Spec, Encrypt
166	17.152661	173.194.34.86	10.0.0.3⑤	TLSv1	New Session Ticket, Change Cipher Spec, Encrypt
167	17.154064	10.0.0.3	173.194.34.86	TLSv1	Application Data
168	17.154412	10.0.0.3	173.194.34.86	TCP	[TCP segment of a reassembled PDU]
169	17.154416	10.0.0.3	173.194.34.86	TLSv1	Application Data
170	17.154515	173.194.34.86	10.0.0.3	TLSv1	Application Data

图 10.28

来看一下 HTTPS 协议的运作原理。由图 10.28 可知，要建立 SSL/TLS 连接，需先完成标准的 TCP 三次握手（图 10.28 中编号为 157~159 的数据包），本次 TLS 握手则始于编号为 160 的数据包。现在，来研究一下本次 TLS 握手的细节。

1. 客户端通过编号为 160 的数据包发出 **Client Hello** 消息，开始 TLS 握手协商。
2. 服务器回之以 **Server Hello** 消息（编号为 162 的数据包）。
3. 服务器向客户端发送证书（编号为 163 的数据包）。
4. 客户端接受证书，生成预主密钥。
5. 服务器创建主密钥，TLS 会话开始（可选消息，编号为 166 的数据包）。



注意

上面提到的步骤 5 是指 RFC 4507 中定义的一种机制，其作用是让 TLS 服务器在无需保存每客户端会话状态的同时，能迅速恢复会话。TLS 服务器可藉此机制，将会话状态封装进 ticket，并通过编号为 166 的数据包转发给客户端。客户端随后可使用自己获取到的 ticket 来恢复 TLS 会话。比方说，当使用浏览器恢复一条通往 webmail (Gmail、hotmail) 账户的 TLS 连接时，就会用到该机制，这种情况在实际的使用中非常常见。

在步骤 4 和 5 之后，服务器与客户端将开始实际的数据通信^①。

现通过 Wireshark 来观察 TLS 握手过程的每一步。

① 译者注：上述对 SSL/TLS 的介绍大多都是作者东拼西凑，抄袭而来，全无逻辑可言（其实整本书都是如此）。由于 SSL/TLS 协议太过复杂，译者只能大致按照原文字面意思翻译，虽会在译文上略作改动，但无力作大篇幅的调整。

在步骤 1 中，客户端发出 **Client Hello** 消息，这也是 TLS 握手过程中生成的首个数据包。

图 10.29 示出了 Client Hello 消息（其数据包编号为 160）所包含的内容。

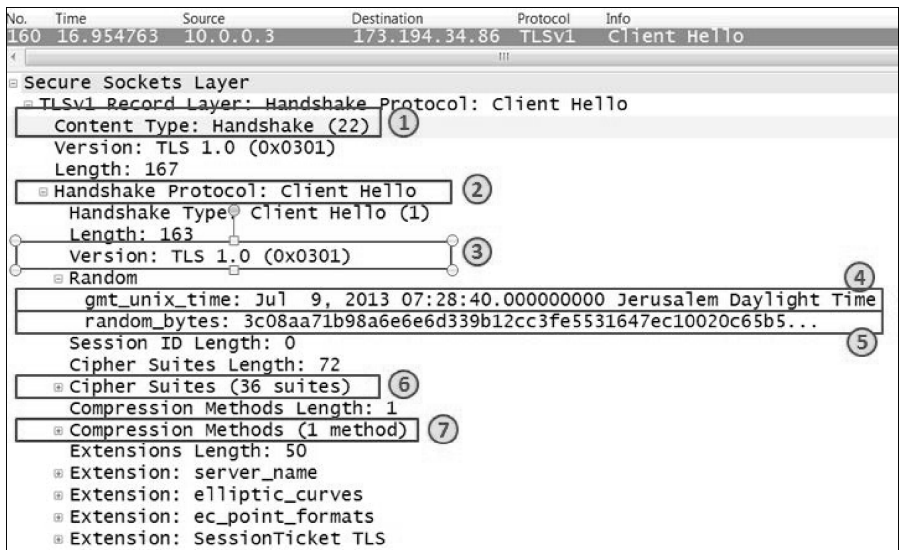


图10.29

- 图 10.29 所示区域①显示的是 TLS 数据包记录层头部的内容类型字段值。该字段值为 22，即表明此 TLS 数据包为握手协议消息（与之对应的显示过滤表达式为 `ssl.record.content_type == 22`）。
- 图 10.29 所示区域②显示的是 TLS 数据包中记录的类型字段值。该字段值为 1，即表明此 TLS 数据包包含由客户端发往服务器的 **Client Hello** 握手协议消息。
- 图 10.29 所示区域③显示的是客户端所支持的 TLS 协议的版本。
- 图 10.29 所示区域④显示的是将会在密钥生成过程中所使用的客户端时间。
- 图 10.29 所示区域⑤显示的是由客户端生成，将会在密钥生成过程中使用的随机数。
- 图 10.29 所示区域⑥显示的是客户端所支持的各种加密算法，优选使用的加密算法排列在先。
- 图 10.29 所示区域⑦显示的是客户端所支持的各种数据压缩方法。

在步骤 2 中，服务器将回复 **Server Hello** 消息。图 10.30 示出了 **Server Hello** 消息（其数据包编号为 162）的内容。

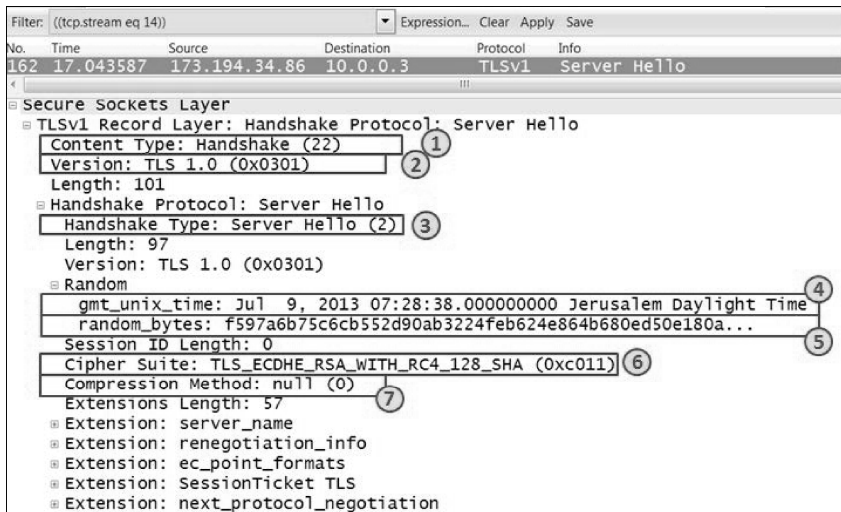


图 10.30

- 图 10.30 所示区域①显示的是 TLS 数据包记录层头部的内容类型字段值。该字段值为 22，即表明此 TLS 数据包为握手协议消息（与之对应的显示过滤表达式为 `ssl.record.content_type == 22`）。
- 图 10.30 所示区域②显示的是本次会话所使用的 TLS 协议的版本。
- 图 10.30 所示区域③显示的是 TLS 数据包中记录数据的类型字段值。该字段值为 2，即表明此 TLS 数据包包含由服务器发往客户端的 **Server Hello** 握手协议消息。
- 图 10.30 所示区域④显示的是将会在密钥生成过程中所使用的服务器端时间。
- 图 10.30 所示区域⑤显示的是由服务器生成，将会在密钥生成过程中使用的随机数。
- 图 10.30 所示区域⑥显示的是本次 TLS 会话所采用的加密算法，从客户端发出的加密算法列表中选取。
- 图 10.30 所示区域⑦显示的是本次 TLS 会话所采用的数据压缩方法。

在步骤 3 中，服务器将会向客户端发送证书（见图 10.31）。

- 图 10.31 所示区域 1 显示的是服务器发送的 **Certificate** 命令，其中包括了服务器的证书。点击左边的“+”号，可以了解到证书的发布者、有效期以及其他信息。
- 图 10.31 所示区域 2 显示的是服务器发送的 **Server Key Exchange** 消息（通常都使用 Diffie-Hellman 算法来完成密钥的交换），其中会包括必要的参数（公钥、签名等信息）。

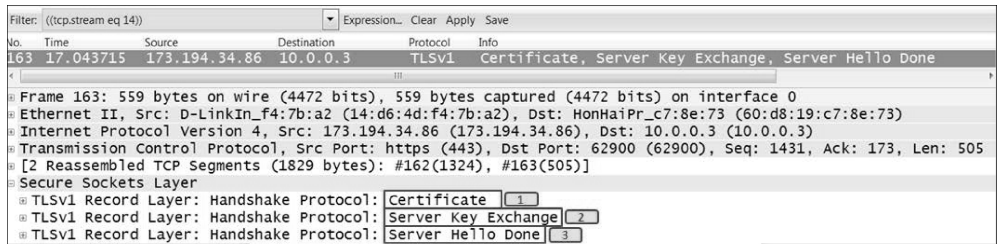


图10.31

- 图 10.31 所示区域 3 显示的是服务器发送的 **Server Hello Done** 消息，表示服务器完成了本阶段的 TLS 握手。

在步骤 4 中，客户端将接受服务器发送的证书，同时生成预主密钥（见图 10.32）。

- 图 10.32 所示区域 1 显示的是客户端发送的 **Client Key Exchange** 消息，其中包含有由客户端创建的预主密钥，该预主密钥在发送时会以服务器发送的公钥加密。服务器和客户端会根据 **Client Hello** 和 **Server Hello** 消息中的数据，来生成对称加密密钥。
- 图 10.32 所示区域 2 显示的是客户端向服务器发出的 **Change Cipher Spec** 消息。这表示客户端要求服务器在后续通信中启用加密模式。

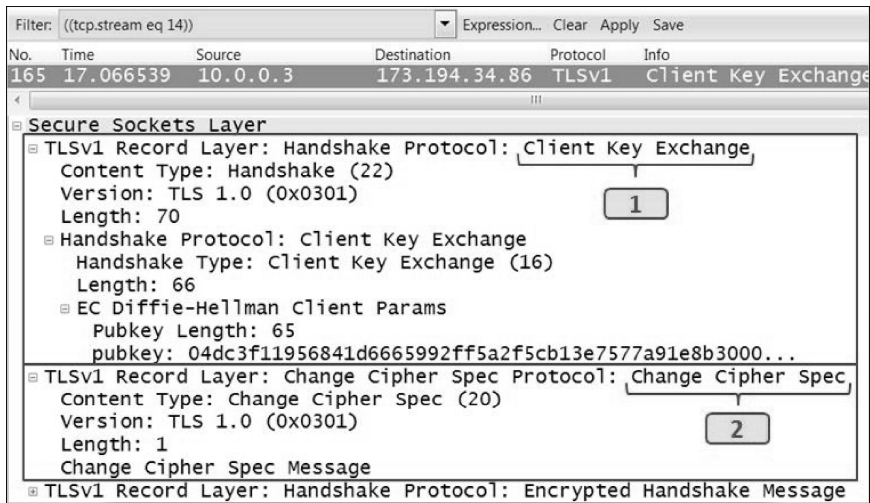


图10.32

在步骤 5 中(这一步为可选)，服务器向客户端发送 **New Session Ticket** 消息，如图 10.33 所示。

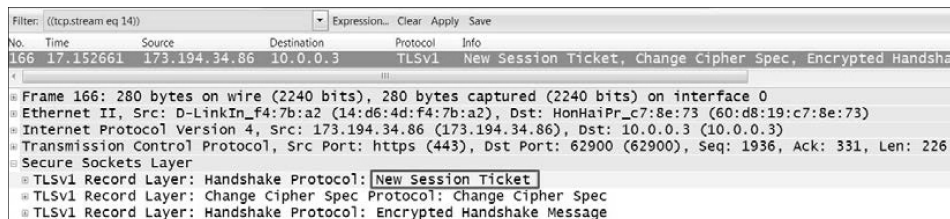


图10.33

10.10.4 拾遗补缺

有个问题作者曾被多次问及，该问题是：用 SSL/TLS 加密的信息有被破解的可能吗？只要握有目标服务器所提供的私钥，当然可以破解通过 SSL/TLS 加密的信息，不过，要想得到这一私钥却不太容易。

窃取这一私钥的方法有很多，在某些情况下，这些方法还非常奏效。显而易见，与此有关的内容不在本书探讨范围之列。如真能获取这一私钥，请点击 **Edit** 菜单中的 **Preferences** 菜单项，在弹出的 **Preferences** 窗口中，点击左边的 **Protocol** 配置选项前的“+”号，单击 **SSL** 协议，在右边的 **Pre-Shared-Key** 对话框中输入该密钥，就能解密 Wireshark 抓到的经过 SSL/TLS 加密的数据包了。与此有关的内容详见链接 <http://wiki.wireshark.org/SSL> 及其他 Web 站点和 blog。

第 11 章

企业网应用程序行为分析

本章涵盖以下主题：

- ▶ 摸清流淌于网络中的流量的类型；
- ▶ FTP 故障分析；
- ▶ Email 协议（POP、IMAP 及 SMTP）故障分析；
- ▶ MS-TS 和 Citrix 故障分析；
- ▶ NetBIOS 协议故障分析；
- ▶ 数据库流量及常见故障分析。

11.1 简介

Wireshark 存在的最大价值之一，就是能用它来分析流淌于网络中的各种应用程序的流量，从而为排除应用程序故障提供依据。当应用程序运行缓慢时，可能的原因包括：LAN 问题（一般都与无线 LAN 有关）、WAN 问题（WAN 链路带宽不足或延迟过高）、安装应用程序服务器端/客户端软件的主机运行缓慢（TCP 窗口问题），以及应用程序自身问题等。

本章将深入探讨企业网内常用的应用程序的运作方式，同时会介绍如何定位及解决与这些应用程序有关的故障。首先，会介绍如何探查及归类流淌于网络中的流量的类型。其次，会分析各种企业网应用程序的运作方式，介绍不同的网络状况对那些应用程序的影响。

本章还会介绍如何使用 Wireshark 解决常见诸于企业网内的各种应用程序的故障。这些应

用程序包括：FTP 应用、各种 Email 应用、数据库、NetBIOS 应用、Microsoft 终端服务器和 Citrix 等。

11.2 摸清流淌于网络中的流量的类型

在一个全新的网络环境中处理故障时，第一件事就是要摸清流淌于网络中的流量的类型。运行于网络中的应用程序五花八门，支撑应用程序的协议类型多种多样，它们之间还会彼此影响，相互干扰。

在某些情况下，网络中还会有多个 VLAN，存在多个虚拟转发实例（VRF），部署多座刀箱（**Bladeserver**），每个刀箱里还会配备连接到虚拟背板交换机端口的刀片服务器。在同一网络基础设施里运行的每一样东西都有可能会相互影响。



注意

很多人都分不清 VLAN 和 VRF 的区别。两者的用途虽大致相同，但使用场合却全然不同。VLAN 用在 LAN 中行使第一、二层的隔离任务；VRF 则是指共存于同一台路由器上的多个路由表实例，常用在 SP 网络中行使第三层的隔离任务。在 SP 网络中，VRF 一般都会跟多协议标签交换（MPLS）技术结合使用，用来提供同一客户不同站点间的 IP（第三层）连通性，同时还能实现不同客户间的网络隔离。

本节会探讨如何借助 Wireshark，来洞察流淌于网络中的流量的类型。



注意

术语“刀箱”（**Bladeserver**）是指刀片服务器的机箱，其正面安装的是刀片服务器，后面则配有 LAN 交换机。这一称谓随厂商而异，IBM 称其为刀片中心（**Bladecenter**），HP 则将其命名为刀片系统（**Bladesystem**）。

11.2.1 准备工作

在一个全新的网络环境中处理故障时，首先应把 Wireshark 主机接入网络，通过抓包来弄清网络中究竟流淌着哪些流量。在执行上述操作时，请按以下原则行事。

- 若故障涉及一台服务器，则应在 LAN 交换机上开启端口镜像功能，将连接该服务器的端口的所有流量重定向给 Wireshark 主机，以此来弄清该服务器都接收或发送了何种类型的流量。
- 若故障涉及一远程分支机构网络，则应设法将路由器上连接该分支机构网络的 WAN 端口的流量重定向给 Wireshark 主机，以此来弄清进、出该分支机构网络的流量的类型。

- 若故障涉及 Internet 链路（比如，用户普遍反应网页打开缓慢），则应设法将奔流于该 Internet 链路的所有流量重定向给 Wireshark 主机，以此来弄清该 Internet 链路究竟承载了何种类型的流量。

本节会介绍如何借助 Wireshark 软件，来摸清流淌于网络中的流量的类型，以及如何排除相应的网络故障。

11.2.2 操作方法

要借助 Wireshark 软件来分析流淌于网络中的流量的类型，请按以下步骤行事。

1. 按照之前提及的三原则之一，将相关流量重定向给 Wireshark 主机。
2. 可利用 Wireshark 软件自带的下列工具，来分析所抓流量。
 - 利用 **Statistics** 菜单下的 **Protocol Hierarchy** 工具，来观察并了解所抓流量的协议类型及占比情况。
 - 利用 **Statistics** 菜单下的 **Conversations** 工具，来了解所抓流量在各个层级上的源头和归宿。
3. 凭借 **Protocol Hierarchy** 工具所生成的信息，能获知流淌于网络中的流量的类型，如图 11.1 所示。

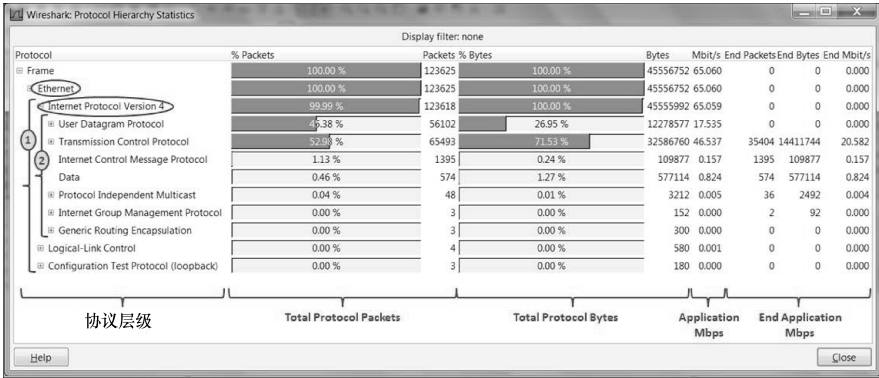


图 11.1

4. 通过图 11.1，可了解到穿梭于网络中的各种流量的分布情况。
 - **以太网帧**：其麾下的子协议流量包括：IPv4 流量、逻辑链路控制协议（Logical-Link Control）流量，以及配置测试协议流量（Configuration Test Protocol [loopback]）。

- **IPv4 (Internet Protocol Version 4) 数据包**: 其麾下的子协议流量包括: UDP 流量、TCP 流量、PIM 流量、IGMP 以及 GRE 流量等。

5. 若点击各 **Protocol** 左边的“+”号, 则会显示出其麾下所有类型的上层协议流量。
6. 要想了解某种协议流量的吞吐量, 请点击其“父协议”左边的“+”号, 直至该协议暴露为止, 如图 11.2 所示。通过图 11.2, 可获知 HTTP 流量在本次抓包时间段内的平均传输速率。

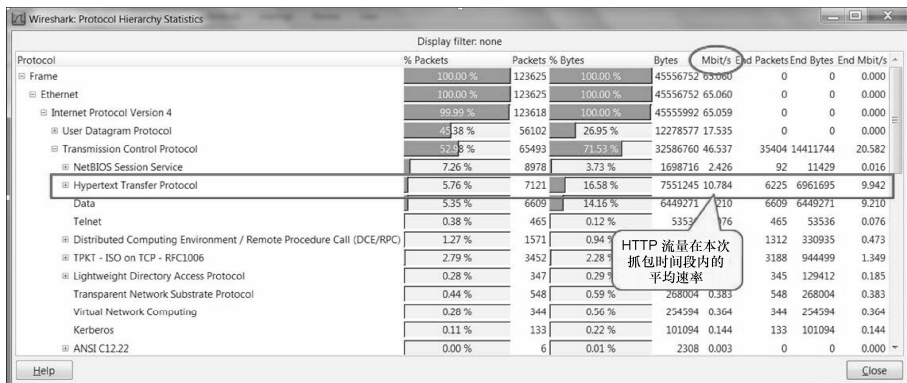


图 11.2

7. 若再点击 Hypertext Transfer Protocol 协议左边的“+”号, 则可了解到经由 HTTP 封装的各种数据 (XML、MIME、JavaScripts 等) 在本次抓包时间段内的平均传输速率。

11.2.3 拾遗补缺

在某些情况下 (比如, 准备网络统计报表, 向领导汇报时), 需要以图形方式来呈现与网络的各种性能指标有关的统计信息。可使用以下工具来完成这项工作。

- **Etherape (用于 Linux)**: <http://etherape.sourceforge.net/>。
- **Compass (用于 Windows)**: http://download.cnet.com/Compass-Free/3000-2085_4_-75447541.html?tag=mncol;1 (来自 Wildpackets)。

11.3 FTP故障分析

文件传输协议 (**FTP**) 的主要用途是跨 TCP/IP 网络传输文件。它是一种运行于 TCP 20 和 21 端口的协议, 这两个端口分别用来传输数据和发送控制命令。

FTP 有以下两种操作模式。

- **Active 模式 (ACTV):** 当 FTP 以此模式运行时, FTP 客户端会先发起一条通往 FTP 服务器端的控制连接, 后者随后会主动发起一条通往前者的数据连接。
- **Passive 模式 (PASV):** 当 FTP 以此模式运行时, FTP 客户端会先后发起通往 FTP 服务器端的控制和数据连接各一。

FTP 的这两种模式都很常用, 11.3.4 节将细述各自的原理。

11.3.1 准备工作

处理 FTP 故障时, 若怀疑存在连通性故障, 或发现数据传输缓慢等迹象, 请在 LAN 交换机上启用端口镜像功能, 将下列端口的流量重定向至 Wireshark 主机:

