**暨南大学本科实验报告专用纸**

课程名称 计算机网络实验 成绩评定

实验项目名称 Internet应用与应用层协议分析 指导教师 潘冰

实验项目编号 3 实验项目类型 验证 实验地点 计算机网络实验室

学生姓名 袁霖 学号 2019051099

学院 智能科学与工程 系 专业 信息安全

实验时间 2021 年 9 月 28 日 上 午～ 10 月 19 日 下 午

1. **实验目的**
2. 理解WWW 、 DNS服务、FTP服务、SMTP的作用和原理；
3. **学会使用wireshark分析HTTP、FTP、SMTP和DNS协议的工作过程，加深对协议格式和工作原理的理解。**
4. **实验内容和要求**
5. 通过域名访问WWW、FTP服务器，分析DNS、WWW、FTP工作过程，并使用WireShark分析相关协议格式；
6. 在客户端访问SMTP服务器，使用wireshark分析SMTP、POP3协议的工作过程。（可以在客户端安装outlook或使用QQ邮件服务器或自己编程）
7. **主要仪器设备**

局部网环境，计算机若干台。本实验不分组，独立完成。

1. **实验步骤与原理**
2. **HTTP协议分析**

本次实验访问站点ip地址：111.200.241.244:53068

首先，我们清空浏览器的高速缓存及客户端的DNS高速缓存。该网页的界面如下图所示，界面较为简洁，便于我们做实验分析：

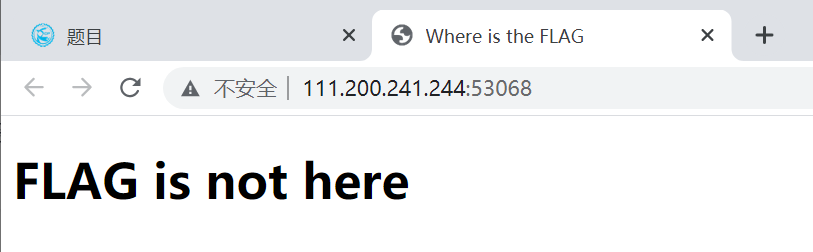


图 1

1. **利用Wireshark分析HTTP协议的工作过程**

由于本次实验是利用IP地址访问网站，故不需要利用DNS协议来解析地址。则本次实验的HTTP协议工作过程如下：

* 1. 封装HTTP请求

将协议类型、IP地址、端口号、请求文件名、请求参数等信息封装成一个HTTP请求数据包。

* 1. 封装TCP包

本步骤需要封装TCP包，并建立TCP连接，即“TCP三次握手”。由于HTTP协议位于最上层的应用层，故HTTP协议在工作前需要先由TCP和IP协议建立连接。

* 1. 客户端发送请求命令

在建立连接后，客户端会将HTTP请求包发送给服务器端，其中包括请求行、请求头部、请求正文。图2即为本机向服务器端发送的HTTP请求包：

请求包大小

请求方法

协议版本

URL

协议名称

目的IP地址

请求方IP地址

捕获到请求包的时间

请求包对应的帧数



图 2

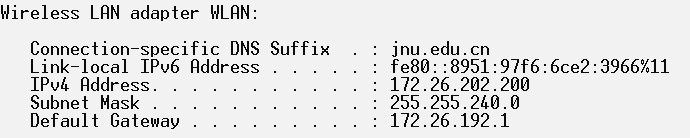


图 3

* 1. 服务器端响应

服务器在收到请求后，会根据客户端的请求发送相应的信息给客户端。相关信息都会放在状态行、响应头部及响应正文中。图4即为服务器端向本机发送的HTTP响应包：

响应包帧数 捕获响应包时间 服务器IP 客户端IP 协议名 响应包大小 协议版本 状态码 状态码描述 返回文本类型



图 4

* 1. 关闭连接

服务器端在发送完响应之后，就会关闭连接。但如果客户端的请求头部中有keep-alive（如图5所示），则服务器端在响应完该请求后不会关闭连接，直到客户端所有请求都响应完毕（超过最大响应时间，或最大请求次数也会关闭连接）。这会节省带宽和I/O资源。



图 5

1. **利用Wireshark分析HTTP协议的格式**

**HTTP报文格式简介**



图 6 HTTP请求报文格式

图2中的第1行为“请求行”，中间部分为“请求头部”，最底下为“请求正文”，在“请求头部”与“请求正文”之间还有一行“空行”。下面分别介绍以上这四部分的含义：

1. 请求行：它由三部分组成，分别为：请求方法、URL、协议版本，中间用空格隔开，并以回车符及换行符结束。

其中，请求方法主要有以下8种：

|  |  |  |
| --- | --- | --- |
| 序号 | 方法 | 描述 |
| 1 | GET | 请求指定的页面信息，并返回实体主体 |
| 2 | HEAD | 类似于GET请求，但是返回的响应中没有具体内容，用于获取报头 |
| 3 | POST | 向指定资源提交数据进行处理请求（提交表单、上传文件），数据被包含在请求体中。POST请求可能会导致新的资源的建立/已有资源的修改 |
| 4 | PUT | 从客户端向服务器传送的数据取代指定的文档的内容 |
| 5 | DELETE | 请求服务器删除指定的页面 |
| 6 | CONNECT | HTTP/1.1协议中预留给能够将连接改为管道方式的代理服务器 |
| 7 | OPTIONS | 允许客户端查看服务器的性能 |
| 8 | TRACE | 回显服务器收到的请求，主要用于测试或判断 |

协议版本主要有以下4种，格式为HTTP/主版本号.次版本号：

|  |  |  |
| --- | --- | --- |
| 序号 | 版本 | 描述 |
| 1 | HTTP/0.9 | 原型版本，功能简陋，只有一个命令GET,服务器只能回应HTML格式字符串，该版本已过时。 |
| 2 | HTTP/1.0 | 任何格式的内容都可以发送。  除了GET命令，还引入了POST命令和HEAD命令。  每次通信都必须包括头信息。  每个TCP连接只能发送一个请求，发送数据完毕，连接就关闭，如果还要请求其他资源，就必须再新建一个连接。 |
| 3 | HTTP/1.1 | 引入了持久连接（ persistent connection），即TCP连接默认不关闭，可以被多个请求复用，不用声明Connection: keep-alive。  引入了管道机制，即在同一个TCP连接里，客户端可以同时发送多个请求改进了HTTP协议的效率。  新增了4种方法：PUT、 PATCH、 OPTIONS、 DELETE。  HTTP协议不带有状态，每次请求都必须附上所有信息。 |
| 4 | HTTP/2 | 头信息和数据体都是二进制。  复用TCP连接，在一个连接里，客户端和浏览器都可以同时发送多个请求或回应，且不用按顺序一一对应。  HTTP/2 允许服务器未经请求，主动向客户端发送资源，即服务器推送。 |

1. 请求头部：它包含了很多客户端环境以及请求正文的有用信息，由“关键字：值”构成，各主要头部类型及所包含的信息会在下文进行描述。
2. 空行：表示请求头部的结束，下面是请求正文。
3. 请求正文：可选，GET请求没有请求正文，POST请求可以以提交表单数据方式为请求正文。

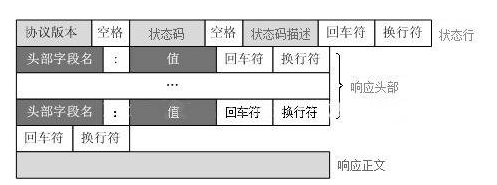


图 7 HTTP响应报文格式

图3中的第1行为“状态行”，中间部分为“响应头部”，最底下为“响应正文”，在“响应头部”与“响应正文”之间还有一行“空行”。下面分别介绍以上这四部分的含义：

1. 状态行：它由三部分组成，分别为：协议版本、状态码、状态码描述，中间用空格隔开，并以回车符及换行符结束。

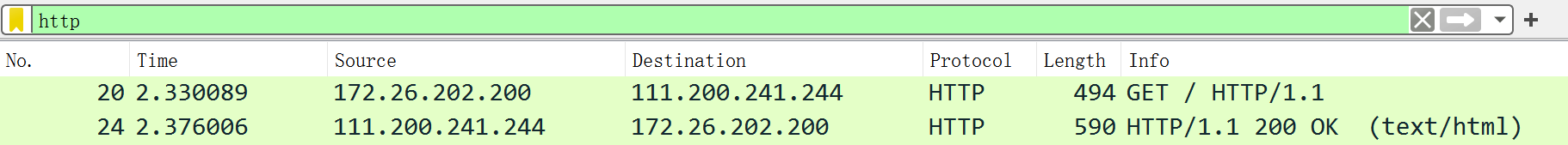
其中，协议版本上文已做过介绍，不再赘述。状态码主要有5类：

|  |  |  |
| --- | --- | --- |
| 序号 | 分类 | 描述 |
| 1 | 100~199 | 信息，服务器收到请求，需要请求者继续执行操作 |
| 2 | 200~299 | 成功，操作被成功接收并处理 |
| 3 | 300~399 | 重定向，需要进一步的操作以完成请求 |
| 4 | 400~499 | 客户端错误，请求包含语法错误或无法完成请求 |
| 5 | 500~599 | 服务器错误，服务器在处理请求的过程中发生了错误 |

1. 响应头部：它包含了很多有用信息，由“关键字：值”构成，各主要头部类型及所包含的信息会在下文进行描述。
2. 空行：表示响应头部的结束，下面是响应正文。
3. 响应正文：由服务器返回的文档，常见的为HTML网页。

**分析捕获到的HTTP数据包**

**请求方法 URL HTTP协议版本**



**响应行信息**

**请求行信息**

图 8

**HTTP协议版本 状态码 状态码描述 返回的文本类型**

由Wireshark捕获到的HTTP协议及本机IP地址如上图所示。

通过图4可知，本机通过Get方法向目的IP地址111.200.241.244发送了一个长度为494字节的HTTP请求包。红框中的GET表示HTTP的请求方法，/表示URL，HTTP/1.1表示HTTP协议版本。

同时，目的IP地址向本机返回了一个长度为590字节的HTTP响应包。蓝框中的HTTP/1.1表示HTTP协议版本，200表示状态码，OK表示状态码描述，(text/html)表示返回的文本类型。

