本 科 实 验 报 告

## 实验名称：利用Wireshark分析验证网络协议

学 员： 于鹄杰 学 号： 201506021075

专 业：网络工程（军事物联网方向） 所属学院： 计算机学院

**国防科学技术大学训练部制**

### 一、实验目的

通过实验，了解和掌握报文捕获工具Wireshark的使用方法和基本特点，使用Wireshark捕获网络报文，并分析各种网络协议的报文格式和工作过程。

### 二、实验内容

使用Wireshark捕获网络报文，分析以太网、ARP、IP、TCP、DNS和HTTP等协议的报文格式和工作过程。

### 实验步骤

1. ICMP分析：

在cmd命令行下通过发ping [www.baidu.com](http://www.baidu.com) 和traceroute [www.baidu.com](http://www.baidu.com) 命令，然后使用Wireshark捕获报文并分析ICMP报文格式和工作过程。

2、 DNS分析和网页请求过程分析：

在浏览器输入www.baidu.com的URL地址后回车，分析DNS过程（涉及到DNS，UDP，IP，ARP和以太网），网页请求过程（涉及到HTTP，TCP，IP，ARP和以太网）。

### 实验结果及分析

（一）实验的基本参数

实验主机的IP地址为：192.168.8.106

实验主机的MAC地址为：84-EF-18-9A-3E-81

子网掩码：255.255.255.0

默认网关：192.168.8.1

默认网关MAC地址为：b0-e5-ed-d9-24-6f

DHCP服务器：192.168.8.1

DNS服务器：192.168.8.1

1. ICMP分析
2. **ping [www.baidu.com](http://www.baidu.com)**

首先，我在自己电脑（实验主机）上通过命令行cmd命令ping www.baidu.com,得到四个数据包：



图 1

同时，利用wireshark软件进行抓包，利用协议筛选框对ICMP报文进行筛选，捕获分组结果非常理想：

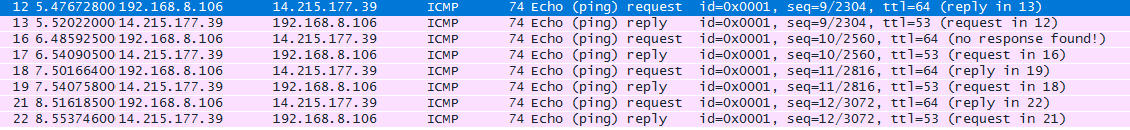
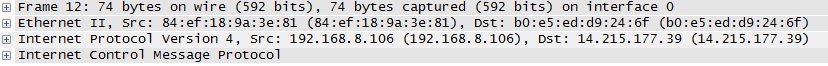


图 2

下面，我们就其中的一条ICMP报文来分析ICMP报文的报文格式以及工作过程，我选取了NO.12这条ICMP报文（标蓝），下面是这条ICMP报文的首部明细：



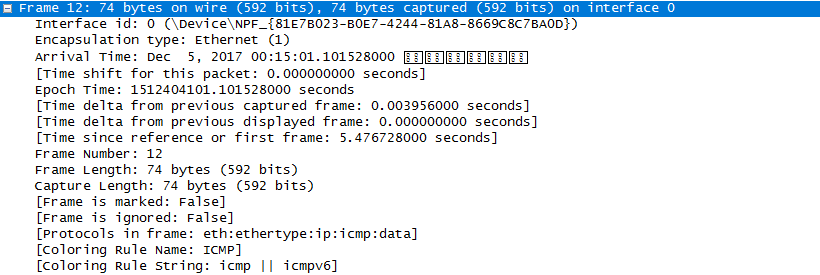


图 3

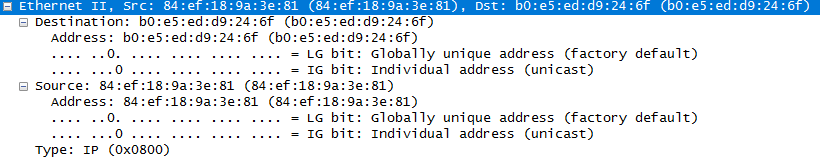


图 4

我们可以在此条ICMP报文发现，源MAC地址为实验主机的MAC地址为：84-EF-18-9A-3E-81，目的MAC地址为默认网关的MAC地址为：b0-e5-ed-d9-24-6f。