- 连接 FTP 服务器的端口;
- 连接 FTP 客户机的端口;
- 连接 FTP 流量穿越链路的端口。

必要时, 还需在 Wireshark 中应用显示及抓包过滤器。

11.3.2 分析方法

排查 FTP 性能问题时, 应遵循以下排障思路。

1. 首先, 应确保在第二、三、四层方面 (以太网、IP、TCP) 没有故障, 前面几章已经介绍过了解决第二、三、四层故障之法。在许多情况下, 数据传输缓慢都与第二、三、四层故障有关, 跟 FTP 应用本身无关。
2. 检查 Wireshark 抓包文件中是否有身背 TCP retransmissions 和 duplicate ACKs 字样的数据包。若有, 则需进一步查明这样的数据包是只涉及 FTP 流量, 还是涉及所有 TCP 协议流量^①。若涉及所有 TCP 协议流量, 则说明不单是 FTP 流量传输缓慢, 而是整个网络都有故障。若身背 TCP retransmissions 和 duplicate ACKs 字样的数据包只涉及 FTP 流量, 且那些数据包的源或目的 IP 地址相同, 则说明 FTP 流量传输缓慢很可能是拜 FTP 服务器或客户机反应慢所赐。
3. 通过 FTP 传输文件时, 若使用 Wireshark 抓包, 则相应的流量在 IO Graphs 窗口中将会呈直边梯形的趋势, 在 TCP Graph (time-sequence) 窗口中将会出现一条笔直的斜线。

^① 译者注: 原文是 “Check for TCP retransmissions and duplicate ACKs. Check if they are on the entire traffic or only on the FTP connection”。

4. 图 11.3 (TCP graph[time-sequence]窗口) 呈现了一次有问题的 FTP 传输过程。

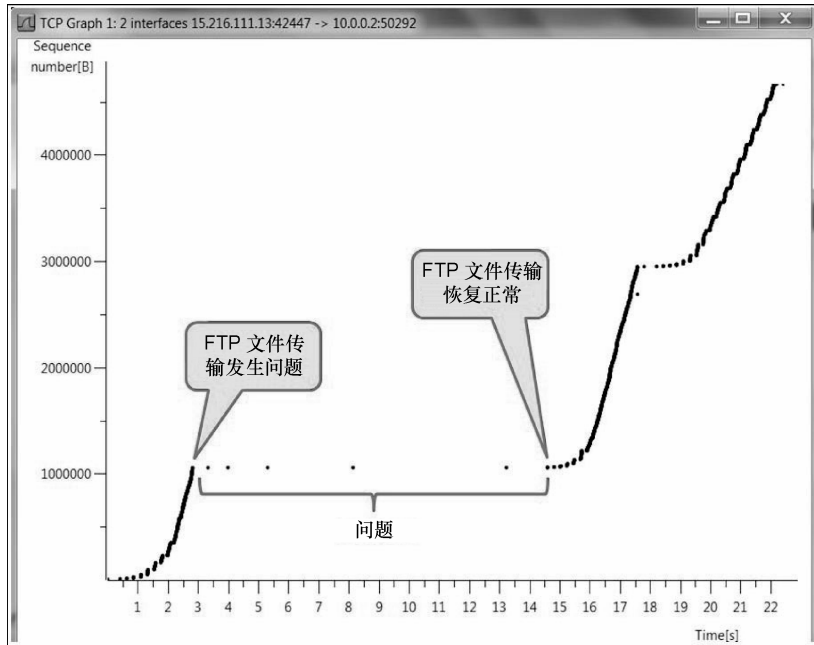


图 11.3

5. 现在来看看这次有问题的 FTP 文件传输在应用了适当的显示过滤器的 IO Graphs 窗口中呈现出的图形 (见图 11.4)。

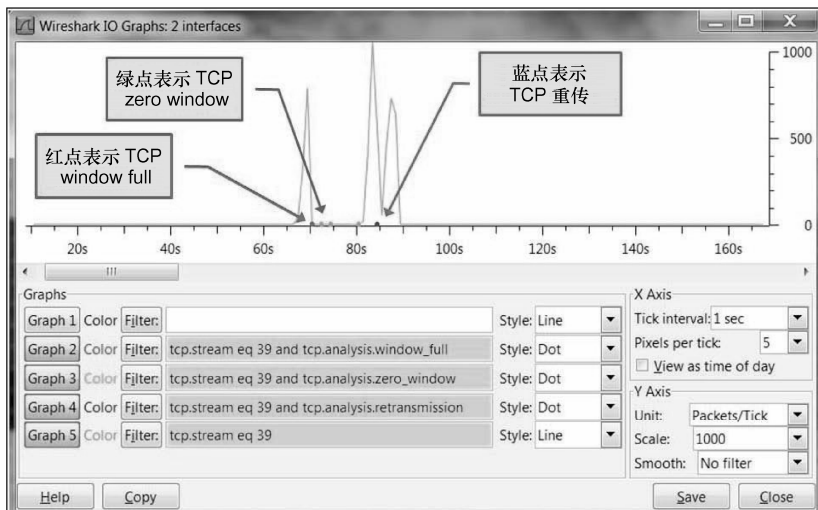


图 11.4

6. 图 11.5 所示为由这次 FTP 文件传输所生成的数据包在 Wireshark 抓包主窗口的数据包列表区域中的真面目。由图 11.5 可知, 这些数据包被 Wireshark 识别为存在以下 TCP window 问题。

- FTP 服务器 15.216.111.13 向 FTP 客户端 10.0.0.2 发出了身背 TCP Window Full 字样的 TCP 报文段 (编号为 5763 的数据包), 也就是说, Wireshark 已识别出前者发出的这一 TCP 报文段, 将会是填满后者的接收缓存的最后一个 TCP 报文段。
- FTP 客户端 10.0.0.2 向 FTP 服务器 15.216.111.13 发出了窗口大小字段值为 0 的 TCP 报文段 (编号为 5778 的数据包, 身背 TCP Zero Window 字样), 其目的是让后者停止传送数据。
- FTP 服务器持续向 FTP 客户端发出了身背 TCP Zero Window Probe 字样的 TCP 报文段, 意在探测后者的 TCP 接收窗口是否仍然为 0。FTP 客户端发出窗口大小字段值为 0 的 TCP 确认报文段 (身背 TCP Zero Window Probe Ack 字样), 其目的是请求 FTP 服务器暂停发送数据。FTP 服务器和客户端之间的上述“交流”过程, 详见 Wireshark 抓到的编号范围为 5793~5931 的数据包。
- 片刻之后, FTP 客户端向 FTP 服务器发出了窗口大小字段值不为 0 的 TCP 确认报文段 (编号为 5939 的数据包, 身背 TCP Window Update 字样), 让后者继续传输数据。

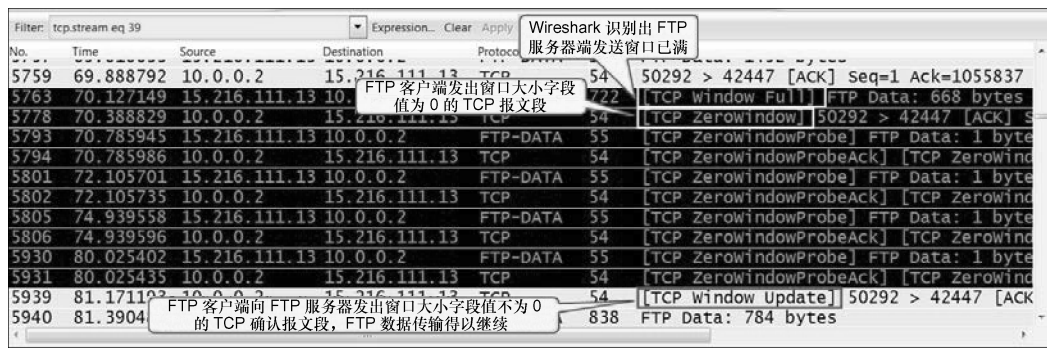


图 11.5

7. 图 11.5 所描绘的这一有问题的 FTP 数据传输过程, 是因 FTP 客户端主机“反应迟钝”所致。仔细检查那台主机, 关停其上所运行的某些进程之后, 便解决了问题。

导致 FTP 连通性故障的原因包括: FTP 服务器无法提供服务 (比如, 主机宕机或 FTP 服务未能正常启动等)、防火墙封锁了 FTP 连接、安装在 FTP 服务器或客户端主机上的某些软件

破坏了 FTP 协议的正常运作等。排查此类故障时，应遵循以下思路。

1. 核实 FTP 文件传输所依赖的 TCP 连接能否正常建立，为此，应检查 Wireshark 是否抓全了与此 FTP 连接配套的 TCP “三次握手” (SYN/SYN-ACK/ACK) 报文段。若与 FTP 文件传输配套的 TCP 连接未能正常建立，则不出以下原因。
 - 防火墙封锁了支撑 FTP 文件传输的 TCP 连接。此时，应赶紧去找管理防火墙的网络管理员。
 - FTP 服务器无法提供服务，需登录安装 FTP 服务器端软件的主机，检查主机自身及 FTP 服务器端软件的运行情况。
 - 安装在 FTP 服务器（主机）上的某个软件，干扰 FTP 协议的正常运作。杀毒软件、VPN 客户端软件、防火墙软件或其他安全防护软件都有可能阻挡来自 FTP 客户端的 FTP 连接。
 - 检查安装 FTP 客户端软件的主机，看看是不是 VPN 客户端软件或防火墙软件阻碍其发起 FTP 连接。
2. 当 FTP 运行于 active 模式时，客户端会先向服务器发起一条控制连接，后者再向前者发起一条数据连接。因此，需配置防火墙，令其在两个方向上放行必要的 FTP 流量，或将 FTP 的运行模式改为 passive 模式。

11.3.3 幕后原理

FTP 有两种运行模式：active 模式和 passive 模式。以 active 模式运行时，控制连接建立之后，FTP 服务器会主动打开一条通往 FTP 客户端的数据连接；运行于 passive 模式时，数据连接则会由后者向前者主动发起。现在来了解一下上述两种 FTP 模式的原理。

图 11.6 所示为 FTP passive 模式的原理。

1. FTP 客户机发起通往 FTP 服务器的 FTP 控制连接，这条 TCP 连接的源端口号为 FTP 客户端随机选定（本例为 TCP 1024 端口），目的端口号为众所周知的 TCP 21 端口。
2. FTP 服务器以 TCP 21 端口进行回应，与 FTP 客户机的 TCP 1024 端口建立 FTP 控制连接。
3. 控制连接成功建立之后，需要传递数据时，FTP 客户机将再次发起通往 FTP 服务器的 FTP 数据连接，这条 TCP 连接的源端口号为“P(FTP 控制端口号)+1”(本例为 1025)，目的端口号为 FTP 服务器指定的高于 1024 的任一 TCP 端口号（本例为 TCP 2000 端口），这一端口号会通过先前建立的控制连接通告给 FTP 客户机。

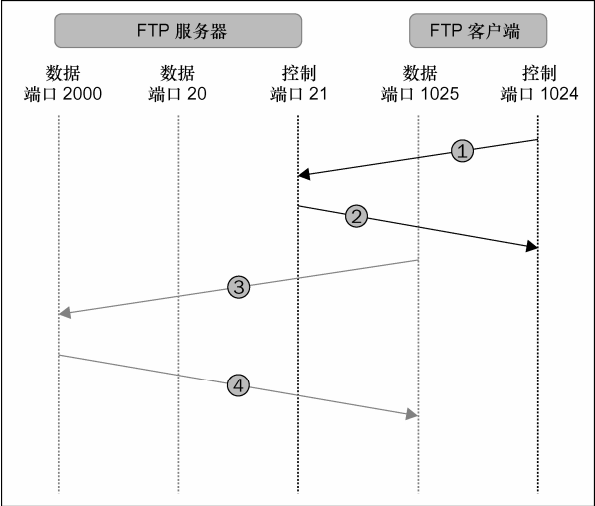


图11.6

4. FTP 服务器以 TCP 2000 端口进行回应，与 FTP 客户机的 TCP1025 端口建立 FTP 数据连接。

当 FTP 以 active 模式运行时，情况会略有不同。

1. FTP 客户机发起通往 FTP 服务器的 FTP 控制连接，这条 TCP 连接的源端口号为 FTP 客户端随机选定(本例为 TCP 1024 端口)，目的端口号为众所周知的 TCP 21 端口(见图 11.7)。

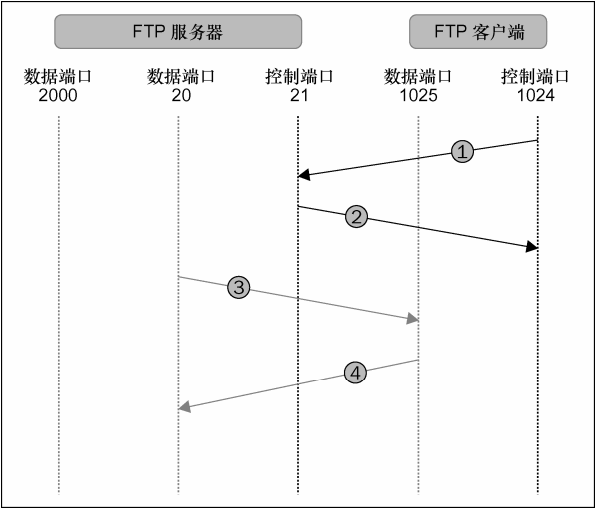


图11.7

2. FTP 服务器以 TCP 21 端口进行回应，与 FTP 客户机的 TCP 1024 端口建立 FTP 控制连接。
3. 控制连接成功建立之后，需要传递数据时，FTP 服务器将主动发起通往 FTP 客户机的 FTP 数据连接，这条 TCP 连接的源端口号为众所周知的 TCP 20 端口，目的端口号为“P（FTP 控制端口号）+1”（本例为 1025）。
4. FTP 服务器以 TCP 20 端口进行回应，与 FTP 客户机的 TCP 1025 端口建立 FTP 数据连接。

11.3.4 拾遗补缺

FTP 是一种比较简单的应用层协议，在大多数情况下，FTP 故障都不难排查。下面举几个作者参与排除过的 FTP 故障的实例：

- **故障实例 1：**FTP 服务器和客户端之间通过一条国际专线互连，用户普遍感觉 FTP 文件传输缓慢，并指责提供专线的运营商不给力。但据运营商反映，那条国际专线的带宽利用率只有不到 20%。经作者核查，确认了这一点。作者使用 Wireshark 进行抓包分析，分析结果表明不存在任何与 TCP 有关的问题（比如，TCP 重传或窗口问题）。无奈之下，作者登录进安装 FTP 服务器端软件的主机，卸载掉了原有的 FTP 服务器端软件，安装了另一款软件（免费的 FTP 服务器端软件遍地皆是），FTP 文件传输的性能就得到了极大地改善。看来，这只不过是 FTP 服务器端软件效率低下所引发的 FTP 文件传输缓慢问题。
- **故障实例 2：**有一用户反映，他在连 FTP 服务器时，每次都是尝试了 5、6 次之后，遭到了 FTP 服务器的拒绝。动用 Wireshark 之后，作者抓到了很多身背 FTP connection refused 字样的数据包（用户投诉的原因也正在于此），这看起来像是 FTP 服务器不响应。作者登录进安装 FTP 服务器端软件的主机，重启了 FTP 服务之后，问题便无影无踪。经过检查，原来是主机上安装的杀毒软件干扰了 FTP 服务器端软件的运行。

作者通过以上两个实例是想传达这样一个理念，那就是排除网络故障时，即便有 Wireshark（或其他软件）助一臂之力，也得具备最基本的 IT 常识。

11.4 Email 协议（POP、IMAP、SMTP）流量及故障分析

常用的 Email 协议包括 POP3（邮局协议版本 3）、简单邮件传送协议（SMTP）以及 Internet 消息访问协议版本 4（IMAP4），这几种协议不是运行于 Email 服务器与客户端之间，就是运

行于 Email 服务器之间。

还有一种收取邮件的方法是，通过 Web 方式来访问邮箱，像 Gmail、Yahoo!以及 Hotmail 之类的邮件服务器都可以通过 Web 方式来访问。

本节将重点关注最常用的 Email 客户端/服务器协议和 Email 服务器/服务器协议：POP3 和 SMTP。此外，还将讨论在使用其他方法访问邮件服务器时，如何处理遇到的问题。

11.4.1 准备工作

当有用户反映存在邮件收发故障时，应首先核实故障是否因弱智级错误而起。所谓的弱智级错误包括：用户名/密码错误、认证协议未配置等。若未发现上述弱智级错误，请把 Wireshark 主机接入交换机；然后，在交换机上启用端口镜像功能，把连接了存在邮件收发问题的用户主机流量重定向至 Wireshark 主机。若遭遇大面积用户投诉，则需要将通往邮件服务器的通信链路的流量重定向至 Wireshark 主机。

11.4.2 操作方法

POP3 是一种运行于 Email 客户端与服务器之间的协议；SMTP 则是运行于 Email 服务器之间的协议。

POP3 协议运作机制

POP3 协议的主要作用是让 Email 客户端从 Email 服务器收取邮件。当 Email 客户端无法从 Email 服务器收取邮件时，请按以下步骤行事。

1. 检查 Email 客户端的用户名/密码是否配置正确。
2. 检查 Email 客户端能否顺利通过 Email 服务器的验证。由图 11.8 所示的 Wireshark 截屏可知，该 Email 账户的用户名和密码分别以 **doronn@**和 **u6F** 打头（用户名和密码的后半部分未予显示）。
3. 要想看到图 11.8 所示的内容，需先在 Wireshark 抓包主窗口的数据包列表区域内选择一个隶属于 POP3 数据流的数据包，点击右键，在弹出的右键菜单中点击 **Follow TCP Stream** 菜单项。
4. 在 POP 用户验证阶段，只要用户名/密码有误，POP3 连接便会中断。图 11.9 所示的 Wireshark 截屏展示了一次 POP3 认证失败的例子。由图可知，收到服务器的 **Logon failure** 提示之后，客户端主动断开了相应的 TCP 连接。

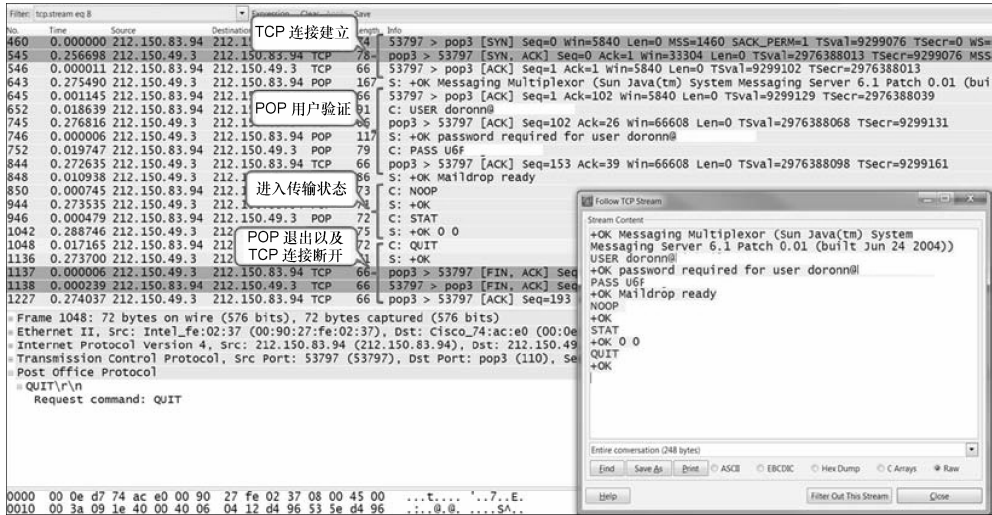


图 11.8

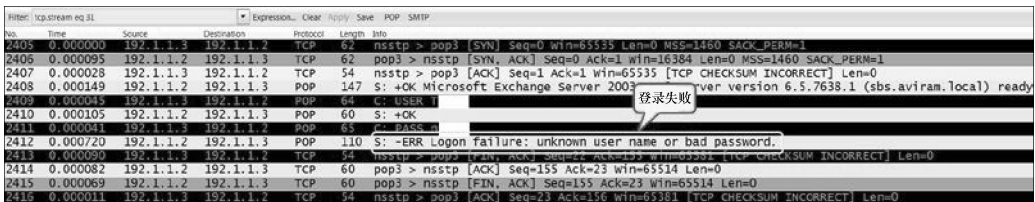


图 11.9

5. 客户端从邮件服务器下载邮件时，很容易导致通信链路过载，这一点请务必关注。可借助于之前介绍过的 IO Graphs 工具，并配搭显示过滤表达式“POP”，来检查通信链路的带宽利用率。
6. 应关注 Wireshark 是否抓到了身背 TCP retransmissions、TCP zero-window、TCP window-full 等字样的数据包。若身负上述字样的数据包全都隶属于 POP3 数据流，则表示存在通信链路过载、服务器反应慢等问题。这些问题有可能导致 Email 客户端无法从 Email 服务器收取邮件。

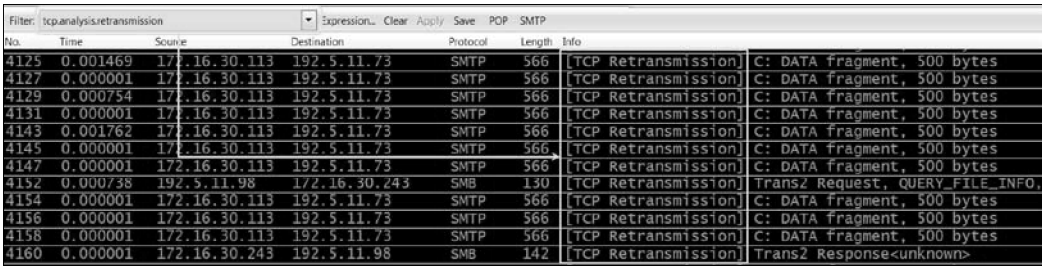
SMTP 协议运作机制

SMTP 协议的用途如下所列。

- SMTP 是一种运行于服务器之间的邮件传输协议。
- 对某些邮件客户端软件而言，可被配置为用 SMTP 来发送邮件（把待发邮件从客户端交付给服务器），用 POP3 或 IMAP4 从服务器收取邮件。