**分析HTTP请求包**

双击HTTP请求包，具体内容如下图所示（仅HTTP协议）：



**应答是第24帧**

**HTTP请求**

**请求的编码格式**

**请求语言**

**（GET方法没有请求正文）**

**请求的URI**

**请求的类型**

**请求用户信息**

**请求方法**

**URI**

**表示可以把所属本站的所有http连接升级为https连接**

**表示是否需要持久连接，HTTP/1.1默认进行持久连接**

**请求的服务器的域名和端口号**

**空行**

**协议版本**

**请求头部**

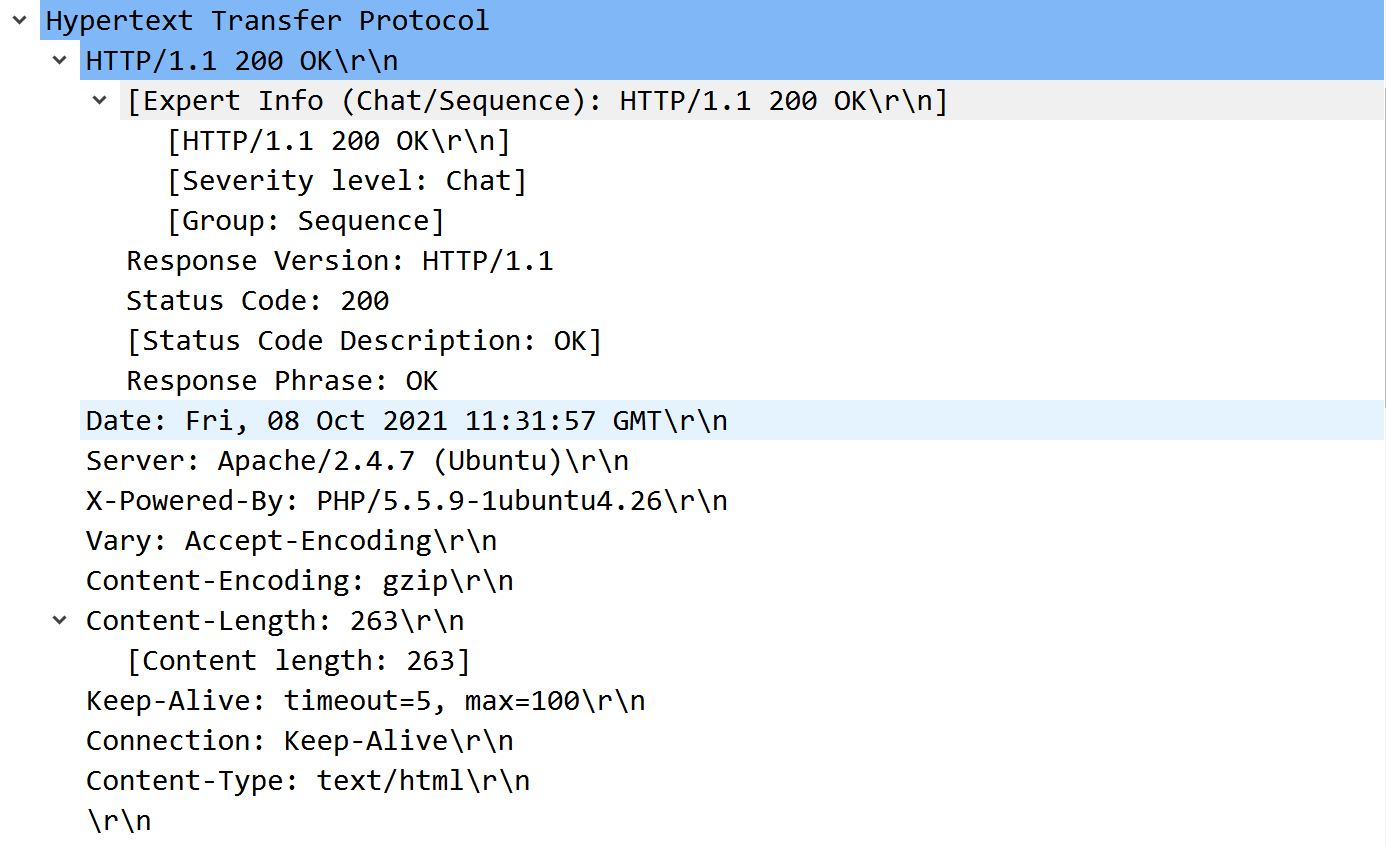
**请求行**

图 9

对照图3所示的HTTP请求报文格式，该请求包各部分的含义注释如上图。

**分析HTTP响应包**

根据请求包的信息，可知对应的响应包是第24帧，如下图所示：



**告知下游代理是使用缓存响应还是从原始服务器请求**

**服务器消息发出时间**

**响应版本**

**状态码：响应成功**

**空行**

**相应内容类型**

**使用持久连接**

**Keep-Alive的超时时间，最大请求次数**

**响应体长度**

**返回内容压缩编码类型**

**网站编写语言**

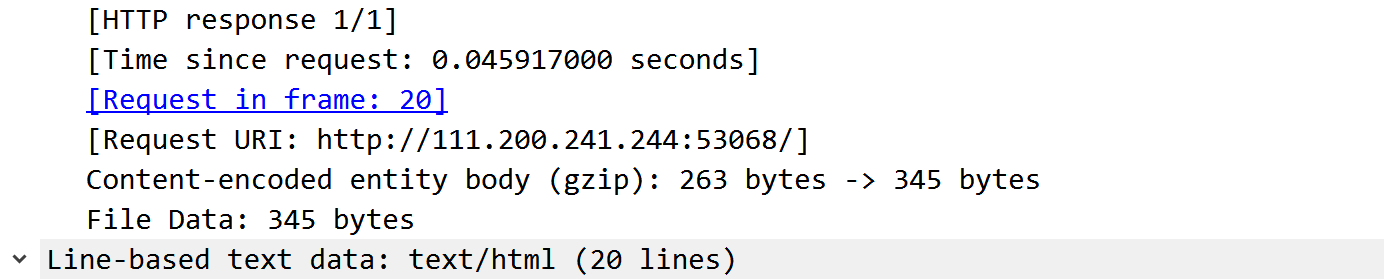
**服务器名称**

**状态描述**

**响应头部**

**响应行**

图 10



**HTTP响应**

**内容大小**

**内容编码**

**请求的URI**

**请求的帧号为20**

**响应使用的时间**

图 11

1. **DNS协议分析**
2. **利用Wireshark分析DNS的工作过程**

本次实验访问的网站是百度，捕获到的DNS数据包如下图所示：

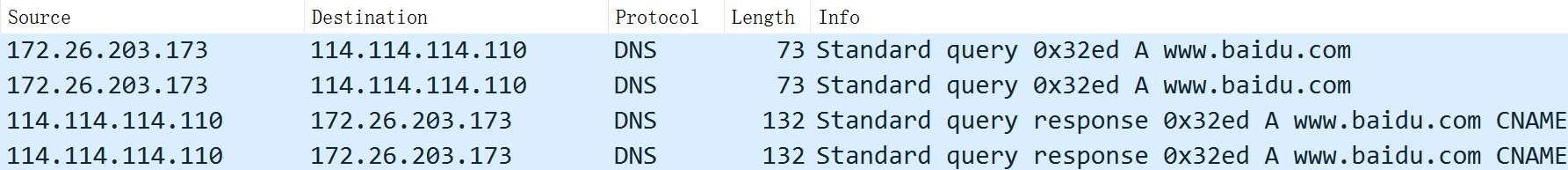


图 12

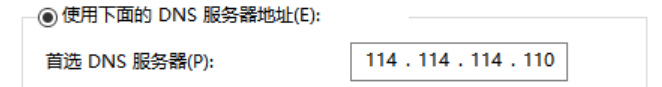


图 13

可以看到，本次实验用到的本地域名服务器为：114.114.114.110。前两个数据包为本机向本地域名服务器发送的两次DNS查询请求，而后两个数据包为本地域名服务器向本机发送的两次DNS应答。

分别观察两个请求包和应答包的区别，发现第二个请求包与第二个应答包都有两行重发的信息，如下图所示。这表示两个请求包与两个应答包的内容都是相同的，因此只需要分析一个即可。

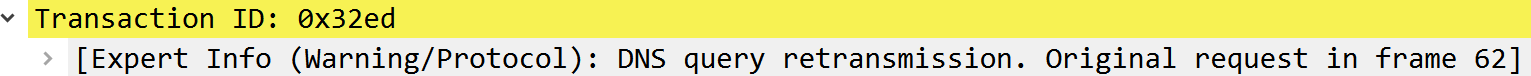


图 14 重发请求信息



图 15 原始请求在第62帧

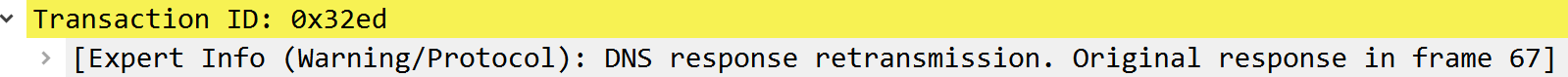


图 16 重发应答信息



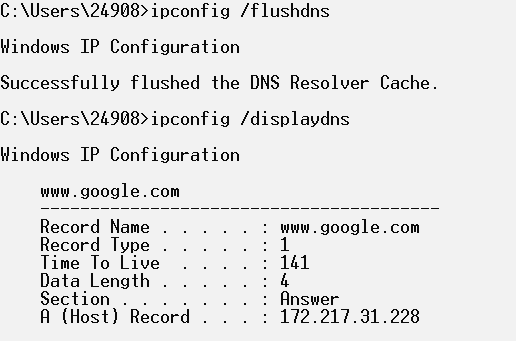
图 17原始应答在第67帧

**DNS工作过程分析**

1. 在浏览器中输入域名www.baidu.com，操作系统会先检查自己本地的hosts文件是否有这个网址映射关系，如果有，就先调用这个IP地址映射，完成域名解析。
2. 由于本机hosts文件中没有www.baidu.com的IP地址映射，因此需要查找本地DNS解析器缓存，看是否有这个网址在生存时间内的有效映射关系，如果有，直接返回，完成域名解析。

为了防止本地DNS解析器缓存中有www.baidu.com的有效IP地址映射，从而无法捕获DNS数据包，故本实验利用命令“ipconfig /flushdns”来刷新DNS缓存，再利用命令“ipconfig /displaydns”来确认是否刷新DNS缓存。

在未刷新DNS缓存前，缓存文件中可能可以看到www.baidu.com的IP地址映射，但是该映射可能已经过了生存时间，因此它是一条无效映射，本机仍然需要发送一个DNS查询请求报文给本地域名服务器。



**在生存时间内的域名到IP地址的映射才有效**

图 18

查看得知DNS解析器缓存中没有www.baidu.com的IP地址映射，故可以利用Wireshark捕获到DNS数据包。

1. 如果hosts与本地DNS解析器缓存都没有相应的网址映射关系，本机会发送一个DNS查询请求报文给本地域名服务器，如下图所示：



图 19

如果要查询的域名包含在本地域名服务器中，则会返回解析结果给客户机，完成域名解析，此解析具有权威性。

1. 如果要查询的域名在本地域名服务器中已缓存了映射关系，从而不需要利用本地域名服务器向根域名服务器发出查询请求，则本地域名服务器会调用这个IP地址映射，完成域名解析，此解析不具有权威性，即此解析没有经过权限服务器的解析。
2. 如果本地域名服务器本地区域文件与缓存解析都没有对应的IP 地址映射，则会向13台根域名服务器中的一台发出查询请求。（以迭代查询为例）
3. 在根域名服务器收到请求后，会返回com这个顶级域名的NS记录
4. 本地域名服务器会向NS记录中的其中一台主机发出查询请求，则com域的服务器会发现该请求是baidu.com这个域的，随后该权限域名服务器会返回该域的NS记录
5. 随后，本地域名服务器会向NS记录中的baidu.com域的一台权限域名服务器发出查询请求，该权限域名服务器收到请求之后，会查找有www的主机，并将其IP地址返回给本地域名服务器，下图即为拥有对应IP地址映射的权限域名服务器：

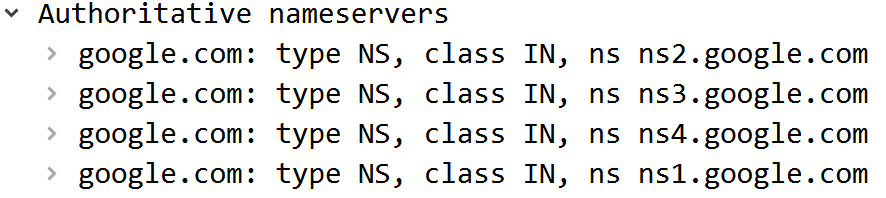


图 20

1. 最后，域名对应的IP地址会返回给本地域名服务器，本地域名服务器再将结果返回给本机，本机即可捕获到DNS查询应答报文，如下图所示：



图 21

1. **分析捕获不到DNS解析过程的原因**

若本地hosts文件或本地DNS解析器缓存中有域名相对应的有效IP地址映射（在生存时间内），则Wireshark捕获不到DNS数据包。现作如下尝试：

本机的hosts文件如下图所示：

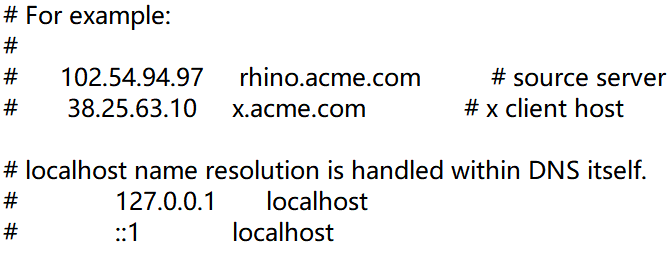


图 22

刷新后的DNS缓存如下图所示：

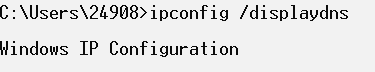
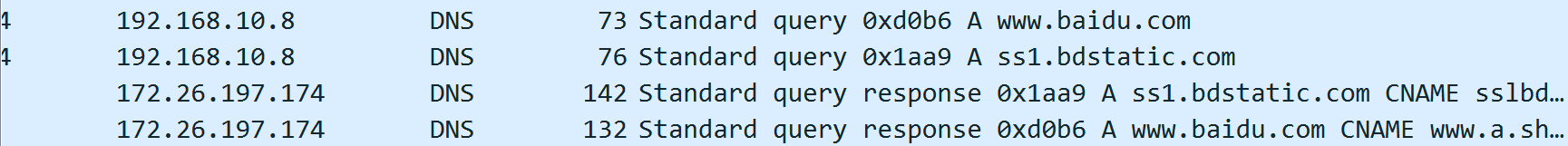