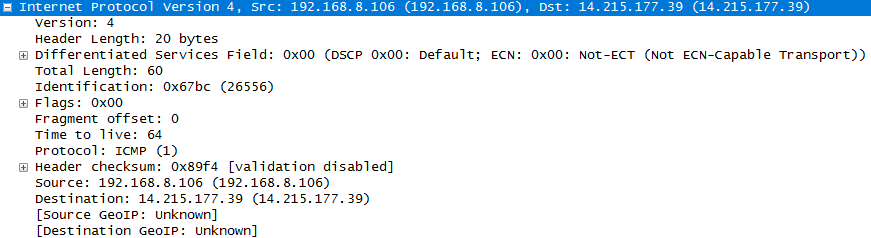


图 5

我们也可以在此条ICMP报文发现，源IP地址为实验主机的IP地址为：192.168.8.106，目的IP地址为www.baidu.com的IP地址：14.215.177.39。通过以上结果，对我们先前的实验参数进行了验证。

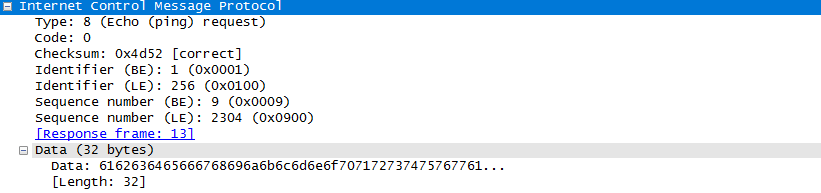


图 6

通过以上结果，我们可以分析出ICMP报文的报文格式：

ICMP（Internet Control Message Protocol）因特网控制报文协议。它是IPv4协议族中的一个子协议，用于IP主机、路由器之间传递控制消息。控制消息是在网络通不通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然不传输用户数据，但是对于用户数据的传递起着重要的作用。   
ICMP协议与ARP协议不同，ICMP靠IP协议来完成任务，所以ICMP报文中要封装IP头部。它与传输层协议（如TCP和UDP）的目的不同，一般不用来在端系统之间传送数据，不被用户网络程序直接使用，除了像Ping和Tracert这样的诊断程序。

**（1）NO.12的icmp报文的报文格式：**

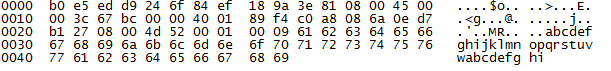


图 7

其中：

0~5字节：目的MAC地址；

6~11字节：源MAC地址；

12~13字节：Type：IP

14字节：header length

15字节：Explicit Congestion Notification

16~17字节：Total Length

18~19字节：Identification

20 ~21字节：Fragment offset

22字节：Time to live

23字节：Protocol

24~25字节：Header Checksum

26~29字节：源IP地址

30~33字节：目的IP地址

34~73字节：ICMP报文

34字节：类型号

35字节：代码号

36~37字节：校验和

38~39字节：标识符

40~41字节：序列号

42~73字节：数据

根据百度得到的信息，各种ICMP报文的前32bits都是三个长度固定的字段：type类型字段(8位)、code代码字段(8位)、checksum校验和字段(16位)。

8bits类型和8bits代码字段：一起决定了ICMP报文的类型。常见的有：　  
　　 类型8、代码0：回射请求。  
　　 类型0、代码0：回射应答。  
　　 类型11、代码0：超时。

ICMP报文包含在IP数据报中，IP报头在ICMP报文的最前面。一个ICMP报文包括IP报头（至少20字节）、ICMP报头（至少八字节）和ICMP报文（属于ICMP报文的数据部分）。当IP报头中的协议字段值为1时，就说明这是一个ICMP报文。ICMP报头如下图所示。



图 8

类型Type：占一字节，标识ICMP报文的类型，目前已定义了14种，从类型值来看ICMP报文可以分为两大类。第一类是取值为1~127的差错报文，第2类是取值128以上的信息报文。在NO.12这条icmp报文中，类型值为8。