当感觉到邮件服务器之间沟通缓慢时，请按以下步骤行事。

1. 核实两台邮件服务器是否都位于同一站点。
 - 若位于同一站点，则很可能是邮件服务器反应慢或应用程序问题。一般不太可能是网络传输问题，若两台邮件服务器都部署在通过高速 LAN 链路互连的数据中心内，则更是如此。
 - 若位于不同的站点（比如，两台邮件服务器分别位于通过低速 WAN 链路互连的异地站点内），请检查 WAN 链路的负载状况。当发送包含大附件的邮件时，极有可能会导致低速 WAN 链路的拥塞。
2. 检查 Wireshark 抓包文件，看看是否抓到了身背 TCP retransmissions、TCP zero-window、TCP window-full 等字样，且隶属于 SMTP 数据流的数据包。图 11.10 所示为 Wireshark 抓到了众多身负 TCP retransmissions 字样，且隶属于 SMTP 数据流的数据包时的情形。



No.	Time	Source	Destination	Protocol	Length	Info
4125	0.001469	172.16.30.113	192.5.11.73	SMTP	566	[TCP Retransmission] C: DATA fragment, 500 bytes
4127	0.000001	172.16.30.113	192.5.11.73	SMTP	566	[TCP Retransmission] C: DATA fragment, 500 bytes
4129	0.000754	172.16.30.113	192.5.11.73	SMTP	566	[TCP Retransmission] C: DATA fragment, 500 bytes
4131	0.000001	172.16.30.113	192.5.11.73	SMTP	566	[TCP Retransmission] C: DATA fragment, 500 bytes
4143	0.001762	172.16.30.113	192.5.11.73	SMTP	566	[TCP Retransmission] C: DATA fragment, 500 bytes
4145	0.000001	172.16.30.113	192.5.11.73	SMTP	566	[TCP Retransmission] C: DATA fragment, 500 bytes
4147	0.000001	172.16.30.113	192.5.11.73	SMTP	566	[TCP Retransmission] C: DATA fragment, 500 bytes
4152	0.000738	192.5.11.98	172.16.30.243	SMB	130	[TCP Retransmission] Trans2 Request, QUERY_FILE_INFO
4154	0.000001	172.16.30.113	192.5.11.73	SMTP	566	[TCP Retransmission] C: DATA fragment, 500 bytes
4156	0.000001	172.16.30.113	192.5.11.73	SMTP	566	[TCP Retransmission] C: DATA fragment, 500 bytes
4158	0.000001	172.16.30.113	192.5.11.73	SMTP	566	[TCP Retransmission] C: DATA fragment, 500 bytes
4160	0.000001	172.16.30.243	192.5.11.98	SMB	142	[TCP Retransmission] Trans2 Response<unknown>

图11.10

3. 核实问题是否是出在 Email(SMTP)服务器反应慢上面。当读者看见图 11.11 所示 Wireshark 截屏时，应该知道作者使用了 **Statistics** 菜单下 **Conversation** 工具中的 **TCP** 标签功能。在勾选 **Limit to display filter** 复选框（让显示过滤参数 tcp.analysis.retransmission 生效），点击过 **Conversation** 窗口中的 **Packets** 一列（按数据包数量的多少重新排序）之后，可以看到有 793 个重传的 SMTP (TCP 25 端口) 数据包，而 172.16.30.247 和 172.16.30.2 TCP 445 端口 (Microsoft DS) 之间、172.16.30.180 和 192.5.11.198 TCP 80 端口 (HTTP) 之间的 TCP 重传数据包则分别高达 9014 和 2139 个。由此可以判断出 SMTP 协议本身并无问题，邮件发送不畅只是受到了恶劣网络环境的影响。
4. 检查 Wireshark 抓包文件，看看是否抓到了身背 SMTP 状态代码的数据包。由图 11.12 可知，Wireshark 抓到了身背 SMTP 状态代码 451 的数据包。状态代码 451 表示邮件 (SMTP) 服务器发生了 local error in processing 错误。

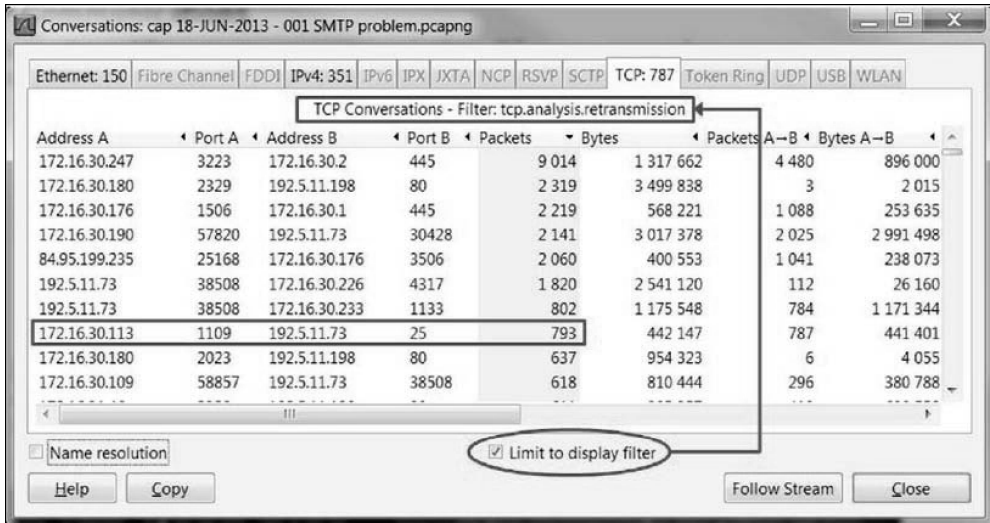


图 11.11

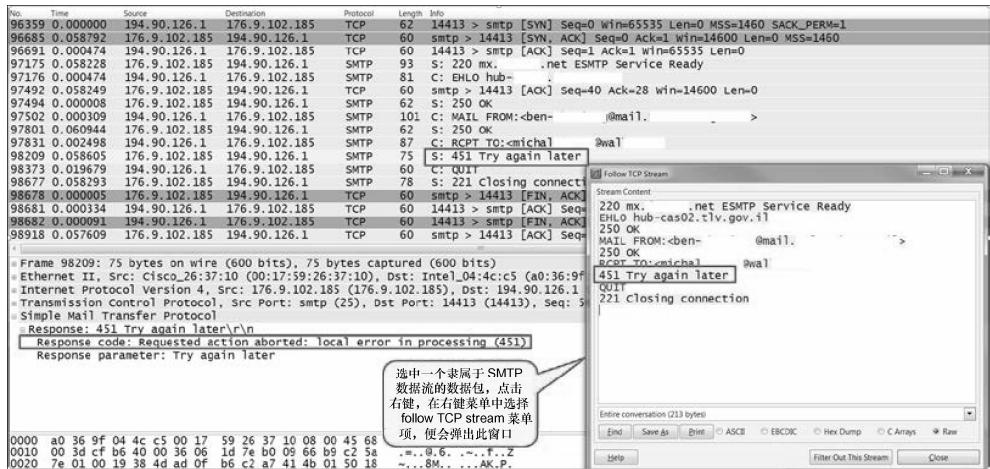


图 11.12

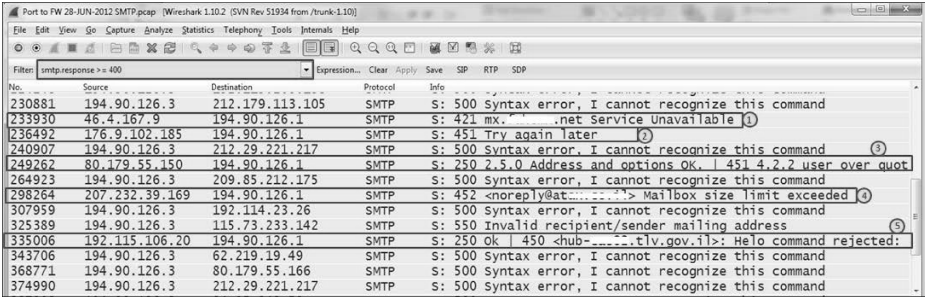


注意

当某种应用（层协议）发生故障时，其客户端或服务器端主机一般都会生成与故障有关的日志信息。通过阅读这些信息或基于这些信息执行 Google 搜索，一般都能得出导致故障的原因。本文后文会举几个这方面的例子。

SMTP 状态码列表详见 RFC 1893 (<http://www.ietf.org/rfc/rfc1893.txt>)。

1. 要想弄清 SMTP 对话双方所发数据包是否包含 SMTP 错误状态码，请配置图 11.13 所示的 Wireshark 显示过滤器。



No.	Source	Destination	Protocol	Info
230881	194.90.126.3	212.179.113.105	SMTP	S: 500 Syntax error, I cannot recognize this command
233930	46.4.167.9	194.90.126.1	SMTP	S: 421 mx... .net Service Unavailable ①
236492	176.9.102.185	194.90.126.1	SMTP	S: 451 Try again later ②
240907	194.90.126.3	212.29.221.217	SMTP	S: 500 Syntax error, I cannot recognize this command ③
249262	80.179.55.150	194.90.126.1	SMTP	S: 250 2.5.0 Address and options OK, I 451 4.2.2 user over quot
264923	194.90.126.3	209.85.212.175	SMTP	S: 500 Syntax error, I cannot recognize this command
298264	207.232.39.169	194.90.126.1	SMTP	S: 452 <noreply@at...> Mailbox size limit exceeded ④
307959	194.90.126.3	192.114.23.26	SMTP	S: 500 Syntax error, I cannot recognize this command
325389	194.90.126.3	115.73.233.142	SMTP	S: 550 Invalid recipient/sender mailing address
335006	192.115.106.20	194.90.126.1	SMTP	S: 250 Ok 450 <hub-cas02.tlv.gov.il>: Helo command rejected: ⑤
343706	194.90.126.3	62.219.19.49	SMTP	S: 500 Syntax error, I cannot recognize this command
368771	194.90.126.3	80.179.55.166	SMTP	S: 500 Syntax error, I cannot recognize this command
374990	194.90.126.3	212.29.221.217	SMTP	S: 500 Syntax error, I cannot recognize this command

图11.13

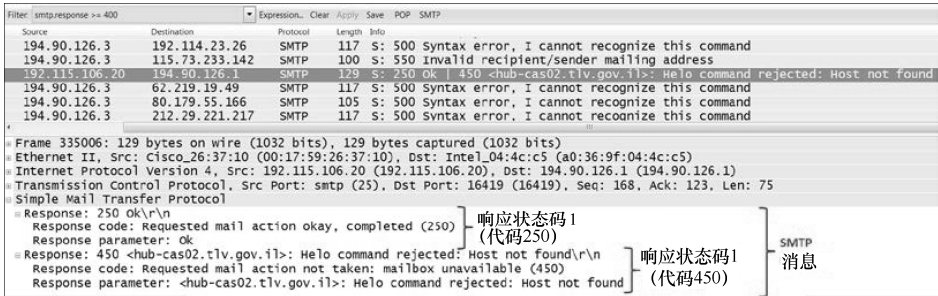
2. 现在来解读一下在图 11.13 中现身的几个 SMTP 错误状态码。

- **状态码 421:** 表示 Mail 服务不可用（图 11.13 中标记为①的区域）。
- **状态码 451:** 表示 SMTP 服务器无法响应，请稍后再试，这可能预示着服务器负载过高或服务器故障（图 11.13 中标记为②的区域）。
- **状态码 451:** 表示用户配额超限（图 11.13 中标记为③的区域，在该区域内还出现了 SMTP 状态码 250，具体原因稍后解释）。
- **状态码 452:** 表示用户邮箱大小超限（图 11.13 中标记为④的区域）。
- **状态码 450:** 表示主机未发现（图 11.13 中标记为⑤的区域，在该区域内还出现了 SMTP 状态码 250，具体原因稍后解释）。



注意

在同一条 SMTP 协议消息中可以包含多个状态码，某些其他（应用层）协议的协议消息也是如此。在 Wireshark 抓包主窗口的数据包列表区域内，只能看见 SMTP 协议消息中的第一个或头几个状态码。要想查看一条 SMTP 协议消息中所包含的全部状态码，请按图 11.14 那样先在数据包列表区域内选中某个 SMTP 数据包，再到数据包结构区域内查看。



Source	Destination	Protocol	Length	Info
194.90.126.3	192.114.23.26	SMTP	117	S: 500 Syntax error, I cannot recognize this command
194.90.126.3	115.73.233.142	SMTP	100	S: 550 Invalid recipient/sender mailing address
192.115.106.20	194.90.126.1	SMTP	129	S: 250 Ok 450 <hub-cas02.tlv.gov.il>: Helo command rejected: Host not found
194.90.126.3	62.219.19.49	SMTP	117	S: 500 Syntax error, I cannot recognize this command
194.90.126.3	80.179.55.166	SMTP	105	S: 500 Syntax error, I cannot recognize this command
194.90.126.3	212.29.221.217	SMTP	117	S: 500 Syntax error, I cannot recognize this command

Frame 335006: 129 bytes on wire (1032 bits), 129 bytes captured (1032 bits)
Ethernet II, Src: Cisco_26:37:10 (00:17:59:26:37:10), Dst: Intel_04:4c:c5 (a0:36:9f:04:c5)
Internet Protocol Version 4, Src: 192.115.106.20 (192.115.106.20), Dst: 194.90.126.1 (194.90.126.1)
Transmission Control Protocol, Src Port: smtp (25), Dst Port: 16419 (16419), Seq: 168, Ack: 123, Len: 75
Simple Mail Transfer Protocol
Response: 250 Ok\r\n
Response code: Requested mail action okay, completed (250)
Response parameter: Ok
Response: 450 <hub-cas02.tlv.gov.il>: Helo command rejected: Host not found\r\n
Response code: Requested mail action not taken: mailbox unavailable (450)
Response parameter: <hub-cas02.tlv.gov.il>: Helo command rejected: Host not found

SMTP 消息

图11.14

若发现有很多条 SMTP 消息都包含不止一个状态码，则表示 Mail 服务器不可用。此时，需联系 Mail 服务器管理员。

其他收取邮件（访问邮箱）的方法和问题

下面简单介绍一下之前提到过的其他收取邮件（访问邮箱）的常用方法，这包括 Web Mail 和基于 HTTP 的 RPC。

- 使用 Web Mail 时，实际上是通过 HTTPS 连接到 Web 服务器，HTTPS 协议在第 10 章已做详述。以 Web Mail 的方式登录邮箱时，如遇到问题，一般都是 HTTPS 问题。
- 基于 HTTP 的 RPC 应用也与上面这种方法相仿。RPC 协议流量可轻松导致网络拥塞，对网络的带宽要求很高，因此基于 HTTP 的 RPC 应用对网络的延迟和抖动非常敏感。Microsoft Outlook 应用运行于 HTTPS 之上，并非基于标准的 RPC 协议。所以说此类应用一旦发现问题，多半为 HTTPS 问题。

11.4.3 幕后原理

邮件客户端软件与邮件服务器沟通时，大多都是借助于 POP3 协议，偶尔也会使用 SMTP 协议。当需要对邮件服务器执行某些操作时（比如，需要在客户端软件界面中查看远程邮件服务器上邮件的主题，但并不下载那些邮件时），将会用到 IMAP4 协议。邮件服务器之间的“沟通”大都使用 SMTP 协议。



注意

IMAP4 和 POP3 协议的区别是，前者允许用户直接通过邮件客户端软件对服务器上的邮件进行操作，而不用将邮件事先全部下载；当使用后者收取邮件时，所有邮件都会从邮件服务器上清除，并下载至本机客户端软件。也就是说，IMAP4 协议提供的是邮件客户端与服务器之间的双向通信，在邮件客户端软件内执行的操作，都会被反馈至邮件服务器；而使用邮件客户端软件操纵通过 POP3 协议收取到邮件时，操作结果不会被反馈至邮件服务器。

SMTP 状态码由三个部分构成，其构造方式使得 SMTP 消息中的错误一目了然。SMTP 状态码的构造方法和细节将在 11.4.4 节讨论。

POP3 协议

POP 是一种邮件客户端用来从服务器收取邮件的应用层协议。图 11.15 中的 Wireshark 截屏展示出了一次典型的 POP3 会话建立过程。

1. 客户端主机与服务器之间以三次握手的方式建立 TCP 连接。

No.	Time	Source	Destination	Protocol	Info
460	0.000000	212.150.83.94	212.150.49.3	TCP	53797 > pop3 [SYN] Seq=0 win=5840 Len=0
545	0.256698	212.150.49.3	212.150.83.94	TCP	pop3 > 53797 [SYN, ACK] Seq=0 Ack=1 win=5840
546	0.000011	212.150.83.94	212.150.49.3	TCP	53797 > pop3 [ACK] Seq=1 Ack=1 win=5840
643	0.275490	212.150.49.3	212.150.83.94	POP3	S: +OK Messaging Multiplexor (Sun Java(tm) Mail) (128 MB)
645	0.001145	212.150.83.94	212.150.49.3	TCP	53797 > pop3 [ACK] Seq=1 Ack=102 win=5840
652	0.018639	212.150.83.94	212.150.49.3	POP3	C: USER ddron@shtil.com
745	0.276816	212.150.49.3	212.150.83.94	TCP	pop3 > 53797 [ACK] Seq=102 Ack=26 win=666
746	0.000006	212.150.49.3	212.150.83.94	POP3	S: +OK password required for user doronn@
752	0.019747	212.150.83.94	212.150.49.3	POP3	C: PASS U6FU6F
844	0.272635	212.150.49.3	212.150.83.94	TCP	pop3 > 53797 [ACK] Seq=153 Ack=39 win=666
848	0.010938	212.150.49.3	212.150.83.94	POP3	S: +OK Maildrop ready
850	0.000745	212.150.83.94	212.150.49.3	POP3	C: NOOP
944	0.273535	212.150.49.3	212.150.83.94	POP3	S: +OK
946	0.000479	212.150.83.94	212.150.49.3	POP3	C: STAT
1042	0.288746	212.150.49.3	212.150.83.94	POP3	S: +OK 0 0
1048	0.017165	212.150.83.94	212.150.49.3	POP3	C: QUIT
1136	0.273700	212.150.49.3	212.150.83.94	POP3	S: +OK
1137	0.000006	212.150.49.3	212.150.83.94	TCP	pop3 > 53797 [FIN, ACK] Seq=192 Ack=58 win=0
1138	0.000239	212.150.83.94	212.150.49.3	TCP	53797 > pop3 [FIN, ACK] Seq=58 Ack=193 win=0
1227	0.274037	212.150.49.3	212.150.83.94	TCP	pop3 > 53797 [ACK] Seq=193 Ack=59 win=666

图11.15

2. 邮件服务器向客户端主机发送身负 **OK Messaging Multiplexor** 字样的 POP3 数据包。
3. 操作客户端主机的用户输入邮箱的用户名/密码。
4. POP3 协议运作的帷幕正式拉开。邮件客户端主机发出身背 **NOOP** (no operation) 字样的 POP3 数据包，测试邮件服务器是否已经打开了 POP3 连接。邮件客户端主机发出身负 **STAT** (status) 字样的 POP3 数据包，要求邮件服务器回发有关邮箱的统计信息，比如，邮件数和总字节数。邮件服务器发出身背 **OK 0 0** 字样的 POP3 数据包（编号为 1042）作为应答，表示邮件数和总字节数均为 0。
5. 当发现服务器上的邮箱无邮件可供收取时，邮件客户端主机会发出身背 **QUIT** 字样的 POP3 数据包（编号为 1048），邮件服务器会发出身负 **OK** 字样的 POP3 数据包（编号为 1136）进行确认。随后，邮件客户端主机会拆除先前建立起的 TCP 连接（编号为 1137、1138 和 1227 的数据包）。

在启用了加密连接的情况下，邮件客户端主机和服务端之间的交流过程应如图 11.16 中的 Wireshark 截屏所示。以三次握手方式建立起 TCP 连接之后（①），邮件客户端主机和服务端之间会彼此交换几个 POP 数据包（②），然后双方会建立起 TLS 连接（③），以加密方式传输数据。

SMTP 协议和 SMTP 状态码

SMTP 状态码的构造方式如下所列。

class. subject. detail

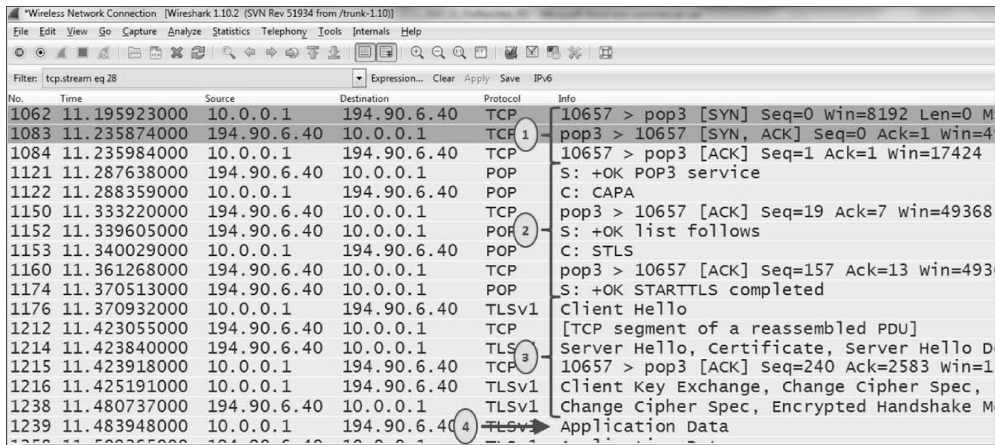


图11.16

举个例子，一条包含状态码 450 的 SMTP 消息具有如下含义。

- **class 4:** 表示存在临时性问题。
- **subject 5:** 指明了邮件递送协议的状态。
- **detail 0:** 表示其他或未定义的协议状态（RFC 3463 3.6 节）

下表所列为 SMTP 状态码的各种类别码（class）。

SMTP 状态码	含 义	描 述
2.x.xxx	成功	操作成功
4.x.xxx	持续出现暂时性的故障	临时情况（暂时性的故障）导致邮件服务器无法发送邮件。暂时性的故障是因服务器负载过重或网络拥塞所致。通常，（在邮件发送失败之后，）重新发送将会取得成功
5.x.xxx	持久性的故障	持久性的故障导致邮件服务器无法发送邮件。所谓持久性的故障一般是指服务器故障或兼容性问题