图 23

在DNS缓存中没有数据时，先访问一次www.baidu.com，发现可以捕获到DNS数据包：

**请求包**



**应答包**

图 24

再查看DNS缓存，出现www.baidu.com对应的IP地址映射：



图 25

再次访问www.baidu.com，发现捕获不到www.baidu.com的数据包。说明该映射记录还在有效时间内，本地域名服务器直接返回对应的IP地址给本地，且此应答不具有权威性。

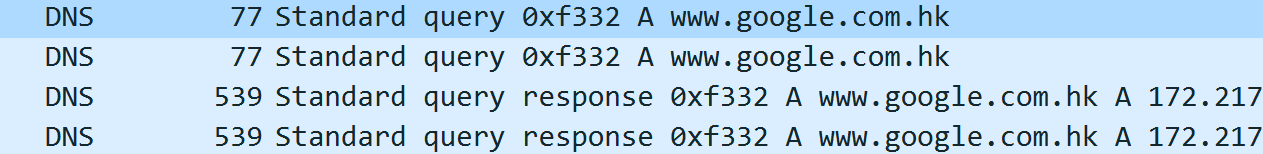
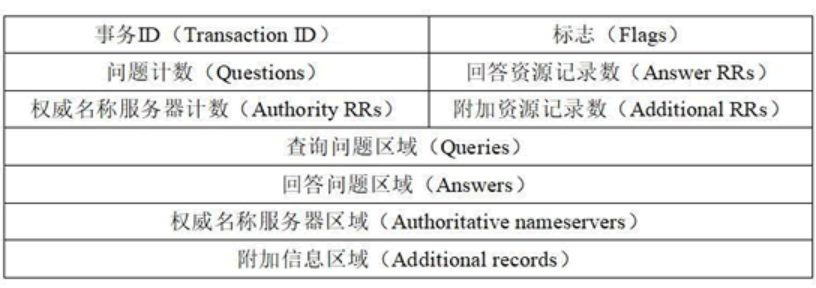


图 26

1. **利用Wireshark分析DNS的格式**

**DNS报文格式简介**

DNS分为查询请求和查询响应两种报文，它们的结构基本相似，结构如下图所示：



**资源记录部分**

**问题部分**

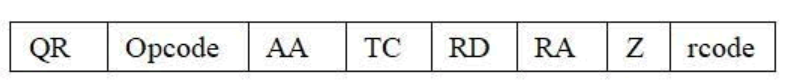
**基础结构部分**

**（报文首部）**

图 27 DNS报文格式

接下来对各部分作简要介绍：

1. 基础结构部分
   1. 事务ID：DNS 报文的 ID 标识。对于请求报文和其对应的应答报文，该字段的值是相同的。通过它可以区分 DNS 应答报文是对哪个请求进行响应的。
   2. 标志：DNS 报文中的标志字段。它又分为若干个字段，如下图所示：



标志字段中每个字段的含义如下：

* + 1. QR(Query/Response)：指明报文是DNS查询还是响应，占1个比特位。为1代表响应，0代表查询。
    2. OpCode：操作码。0 表示标准查询；1 表示反向查询；2 表示服务器状态请求。
    3. AA(Authoritative Answer)：授权应答，该字段在响应报文中有效。值为 1 时，表示名称服务器是权限域名服务器；值为 0 时，表示不是权限域名服务器。
    4. TC（Truncation）：表示是否被截断。值为 1 时，表示响应已超过 512 字节并已被截断，只返回前 512 个字节。
    5. RD（Recursion Desired）：递归查询标志。1表示执行递归查询，0表示执行迭代查询。
    6. RA（Recursion Available）：可用递归。该字段只出现在响应报文中。 1 表示服务器支持递归查询，0表示服务器不支持递归查询。
    7. Z：保留字段，在所有的请求和应答报文中，它的值必须为 0。
    8. rcode（Reply code）：返回码字段，表示响应的差错状态。各差错状态如下图所示：

|  |  |
| --- | --- |
| 返回码字段 | 响应差错状态 |
| 0 | 没有错误 |
| 1 | 报文格式错误，服务器不能理解请求的报文 |
| 2 | 域名服务器失败，因为服务器的原因导致没办法处理这个请求 |
| 3 | 名字错误，只对权限域名服务器有意义，指解析的域名不存在 |
| 4 | 查询类型不支持，即域名服务器不支持查询类型 |
| 5 | 拒绝，服务器由于设置的策略拒绝给出应答 |

* 1. 问题计数：DNS 查询请求的数目。
  2. 回答资源记录数：DNS 响应的数目。
  3. 权威名称服务器计数：权威名称服务器的数目。
  4. 附加资源记录数：额外的记录数目（权威名称服务器对应 IP 地址的数目）。

1. 问题部分

这部分指的是报文格式中的“查询问题区域（Queries）”部分。它用来表示DNS查询请求的问题，通常只有一个。它包含了查询名（被查询主机名字）、查询类型、查询类三个部分。

1. 查询名：要查询的域名，或IP地址，用于反向查询。
2. 查询类型：DNS查询请求的资源类型，主要有以下几种：
   1. 类型A：值是1，表示获取目标主机的IP地址。
   2. 类型CNAME：值是5，表示获得目标主机的别名。
   3. 类型PTR：值是12，表示反向查询。
3. 查询类：地址类型，通常为互联网地址，值为1。
4. 资源记录部分

这部分指的是DNS报文格式中的最后三个字段，包括回答问题区域、权限域名服务器、附加信息三个部分。它采用资源记录的格式，如下图所示：

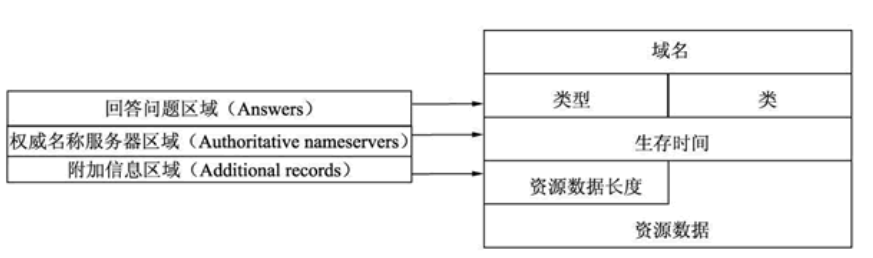


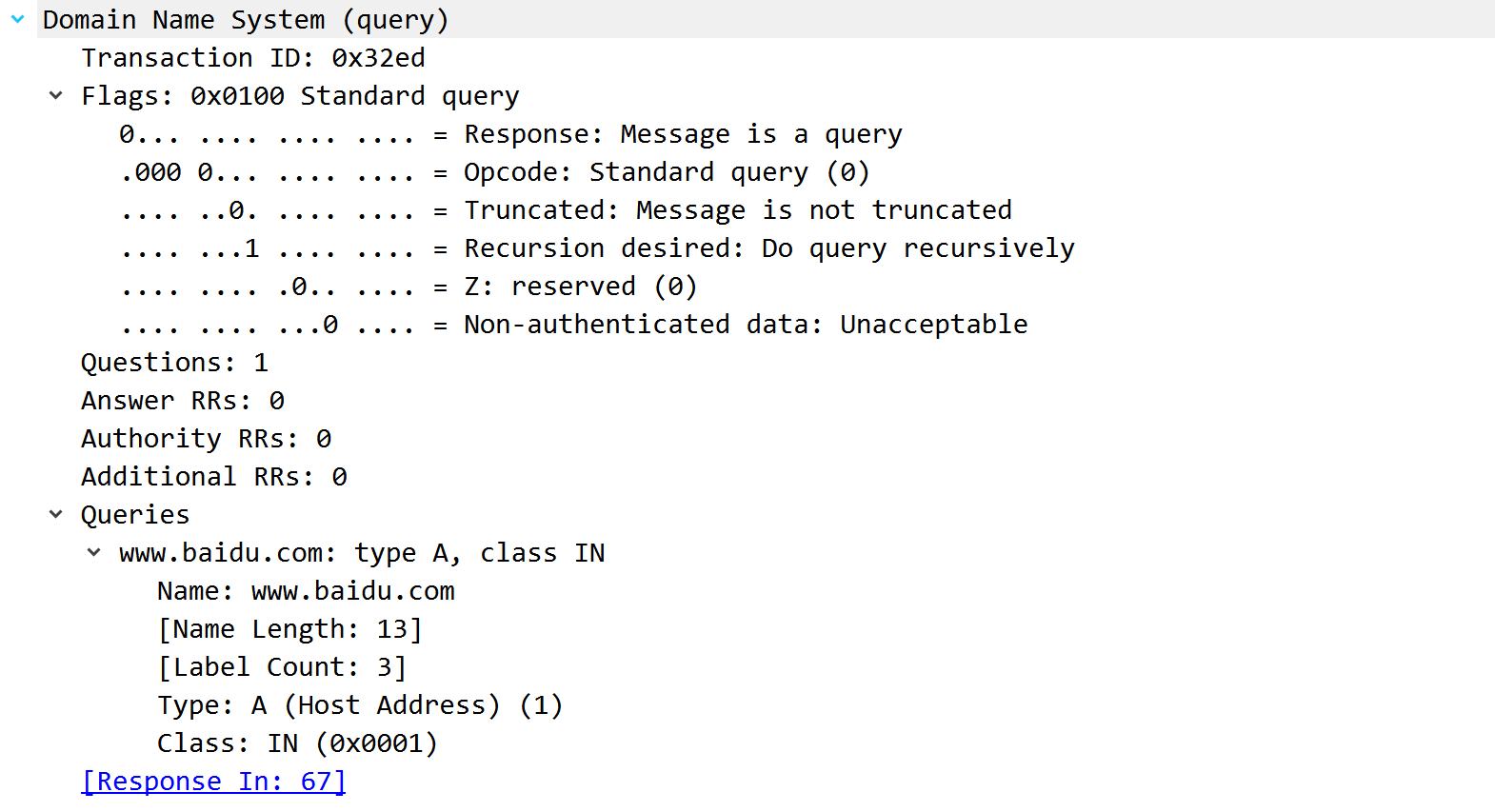
图 28

其中，资源记录格式各字段的含义如下：

* 1. 域名：DNS 请求的域名。
  2. 资源数据类型，类型：与问题部分中的查询类型值是一样的。
  3. 资源数据类别，类：地址类型，与问题部分中的查询类值是一样的。
  4. 生存时间：资源记录的生命周期，一般用于当地址解析程序取出资源记录后决定保存及使用缓存数据的时间。
  5. 资源数据长度：资源数据的长度。
  6. 资源数据：表示按查询段要求返回的相关资源记录的数据。