代码Code：占一字节，标识对应ICMP报文的代码。它与类型字段一起共同标识了ICMP报文的详细类型。在NO.12这条icmp报文中，代码值为0。

校验和Checksum：这是对包括ICMP报文数据部分在内的整个ICMP数据报的校验和，以检验报文在传输过程中是否出现了差错。

标识Identifier：占两字节，用于标识本ICMP进程，但仅适用于回显请求和应答ICMP报文，对于目标不可达ICMP报文和超时ICMP报文等，该字段的值为0。

**（2）工作过程：**

我们进行的Ping操作中就包括了相应请求（类型字段值为8）和应答（类型字段值为0）ICMP报文。类型8、代码0决定了NO.12 ICMP报文是一条回射请求报文。

Ping数据与www.baidu.com建立通信的过程如下：

1. 由客户端PC发出ARP以确认DNS服务器并建立连接；
2. 再通过DNS服务器解析出域名www.baidu.com的IP地址；
3. 客户端通过与解析出来的目的IP地址对比目的IP地址是否与主机IP地址在同一个网段；
4. 主机IP地址与目的IP地址不在同一个网段，则PC向网关发出ARP请求；
5. 网关回复PC的ARP请求；
6. Ping请求包通过目的IP地址转发并收到目的主机的回复包完成ping流程。
7. **tracert [www.baidu.com](http://www.baidu.com)**