下表所列为 SMTP 状态码的各种主题码（subject）。

SMTP 状态码	含 义	原 因
x.0.xxx	其他或未定义状态	
x.1.xxx	地址状态	
x.2.xxx	邮箱状态	
x.3.xxx	邮件系统状态	
x.4.xxx	网络及路由状态	
x.5.xxx	邮件递送协议状态	
x.6.xxx	邮件内容或媒体状态	
x.7.xxx	安全或策略状态	

SMTP 状态码的细节码（detail）着实太多，难以全部罗列，详情请参考 RFC 3463（<http://tools.ietf.org/html/rfc3463>）。

下表所列为常见的 SMTP 状态码。

SMTPu 状态码	含 义	原 因
220	服务就绪	（邮件）服务正在运行，随时准备执行邮件“投递”任务
221	服务关闭	一般为正常，表示邮件服务器上的服务在不用时关闭
250	对邮件的操作请求已执行完毕	邮件发送成功
251	非本地用户，邮件将会被转发	一切正常
252	用户通不过验证	用户通不过服务器的验证，邮件无法递送
421	服务不可用	邮件发送服务不可用，服务器无法处理收到的邮件，这可能是由于服务器问题（邮件递送服务未开启）或服务器限制问题
422	邮件大小问题	收件人邮箱已满或邮件服务器对所收邮件的大小有限制
431	内存溢出或磁盘空间已满	邮件服务器内存溢出或磁盘空间已满
432	邮件接收队列被停止	服务器故障（邮件递送服务未开启）
441	收件服务器无响应	发件服务器表示收件服务器无响应
442	连接故障	通往收件服务器的连接（链路）存在故障
444	无法路由	服务器无法确定递送邮件的下一跳
445	邮件服务器拥塞	邮件服务器临时性拥塞
447	邮件递送逾期	收件系统认为邮件太“旧”，这通常都表示队列和传输问题
450	操作请求未被接受	邮件可能未被递送，这通常都归咎于远程邮件服务器上的邮件递送服务有问题
451	无效命令	表示命令乱序或不被支持。收件服务器终止执行操作请求，主要原因是其负载过重
452	操作请求未被接收	收件服务器存储空间不足
500	语法错误	由服务器发出的命令未被识别为有效的 SMTP 或 ESMTP 命令
512	DNS 错误	无法定位收件服务器
530	认证问题	收件服务器要求认证，或已将发件服务器录入了黑名单
542	收件地址遭拒	表示收件人地址遭到收件服务器的拒绝。这通常是因为收件人地址通不过反垃圾邮件系统、IDS/IPS、智能防火墙或其他安全设备的检查

11.4.4 拾遗补缺

Email 应用有时也会成为网络中的杀手级应用，对使用非对称线路接入 Internet 的小型

企业网络来说则更是如此。在发送文本格式的邮件时，并不会消耗多少网络带宽；可要是在发送邮件时携带几兆甚至几十兆的附件，且 Internet 链路的上行带宽又过窄，网络中的其他用户就会在相当长的一段时间内感觉上网卡顿。这样的事儿在小型企业网络中可谓屡见不鲜。

另一个问题是，有些用户会把邮件客户端软件配置为在启动时自动从邮件服务器下载新邮件。在长假（或双休日）后的第一个工作日，要是有人一来上班就反映上网卡顿，则很有可能是公司所有员工正同时打开邮件客户端软件从邮件服务器收取邮件。

11.5 MS-TS 和 Citrix 故障分析

微软终端服务器（Microsoft Terminal Server, MS-TS）基于远程桌面协议（Remote Desktop Protocol, RDP）和 Citrix Metaframe 独立计算架构（Independent Computing Architecture, ICA），是一种在本地和远程 PC（或瘦客户端）之间实施桌面控制的应用程序。需要注意的是，这种桌面控制型应用程序会通过网络来传递远程主机的屏幕的变化情况。若屏幕变化不大，则 MS-TS 应用基本不会占用多少带宽，否则，便会吃掉大量带宽。

还有一事值得关注，那就是由安装桌面控制型应用程序的主机生成的流量是完全不对称的。对主控端而言，下行流量速率少则数十 K（kbit/s），多则数 M（Mbit/s）；而上行流量速率则最多只有几 K（kbit/s）。在设计运行此类应用程序的网络时，切莫遗忘这一点。

本节会举几个与桌面控制型应用有关的典型故障案例，同时会细述故障解决方法。为求表述简洁，作者会把上述此类应用统称为 Microsoft TS，简写为 MS-TS。

11.5.1 准备工作

当有用户反应 MS-TS 应用程序运行缓慢时，应首先问清楚慢在什么地方。然后，再到连接客户主机或受控端主机的 LAN 交换机上开启端口镜像功能，将相关端口的流量重定向至 Wireshark 主机。

11.5.2 操作方法

要想准定位 MS-TS 相关故障，首先应向一线用户询问具体遇到的是什么问题，并按以下步骤行事。

1. 当用户反应与 MS-TS 有关的应用程序运行速度缓慢时，应问他们：是屏幕中的数据呈现速度慢，还是窗口切换速度慢？
2. 若得到的回答是窗口切换速度不慢，则说明并非 MS-TS 问题。要是 MS-TS 出了问题，

就会出现窗口切换速度缓慢、鼠标键盘操作结果回显迟钝等现象。

3. 若得到的回答是基于 MS-TS 运行的应用程序生成报表的速度变慢,则很有可能是该应用程序的后台数据库问题,并非 MS-TS 或 Citrix 问题。
4. 若通信链路负载较高,则使用 MS-TS 相关应用的最终用户势必会反映,输入回显延迟较高。这是因为 MS-TS 应用的主要用途就是传输远程主机的屏幕变化情况,只要用户打字速度过快,在通信链路拥塞的情况下,就会感觉输入回显迟钝。
5. 要想利用 Wireshark 获知通信链路的负载状况,请按以下步骤行事。
 - 启用 Wireshark 自带的 IO Graphs 工具。
 - 在 IO Graphs 窗口中,输入相关 Wireshark 显示过滤器,监控 (MS-TS 主控端的) 上、下行链路的负载状况。
 - 在 IO Graphs 窗口中的 Y 轴坐标区域内,选择 Unit 下拉菜单中的 Bits/Tick 菜单项。
6. 按照上述设置, Wireshark IO Graphs 窗口看起来应该如图 11.17 所示。

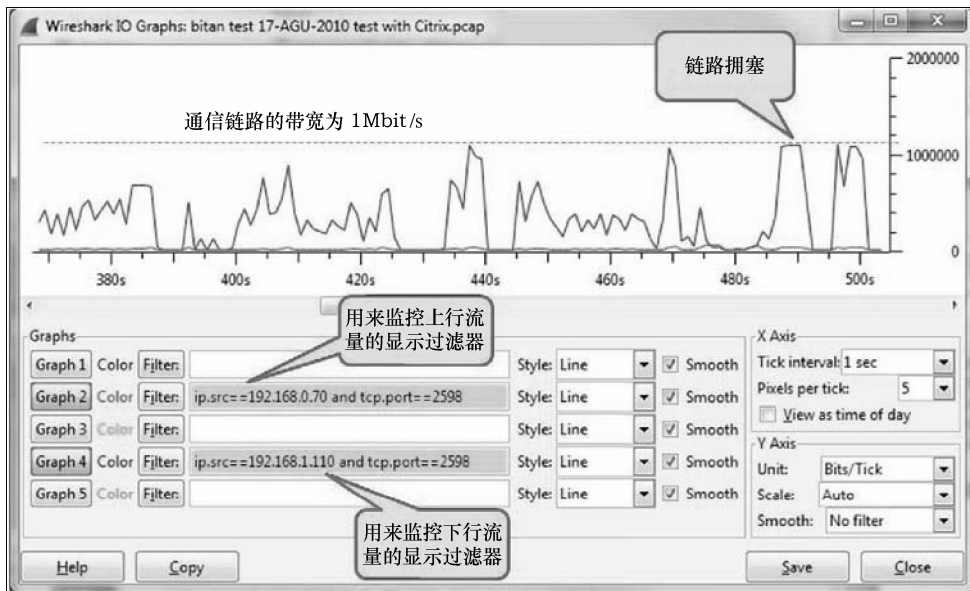
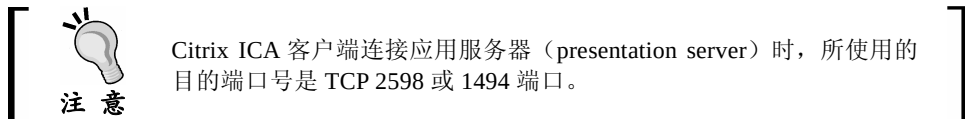


图 11.17

7. 由图 11.17 所示流量模式可知,运行 MS-TS 主控端程序的主机会生成极高的下行流量,但却不怎么生成上行流量。请注意,在图 11.17 所示的 IO Graphs 中, Y 轴坐标的单位为

Bits/Tick，即比特/秒（在 X 轴区域内的 Tick interval 下拉菜单中，选择的是“1 sec”）。在抓包开始后的 485~500 秒那段时间内，通信链路上的流量吞吐量达到了顶峰。也就是在那段时间内，用户感觉到了 MS-TS 应用运行缓慢（比如，鼠标或键盘操作结果回显缓慢等）。



8. 请别忘了，用户通过 MS-TS 主控端连接到 MS-TS 服务器之后，操作的软件其实是安装在 MS-TS 服务器上的某种应用的客户端软件，也就是说 MS-TS 服务器还得作为客户端去连接别的应用服务器。因此，若 MS-TS 服务器不给力，也会让用户产生 MS-TS 应用运行缓慢的错觉。
9. 若确诊为 MS-TS 故障，则有必要弄清故障具体出在网络层面还是系统层面，此时应：
 - 借助于 Wireshark 自带的 IO Graph 工具来了解通信链路的负载状况。如 MS-TS 应用运行缓慢确实是因为通信链路负载过高（如上图所示），则可以通过扩容链路带宽来解决；
 - 检查 MS-TS 服务器的性能状况。运行 MS-TS 应用的服务器的内存消耗量都颇为可观，因此需检查其内存（RAM）使用情况。

11.5.3 幕后原理

MS-TS、Citrix Metaframe 等应用在运行时，需要通过网络来传输（远程服务器的）屏幕变化情况。用户要先从客户端（即安装了 MS-TS 客户端软件的主机或瘦客户端）连接到终端服务器，再使用安装在终端服务器上的各种应用程序客户端软件，来连接各应用服务器。图 11.18 所示为终端服务器的运作原理。

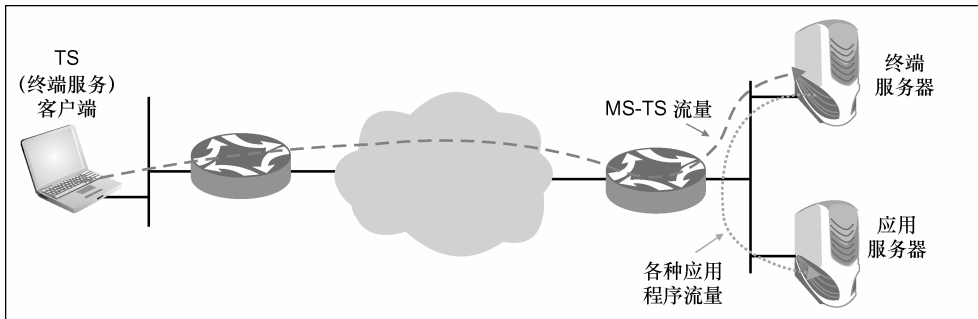


图11.18

11.5.4 拾遗补缺

许多终端服务器软件的生产厂商都吹嘘自己的产品能在两个方面提升企业的生成率。厂商的销售人员首先会说他们的产品能方便 IT 人员对用户终端和各种软件客户端的管理，理由是各种应用程序的客户端软件都只需在终端服务器上安装，这样一来，就不用管理每个用户的 PC 以及其上所安装的各种软件了。其次，那些销售人员还会说，他们的产品一经部署，网络流量就会大大降低。

对于一种说法，作者倒也不准备抬杠，而且该话题与本书的内容无关。不过，作者绝不认可第二种说法。在有多个用户同时连接终端服务器的网络环境中，网络流量的高低要取决于用户操作什么样的应用程序。

1. 在操作基于文本或字符的应用程序（比如，**ERP**[企业资源规划]类应用程序）时，用户都是先输入数据，再点击某个按钮，最后查看结果。对于使用终端客户端的用户而言，要先连接到终端服务器，才能操作安装在终端服务器上的 **ERP** 客户端软件。此时，终端服务器会连接 **ERP** 程序的后台数据库。上述第二种说法能否成立，将取决于应用程序开发人员对后台数据库的设计，预期的流量（终端服务器与数据库服务器之间的流量）约为几十至几百 K（kbit/s）之间。
2. 在终端客户端上操作像 Word、PowerPoint 这样的办公软件时，到底会生成多少流量则完全取决于用户的操作行为。若只是编辑普通的 Word 文档，几十到几百 K（kbit/s）的流量也就够了。可要是编辑 PowerPoint 文档的话，则至少需要几百 K（kbit/s）甚至几 M（Mbit/s）的流量。要是以全屏方式预览 PowerPoint 文档（打开 PPT 文件后，再按一下 **F5** 键），吞吐量将高达 8~10M（Mbit/s）。
3. 通过终端客户端上网冲浪时，所耗带宽将会在几百 K（kbit/s）到几 M（Mbit/s）之间，具体的数字要取决于用户的上网行为。要是在线观看高清视频呢？还是别这么干为妙。

在部署任何终端应用之前，都必须再三测试，以弄清该应用所耗带宽。有一次，有家软件公司为使其软件上的 logo 更为清晰、醒目，把它安置在了软件窗口的右上角，且让它每秒闪烁个 10 来次。该软件作为终端应用一经部署，便让一条 2Mbit/s 的通信链路拥堵不堪，这当然要归功于那个在软件窗口上闪来闪去的 logo。有些东西要是没有事先测试，你就永远不知道它会怎样。

11.6 NetBIOS协议故障分析

网络基本输入/输出系统（**Network Basic Input/Output System, NetBIOS**）是一套开发于 20 世纪 80 年代初期，用于 LAN 通信的协议。若干年前，这套协议被 Microsoft 公司相中，将

它用于 Windows 操作系统在 LAN 内的网络通信协议。随后, Microsoft 又把它与 TCP/IP 相融合在了一起(RFC 1101 和 RFC 1102, “NetBIOS over TCP/IP”)。

如今, NetBIOS 协议能提供以下三项服务。

- **名字服务** (端口号 137): 用于名称注册以及名称/IP 地址间的解析。
- **数据报发布服务** (端口号 138): 用于客户端和服务器的服务发布。
- **会话服务** (端口号 139): 用于主机间的会话协商、文件访问和打开目录等。

本节将讨论 NetBIOS 协议族的常见问题以及解决问题的方法。NetBIOS 协议族极其复杂, 故障点极多, 本节只介绍该协议容易在什么地方出故障, 以及解决常见故障的方法。

11.6.1 准备工作

NetBIOS 协议主要用于 Windows 主机间的通信, 也可以用于 MAC/Linux 主机与 Windows 主机间的通信。如果在部署了上述机型的网络环境中出现断网、网速慢等问题, NetBIOS 协议故障也有可能是原因之一。Wireshark 是定位上述问题的绝佳工具, 它能显示出奔流于网络中的流量, 而 Windows 自带的某些工具则能显示出客户机与服务器间的操作情况^①。

11.6.2 操作方法

要解决 NetBIOS 协议故障, 请在 LAN 交换机上开启端口镜像功能, 将连接了相关客户机或服务器的端口的流量重定向至 Wireshark 主机。接下来, 作者会举几个 NetBIOS 协议故障场景的例子。

Wireshark 内置有若干显示过滤表达式, 可专门用来筛选 NetBIOS 协议流量, 在 Filter Expression 窗口中找到这些显示过滤表达式(在 Wireshark 抓包主窗口中点击 Filter 输入栏右边的 Expression 按钮, 就会弹出 Filter Expression 窗口)。

1. 以 netbios 打头的显示过滤表达式, 作用于包含 NetBIOS 命令的流量。
2. 以 nbns 打头的显示过滤表达式, 作用于 NetBIOS 名字服务流量。
3. 以 nbds 打头的显示过滤表达式, 作用于 NetBIOS 数据报服务流量。
4. 以 nbss 打头的显示过滤表达式, 作用于 NetBIOS 会话服务流量。
5. 以 smb 打头的显示过滤表达式, 作用于 SMB 流量。

^① 译者注: 原文是 “It will show you what runs over the network, and Windows tools will show you what runs in the clients and servers”。

常规测试

首先，要了解一下网络流量的整体状况；然后，在其中寻找异常流量。

- 1. 将 Wireshark 主机与 LAN 交换机相连，连哪个端口都可以，只要与连接了故障主机的端口隶属同一 VLAN（同一广播域）。
- 2. 在 Wireshark 抓包主窗口的 Filter 输入栏内输入显示过滤表达式 `nbns.flags.response == 0`，点击 **Apply** 按钮。所有 NetBIOS 名字服务请求数据包会立刻浮出水面，此类数据包都以广播方式发送，如图 11.19 所示。
- 3. 在图 11.19 所示的 Wireshark 截屏中，能看见以下几种 NetBIOS 名字服务请求数据包。

No.	Time	Source	Destination	Protocol	Length	Info
4994	0.000002	10.0.0.103	10.0.0.255	① NBNS	110	Registration NB WORKGROUP<1e>
4997	0.749962	10.0.0.103	10.0.0.255	NBNS	110	Registration NB ETTI<20>
4998	0.000002	10.0.0.103	10.0.0.255	NBNS	110	Registration NB WORKGROUP<1e>
5057	10.255261	10.0.0.102	10.0.0.255	② NBNS	92	Name query NB WPAD<00>
5075	0.763927	10.0.0.102	10.0.0.255	NBNS	92	Name query NB WPAD<00>
5088	0.512027	10.0.0.138	10.0.0.105	NBNS	92	Name query NBSTAT *<00><00><00><00><00>
5091	0.252327	10.0.0.102	10.0.0.255	NBNS	92	Name query NB WPAD<00>
5141	16.912745	10.0.0.105	10.0.0.255	③ NBNS	92	Name query NB YORASM-NDI<1c>
5144	0.749377	10.0.0.105	10.0.0.255	NBNS	92	Name query NB YORASM-NDI<1c>
5147	0.750008	10.0.0.105	10.0.0.255	NBNS	92	Name query NB YORASM-NDI<1c>
5265	4.215407	169.254.26.83	169.254.255.255	④ NBNS	92	Name query NB UM23.ESET.COM<00>
5287	0.744925	169.254.26.83	169.254.255.255	NBNS	92	Name query NB UM23.ESET.COM<00>
5297	0.749979	169.254.26.83	169.254.255.255	NBNS	92	Name query NB UM23.ESET.COM<00>
5298	0.751111	169.254.26.83	169.254.255.255	NBNS	92	Name query NB UM23.ESET.COM<00>

图11.19

- 主机 10.0.0.3 发出的包含用名称 **WORKGROUP** 和 **ETTI** 来注册的 name registration request 数据包(①)。NetBIOS 名字服务(NBNS)服务器会发出 positive name registration response 或 negative name registration response 数据包对发出 name registration request 数据包的主机进行回应（拒绝或接受注册）。主机发出 name registration request 数据包若收不到任何回应，将会假定注册成功。
- 用来查询（具有）指定名称（的主机）的 name query request 数据包（②、③、④）。若网络中部署了 NBNS 服务器（域控制器），Wireshark 还将抓到以下两种 NBNS 响应数据包中的一种。
 - 代码字段值为 3，表示所查名称不存在的 negative name registration response 数据包。
 - 包含所查名称的 positive name registration response 数据包，其代码字段值为 0。
- 4. 不应抓到由源 IP 地址为 169.254 打头的主机发出的任何 NBNS 请求数据包（⑤）。

以 169.245 打头的 IP 地址是自动私有 IP 地址（APIPA）。当（Windows）主机被配置为以 DHCP 方式获取 IP 地址,但获取不到任何 IP 地址时,便会自行配置一个 APIPA 地址。

5. 在 Wireshark 抓包主窗口的 Filter 输入栏内输入显示过滤表达式 `tcp.port == 138 or udp.port == 138`, 点击 **Apply** 按钮, 所有 NetBIOS 数据报服务数据包会立刻浮出水面^①。NetBIOS 数据报服务数据包的目的 IP 地址一定是广播地址, 源、目端口则是 TCP 或 UDP 138 端口。图 11.20 所示为 Wireshark 抓到的 NetBIOS 数据报服务数据包, 主机发出此类数据包的目的, 是通告本机所行使的功能（是工作站、数据库服务器还是打印机服务器等）。通过图 11.20, 可以看见:

- IP 主机 172.16.100.10 的主机名是 FILE-SRV, 其所行使的功能包括: 工作站、服务器和 SQL server (workstation, server, SQL server) (①)。
- IP 主机 172.16.100.204 的主机名是 GOLF, 其所行使的功能包括: 工作站、服务器和打印排队服务器(workstation, server, print queue server)(②)。

No.	Source	Destination	Protocol	Length	Info
10119	172.16.100.176	172.16.100.255	BROWSER	243	Host Announcement ZIVAK, workstation, Server, windows for W
10179	172.16.100.119	172.16.100.255	BROWSER	243	Host Announcement MERAVTI, workstation, Server, NT workstat
10332	172.16.100.16	172.16.100.255	BROWSER	244	Host Announcement GNETAPP, workstation, Server, NT workstat
10424	172.16.100.96	172.16.100.255	BROWSER	243	Host Announcement HAGITA, workstation, Server, NT Workstati
1047①	172.16.100.10	172.16.100.255	BROWSER	243	Host Announcement FILE-SRV, workstation, Server, SQL Server
10542	172.16.100.94	172.16.100.255	BROWSER	243	Host Announcement ORNAPI, workstation, Server, NT workstat
1072②	172.16.100.204	172.16.100.255	BROWSER	264	Host Announcement GOLF, workstation, Server, Print Queue Se
10721	172.16.100.204	172.16.100.255	BROWSER	264	Host Announcement GOLF, workstation, Server, Print Queue Se
10766	172.16.100.124	172.16.100.255	BROWSER	243	Host Announcement ADIP, workstation, Server, NT workstation
10768	172.16.100.170	172.16.100.255	BROWSER	243	Host Announcement MICHALA, workstation, Server, NT workstat
10929	172.16.100.106	172.16.100.255	BROWSER	258	Domain/workgroup Announcement NDI, NT workstation, Domain E
10930	172.16.100.106	172.16.100.255	BROWSER	258	Domain/workgroup Announcement NDI, NT workstation, Domain E