**分析DNS请求包**

双击DNS请求包，具体内容如下图所示：



**对应响应包的帧数**

**查询类，这里为互联网地址IN**

**查询类型，获取目标主机的IP地址**

**查询名，这里为请求域名**

**附加资源记录数**

**授权资源记录数**

**回答资源计数，没有DNS响应**

**问题计数，有1个查询请求**

**保留位，必须为0**

**递归查询**

**没有被截断**

**操作码，标准查询**

**Q/R，查询请求**

**事务ID**

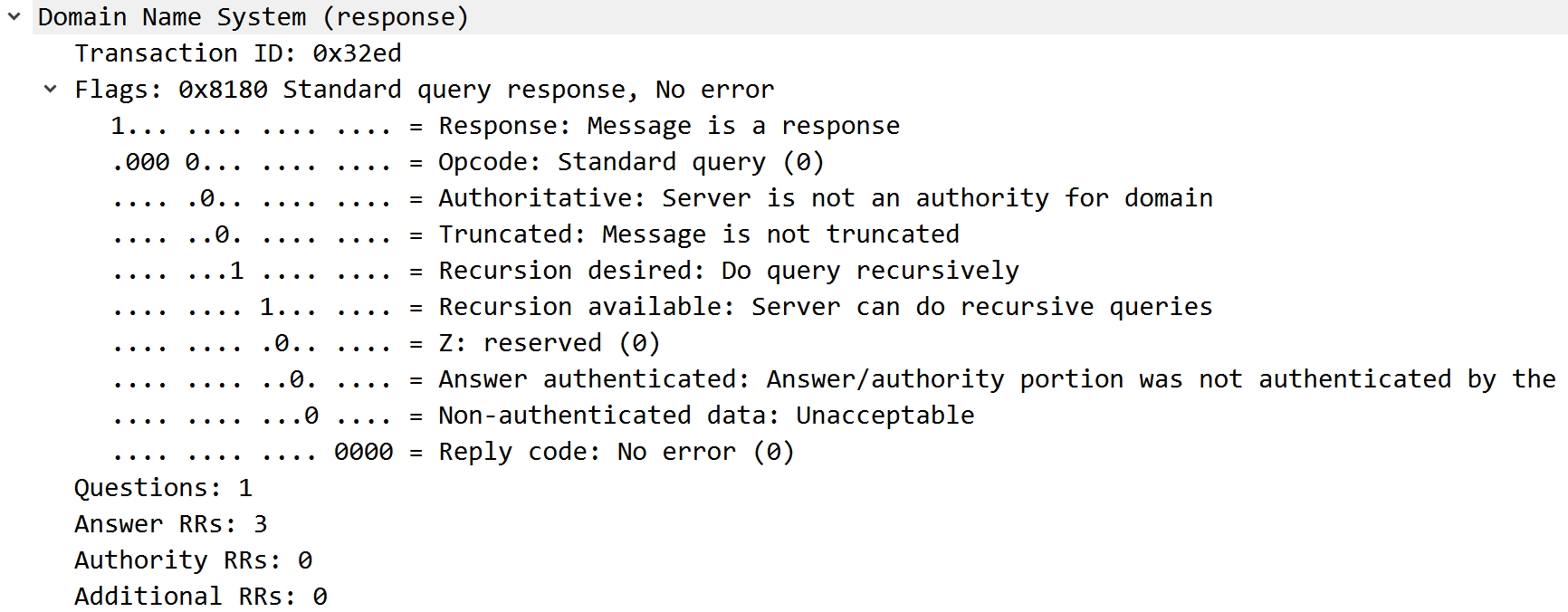
**标志**

**问题部分**

**基础结构部分**

**分析DNS应答包**

双击DNS应答包，具体内容如下图所示：



**附加资源记录数**

**授权资源记录数**

**回答资源记录数，有3个DNS响应**

**问题计数，有1个查询请求**

**返回码，没有错误**

**保留位**

**可以支持递归查询**

**递归查询**

**没有被截断**

**授权应答，不是权限域名服务器**

**操作码，标准查询**

**Q/R，查询响应**

**标志**

**事务ID**

**基础结构部分**



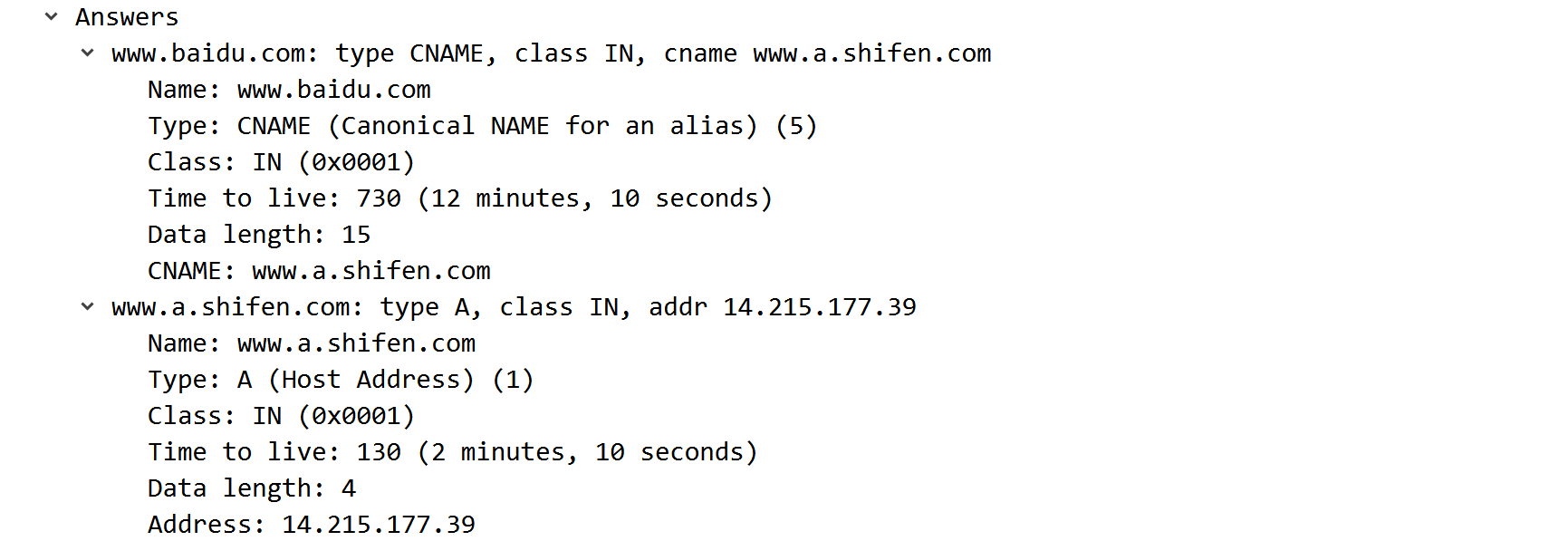
**问题回答区域**

**查询类，这里为互联网地址IN**

**查询类型，获取目标主机的IP地址**

**查询名，这里为请求域名**

**问题部分**



**回答3**

**回答2**

**资源数据**

**资源数据长度**

**生存时间**

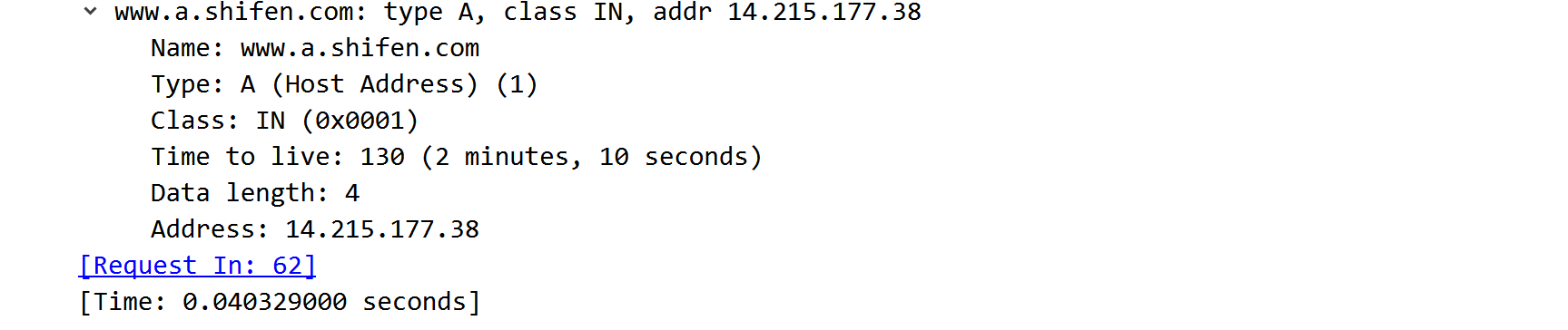
**类，与查询类相同**

**资源类型，与查询类型值相同**

**DNS请求域名**

**回答1**

**资源记录部分**



**应答对应的请求包在62帧**

由于捕获到的应答包中权限域名服务器计数（Authority RRs）与附加资源记录数（Additional RRs）均为0，即baidu.com即作为了顶级域名服务器，又作为了权限域名服务器，因此它本身就可以查找到www.baidu.com对应的IP地址映射，故本地域名服务器不会有访问权限域名服务器的记录。

为了让本实验捕获到权限域名服务器计数（Authority RRs）与附加资源记录数（Additional RRs）的信息，本次实验再次尝试捕获www.google.com的DNS数据包，如下图所示：

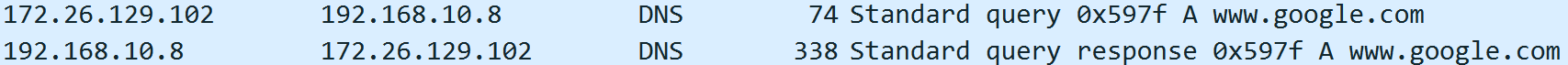
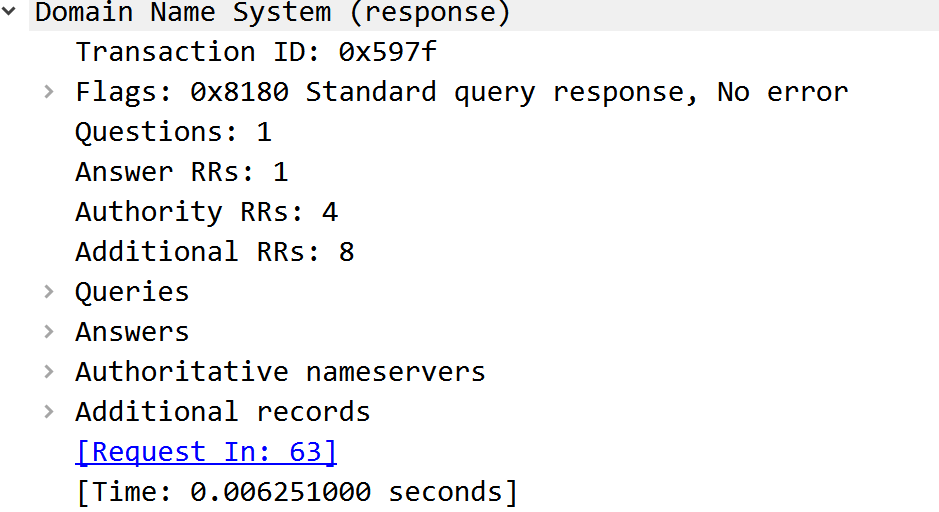