* Frame 10119: 243 bytes on wire (1944 bits), 243 bytes captured (1944 bits) * Ethernet II, Src: 3com_82:9a:c7 (00:50:da:82:9a:c7), Dst: Broadcast (ff:ff:ff:ff:ff:ff) * Internet Protocol Version 4, Src: 172.16.100.176 (172.16.100.176), Dst: 172.16.100.255 (172.16.100.255) * User Datagram Protocol, Src Port: netbios-dgm (138), Dst Port: netbios-dgm (138) * NetBIOS Datagram Service * SMB (Server Message Block Protocol) * SMB MailSlot Protocol * Microsoft Windows Browser Protocol	协议结构
--	------

图 11.20

6. 某些蠕虫或病毒也会设法让主机发出 NetBIOS 名字服务数据包, 来达成扫描网络的目的。要想得知网络中是否有主机感染了蠕虫或病毒, 需要仔细分析抓包文件, 看看 NetBIOS 名字服务流量是否高的离谱。
7. 仔细分析抓包文件, 关注广播数据包的数量。正常情况下, 一台主机每分钟发出广播数据包不应超过 10 个, 若超过了这一数字, 就要找到发包主机, 对其仔细检查。

① 译者注: 原文只有短短一句 “There are many announcement packets as well”。译文酌改。



注意

分析 Wireshark 抓包文件时，对异常流量模式的判断并无定式可循，需借助于 Wireshark 专家系统（Expert system）工具、百度（Google）和常识来加以判断。

具体问题

在日常运维过程中，可能会碰到以下问题。

1. 需要使用 SMB（Server Message Block，服务器消息块）协议的网络中，Wireshark 可能会抓到包含 SMB 错误代码（完整的 SMB 错误代码列表详见 Microsoft MSDN: <http://msdn.microsoft.com/en-us/library/ee441884.aspx>）的 NetBIOS 会话服务数据包。SMB 协议是一种能让主机跨网络浏览（其他主机的）目录，（从其他主机）复制文件或执行其他操作的协议^①。
2. 若 SMB 错误代码为 0，则表示 SMB 协议或 SMB 应用程序的状态正常。在检查 Wireshark 抓到的 NetBIOS 会话服务数据包时，如发现其（SMB 头部中的）错误代码不为 0，就表示 SMB 协议或 SMB 应用程序存在问题，应仔细检查相关应用^②。
3. 在图 11.21 所示的 Wireshark 截屏中，可以看到一个 SMB 错误代码为 0xC0000022（STATUS_ACCESS_DENIED）的 NetBIOS 会话服务数据包^③。这一 SMB 错误代码是网管人员理应关注的众多 SMB 错误代码之一。由图 11.21 所示的 Wireshark 截屏可知，一台主机（IP 地址以 203 打头，IP 地址的后 3 个字节因故省略）在访问一台服务器（IP 地址为 10.1.70.95）上的目录\\NAS01\\HOMEDIR 时，遭到了拒绝。
4. 使用基于 NetBIOS 协议的应用程序以远程访问的方式，浏览一台 Windows 主机的主目录时，弹出了标有“拒绝访问”（ACCESS DENIED）或类似字样的对话框。当然，该应用程序也可能会有自己的报错方式。执行上述操作的同时，若使用 Wireshark 抓包，在抓到的相关 NetBIOS 会话服务数据包中必能看见精确的 SMB 错误状态码，只要将 NT Status: 后的内容复制进 Google 里进行搜索，必能搜到访问遭拒的具体原因。

① 译者注：原文是“Using Server Message Block (SMB), which is the protocol that is used for browsing directories, copying files, and other operations over the network, you might see some error codes. The full list of error codes is listed in Microsoft MSDN:<http://msdn.microsoft.com/en-us/library/ee441884.aspx>”。译文酌改，如有不妥，请指正。依译者拙见，在未事先介绍 NetBIOS 协议数据包的种类和结构的情况下，本节的内容对许多初学者来说是都在“对牛弹琴”。

② 译者注：原文是“Code 0 means STATUS_OK, when everything works fine and there is no problem. Any other code should be examined”。译文酌改，如有不妥，请指正。

③ 译者注：原文是“In the following example, you can see a message STATUS_ACCESS_DENIED”。作者只会看表象，真不知道他是怎么排除网络故障的。

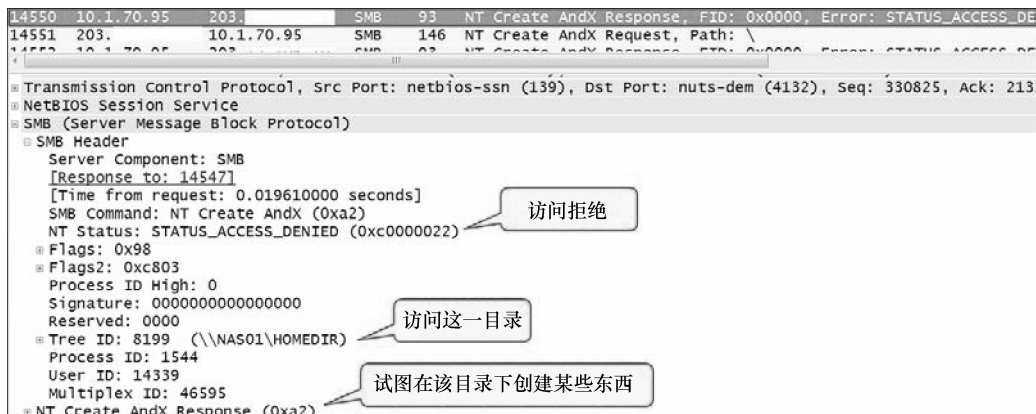


图 11.21

5. 在图 11.22 所示的 Wireshark 截屏中，可以看到一个 SMB 错误代码为 0xc0000016 (STATUS_MORE_PROCESSING_REQUIRED) (②) 的 NetBIOS 会话服务数据包。由图 11.22 所示的 Wireshark 截屏可知，那台 IP 地址以 203 打头的主机试图访问 (①) IP 地址为 10.1.70.95 的服务器上的目录 \\NAS01\ \SAMIM (③) 时，发生了错误。

- 根据上述 SMB 错误代码，查询之前提及的链接 (<http://msdn.microsoft.com/en-us/library/ee441884.aspx>)，可以获知出错的原因为 “There is more data available to read on the designated named pipe” (在指定的命名管道里无更多的数据可供读取)。
- 将 (②) 处 NT Status: 后的内容或出错的原因复制进百度 (Google)，搜索后得知是访问凭证问题，需与系统管理员协商解决。

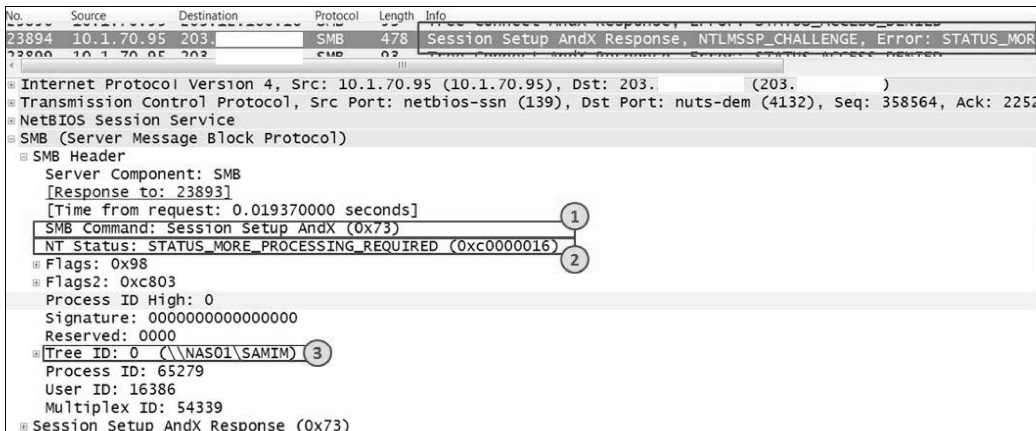


图 11.22

6. 要让 Wireshark 只显示出抓包文件中 SMB 错误码为非 0 的 NetBIOS 会话服务数据包, 请先在 Filter 输入栏内输入显示过滤表达式 `smb.nt_status != 0x0`, 再点击 **Apply** 按钮, 如图 11.23 所示^①。

Destination	Protocol	Length	Info	Command	Sub-command	Error
203.12.106.10	SMB	93	NT Create AndX Response, FID: 0x0000			STATUS_ACCESS_DENIED
203.12.106.10	SMB	126	Trans2 Response, FIND_FIRST2			STATUS_ACCESS_DENIED
203.12.106.10	SMB	126	Trans2 Response, FIND_FIRST2			STATUS_ACCESS_DENIED
203.12.106.10	SMB	93	NT Trans Response, FID: 0x0001, NT NOTIFY			STATUS_CANCELLED
203.12.106.10	SMB	478	Session Setup AndX Response, NTLMSSP_CHALLENGE			STATUS_MORE_PROCESSING_REQUIRED
203.12.106.10	SMB	93	Tree Connect AndX Response			STATUS_ACCESS_DENIED
203.12.106.10	SMB	478	Session Setup AndX Response, NTLMSSP_CHALLENGE			STATUS_MORE_PROCESSING_REQUIRED
203.12.106.10	SMB	93	Tree Connect AndX Response			STATUS_ACCESS_DENIED
203.12.106.10	SMB	478	Session Setup AndX Response, NTLMSSP_CHALLENGE			STATUS_MORE_PROCESSING_REQUIRED
203.12.106.10	SMB	93	Tree Connect AndX Response			STATUS_ACCESS_DENIED
203.12.106.14	SMB	93	Trans2 Response, GET_DFS_REFERRAL			STATUS_NO_SUCH_DEVICE
203.12.106.14	SMB	93	NT Create AndX Response, FID: 0x0000			STATUS_ACCESS_DENIED

图 11.23

11.6.3 幕后原理

如 11.1 节所述, NetBIOS 协议能提供三种服务: NetBIOS 名字服务 (NBNS)、NetBIOS 数据报发布服务 (NBDS) 以及 NetBIOS 会话服务 (NBSS)。

NBNS 提供注册以及将 NetBIOS 名转换为 IP 地址的功能。所谓注册是指客户机以其 NetBIOS 名来注册域控制器。客户机在注册时, 会发出 NetBIOS 注册请求数据包 (即前面提到的 name registration request 数据包), 服务器会回复注册响应数据包 (即前面提到的 positive name registration response 或 negative name registration response 数据包) 来告知其是否注册成功^②。

客户端和服务端会利用 NBDS, 来行使服务宣告功能。网络中的主机可借助于 NBDS 来宣告: 本机 NetBIOS 名称、本机可向网络中的其他主机提供那些服务, 以及如何访问这些服务等。NBDS 运行于 UDP 138 端口。

NBSS 的作用包括: 在主机间建立会话, 跨网络打开、保存或执行远程 (主机上的) 文件^③。NBSS 运行于 UDP 138 端口。

NetBIOS 协议族中还包括 SMB (服务器消息块) 和 SPOOLS 等协议。当 SMB 协议运行于 NBSS 之上时 (即 SMB 数据由 NetBIOS 会话服务数据包来承载), 该协议所起作用为事务操纵; 当 SMB 协议运行于 NBDS 之上时 (即 SMB 数据由 NetBIOS 数据报服务数据包来承载), 该协议将用于服务通告。SPOOLS 协议则用于打印服务。对 NetBIOS 协议族的深层次讨论,

^① 译者注: 原文是“6. To see all SMB error messages, type the filter `smb.nt_status != 0x0`. You will get all error responses, as shown in the following screenshot”。如此粗制滥造的文字译者实在不能忍受, 译文酌改。

^② 译者注: 原文还有两句“Microsoft environment was implemented with WINS when most networks did not use it, and later it was replaced by DNS. It works over UDP port 137”。译者认为作者是在胡言乱语, 保留原文, 不予翻译。

^③ 原文是“NBSS is used to establish sessions between hosts, open or save files, and execute remote files and other sessions over the network”。原文粗制滥造, 实在没法翻译。

跟本书的内容无关。要想排除 NetBIOS 协议故障，需要遵照本节内容里的指示——注意观察（Wireshark 抓到的）NetBIOS（会话服务）数据包所含的 SMB 错误代码。

11.6.4 拾遗补缺

为了让读者更好的理解 NetBIOS 协议，本节会举几个与该协议有关的抓包示例。

示例 1——应用程序卡顿

通过图 11.24 所示的 Wireshark 截屏，可发现该应用程序卡顿的原因。

仔细观察 Wireshark 截屏中的数据包，不难发现下述情况。

- 1. 一台主机（IP 地址以 203 打头）在请求访问一台服务器（IP 地址为 10.1.70.95）上的目录\\NAS01\\SAMIM 时（对应于身背 PATH\\NAS01\\SAMIM 字样的数据包），遭到了服务器的拒绝（对应于身背 STATUS_ACCESS_DENIED 字样的数据包）。

No.	Time	Source	Destination	Protocol	Info
26562	362.699257	203.	10.1.70.95	SMB	Tree Connect AndX Request, Path: \\NAS01\\SAMIM
26563	362.717483	10.1.70.95	203.	SMB	Tree Connect AndX Response, Error: STATUS_ACCESS_DENIED
26564	362.717635	203.	10.1.70.95	SMB	Logoff AndX Request
26565	362.734572	10.1.70.95	203.	SMB	Logoff AndX Response
26572	362.853441	203.	10.1.70.95	TCP	nuts-dem > netbios-ssn [ACK] Seq=226260 Ack=359968 Win=
36000	482.813425	10.1.70.95	203.	TCP	netbios-ssn > nuts-dem [ACK] Seq=339967 Ack=226260 Win=
36001	482.813508	203.	10.1.70.95	TCP	[TCP Dup ACK 26572#1] nuts-dem > netbios-ssn [ACK] Seq=
44869	602.799670	10.1.70.95	203.	TCP	[TCP Keep-Alive] netbios-ssn > nuts-dem [ACK] Seq=35996
44872	602.800321	203.	10.1.70.95	TCP	[TCP Keep-Alive ACK] nuts-dem > netbios-ssn [ACK] Seq=2
55372	722.786747	10.1.70.95	203.	TCP	[TCP Keep-Alive] netbios-ssn > nuts-dem [ACK] Seq=35996
55375	722.787380	203.	10.1.70.95	TCP	[TCP Keep-Alive ACK] nuts-dem > netbios-ssn [ACK] Seq=2
59751	798.181386	10.1.70.95	203.	NBSS	Session keep-alive
59758	798.390573	203.	10.1.70.95	TCP	nuts-dem > netbios-ssn [ACK] Seq=226260 Ack=359972 Win=
60622	816.812860	203.	10.1.70.95	SMB	Tree Disconnect Request
60623	816.829093	10.1.70.95	203.	SMB	Tree Disconnect Response
60627	816.984481	203.	10.1.70.95	TCP	nuts-dem > netbios-ssn [ACK] Seq=226299 Ack=360011 Win=
64565	936.948575	10.1.70.95	203.	TCP	[TCP Keep-Alive] netbios-ssn > nuts-dem [ACK] Seq=36001
64568	936.949116	203.	10.1.70.95	TCP	[TCP Keep-Alive ACK] nuts-dem > netbios-ssn [ACK] Seq=2
75087	1056.936316	10.1.70.95	203.	TCP	[TCP Keep-Alive] netbios-ssn > nuts-dem [ACK] Seq=36001
75088	1056.936568	203.	10.1.70.95	TCP	[TCP Keep-Alive ACK] nuts-dem > netbios-ssn [ACK] Seq=2
84066	1142.229579	10.1.70.95	203.	TCP	netbios-ssn > nuts-dem [RST, ACK] Seq=360011 Ack=226295

图11.24

- 2. 那台主机向服务器发出取消访问的请求（对应于身背 logoff andx request 字样的数据包），得到了服务器的确认（服务器发出了身背 logoff andx Response 字样的数据包）。
- 3. 由于（安装在那台主机上）实际发出目录访问请求的应用程序一直在等待，因此主机和服务器之间会互发身背 TCP keep-alive 字样的数据包，来保持 TCP 连接。
- 4. 片刻之后，那台主机向服务器发出断开连接请求（对应于身背 Tree disconnect request 字样的数据包），征得了服务器的同意（对应于服务器发出了身背 Tree disconnect Response 字样的数据包）。
- 5. （安装在那台主机上）实际发出目录访问请求的应用程序继续等待，主机和服务器之间继续互发身背 TCP keep-alive 字样的数据包，来保持 TCP 连接。

6. 服务器发出 RST 位置 1 的 TCP 数据包，断开之前建立的 TCP 连接。

现在知道为什么最终用户反映该应用程序卡顿了吧？

示例 2——由 SMB 协议造成的广播风暴

有一次，作者接到了一个客户的紧急求助电话。客户在电话里反映：其公司总部（HQ）与某一远程站点之间通信中断。以下所列为该客户的 HQ 网络与那个远程站点网络的基本资料。

- 远程站点网络隶属于 IP 子网 172.30.121.0/24，网关为 172.30.121.254。
- HQ 网络的 IP 地址段为 172.30.0.0/24。
- HQ 网络与远程站点网络之间通过运营商提供的 L3 MPLS IP-VPN 网络互连。

为了查明断网的原因，作者进行了如下操作。

1. 作者在远程站点网络内的一台主机上 ping HQ 网络内的一台服务器，但 ping 不通。
2. 作者致电提供 MPLS IP-VPN 网络的运营商；运营商告知作者，据他们的网管系统显示，那条 MPLS VPN 链路负载为空。
3. 作者继续在那台主机上 ping 其默认网关（远程站点网络内的路由器）的 IP 地址 172.30.121.254，同样 ping 不通。也就是说，问题出在远程站点的本地 LAN 内，主机与默认网关（路由器）之间无法通信。
4. 作者将安装了 Wireshark 的笔记本接入 LAN 交换机，开启端口镜像功能，抓取连接了路由器（内部 LAN 接口）的端口的流量。于是，作者看到了图 11.25 所示的景象。

No.	Time	Source	Destination	Protocol	Info
22	0.000002	172.30.121.1	172.30.121.255	SMB Mailslot	Write Mail Slot
23	0.000001	172.30.121.1	172.30.121.255	SMB Mailslot	Write Mail Slot
24	0.000001	172.30.121.1	172.30.121.255	SMB Mailslot	Write Mail Slot
25	0.000001	172.30.121.1	172.30.121.255	SMB Mailslot	Write Mail Slot
26	0.000002	172.30.121.1	172.30.121.255	SMB Mailslot	Write Mail Slot
27	0.000910	172.30.121.1	172.30.121.255	SMB Mailslot	Write Mail Slot
28	0.000002	172.30.121.1	172.30.121.255	SMB Mailslot	Write Mail Slot
29	0.000001	172.30.121.1	172.30.121.255	SMB Mailslot	Write Mail Slot
30	0.000001	172.30.121.1	172.30.121.255	SMB Mailslot	Write Mail Slot
31	0.000857	172.30.121.1	172.30.121.255	SMB Mailslot	Write Mail Slot

Frame 1: 277 bytes on wire (2216 bits), 277 bytes captured (2216 bits) on interface 1
Ethernet II, Src: Hewlett-2b:5d:e3 (f4:ce:46:2b:5d:e3), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 172.30.121.1 (172.30.121.1), Dst: 172.30.121.255 (172.30.121.255)
User Datagram Protocol, Src Port: netbios-dgm (138), Dst Port: netbios-dgm (138)
NetBIOS Datagram Service
SMB (Server Message Block Protocol)
SMB Mailslot Protocol
Data (65 bytes)

图11.25

5. 由图 11.25 可知，IP 主机 172.30.121.1 几乎是每隔 1 微秒（①）便发出一个协议属性为 SMB Mailslot（④），身背 Write Mail Slot 字样（⑤）的广播数据包（③）。

6. 为弄清身背 Write Mail Slot 字样的广播包的发送频率，作者动用了 Wireshark 自带的 IO Graph 工具，该工具给出的结果是 5000 个数据包/秒，如图 10.26 所示。而那台路由器的内部 LAN 接口只是 10Mbit/s 接口，根本就承受不了那么多蜂拥而来的广播包（即便是 100Mbit/s 或 1000Mbit/s 接口，也同样无法承受）。
7. 作者在 Google 和 Microsoft 站点上搜索了一翻，但一无所获。于是，便开始逐一停止运行于 IP 主机 172.30.121.1 上的可疑服务。在停止可疑服务的同时，保持对 Wireshark 抓包主窗口数据包列表区域的关注（见图 11.26）。当作者关停了一个名为 LS3Bcast.exe 的服务时，Wireshark 就抓不到身背 Write Mail Slot 字样的广播包了。看来这个名叫 LS3Bcast.exe 的服务才是导致断网的罪魁祸首。

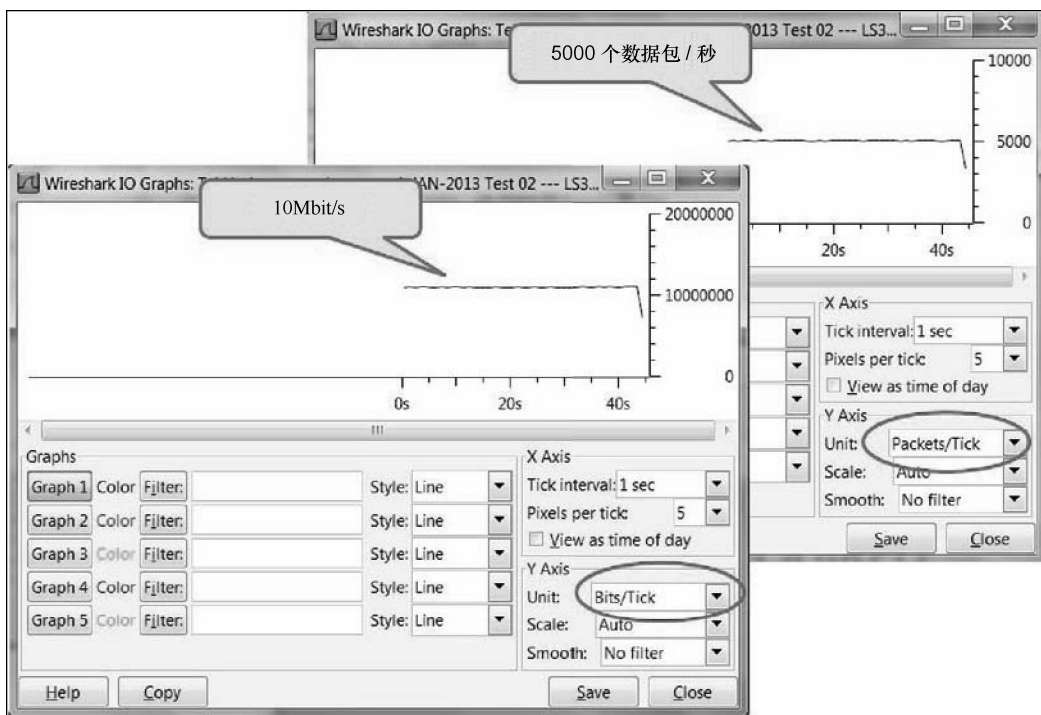


图11.26

11.7 数据库流量及常见故障分析

有些读者一定会对作者在本章安排这么一节内容而感到不解。在 IT 领域里，网络技术与数据库技术毕竟是截然不同的两个分支。精通数据库和应用程序开发技术的人一般都不懂网络技术，反之亦然，当然少数绝世牛人除外。所以说，一般会让网络工程师去排除数据库故障，

也不可能让 DBA 去排除网络故障。不过，通过抓取并分析流淌在网络中的流量，网络工程师可以帮助 DBA 解决某些数据库故障。

在大多数情况下，当应用程序无法使用时，最终用户会首先认定是网络问题，从而对网络技术人员心存不满。因此，网络技术人员必须能准确判断故障是否真的跟网络有关。在某些特殊场合下，网络技术人员或许还要动用 Wireshark 抓取并分析网络中的流量，去配合 DBA 协查数据库故障。

11.7.1 准备工作

当有用户反映“网络太慢”的时候，网络技术人员首先需要判断网络是不是真的太慢。为此，请按下面所述步骤行事。

11.7.2 操作方法

若怀疑是数据库故障，请按以下步骤行事。

1. 当有用户投诉网络太慢时，需要弄清楚以下事宜。

- 遭遇的是大面积用户投诉，还是个别用户投诉？影响的是总部站点网络内的用户，还是某个分支机构网络内的用户？若整个网络内的用户都受到了影响，则可以断定故障并非出现在用来互连总部和分支机构网络的 WAN 线路上面。
- 是所有用户都反映网速慢，还是安装了某种应用程序的那一类用户反映网速慢？若为后一种情况，则应把注意力放到该应用程序上面。
- 是运行该应用程序客户端软件的主机与服务器之间的通信链路过载吗？

2. 在弄清了上述事宜之后，请按以下步骤行事。

- 配置 LAN 交换机，将其连接某台 PC、服务器或路由器（用来连接远程站点网络的 WAN 路由器）的端口的流量，甚至是某个 VLAN 的流量，重定向至 Wireshark 主机。打开 Wireshark 软件，开始抓包。
- 在抓包的同时，点击 **Analyze** 菜单下的 **Expert Info** 菜单项，Expert Infos 窗口会立刻弹出。要注意观察该窗口中各标签栏内与 TCP 有关的各种事件（尤其是 Errors 和 Warnings 事件），了解与这些事件相对应的数据包的源、目 IP 地址和源、目 TCP 端口号信息。这些信息将助于快速定位故障的原因。



注意

用 Wireshark 抓取并分析 Internet 链路的流量时，一定会观察到很多来自/发往 Internet Web 站点或 Email 服务器的 TCP 数据包都身背了 TCP retransmissions 或 TCP duplicate ACK 等字样。没有办法，访问 Internet 就是这样。然而，用 Wireshark 抓取并分析内网流量时，身背 TCP retransmissions 等字样的 TCP 数据包最多只应占总 TCP 流量的 0.1%~0.5%。

3. 若判断是网络中的主机、服务器或其他网络设备发生故障，请按本书之前各章所述来解决故障。不过，某些网络自身的问题也会对数据库应用的行为产生影响。在下面这个例子中，读者可以看到一客户机通过一条往返延迟为 35~40 毫秒的链路^①，与服务器通信时的“举动”^②。

- 由图 11.27 可知，Wireshark 截屏中所示数据包全都归属于编号为 8 的 TCP 数据流(①)，前三个 TCP 数据包是建立 TCP 连接的三次握手(TCP SYN/SYN-ACK/ACK) TCP 报文段。作者已将 Wireshark 显示出的第一个数据包的时间设置为参考时间(②)(选中第一个数据包，点击右键，选择右键菜单中的 **set time reference (toggle)** 菜单项)。构成这条编号为 8 的 TCP 数据流的 TCP 数据包的总数为 371(③)。

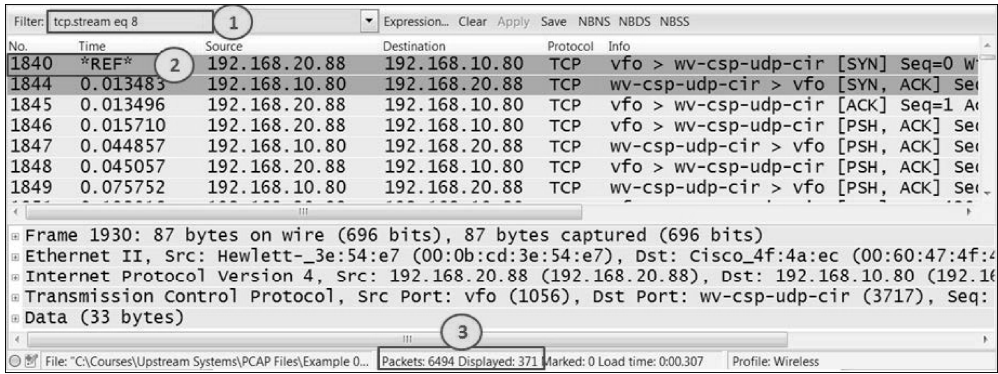


图11.27

- 由图 11.28 可知，客户机从每次发出数据库查询，到收到来自服务器的查询响应所耗时间约为 35 毫秒。

① 译者注：通过后面的抓包文件可以看出，这条链路应该是 wireless 链路，这么重要的信息作者却偏未指明。
② 译者注：原文是 “If you see problems in the network, solve them as we learned in previous chapters. But, there are some network issues that can influence database behavior. In the following example, we see the behavior of a client that works with the server over a communication line with a roundtrip delay of 35 to 40 ms”。作者举的这个例子漏洞百出，由于译者并未参与此次故障排除，译文只能按字面意思直译。

No.	Time	Source	Destination	Protocol	Info
1981	35.833309	192.168.10.88	192.168.10.80	TCP	vfo > wv-csp-udp-cir [PSH, ACK] Seq=...
1982	35.869385	192.168.10.80	192.168.10.88	TCP	wv-csp-udp-cir > vfo [PSH, ACK] Seq=...
1983	35.869930	192.168.10.88	192.168.10.80	TCP	vfo > wv-csp-udp-cir [PSH, ACK] Seq=...
1984	35.905654	192.168.10.80	192.168.10.88	TCP	wv-csp-udp-cir > vfo [PSH, ACK] Seq=...
1985	35.906194	192.168.10.88	192.168.10.80	TCP	vfo > wv-csp-udp-cir [PSH, ACK] Seq=...
1986	35.944428	192.168.10.80	192.168.10.88	TCP	wv-csp-udp-cir > vfo [PSH, ACK] Seq=...
1987	35.953804	192.168.10.88	192.168.10.80	TCP	vfo > wv-csp-udp-cir [PSH, ACK] Seq=...

图11.28

- 由于这条编号为 8 的 TCP 数据流由 371 个数据包构成,因此客户机完成数据库查询所耗时间约为 13 秒 (371×35 毫秒)^①。再加上 (因无线网络质量所导致的) 某些 TCP 重传,使得操纵该客户机的用户要等待 10~15 秒 (或更长的时间) 才能完成本次数据库查询任务。
- 在这种情况下,应与 DBA 协商如何通过改进数据库应用程序来降低过往于网络的数据包的数量。或者,可以让那台客户机不要通过无线网络去访问服务器,改用其他方式 (比如,终端服务器或 Web 方式) 去访问服务器^②。

4. 有时,通过 Wireshark 抓包分析,还可以发现应用程序自身的问题。在图 11.29 所示的 Wireshark 截屏中,可以看见客户机向服务器连续发送了 5 个身背 TCP retransmissions 字样的 TCP 数据包。之后,客户机与服务器之间又重新通过三次握手,建立起了 TCP 连接 (③)。这很像是 TCP 故障,但只有当用户在该应用程序客户端软件的特定窗口内执行某些操作时,才能在 Wireshark 抓包主窗口的数据包列表区域内看见上述景象。出现这样的问题,是由于该应用程序设计失当,每当用户执行上操作时,服务器端软件就会停止响应客户端软件发起的 TCP 连接^③。

No.	Time	Source	Destination	Protocol	Info
274	0.078889	192.168.3.50	192.168.200.227	TCP	http > vrtp [ACK] Seq=1 Ack=59884 Win=...
275	0.380166	192.168.200.227	192.168.3.50	TCP	[TCP Retransmission] [TCP segment of...
276	0.983678	192.168.200.227	192.168.3.50	TCP	[TCP Retransmission] [TCP segment of...
277	2.195589	192.168.200.227	192.168.3.50	TCP	[TCP Retransmission] [TCP segment of...
278	4.604757	192.168.200.227	192.168.3.50	TCP	[TCP Retransmission] [TCP segment of...
279	9.432867	192.168.200.227	192.168.3.50	TCP	[TCP Retransmission] [TCP segment of...
280	18.989050	192.168.200.227	192.168.3.50	TCP	rcts > http [SYN] Seq=0 Win=65535 Len=...
281	18.994054	192.168.3.50	192.168.200.227	TCP	http > rcts [SYN, ACK] Seq=0 Ack=1 Win=...
282	18.994085	192.168.200.227	192.168.3.50	TCP	rcts > http [ACK] Seq=1 Ack=1 Win=655...
283	18.994264	192.168.200.227	192.168.3.50	TCP	[TCP segment of a reassembled PDU]
284	18.994280	192.168.200.227	192.168.3.50	TCP	[TCP segment of a reassembled PDU]
285	19.000271	192.168.3.50	192.168.200.227	TCP	http > rcts [ACK] Seq=1 Ack=537 Win=6...

图11.29

① 译者注: 应该是 371/2*35 毫秒。
② 译者注: 原文是 “In this case, you should consult with the DBA on how to significantly reduce the number of packets that run over the network; or you can move to another way of access, for example, terminal server or web access” 。
③ 译者注: 整段原文是 “Another problem that can happen is that you will have a software issue that will reflect in the capture file. If you have a look at the following screenshot, you will see that there are five retransmissions (1), and then a new connection is opened from the client side (3). It looks like a TCP problem but it occurs only in a specific window in the software. It is simply a software procedure that stopped processing, and this stopped the TCP from responding to the client (2)” 。对原文无语, 译者也只能译到这个份上了。还有就是, 作者描述的这一场景似乎与数据库关系不大。

11.7.3 幕后原理

常言道，隔行如隔山，网络技术人员肯定玩不转数据库。不过，网络技术人员有义务配合 DBA 去排除与数据库有关的故障，这也是作者在本章安排这些内容的目的。

11.7.4 拾遗补缺

在 Wireshark 抓包主窗口的数据包列表区域中，选中一个隶属于某条数据库客户端与数据库服务器间 TCP 会话的数据包，点击右键，选择右键菜单中的 **Follow TCP Stream** 菜单项，**Follow TCP Stream** 窗口会立刻弹出^①。可以让 DBA 通过该窗口所示内容，来了解经由网络传输了哪些数据。

若数据库客户端与数据库服务器之间的网络延迟较高（比如，数据库客户端通过移动网络访问数据库服务器），则实际操纵应用程序的用户就会觉得卡顿。此时，需考虑让用户采用其他方式来访问数据库服务器（比如，采用终端服务器或 Web 方式）。

数据库服务器的运作方式同样重要。若某台数据库服务器需要从另一台文件服务器获取数据，即便数据库客户端与该数据库服务器之间通信正常，但只要前两者之间的通信出现问题，还是会对数据库应用产生影响。在排除数据库相关故障时，请务必弄清待查目标的各个环节^②。

对网络技术人员而言，最重要的是要跟真正有本事的 DBA 处好关系，说不定哪天就能用上他们。

① 译者注：原文是 “When you right-click on one of the packets in the database client to the server session, a window with the conversation will open”。

② 译者注：原文是 “An important issue is how the database works. If the client is accessing the database server, and the database server is using files shared from another server, it can be that the client-server works great; but the problems come from the database server to the shared files on the file server. Make sure you that know all these dependencies before starting with your tests”。

第12章

SIP、多媒体和IP电话

本章将介绍如何使用 Wireshark 去诊断并解决与 IP 电话、语音及视频有关的故障。本章涵盖以下主题：

- ▶ 内置于 Wireshark 的 IP 电话及多媒体流量专用分析工具的用法；
- ▶ SIP 故障分析；
- ▶ RTP/RTCP 故障分析；
- ▶ 视频及视频监控应用排障场景；
- ▶ IPTV 应用排障场景；
- ▶ 视频会议应用排障场景；
- ▶ 排除 RSTP 故障。

12.1 简介

包括 IP 电话、视频会议、视频监控以及在线教育在内的各类多媒体应用是现代化网络通信的重要组成部分。

这些多媒体应用对网络的要求与前面介绍的众多企业网应用有所不同。像 HTTP、E-mail 以及文件共享这样的企业网应用需要较高的带宽，而 IP 电话等多媒体应用所需带宽不高（只需要区区 100kbit/s 的带宽），但对网络的延迟及抖动（延迟的差异）却非常敏感。对于那些拥有远程分支机构的企业而言，在各分支机构网络内，绝大多数企业网应用都会引发较高的下载流量，而诸如视频监控系统之类的多媒体应用却会触发较高的上传流量，因为总部要想监控各个分支机构，就得从各分支机构网络下载视频监控图像。

本章会关注包括语音、视频在内的多媒体应用以及它们在网络中的运作方式，同时还会介绍当这些应用发生故障时，如何利用 Wireshark 排除故障。本章涉及：会话初始协议（Session Initiation Protocol, SIP）、实时协议/实时传输控制协议（Real Time Protocol /Real-time Transport Control Protocol, RTP/RTCP）、实时流协议（Real Time Streaming Protocol, RTSP）等常用多媒体协议。

本章首先会介绍内置于 Wireshark 的 IP 电话及多媒体流量专用分析工具。然后，将介绍如何解决 SIP 故障，以及如何排除 RTP/RTCP 会话故障。最后，会关注如何排除各种视频系统（比如，视频会议及视频监控系统）故障。

12.2 使用内置于Wireshark 的IP电话及多媒体流量专用分析工具

先来了解一下内置于 Wireshark 的语音、视频及多媒体专用分析工具。

12.2.1 准备工作

处理语音通话、视频电话会议或其他多媒体会话应用故障时，请按图 12.1 所示安置 Wireshark 主机并在 LAN 交换机上开启端口镜像功能。

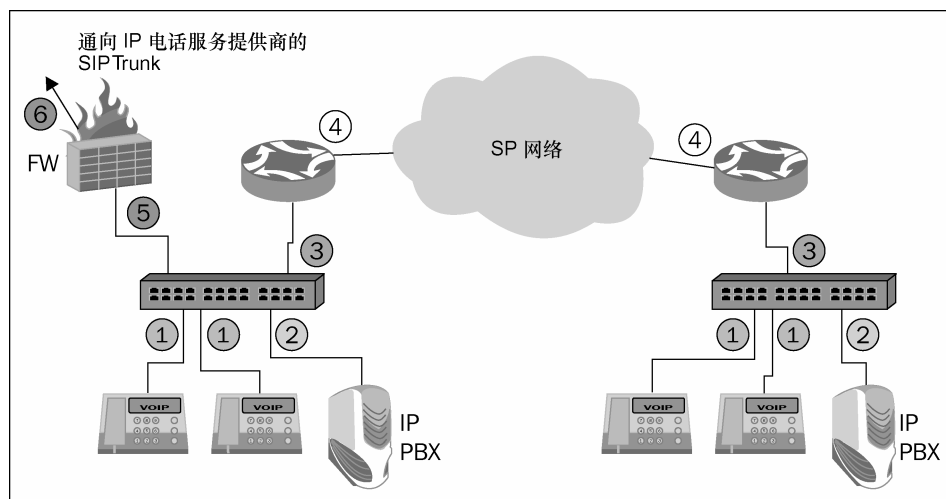


图12.1

1. 若怀疑是用户使用的 IP 电话（图 12.1 中的①处）故障，请把 Wireshark 主机接入交换机，开启交换机的端口镜像功能，将连接 IP 电话的端口的流量重定向至 Wireshark 主机。

2. 若怀疑是各站点内的整个 IP 电话网络故障（IP PBX 故障，图 12.1 中的②处），请把 Wireshark 主机接入交换机，开启交换机的端口镜像功能，将连接 IP PBX 的端口的流量重定向至 Wireshark 主机。
3. 若怀疑站点间网络存在连通性故障（WAN 线路故障，图 12.1 中的③、④处），请把路由器上 WAN 端口的流量重定向给 Wireshark 主机^①。
4. 若怀疑 SIP Trunk 故障，请设法将防火墙的内口（图 12.1 中的⑤处）或外口（图 12.1 中的⑥处）流量重定向至 Wireshark 主机。防火墙的外口所连线路既有可能是 Internet 线路，也有可能是通向 IP 电话服务提供商的专线。

12.2.2 操作方法

请在 Wireshark 主抓包窗口中点击 **Telephony** 菜单，如图 12.2 所示。

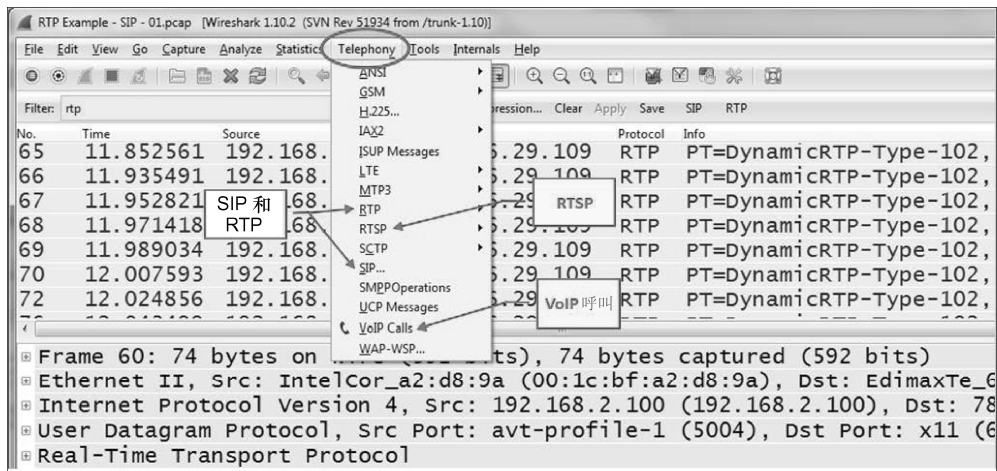


图 12.2

要分析 RTP 流量，需使用 **Telephony** 菜单中的以下菜单项功能。

1. 可按图 12.3 所示，选择 **Telephony** 菜单，点击 **RTP** 菜单项下的 **Show All Streams** 子菜单项。
2. RTP Streams 窗口会立刻弹出，如图 12.4 所示。
3. 通过 RTP Streams 窗口，可以获知以下细节。

^① 译者注：若图中的那对路由器通过串行 SDH 线路互连，请问怎么用 Wireshark 抓包？退一步来讲，即便那对路由器通过 SMTP 线路互连，如果存在连通性故障，无需使用 Wireshark 抓包应该也能判断故障所在。

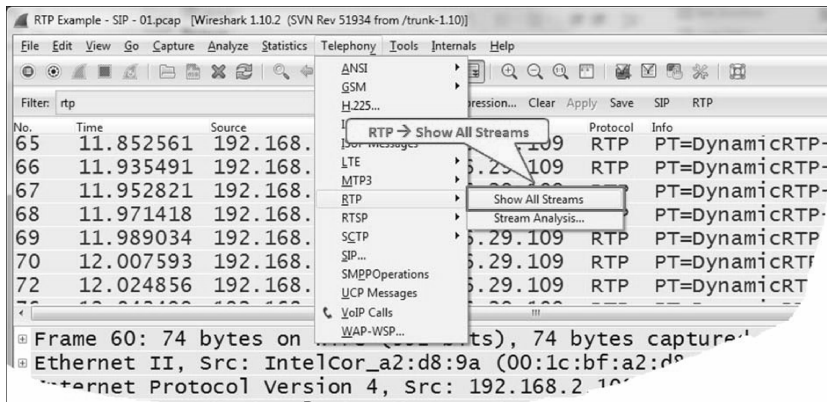


图 12.3

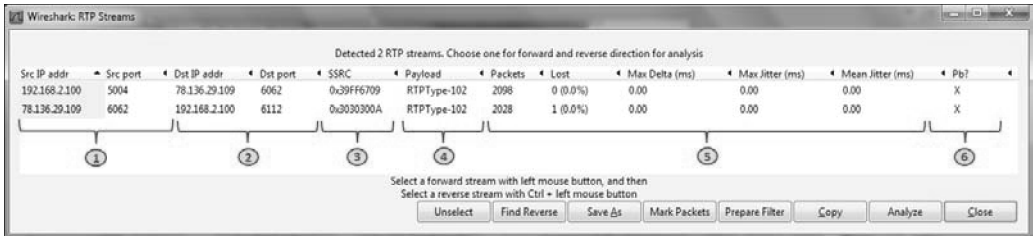


图 12.4

- RTP 流的源、目 IP 地址及源、目 UDP 端口号。
- 同步源（SSRC）标识符（RTP 流的标识符）。
- RTP 流量的净载类型（通常所指为编码类型）。
- RTP 流的属性信息，包括：抓到的数据包的总数、丢包数、数据包间的最长间隔时间、最长抖动时间及平均抖动时间。
- Pb?: 表示 RTP 流中发生的常见问题。