图 29

查看其中的DNS应答包：



**权限域名服务器区域，此处有4个权限域名服务器**

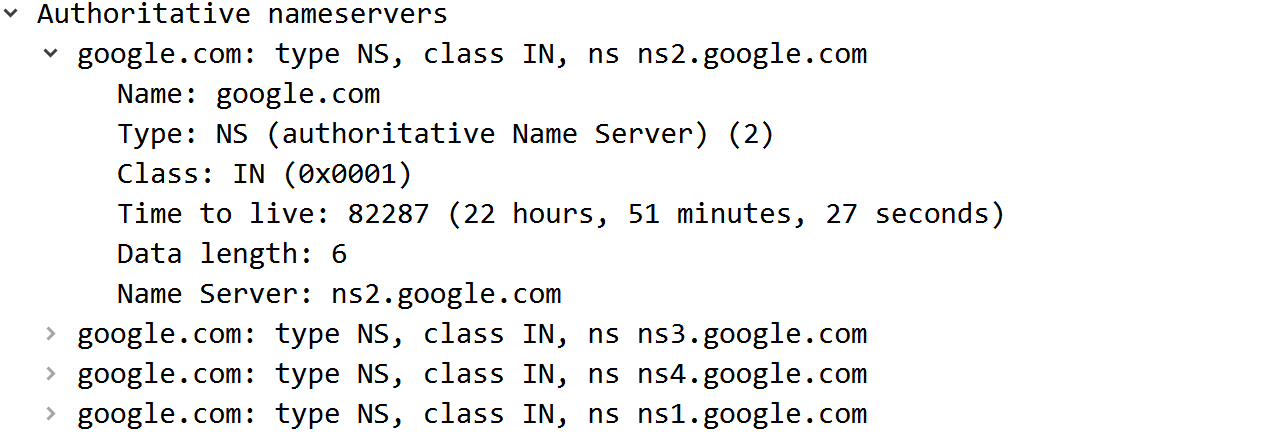
**基础结构部分**

**资源记录部分**

**问题部分**

**附加资源记录数**

**授权资源记录数**



**资源记录部分**

**资源数据，权限域名服务器**

**资源数据长度**

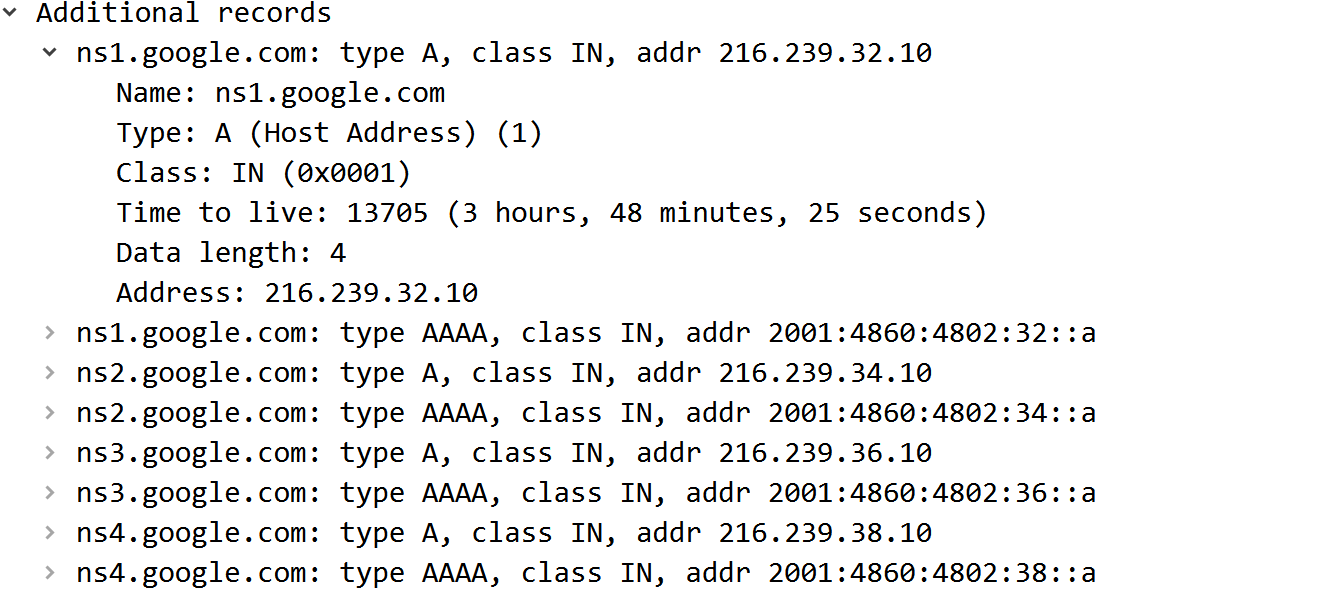
**生存时间**

**类**

**类型，代表权限域名服务器**

**域名，权限域名服务器名称**

**附加信息区域，说明了授权域名服务器的地址信息**



**资源数据长度**

**资源记录部分**

**资源数据**

**生存时间**

**类**

**类型，由域名获得IPv4地址**

**域名，权限域名服务器名称**

1. **FTP协议分析**
2. **利用Wireshark分析FTP的工作过程**

**1.1被动连接（PASV）**

首先，我们先通过浏览器来访问暨南大学的FTP服务器：

1. 通过TCP三次握手来建立TCP连接。

**建立控制连接**

1. 客户端通过控制进程（控制连接端口号：49241）向服务器进程（控制连接端口号：21号）发出连接请求，并请求以被动模式（PASV）工作
2. 服务器端同意利用被动模式工作，并返回IP地址，及数据连接的端口号，如下图所示。计算方式为：194\*256+137=49801

**数据连接端口号**

**IP地址**



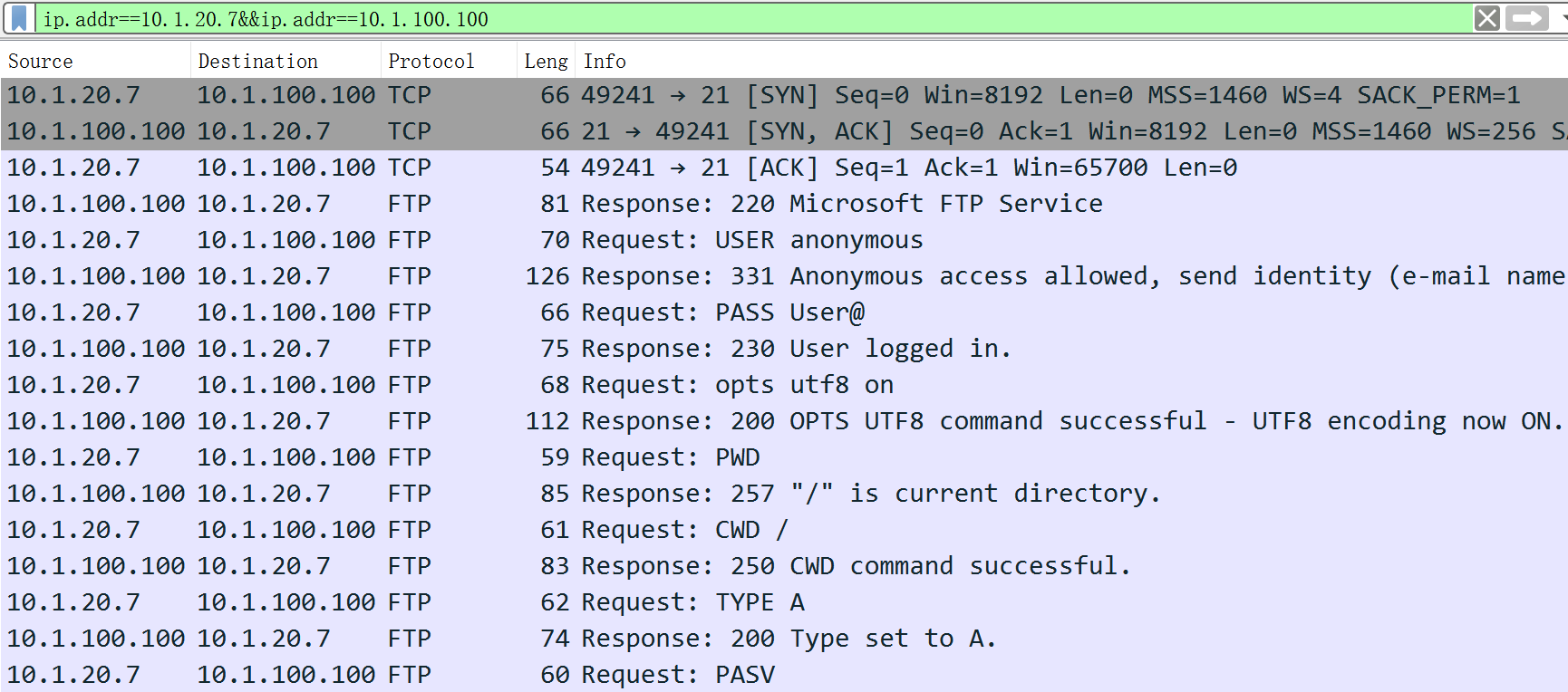
**建立数据连接**

1. 客户进程（数据连接端口号：49242）向服务器进程（数据连接端口号：49801）建立数据连接（TCP三次握手）
2. 服务器进程（数据连接端口号：49801）向客户进程（数据连接端口号：49242）发送数据



**关闭数据连接**

1. 在数据传输完毕后，服务器端提示断开数据连接
2. TCP四次挥手断开TCP连接，数据连接断开，但是控制连接没有断开



**服务器端响应被动模式**

**客户端请求被动模式**

**服务器端设定传输模式为ASCII**

**客户端设定传输模式为ASCII**

**服务器端返回请求操作完成的响应代码**

**客户端请求更改目录**

**服务器端返回当前路径**

**客户端请求获取当前路径**

**服务器端返回登陆成功信息**

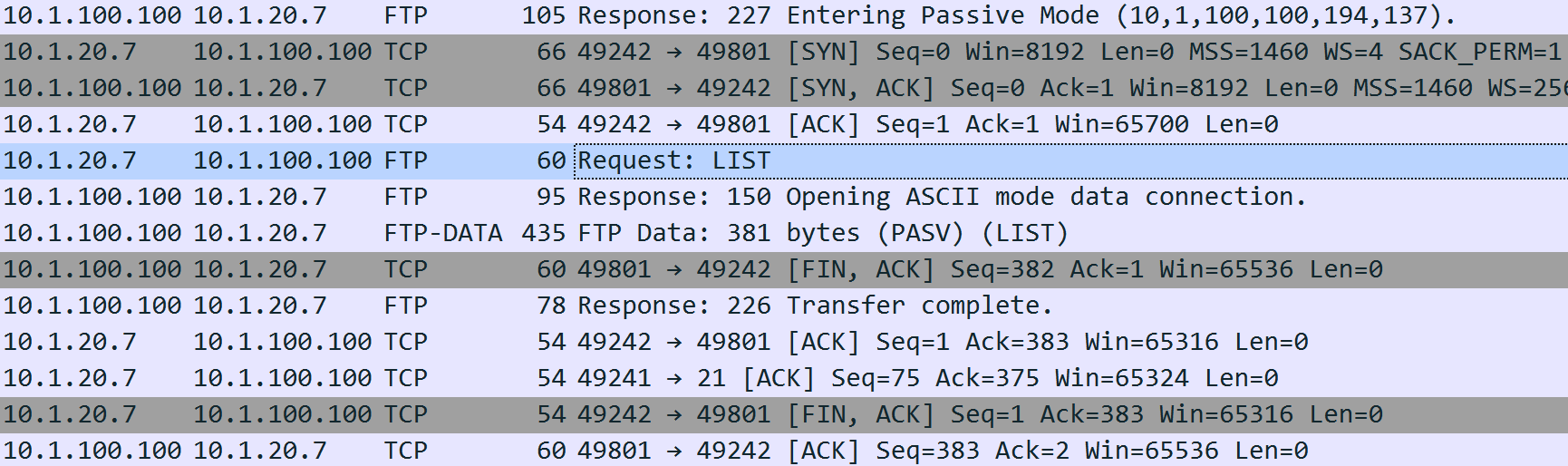
**服务器端允许匿名访问，并要求输入密码**

**客户端输入密码**

**客户端（49241端口）请求匿名访问**

**服务器端（21端口）响应FTP连接**

**TCP三次握手建立控制连接**



**传输的数据**

**客户端确认数据**

**服务器端传输数据完毕，提示断开数据连接**

**服务器端打开ASCII模式的数据连接**

**客户端请求指定的文件或目录**

**TCP三次握手建立数据连接**

**TCP四次挥手断开数据连接**

**1.2主动连接（PORT）**

其次，我们通过交互式的方式主动连接暨南大学的FTP服务器，如下所示：

1. 通过TCP三次握手来建立TCP连接

**建立控制连接**

1. 客户端通过控制进程（控制连接端口号：5233）向服务器进程（控制连接端口号：21号）发出以主动模式（PORT）工作的连接请求，并将客户端的IP地址，数据连接端口号发送给服务器端。数据连接端口计算方式为：20\*256+116=5236

**数据连接端口号**

**客户端IP地址**



**建立数据连接**

1. 服务器进程（数据连接端口号：20）通过TCP三次握手，发起与客户进程（数据连接端口号：5236）的数据连接
2. 服务器进程（数据连接端口号：20）向客户进程（数据连接端口号：5236）发送数据



**关闭数据连接**

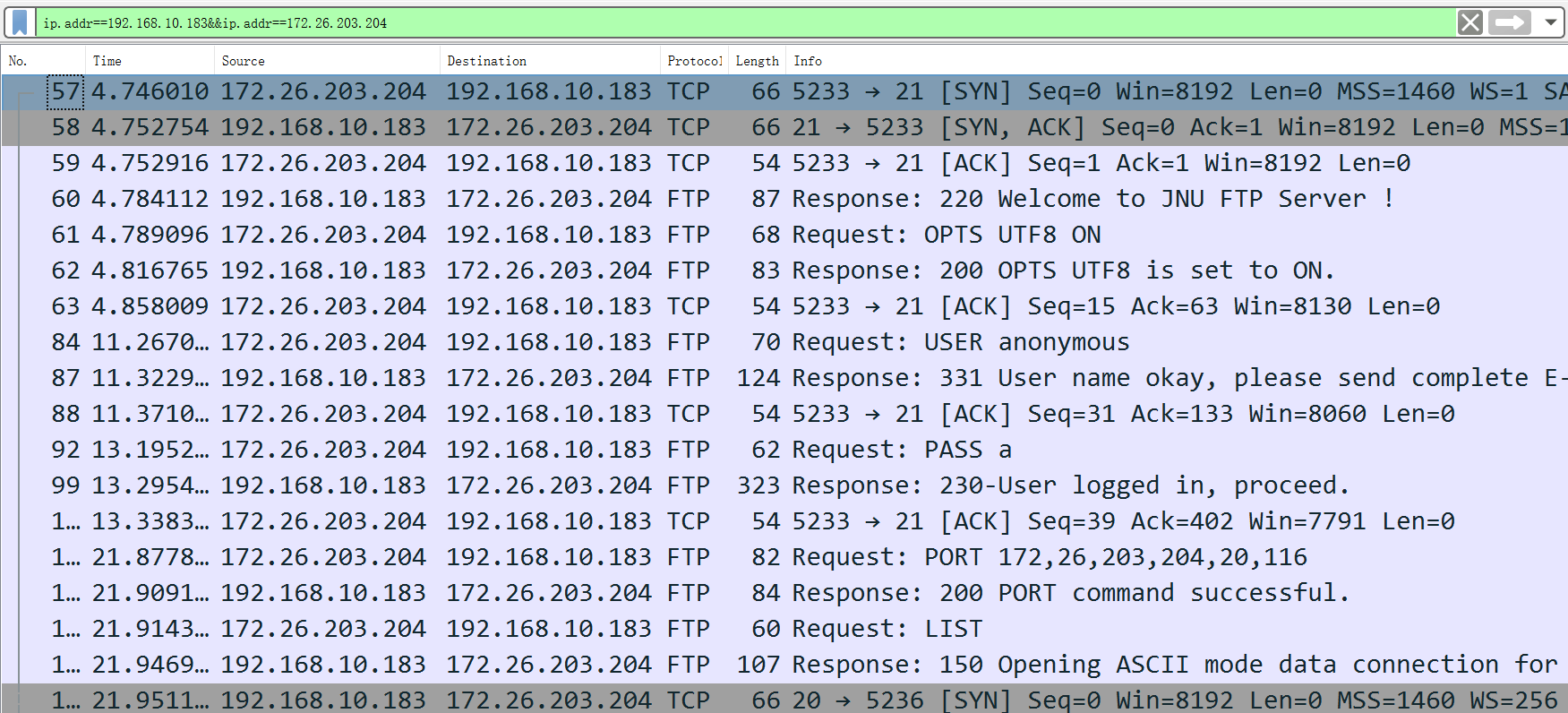
1. 数据传输完毕后，服务器端提示断开数据连接
2. TCP四次挥手断开TCP连接，数据连接断开，但是控制连接没有断开