在 RTP Streams 窗口的底部，还有如下按钮。

- **Unselect:** 选中 RTP Streams 窗口中的一条语音流之后，点击此按钮，便会取消之前的选择。
- **Find reverse:** 由语音通话（或多媒体呼叫）所生成的 RTP 数据流必然是一来一回，选中 RTP Streams 窗口中的一条 RTP 流之后，点击此按钮，该 RTP 流便会与与之配套的逆向 RTP 流同时以亮灰色面目示人。
- **Save as:** 点此按钮，可将选中的 RTP 流以 rtpdump 文件格式存盘。欲知更

多与 rtpdump 文件格式有关的内容,请访问 <http://www.cs.columbia.edu/irt/software/rtptools/>。

- **Mark packets:** 不起作用。
 - **Prepare filter:** 点此按钮,在 Wireshark 抓包主窗口的 Filter 输入栏内,便会基于选中的 RTP 流,创建与之配套的显示过滤器。
 - **Copy:** 先点此按钮,再打开一个文本编辑器,按右键,点“粘贴”,便可将 RTP Streams 窗口中的内容复制进该文本编辑器。
 - **Analyze:** 选择 RTP Streams 窗口中的 RTP 流之后,点此按钮,RTP Stream Analysis 窗口会立刻弹出。在 Wireshark 抓包主窗口中先选中一个 RTP 数据包,再点击 **Telephony** 菜单,选择 RTP 菜单项中的 **Stream** 子菜单项,也会导致相同窗口的弹出。
 - **Close:** 点此按钮,RTP Streams 窗口将会关闭。
4. 要分析 SIP 流量,请在 Wireshark 抓包主窗口内点击 **Telephony** 菜单中的 **SIP** 菜单项,**SIP Packet Counter** 窗口会立刻弹出。在该窗口的 Filter 输入栏里输入 **ip (udp/sip)**,再点击 **Create Stat** 按钮,**SIP statistics with filter: ip (udp/sip)** 窗口会立刻弹出,如图 12.5 所示。

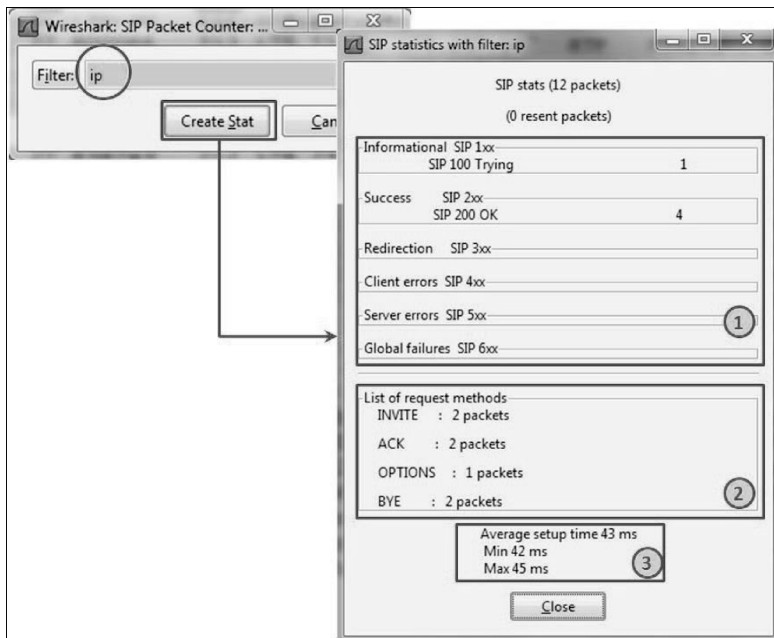


图 12.5

5. 通过 SIP statistics with filter: ip (udp/sip) 窗口，可了解到以下与 SIP 流量有关的统计信息。

- 含各种状态码的 SIP 响应数据包的个数（显示在图 12.5 中标以①的区域内）。
- 含各种 SIP 方法的 SIP 请求数据包的个数（显示在图 12.5 中标以②的区域内）。
- SIP 会话的建立时间（包含最大、最小及平均值）（显示在图 12.5 中标以③的区域内）。

6. 要分析 RTSP 流量，请在 Wireshark 抓包主窗口内选择 **Telephony** 菜单，点击 SIP 菜单项中的 **Packet Counter** 子菜单项，RSTP/Packet Counter 窗口会立刻弹出。在该窗口的 Filter 输入栏里输入 ip（或 rtsp，也可以什么都不输），再点击 **Create Stat** 按钮，**RTSP/Packet counter with filter: ip (rtsp)** 窗口会立刻弹出，如图 12.6 所示。

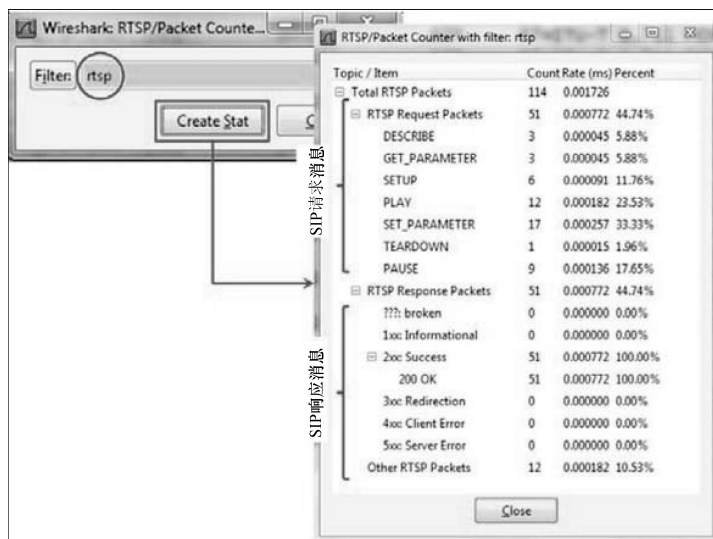


图 12.6

7. 要分析已抓取的数据包中的 VoIP 通话状况，请在 Wireshark 抓包主窗口内点击 **Telephony** 菜单中的 **VoIP Calls** 菜单项，**VoIP Calls** 窗口会立刻弹出，如图 12.7 所示。

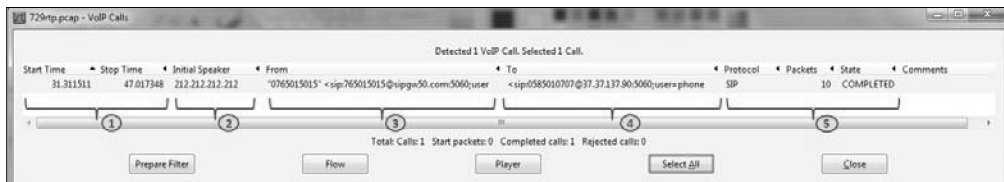


图 12.7

8. 通过 VoIP Calls 窗口，可以了解到以下 VoIP 通话状况。

- VoIP 呼叫的开始和结束时间（显示在图 12.7 中标以①的区域内）。
- 发起呼叫的 IP 地址（显示在图 12.7 中标以②的区域内）。
- 发起呼叫的 SIP 地址（主叫方）（显示在图 12.7 中标以③的区域内）。
- 接收呼叫的 SIP 地址（被叫方）（显示在图 12.7 中标以④的区域内）。
- 建立呼叫的信令协议（一般都是 SIP 协议）、信令协议（SIP 协议）数据包个数以及信令协议最终的状态。

12.2.3 幕后原理

应把语音、视频及多媒体流量的传递一分为二来看待。首先，是媒体流量（即实际的语音或视频流量）的传递；其次，是信令流量的传递，信令的作用包括：语音/视频呼叫的建立和终止、邀请参与者参与通话/视频等。一直以来，有以下两个协议族可用来提供语音或视频的信令功能。

- **ITU-T** 协议族，包括框架性协议 H.323、起注册及地址解析作用的 H.225 协议，以及起控制作用的 H.245 协议。
- **IETF** 协议族，包括信令协议 SIP（RFC 6878）以及用来描述会话参数的 SDP（会话描述协议）（RFC 4566）。

在过去几年，ITU-T 协议族已被逐步淘汰，如今的大多数多媒体应用使用的都是 IETF 协议族，这也是本章重点关注的内容。图 12.8 所示为供多媒体应用使用的 IETF 协议族的架构。

无论是 ITU-T 协议族，还是 IETF 协议族都是采用 RTP 和 RTCP 来传输实际的媒体流量，前者用于媒体流的传输，后者则用来控制并保证媒体流的质量。

图 12.8 所示为在 IP 网络中用来传递多媒体流量的 IETF 协议族的架构。

SIP 是一种作用于末端用户之间的信令协议。SDP 数据由 SIP 消息承载，用来描述会话。RSTP 用来行使流量（一般为视频流量，比如，IPTV 流量）控制功能。

RTP 是一种用来传递实际的媒体流的协议。在 RTP 之上，则运行着若干种用来压缩及解压缩语音、视频信号的编解码器。

RTCP 用来控制媒体流的质量，RSVP 则是一种在网络中确保服务质量的协议。

图 12.8 中呈现的所有协议都基于 TCP/IP 协议族，本章稍后会详尽探讨其中的大多数协议。

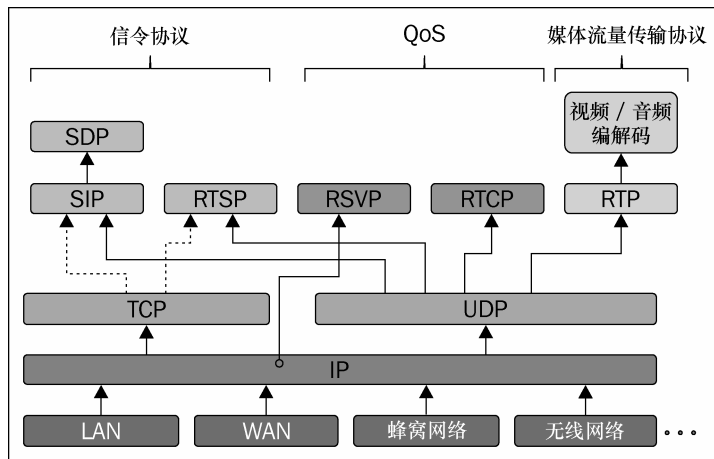


图 12.8

12.2.4 拾遗补缺

由于 SIP 协议所使用的端口号固定，因此 Wireshark 总是把目的端口为 5060 (SIP) 或 5061 (SIP over TLS，即受 TLS 保护的安全 SIP) 的 TCP 或 UDP 数据包识别为 SIP 协议数据包。虽然 IETF 标准规定 SIP 协议报文可封装在 TCP 报文段或 UDP 数据报内，但在一般情况下，SIP 协议报文总是以 UDP 数据报的面目示人。

RTP 和 RTCP 协议所使用的端口号并不固定，RTP 协议流量的源和目的端口号总是为偶数，而与之对应的 RTCP 协议流量的源和目的端口号则总是分别采用下一个奇数。比方说，如 RTP 协议流量的源和目的端口为 5004 和 2006，那么与之对应的 RTCP 协议流量的源和目的端口号将会是 5005 和 2007。所以说，在默认情况下，Wireshark 是识别不出 RTP 和 RTCP 协议流量的，只能将那些流量解析为 UDP 数据包。

要想让 Wireshark 自动识别 RTP 和 RTCP 协议流量，有以下两种方法。

1. 在 Wireshark 抓包主窗口的数据包列表区域内选择一个隶属于 RTP 流的 UDP 数据包，点击右键，选择右键菜单中的 **Decode as** 菜单项，在弹出的 Decode as 窗口的右边找到 **RTP** 协议，点击 **OK** 按钮，关闭窗口。
2. 在 Wireshark 抓包主窗口中，点击 **Edit** 菜单里的 **Preferences** 菜单项。在弹出的 Preferences 窗口中，点击左边 **Protocol** 配置选项之前的+号，选择其中的 **RTP** 协议，勾选右边与 **Try to decode RTP outside of conversations** 挂钩的复选框。在大多数情况下，可让 Wireshark 以这种方式来自动解码 RTP 协议数据包。

上述方法同样适用于 RTCP 协议流量的自动识别。

12.3 SIP故障分析

如上一节所述，**SIP**（RFC 6878）是一种信令协议，可用在一或多个参与者之间创建、修改及终止会话。在建立媒体会话交换 SIP 消息的过程中，包含会话参数信息的 **SDP** 消息也会被封装在 SIP 消息中传递。媒体会话创建之后，实际的语音或视频流才会被封装进 RTP 数据包中传输，可通过 **RTCP** 协议来进行控制（RTCP 为可选协议，可用于多媒体应用）。

在 IETF 制定的 SIP 标准文档中，把 SIP 会话的端点命名为用户代理（**User Agent, UA**）。因此，创建 SIP 会话的过程实际上就是 UA 之间进行协商，对会话参数达成一致意见的过程。至于提供定位会话参与者、注册、呼叫转发以及其他服务的网络主机，则被称为服务器（**server**），UA 可以向服务器发送注册请求消息、会话邀请请求消息以及其他各种请求消息。

本节将介绍用来传递媒体流量的 IETF 协议族中的信令协议——SIP，同时会详述如何使用 Wireshark 来排除 SIP 故障。下一节将介绍 RTP 和 RTCP 协议。

可把 IP 上的语音和视频故障分为以下两大类。

- **呼叫建立、修改及终止类故障：**诸如提起听筒后听不见拨号音、听得见拨号音但不能拨号、能拨号但被叫方电话无法振铃之类的情況都属于此类故障。此类故障一般都跟信令协议有关。
- **（语音或视频）信号质量类故障：**此类故障与信号质量有关，比如，语音通话过程中的信号干扰问题（有噪音）、视频画面停顿问题等。此类故障通常都是由网络问题、RTP 协议问题或各种介质问题所导致。

本节讨论的是如何解决第一类故障，第二类故障将放到下一节讨论。



以上两类 IP 上的语音和视频故障并不一定都跟网络故障有关，也有可能由设备（比如，电话交换机或末端设备）配置问题所引发。

12.3.1 准备工作

在处理呼叫建立、修改及终止类故障时，首先应考虑信令协议有无故障。SIP 是在 IP 网络环境中用来传递语音和视频的信令协议，它有两种主要的消息类型：SIP 呼叫控制请求消息和 SIP 响应消息。在 SIP 呼叫控制请求消息中，包含的是 SIP 方法，即 SIP 会话的一方发出的命令；SIP 会话的另一方会发出 SIP 响应消息作为应答。

排除 SIP 协议故障时，应重点关注 SIP 响应消息所包含的内容。

12.3.2 操作方法

把 Wireshark 主机接入网络，在交换机上开启端口镜像功能，将必要的流量重定向至 Wireshark 主机。以下所列的为在排除 SIP 协议故障时，需要关注的地方。

1. 当某一 UA 希望建立多媒体会话时，将会向远端 UA 发送包含 **INVITE** 方法的 SIP 呼叫控制请求消息，后者也会发出 SIP 响应消息回应前者，两者相互交流的过程如图 12.9 所示。

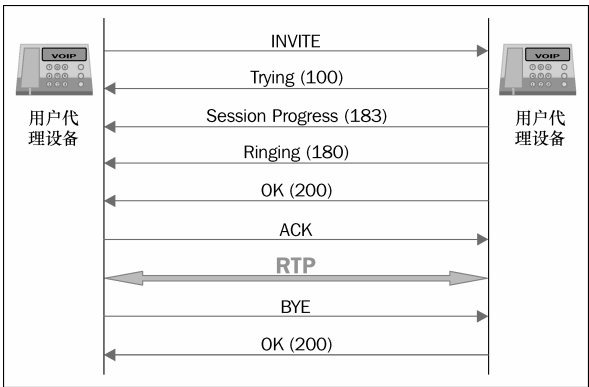


图12.9



注意

按 SIP 协议的行话来讲，末端用户设备被称为用户代理设备（**User Agent, UA**）。UA 既可以是发起呼叫也可以接收呼叫。IP 电话、摄像头、具备语音或视频功能的客户端软件，以及能参与 SIP 会话的任何软、硬件都属于 UA。

2. 图 12.10 中的 Wireshark 截屏清晰地展示了 VoIP 通话双方建立 SIP 会话时的过程。

No.	Time	Source	Destination	Protocol	Info
22	0.000000	93.172.121.146	192.168.3.2	SIP/SDP	Request: INVITE sip:90526613742@213.8.115.84:5068 , with ses
23	0.000000	192.168.3.2	93.172.121.146	SIP	Status: 100 Trying
27	0.960000	192.168.3.2	93.172.121.146	SIP/SDP	Status: 183 Session Progress , with session description
599	3.890000	192.168.3.2	93.172.121.146	SIP	Status: 180 Ringing
1834	9.950000	192.168.3.2	93.172.121.146	SIP/SDP	Status: 200 OK , with session description
1877	10.150000	93.172.121.146	192.168.3.2	SIP	Request: ACK sip:90526613742@213.8.115.84:5068;user=phone
3426	17.850000	93.172.121.146	192.168.3.2	SIP	Request: BYE sip:90526613742@213.8.115.84:5068;user=phone
3427	17.850000	192.168.3.2	93.172.121.146	SIP	Status: 200 OK

图12.10

3. 要想更为详细地了解 VoIP 通话双方相互交流的内情，请在抓包主窗口中点击 **Statistics** 菜单下的 **Flow Graph** 菜单项，Flow Graph 窗口会立刻弹出。同时点选该窗口中的 **Displayed packets**、**General flow** 和 **Standard source/destination addresses** 单选框后，再点击 **OK** 按钮，**Graph Analysis** 窗口会立刻弹出，如图 12.11 所示。
4. 对于一次正常的 SIP 会话建立过程而言，在图 12.11 所示的 Graph Analysis 窗口中，首先应能看到身负 INVITE 字样的 SIP 消息，而从对端发来的身背 Trying、Session

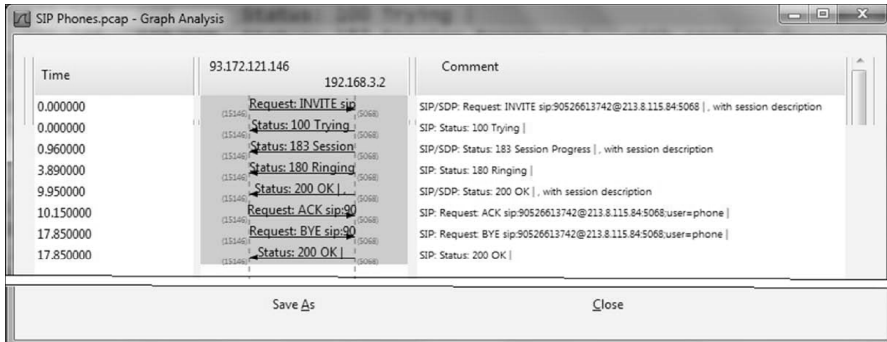


图12.11

Progress、Ringing 字样的 SIP 消息将紧随其后。

5. 通过图 12.11, 可以清楚地了解到 SIP 会话建立过程是怎样在发起方 (主叫方) (93.172.121.146) 和应答方 (被叫方) (192.168.3.2) 之间一步步进行的。

- 会话发起方发出身背 **INVITE** 字样的 SIP 呼叫控制请求数据包, 这一定是建立 SIP 会话的首个数据包。
- 会话应答方会立刻回复身背 **Trying** (状态码 100) 和 **Session progress** (状态码 183) 字样的 SIP 响应数据包, 三秒之后, 还会回复身背 **Ringing** (状态码 180) 字样的 SIP 响应数据包。此后, 当有人提起电话听筒时, 会话应答方还会继续发出身背 **OK** 字样的 SIP 响应数据包。



注意

建立 SIP 会话时, 被叫方未必会回复之前提到的全部三种 SIP 响应消息, 在某些情况下, 可能只会回复其中的一种或两种 SIP 响应消息, 但 SIP 会话仍能正常建立。本章后文会介绍当出现这种情况时, 如何辨别是否存在故障。

- 会话发起方发出身背 **ACK** 字样的 SIP 呼叫控制请求数据包, 至此, SIP 会话成功建立。

6. 在 SIP 会话建立阶段, 若抓到了身背 **ERROR** 字样的 SIP 响应消息, 则 SIP 会话将无法建立。



注意

SIP 消息通过 UDP 报文段来承载, 而 UDP 协议是一种非面向连接、不可靠的传输层协议。在网络质量堪忧的情况下, 包含 **INVITE** 方法的 SIP 呼叫控制请求消息有可能会在传输途中丢失。所以说, 如果 SIP 会话发起方收不到任何 SIP 响应消息, 则很有可能是其发出的包含 **INVITE** 方法的 SIP 呼叫控制请求消息因网络质量问题而未能抵达 SIP 会话接收方。

7. 若 UA 之间的通话要通过电话交换机来转接, 则 SIP 会话的建立过程应该向图 12.12 中 Wireshark Graph Analysis 窗口显示的那个样子。诸如 IP 电话交换机、呼叫管理器、IP PBX 之类的 VoIP 术语, 读者一定都耳熟能详, 其实它们都是指能在末端设备之间“居中调停”的电话交换机。这些 VoIP 术语所指代的设备, 在 SIP 术语系里被统称为“代理服务器”(Proxy Server), 下一节会就此展开讨论。

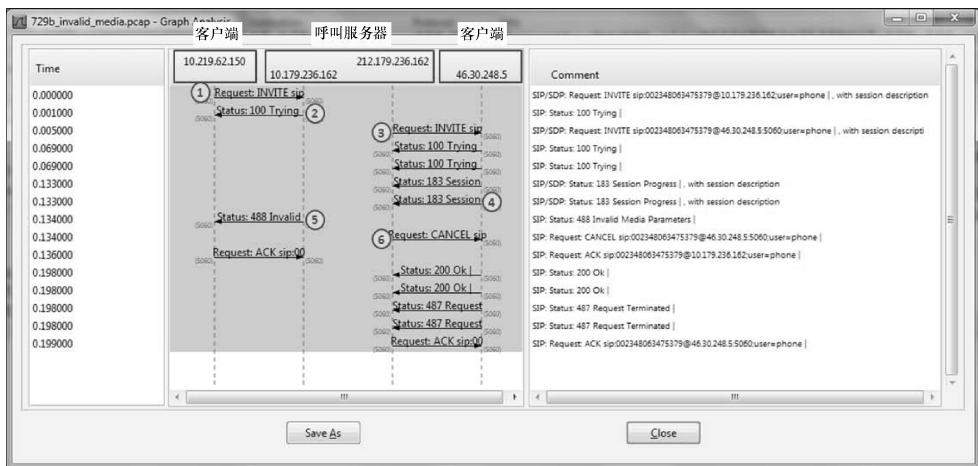


图12.12

8. 通过图 12.12 所示的 Wireshark 截屏, 可以看出, 该 IP 电话交换机有 2 个接口, 分别连接内网和外网, IP 地址分别为 **212.179.236.162** 和 **10.179.236.162**。
- 左侧的 UA (**10.219.62.150**) 向 IP 电话交换机发出身背 **INVITE** 字样的 SIP 控制请求消息 (在图 12.12 中被标记为①)。
 - IP 电话交换机向左侧的 UA 回复身背 **Trying** 字样的 SIP 响应消息 (在图 12.12 中被标记为②)。
 - IP 电话交换机向右侧的 UA 发出身背 **INVITE** 字样的 SIP 控制请求消息 (在图 12.12 中被标记为③)。
 - 右侧的 UA 向 IP 电话交换机回复身背 **Trying** (状态码为 **100**) 和 **session progresses** (状态码为 **183**) 字样的 SIP 响应消息 (在图 12.12 中被标记为④)。
 - 片刻之后, 当 IP 电话交换机感知到右侧的 UA 无进一步的响应, 于是发出包含状态 **488** 的 SIP 响应消息, 通知主叫方 (左侧的 UA) 被叫方 (右侧的 UA) 没有应答, 同时给出了没有应答的原因 (**invalid**) (在图 12.12 中被标记为⑤)。

- IP 电话交换机向右侧的 UA 发出身背 **Cancel** 字样的 SIP 控制请求消息（在图 12.12 中被标记为⑥）。

9. 要想定位 SIP 协议故障，请按下列步骤行事。

- 拥有一张最新的包含所有部件的网络拓扑结构图。
- 检查 Wireshark 抓包文件，看看是否出现了身背 **ERROR** 字样的 SIP 数据包。
- 弄清 VoIP 客户端或服务器为什么会发出身背 **ERROR** 字样的 SIP 数据包。

10. 以下诸表所列为 SIP 响应消息中所包含的各种状态码以及相关解释，它们几乎都定义于 RFC 3261，另有说明的除外。

状态码 1XX——临时/信息状态码

当 SIP 会话接收方收到（SIP 会话发起方发出的）SIP 控制请求消息，且正在做进一步处理时，便会（向 SIP 会话发起方）发出包含状态码 1xx 的 SIP 响应消息（来通报此事）。下表所列为各种临时/信息状态码以及相关描述。

状态码	事件名	描 述
100	Trying	表示服务器已收到并接受了（SIP 会话发起方发出的）SIP 控制请求消息，且正在对本次呼叫做进一步的处理
180	Ringing	表示接收本次呼叫的 UA 正在（发出振铃）提醒用户。当接收本次呼叫的 UA 发出振铃时，将会向主叫端 UA 回发包含此状态码的 SIP 响应消息
181	Call forward	表示呼叫被前传至另一目的地
182	Queued	表示被叫方暂时不便接听（比如，占线），服务器会暂时接管呼叫，以便稍后再试
183	Session progress	表示接收方服务器正在处理会话。与会话处理有关的详细信息，将会由包含此状态码的 SIP 响应消息头部中的某些字段来承载

状态码 2XX——成功状态码

当 SIP 会话接收方成功接收、识别并接受了（SIP 会话发起方发出的）SIP 控制请求消息时，便会（向 SIP 会话发起方）发出包含状态码 2xx 的 SIP 响应消息（来通报此事）。下表所列为各种成功状态码以及相关描述。

状态码	事件名	描 述
200	OK	表示（SIP 会话发起方发出的）SIP 控制请求消息已被（SIP 会话接收方）接受并成功处理。
202 （定义于 RFC 3265）	Accepted	表示（SIP 会话发起方发出的）SIP 控制请求消息已被（SIP 会话接收方）接受且正在处理，但操作尚未完成

状态码 3XX——重定向状态码

当 SIP 会话接收方只有采取重定向操作才能完成 SIP 会话发起方发出的请求时，便会向后者发出包含状态码 3XX 的 SIP 响应消息（来通报此事）。下表所列为各种重定向状态码以及相关描述。

状态码	事件名	描 述
300	Multiple choices	表示接收 SIP 控制请求消息的服务器在解析过消息中所含目的地址之后，发现有很多转发选择，亦即呼叫可被转发至多个指定位置。服务器可在包含此状态码的 SIP 响应消息的消息主体中列出资源的特征及位置，好让（发出 SIP 控制请求消息的）UA 从中选择一个最合适的呼叫目的地址，并将 SIP 控制请求消息重定向至该地址
301	Moved permanently ^①	
302	Moved temporarily	
305	Use proxy	
380	Alternative service	

状态码 4XX——客户端错误状态码

当（SIP 会话发起方发出的）SIP 控制请求消息包含了错误的语法，或消息不能被 SIP 会话接收方（服务器）解析时，后者便会向前者发出包含状态码 4xx 的 SIP 响应消息（来通报此事）。下表所列为各种客户端错误状态码以及相关描述。

状态码	事件名	描 述
400	Bad request	表示 SIP 会话接收方（服务器）因 SIP 控制请求消息中包含了错误的语法而无法对其进行解析
401	Unauthorized	表示 SIP 会话接收方（服务器）收到了 UA 发出的 SIP 控制请求消息，但需要对 UA 执行认证
402	Payment required	预留以供将来使用
403	Forbidden	表示 SIP 会话接收方（服务器）能识别 UA 发出的 SIP 控制请求消息，但拒绝对其进行处理。UA 不应重复发送之前曾发出过的 SIP 控制请求消息
404	Not found ^②	
405	Method not allowed	
406	Not acceptable	

① 译者注：原文对 SIP 状态码 301~380 的描述照抄自 RFC 3261。在译者看来，若未对 SIP 消息的格式做详细介绍，按字面意思翻译这些内容没有任何意义，且作者添加这些内容只是为了凑字数，跟本书主题无关，故而掠过不译。

② 译者注：原文对 SIP 状态码 404~493 的描述照抄自 RFC 3261。在译者看来，若未对 SIP 消息的格式做详细介绍，按字面意思翻译这些内容没有任何意义，且作者添加这些内容只是为了凑字数，跟本书主题无关，故而掠过不译。

续表

状态码	事件名	描 述
407	Proxy authentication required	
408	Request timeout	
410	Gone	
413	Request entity too large	
414	Request-URI too long	
415	Unsupported media type	
416	Unsupported URI scheme	
420	Bad extension	
421	Extension required	
423	Interval too brief	
424	Bad location information	
428	Use Identity header	
429	Provide referrer identity	
433	Anonymity disallowed	
436	Bad identity info	
437	Unsupported certificate	
438	Invalid identity header	
470	Consent needed	
480	Temporarily unavailable	
481	Call/transaction does not exist	
482	Loop detected	
483	Too many hops	
484	Address incomplete	
485	Ambiguous	
486	Busy here	
487	Request terminated	

续表

状态码	事件名	描 述
488	Not acceptable here	
491	Request pending	
493	Undecipherable	

状态码 5XX——服务器错误状态码

若 SIP 会话发起方收到了包含状态码 5xx 的 SIP 响应消息，则表示 SIP 会话接收方（服务器）无法解析本机发出的有效的 SIP 控制请求消息。下表所列为各种服务器错误状态码以及相关描述。

状态码	事件名	描 述
500	Server internal error	表示服务器因异常状况无法处理 UA 发出的 SIP 呼叫控制请求消息
501	Not implemented	表示（接收 SIP 控制请求消息的）服务器不支持（发出此消息的）UA 请求执行的功能
502	Bad gateway ^①	
503	Service unavailable	
504	Server time out	
505	Version not supported	
513	Message too large	

状态码 6XX——全局故障状态码

收到包含状态码 6xx 的 SIP 响应消息，则意味着 SIP 会话发起方之前（通过发出 SIP 控制请求消息）所请求的功能在任何一台服务器都无法执行。下表所列为各种全局故障状态码以及相关描述。

状态码	事件名	描 述
600	Busy everywhere	表示能成功联络上接收方系统，但此时该用户繁忙，无意接受呼叫，且无任何可替代的目标端系统（比如，语音信箱服务器）能够接收呼叫
603	Decline	表示能成功联络上接收端 UA，但该用户明确表态不愿意参与呼叫，且无任何可替代的目标端系统（比如，语音信箱服务器）愿意接收呼叫

① 译者注：原文对 SIP 状态码 502~513 的描述照抄自 RFC 3261。在译者看来，若未对 SIP 消息的格式做详细介绍，按字面意思翻译这些内容没有任何意义，且作者添加这些内容只是为了凑字数，跟本书主题无关，故而掠过不译。

续表

状态码	事件名	描 述
604	Does not exist anywhere	当 SIP 会话发起方希望联络某个用户时，会向服务器发出 SIP 控制请求消息，在消息中会包含该用户所在位置（URI）。若服务器所持权威信息表明，该用户不存在，便会回复包含状态码 604 的 SIP 响应消息
606	Not acceptable	表示 SIP 会话发起方能成功联络上用户的代理（agent），但在前者发出的 SIP 控制请求消息的消息主体中，由 SDP 描述的会话中的某些属性不被后者接受

12.3.3 幕后原理

SIP 是一种应用层协议，可用来建立、维护及终止两个或多个端节点之间的呼叫。

参与 SIP 会话的网络大致被为两类——客户端和服务端。

- 客户端是指发送 SIP 呼叫控制请求消息的实体（或应用程序）。
- 服务器则是发送 SIP 响应消息，对 SIP 呼叫控制请求消息进行回应的实体（或应用程序）。

参与 SIP 会话的另一类网络实体被称为网关（gateway），其作用是把 SIP 网络和公众交换电话网络（PSTN）连接在一起。

如图 12.13 所示，每个 SIP 终端都是由用户代理客户端（User Agent Client，UAC）和用户代理服务器（User Agent Server，UAS）构成，既可以发起呼叫请求（发出 SIP 控制请求消息），也可以对呼叫请求进行应答（发出 SIP 响应消息，作为对收到的 SIP 控制请求消息的回应）。

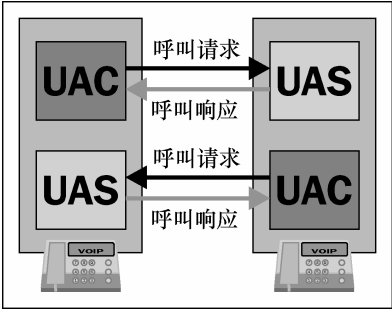


图12.13

SIP 服务器则分为以下几种类型。

- **SIP 代理服务器（Proxy Server）**：作用是先接收来自 UA 或其他代理服务器的 SIP 控

制请求消息，再转发或代为转发至其他位置。

- **SIP 重定向服务器 (Redirect Server):** 作用是接收来自 UA 或其他代理服务器的 SIP 控制请求消息，向它们回复包含状态码 3xx（重定向状态码）的 SIP 响应消息，指明 SIP 控制请求消息实际的归宿。
- **SIP 注册服务器 (Registrar):** 作用是接受来自用户的 SIP 注册请求，将用户的位置信息登记进名为“位置服务”（location service）的数据库（或其他数据库）。

图 12.14 所示为以上几种 SIP 服务器是如何密切合作的。

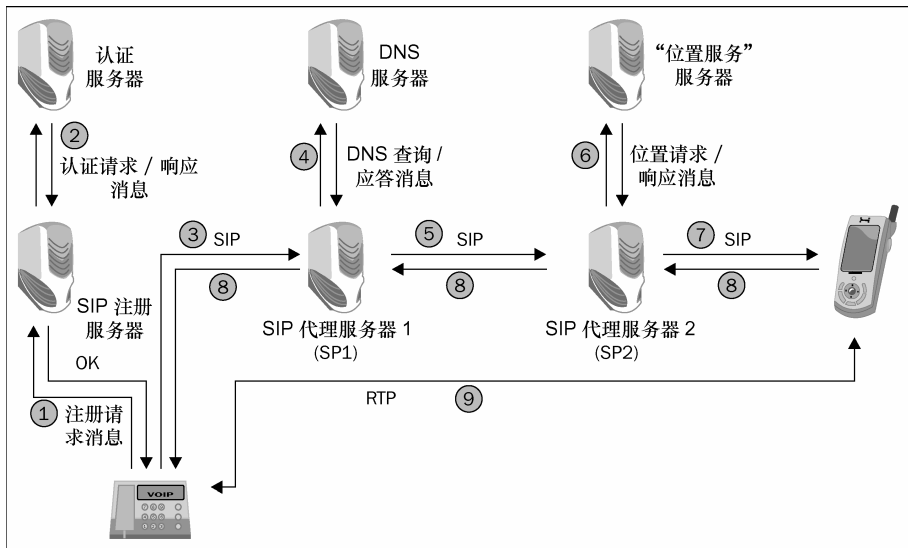


图 12.14

由图 12.14 可知，一部 IP 电话（主叫方）正在向 SIP 注册服务器注册（①）。SIP 注册服务器向认证服务器提交认证请求（②），若通过认证，就会向服务提供商的 SIP 代理服务器发送 SIP 控制请求消息（③）。服务提供商的 SIP 代理服务器与 DNS 服务器进行沟通（④），以解析出被叫方所在域的 IP 地址。若解析成功，则会将 SIP 控制请求消息转发至被叫方所在地 SIP 代理服务器（⑤）。此 SIP 代理服务器无法定位被叫方，于是会找“位置服务”服务器索要被叫方的位置（⑥）。若能获悉被叫方的位置，此 SIP 代理服务器就能将 SIP 控制请求消息转发至被叫方（⑦）。被叫方回复 SIP 响应消息，接受主叫方的呼叫请求（⑧）。若一切 OK，主叫方和被叫方之间将会建立起 RTP 会话（⑨），所使用的 UDP 端口号则取自两者之前发出的 SIP 控制请求消息/响应消息中 SDP 媒体描述部分所包含的媒体端口号字段值。

SIP 控制请求消息的格式如图 12.15 所示。

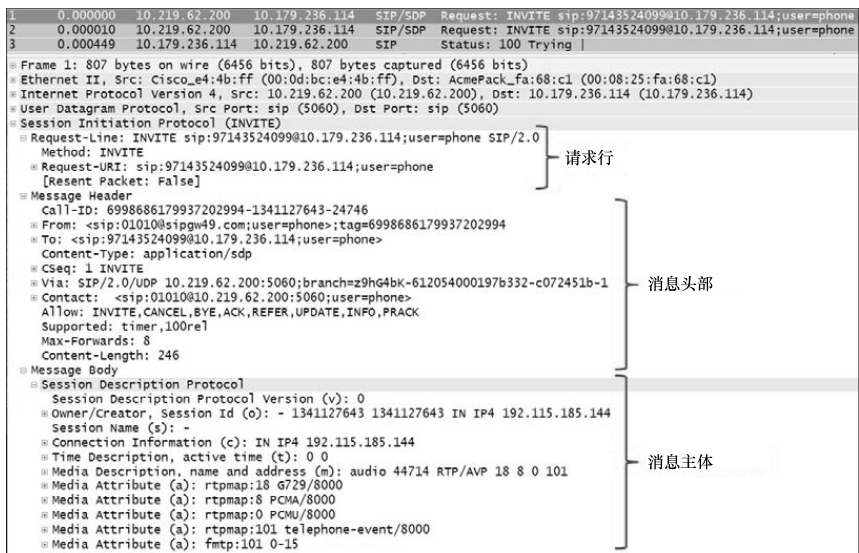


图12.15

由图 12.15 可知，SIP 控制请求消息的第一大块叫做请求行（Request line），由以下两部分组成。

- 请求方法：本例为 **INVITE**。
- 有待建立呼叫请求的被叫方的 URI（The requested URI）。

SIP 控制请求消息的第二大块名为消息头部（Message Header），包括下列字段。

- **Call-ID**：其值为本次呼叫的唯一标识符。
- **From**：用来标识本次呼叫的主叫方。
- **To**：用来标识本次呼叫的被叫方。
- **Cseq**：其内容为一个序列号值（无符号整数）外加本消息所包含的请求方法。该序列号值根据各自的方向（主叫方和被叫方的发出控制请求消息的方向）严格单向按 1 递增，用来区分某条 SIP 控制请求消息是新的消息还是重发的消息。
- **Via**：指出了本 SIP 控制请求消息在网络中穿行的路线（由哪个 UA 发出，穿过了哪台 SIP 代理服务器），与其相对应的 SIP 响应消息也应走相同的路线。
- **Contact**：包含了一或多个 SIP URI，用来向会话接收方提供会话发起方的联络信息。
- **Allow**：用来指明所允许的请求方法。

- **Supported:** 用来指明是否支持诸如计时器之类的参数。
- **Max-Forwards:** 指明了本 SIP 控制请求消息所能途经的 SIP 实体的上限。
- **Content-Length:** 指明了消息主体的字节数。

SIP 控制请求消息的第三大块名为消息主体 (**Message Body**)，包含了主叫方所支持的媒体编码方式等信息。

12.3.4 拾遗补缺

排查 SIP 协议故障时，首先需要根据 SIP 消息头部中的 Call-ID 字段值来筛选出隶属于某次呼叫的所有 SIP 流量。为此，可按以下两种方法之一行事。

- 在 Wireshark 抓包主窗口的数据包列表区域先选择一个隶属于某个呼叫的 SIP 数据包，再到数据包结构区域中点击 **Message Header** 之前的+号，找到 Message Header 字段，点击右键，选择右键菜单中的 **Apply as Filter** 菜单项，如图 12.16 所示。

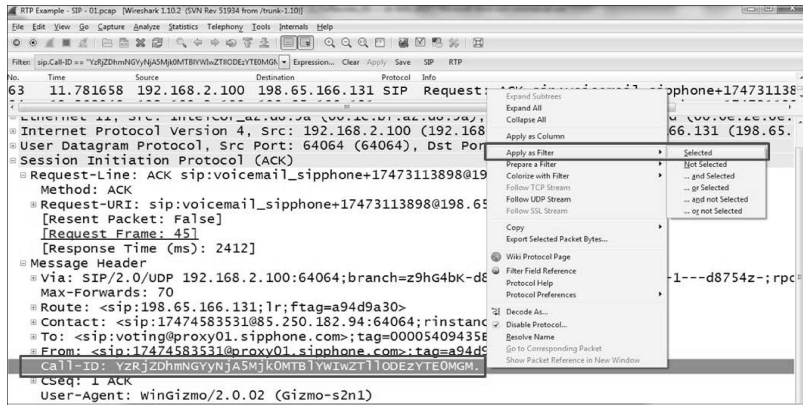


图12.16

- 借助于之前介绍过的 **Telephony** 菜单中的 **VoIP Calls** 菜单项功能。

当排除 SIP Trunk（其作用为在 IP PBX 之间传递信令）故障时，应首先判断是个别用户之间通话有问题，还是所有用户的通话都有问题。排除 VoIP 故障的最佳方法就是仔细观察抓包文件中的 SIP 协议数据包。

12.4 RTP/RTCP故障分析

上一节介绍了在 IP 上传输语音和视频时所使用的信令协议 SIP 和 SDP。本节将介绍用来传递语音和视频数据自身的 RTP 协议，以及排障之法。

对 RTP/RTCP 故障的排查，必然起始于最终用户对语音或视频通话质量的投诉。

此外，请别忘了，要是通过 Wireshark 抓包分析也看不出个所以然来的话，那就只能说需要对 IP PBX 做进一步优化，来增加语音传输容量了。

12.4.1 准备工作

当某个用户反映语音或视频通话质量不佳时，请借助于端口镜像功能，将交换机上连接该用户 UA 的端口的流量重定向至 Wireshark 主机；在处理接入相同链路的多个用户的类似问题时，亦需利用端口镜像功能设法将流淌于该链路的流量重定向至 Wireshark 主机^①。

12.4.2 操作方法

要定位跟语音或视频通话质量有关的问题，请按以下步骤行事。

- 1. 在交换机上开启端口镜像功能，把 Wireshark 主机连接到正确的端口之后，开始抓包。
- 2. 确保有语音或视频通话正在进行。
- 3. 在 Wireshark 抓包主窗口中点击 **Telephony** 菜单，选择 RTP 菜单项中的 **Show All Streams** 子菜单项，RTP Streams 窗口会立刻弹出，如图 12.17 所示。由图可知，该窗口会显示出 Wireshark 从受监控交换机端口抓取到的所有 RTP 流。
- 4. 在图 12.17 所示的 RTP Streams 窗口中，需要重点关注 Lost、Max Delta (ms)、Max Jitter (ms)和 Mean Jitter (ms)这四列的内容。

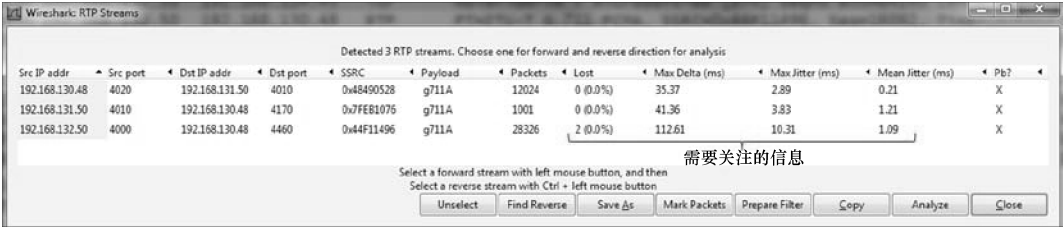


图12.17



对于基于 IP 的交互式语音和视频通话而言，若网络延迟（RTT）和抖动值分别高过 300 和 50 毫秒，便预示着有可能存在通话质量问题，12.4.3 节将对此展开深入探讨。

^① 译者注：原文是 “When facing problems on a specific client, connect Wireshark to the client port with a port mirror. When facing problems with all clients connected to the same link, connect Wireshark to the link with a port mirror”。译文酌改。