**关闭控制连接**

1. 客户端进程（5233）向服务器端（21）发送bye命令，服务器端响应该命令
2. TCP四次挥手断开TCP连接，控制连接断开

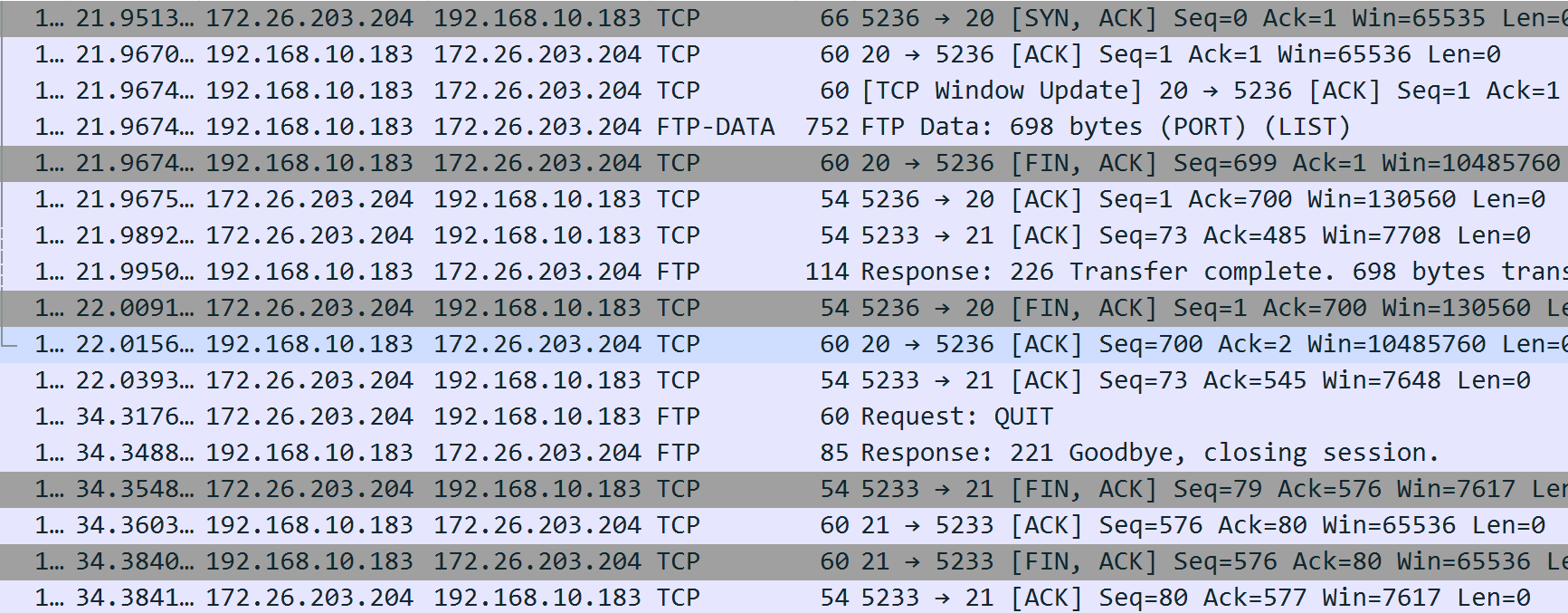
**TCP三次握手**

**建立控制连接**



**客户端请求主动连接，并发送IP地址及数据连接端口号**

**服务器端发起主动数据连接**



**传输数据**

**TCP四次挥手**

**断开数据连接**

**TCP四次挥手**

**断开控制连接**

1. **SMTP协议分析**

**利用Wireshark分析SMTP协议**

**邮件收发过程简述**

一封邮件的发送和接收过程如下：

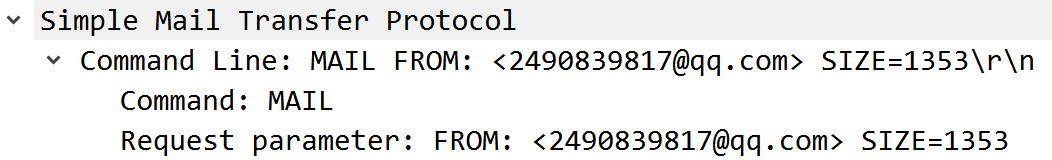
1. 发送方在用户代理（Foxmail/outlook）中编辑邮件
2. 用户代理生成一封符合邮件格式标准的邮件
3. 用户代理（Foxmail/outlook）利用SMTP协议将邮件发送到发送端邮件服务器中
4. 发送端邮件服务器利用SMTO协议将邮件发送到接收端邮件服务器中
5. 收信人调用用户代理（点击“收取”），用户代理利用POP3协议从接收端邮件服务器收取邮件
6. 用户代理解析收到的邮件以供收信人查看

**Wireshark对SMTP的协议分析**

**SMTP协议格式分析**

**SMTP命令格式：**COMMAND [Parameter] <CRLF>

其中，COMMAND是命令名，Parameter是命令参数，CRLF是回车换行符。

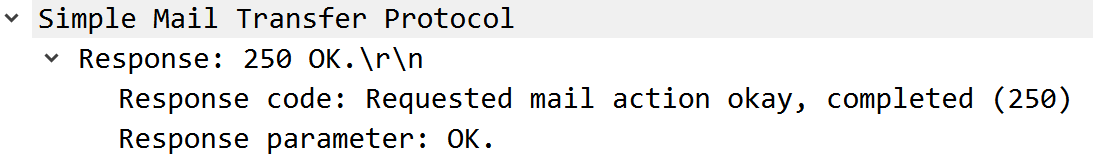


**命令参数**

**命令名**

**回车换行符**

**SMTP响应格式：**①响应码②响应参数 ③回车换行符

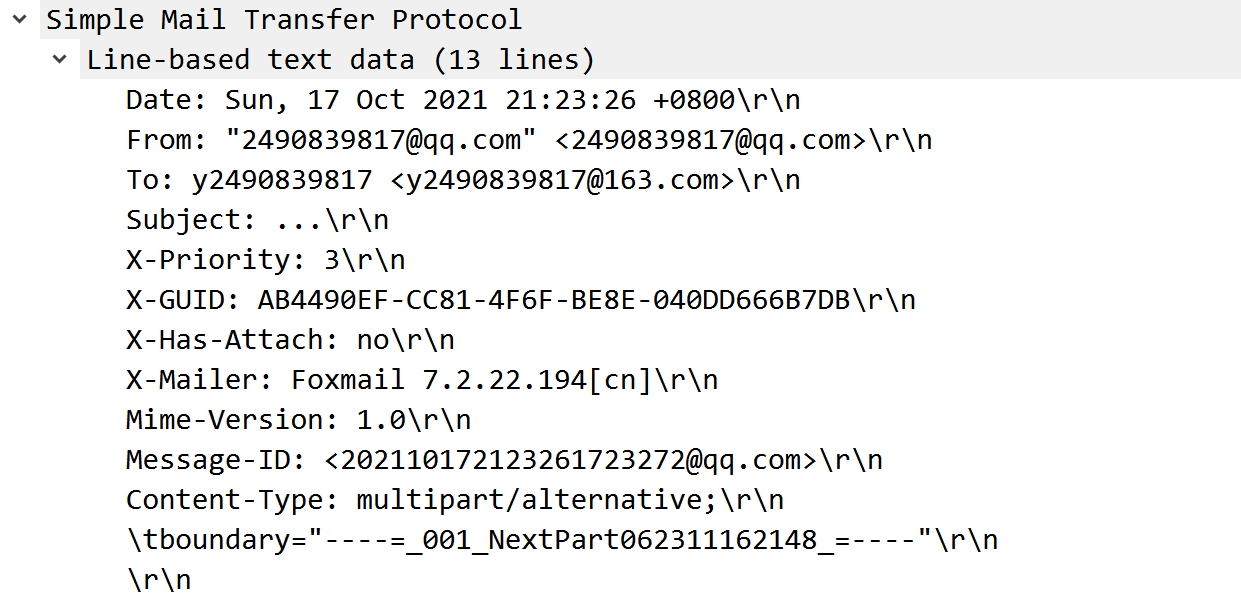


**响应参数**

**响应码**

**回车换行符**

**电子邮件信息格式**



**邮件主题**

**接收地址**

**发信地址**

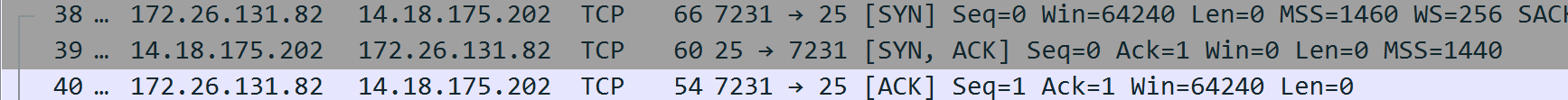
**发信日期**

**SMTP工作过程分析**

由于Wireshark只能捕获经过本机的数据包，因此本文只分析用户代理和发送端邮件服务器之间的SMTP协议工作过程，如下所示：

**建立连接**

1. 用户代理（客户端）与发送端邮件服务器（服务器）的25号端口通过TCP三次握手建立TCP连接



1. 客户端向服务器发送各种命令来请求各种服务；而服务器解析客户端的命令，并作出回应
   1. 服务器利用220响应码表明服务就绪

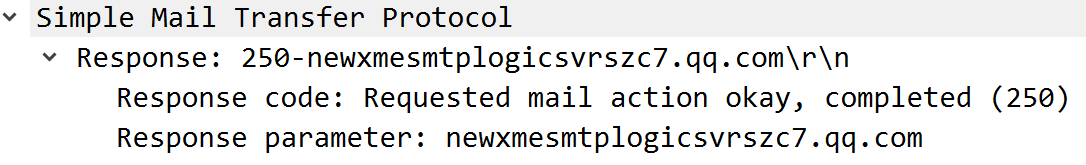


* 1. 客户端利用EHLO命令表明自己的用户身份



* 1. 服务器利用250响应码表明要求的邮件操作完成，并附上了支持的身份验证方式





**支持的身份验证方式**

* 1. 客户端利用AUTH命令表明用户登录，进行身份验证



* 1. 服务器利用235响应码表明身份认证成功



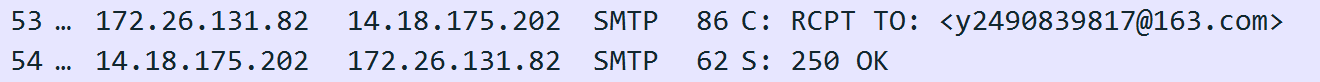
**邮件传送**

* 1. 客户端利用MAIL命令表明发送方身份，服务器收到后会返回OK报文作为应答





* 1. 客户端利用RCPT TO命令来标识该邮件的接收方，并确认邮件是否接收到，服务器若收到则会返回OK报文作为应答，否则会发出拒绝接收的应答

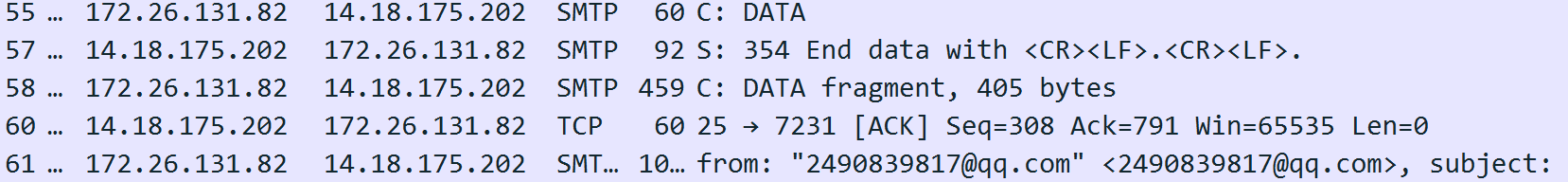


* 1. 客户端与服务器协商结束，客户端以DATA命令告知服务器自己要开始传输邮件的正文内容



* 1. 服务器利用354响应码告知邮件的内容结束部分要以<CR><LF>.<CR><LF>标记

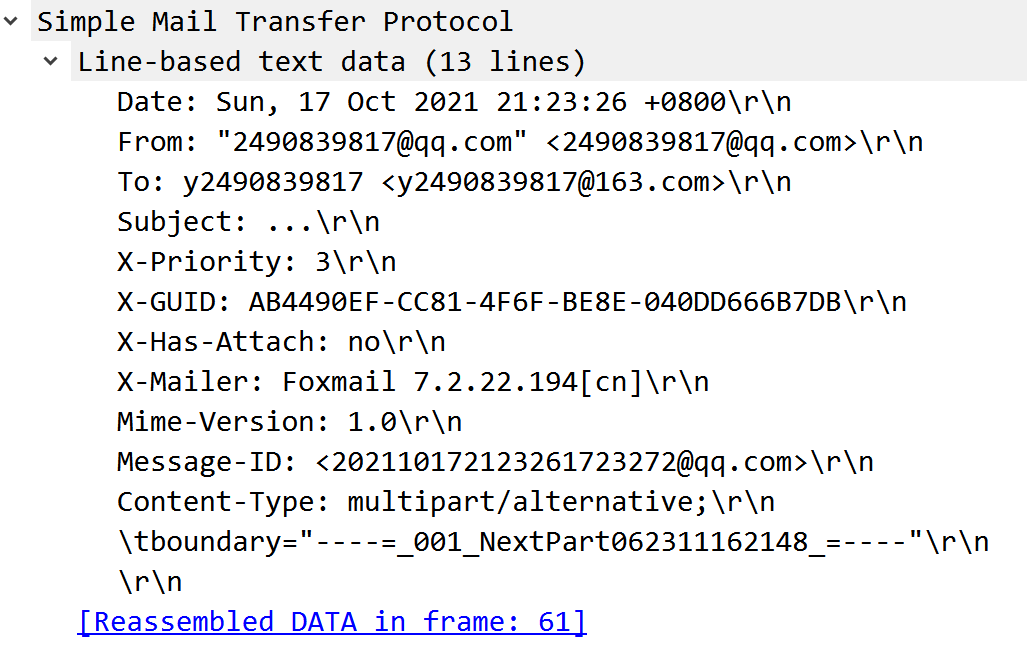
**标记传输终止**

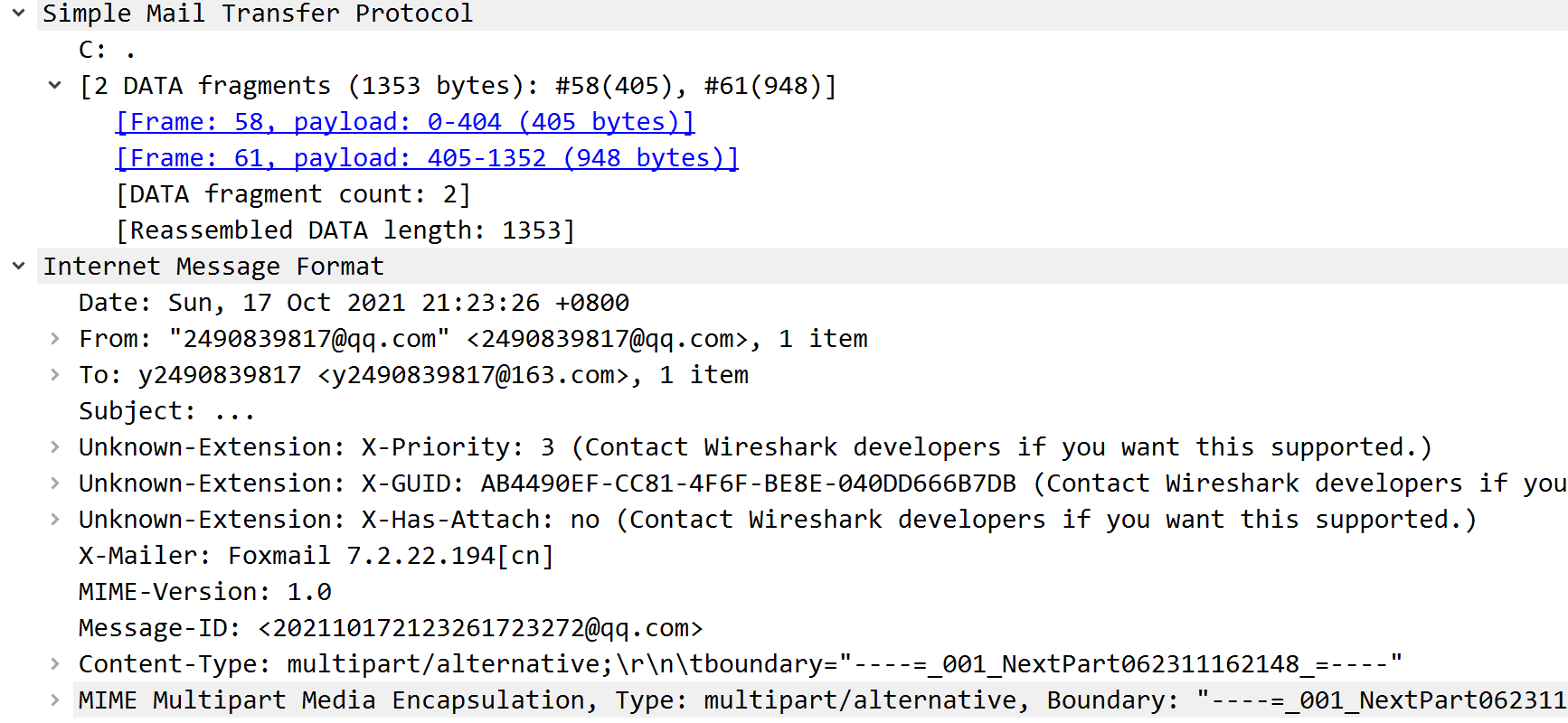


* 1. 客户端在接收到354响应码后，开始传输邮件内容。可以看到，以下的数据包中包含了SMTP和IMF两个协议。可以看到，邮件的内容就是有IMF协议进行传输的。而SMTP协议中的内容为DATA命令的终止标志。









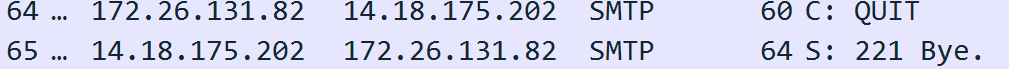
**邮件内容**

**终止标志**

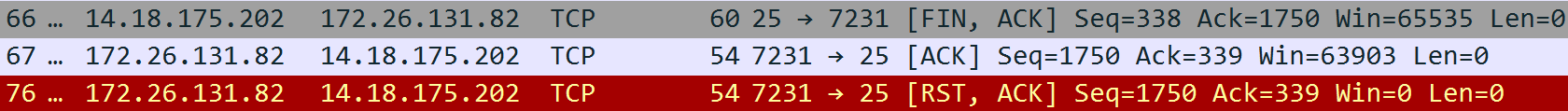
图 30 邮件正文

**释放连接**

* 1. 在客户端发送完邮件内容后，会以QUIT命令结束此次SMTP传输；服务器收到数据后，会返回221响应码表示结束本次SMTP传输

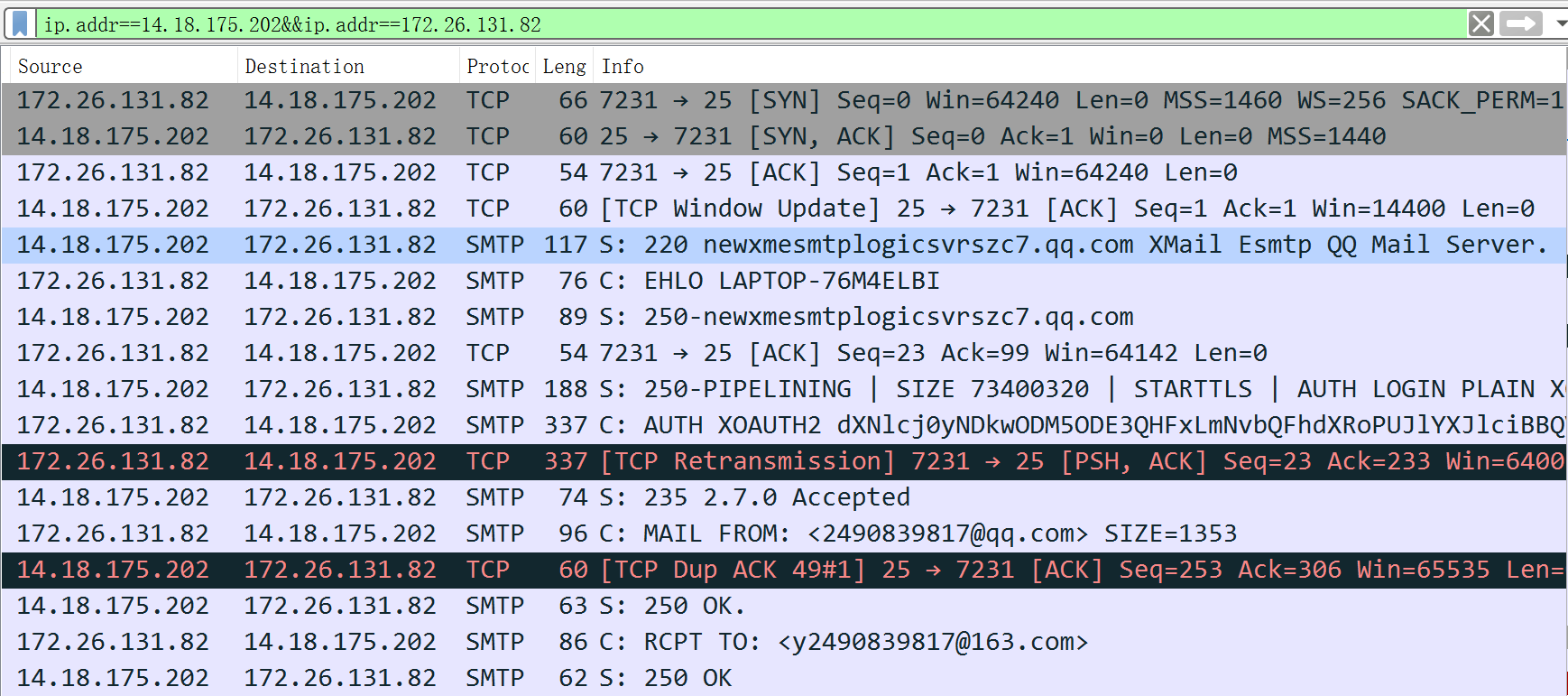


1. 所有邮件发送完毕后，SMTP关闭建立的TCP连接



**TCP三次握手建立用户代理与发送方邮件服务器的TCP连接**

**状态字：220，SMTP服务器连接成功**



**状态字：235，身份认证成功**

**RCPT TO：指明邮件接收方，并询问是否接收到邮件**

**服务器已接收邮件**

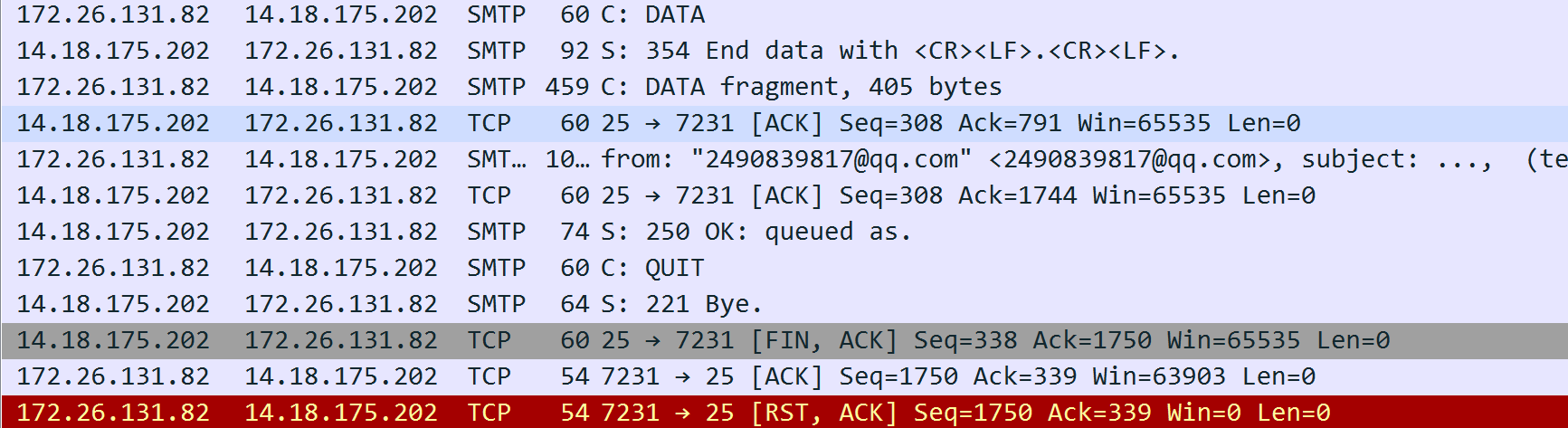
**MAIL：表明发送方身份**

**开始发送邮件**

**EHLO：表明用户身份**

**AUTH：用户登录**

**状态字：250，SMTP服务器握手成功**



**开始邮件输入**

**标志数据结尾**

1. **POP3协议分析**

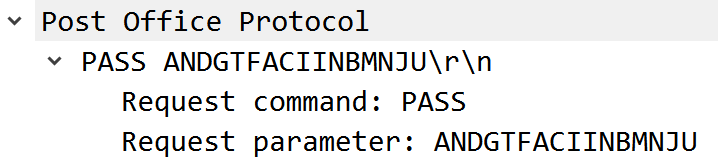
POP3协议是一种邮件读取协议，使用的是客户/服务器工作方式。POP3客户端与POP3服务器的110端口连接。在连接建立好以后，客户端与服务器之间通过交互命令和响应的方式完成邮件的接收。

|  |  |  |  |
| --- | --- | --- | --- |
| 命令 | 参数 | 状态 | 描述 |
| USER | Username | 处理 | 此命令若成功，将导致状态转换 |
| PASS | Password | 处理 | 此命令若成功，状态转化为更新 |
| APOP | Name,Digest | 处理 | Digest是MD5消息摘要 |
| STAT | None | 处理 | 请求服务器发回关于邮箱的统计资料 |
| UIDL | [Msg#]（邮件号） | 处理 | 返回邮件的唯一标识符 |
| LIST | [Msg#]（邮件号） | 处理 | 返回邮件的唯一标识符 |
| RETR | [Msg#]（邮件号） | 处理 | 返回由参数标识的邮件的全部文本 |
| DELE | [Msg#]（邮件号） | 处理 | 服务器将由参数标识的邮件标记为删除 |
| TOP | [Msg#]（邮件号） | 处理 | 服务器将返回由参数标识的邮件的邮件头+前n行内容，n必须是正整数 |
| NOOP | None | 处理 | 服务器返回一个肯定的响应，用于测试连接是否成功 |
| QUIT | None | 处理、认证 | 1)如果服务器处于“处理”状态，么将进入“更新”状态以删除任何标记为删除的邮件，并重返“认证”状态。  2)如果服务器处于“认证”状态，则结束会话，退出连接 |

**POP3协议格式分析**

**POP3命令格式：**COMMAND [Parameter] <CRLF>

其中，COMMAND是命令名，Parameter是命令参数，CRLF是回车换行符。

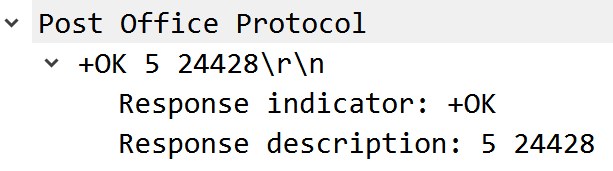


**命令名**

**回车换行符**

**命令参数**

**POP3响应格式**：①响应标志：“+OK”/“-ERR”②响应描述 ③回车换行符



**响应描述**

**响应标志**

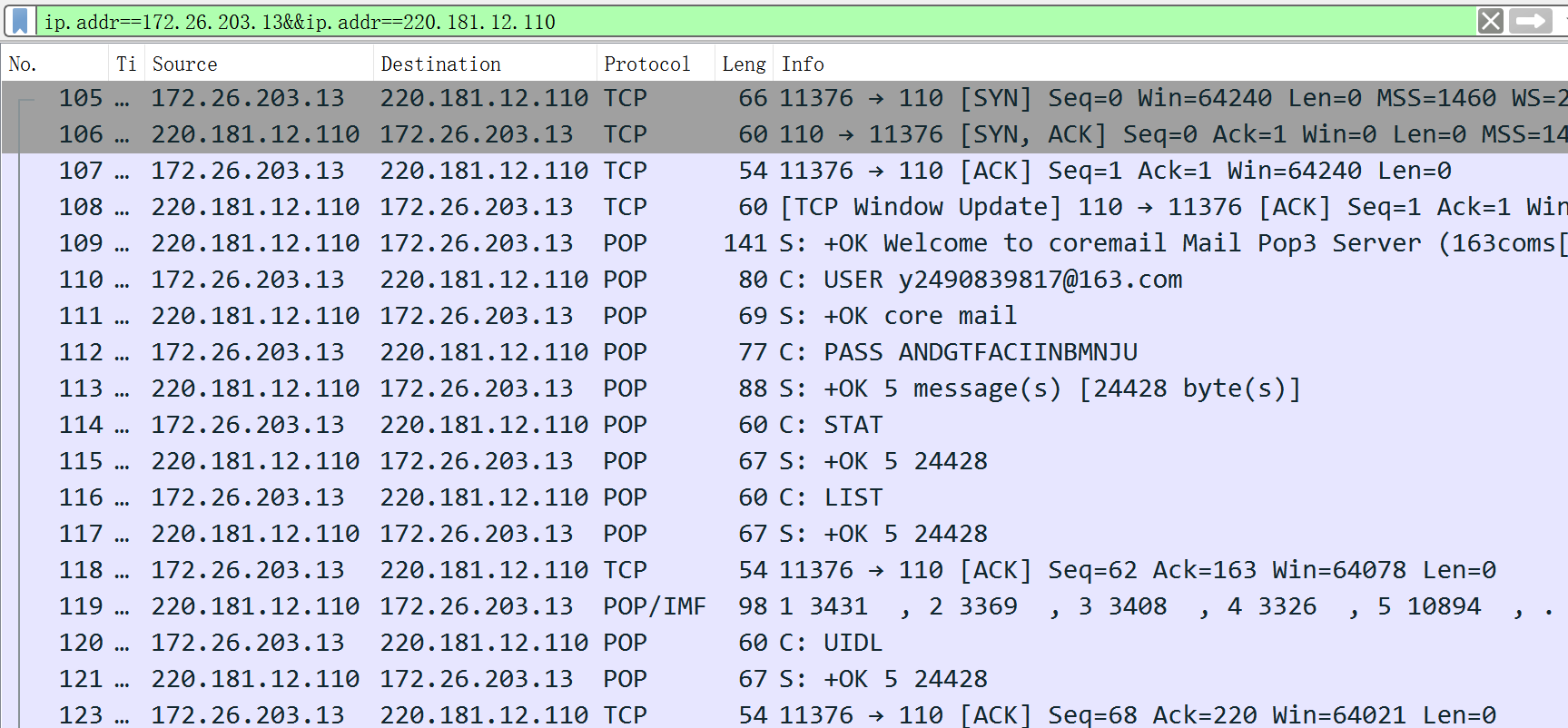
**回车换行符**

**POP3协议工作过程分析**

**三次握手建立接收端邮件服务器与用户代理的TCP连接**

**服务器通知客户端已经准备好**

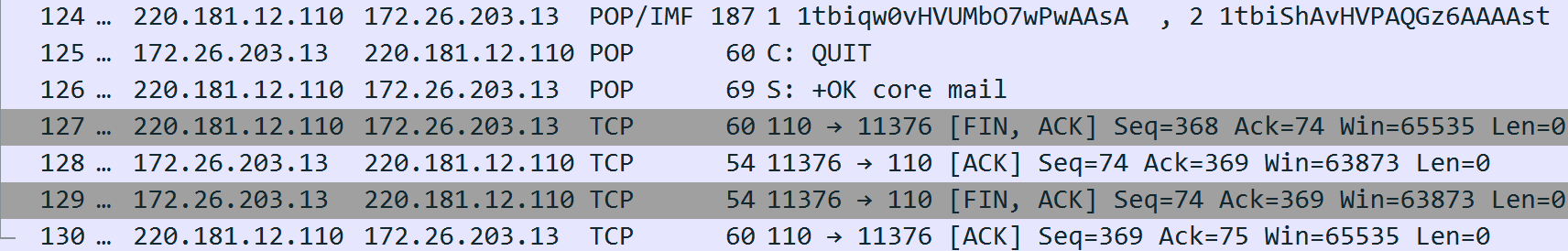
**客户端认证阶段**



**用户名**

**密码**

**请求服务器发送邮箱统计资料**



**TCP四次挥手断开连接**

**客户端向服务器发送命令码LIST，服务器返回命令执行的状态，邮件数量和每个邮件的大小**

**客户端请求退出处理状态，服务器会将客户端标识为删除的邮件删除并退出POP3服务器**

**客户端请求服务器返回邮件的唯一标识，如果没有指定邮件，则返回所有邮件的标识**

**什么情况下可以捕获POP或SMTP数据包？**

1. 在本机安装用户代理（Outlook/Foxmail）
2. 在两个邮箱账号的设置中打开POP3/SMTP服务
3. 在用户代理中创建两个邮箱类型为POP3的账号（与第2步中的账号对应）
4. 将收件服务器、发件服务器中的SSL勾选取消，用以获得不加密的报文

捕获SMTP数据包

在用户代理中编辑好邮件，并打开Wireshark捕获数据包，再在用户代理中点击“发送”即可捕获到SMTP数据包。

捕获POP3数据包

打开Wireshark捕获数据包，再在用户代理中点击“收取”即可捕获到POP3数据包。

1. **实验结果与分析**
2. **HTTP**
   1. HTTP协议本身是无连接的。虽然HTTP使用了TCP连接，但是通信双方在交换HTTP报文之前不需要先建立HTTP连接
   2. HTTP协议是无状态的，服务器不会记得是否曾经访问过某用户
3. **DNS总结**
   1. Wireshark只能捕获到本机到本地域名服务器或本地域名服务器到本机的DNS数据包。在数据包中，我们无法观察到本地域名服务器是用递归方式还是用迭代方式来对域名进行解析；也观察不到究竟访问了多少台权限域名服务器；更不知道究竟是哪一台权限域名服务器完成了域名的解析。通过观察本地域名服务器的应答报文，我们只能知道本机到本地域名服务器是否用迭代的方式进行查询，以及哪几个顶级域名服务器中有域名的IP地址映射。
   2. 本地的DNS缓存中存在的域名对应的IP地址映射不一定有效，其有效时间会以Time to live这项参数作为参考，在执行ipconfig /flushdns后，可以查看本机还在有效期内的IP地址映射
   3. 本地域名服务器中也会保存域名对IP地址的映射，当本地域名服务器利用该映射完成域名解析时，该解析不具有权威性
4. **FTP总结**
   1. 主动被动是根据服务器端来定义的
   2. 主动模式服务器端的数据端口号为20，被动模式服务器端的数据端口号由PASV应答命令指定
   3. 不论是主动模式还是被动模式，服务器控制端口号均为21
   4. 主动模式的控制和数据连接均由客户端发起，这可能使得服务器端到客户端的连接被防火墙过滤
   5. 端口N和N+1并不连续，而是通过相应的应答报文（被动模式）或请求报文（主动模式）确定，端口号=第五位数\*256+第六位数
5. **SMTP总结**
6. 发送端邮件服务器（客户端）与接收端邮件服务器（服务器）的25号端口建立TCP连接
7. 用户代理将邮件发送到发送端邮件服务器的过程与发送端邮件服务器将邮件发送到接收端邮件服务器的过程相同

SMTP协议缺点：

1. SMTP协议只能传送7位的ASCII码，而不能传送二进制文件
2. **POP3总结**
3. POP3客户端与POP3服务器的110端口连接
4. POP3协议的命令和响应报文格式是固定的

**暨南大学本科实验报告专用纸(附页)**