首先，我在自己电脑（实验主机）上通过命令行cmd命令tracert www.baidu.com ，得到如下结果：



图 9

同时，利用wireshark软件进行抓包，利用协议筛选框对ICMP报文进行筛选，捕获分组结果比较理想：

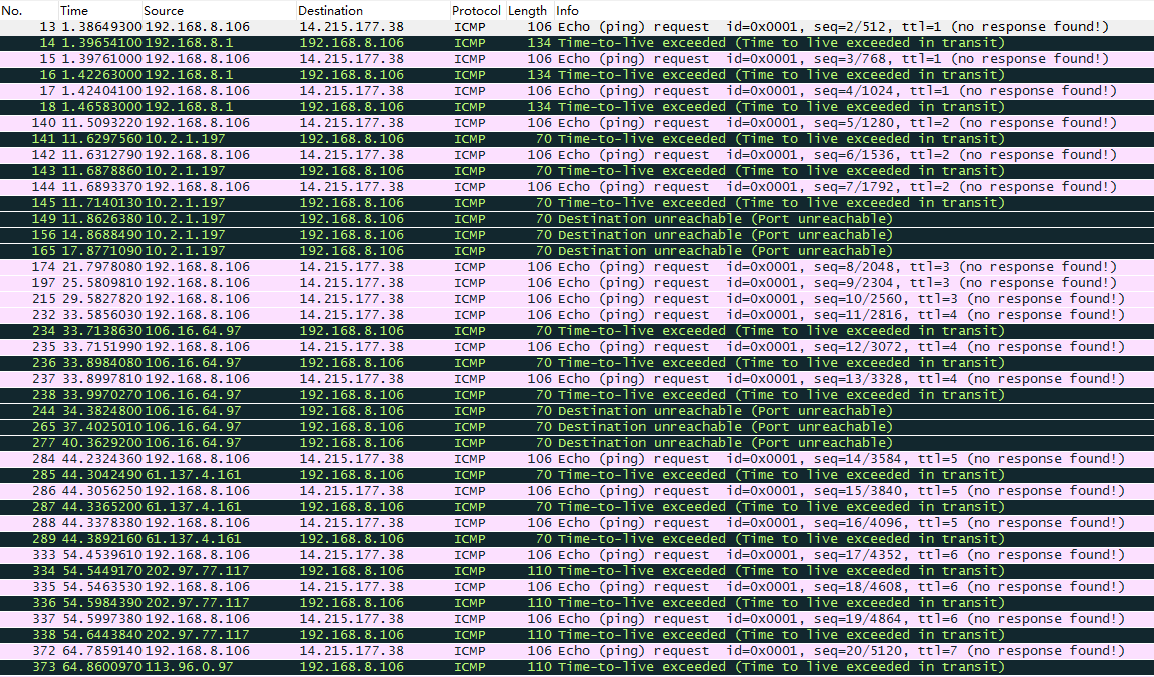


图 10

我们选取其中一条ICMP报文进行分析——NO.591。

这条报文的首部明细结果如下：

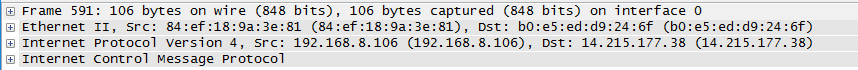


图 11

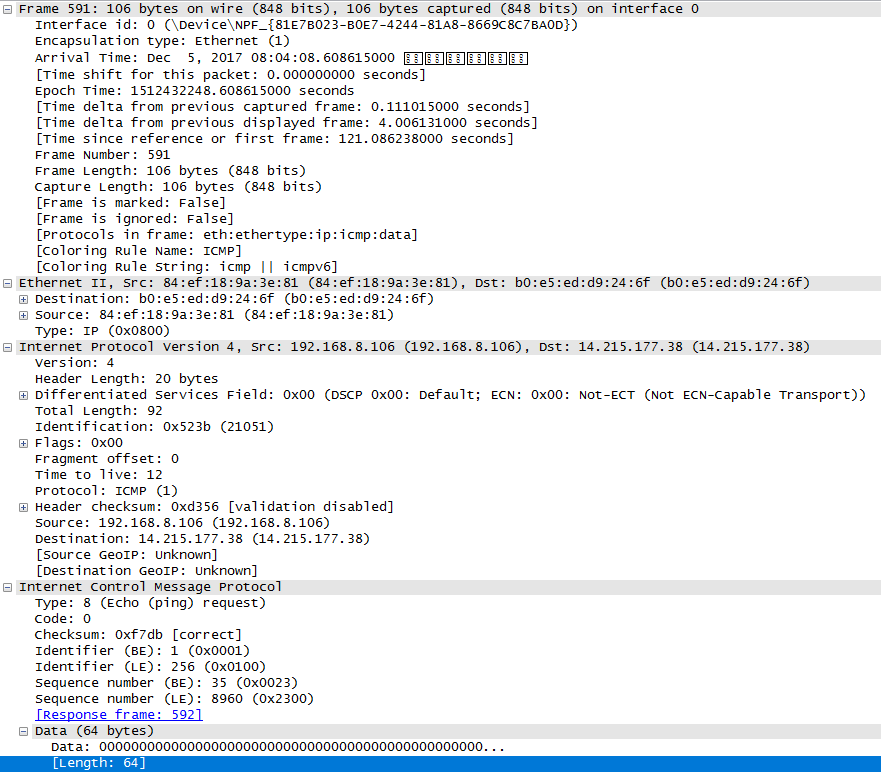


图 12

**（1）ICMP报文格式**

我们可以发现ICMP的报文格式除数据区由32bytes变为64bytes，其他没有发生变动。下面我们主要来分析traceroute的工作过程。

1. **traceroute [www.baidu.com的工作过程](http://www.baidu.com的工作过程)**

Traceroute是用来侦测主机到目的主机之间所经路由情况的重要工具，也是最便利的工具。前面说到，尽管ping工具也可以进行侦测，但是，因为ip头的限制，ping不能完全的记录下所经过的路由器。所以Traceroute正好就填补了这个缺憾。

Traceroute的原理如下，它收到目的主机www.baidu.com的IP后，首先给目的主机www.baidu.com发送一个TTL=1的UDP数据包，而经过的第一个路由器收到这个数据包以后，就自动把TTL减1，而TTL变为0以后，路由器就把这个包给抛弃了，并同时产生一个主机不可达的ICMP数据报给主机。主机收到这个数据报以后再发一个TTL=2的UDP数据报给目的主机，然后刺激第二个路由器给主机发ICMP数据报。如此往复直到到达目的主机。这样，traceroute就拿到了所有的路由器ip，从而避开了ip头只能记录有限路由IP的问题。

Traceroute的工作过程如下，根据得到的ICMP报文，在Tracert过程中前几次发request之后没有收到reply，而是收到exceeded time。而且IP报文中的TTL在不断增加，则可以推断tracert的工作过程是tracert送出一个TTL是1的IP 数据包到目的地，当路径上的第一个路由器收到这个数据包时，它将TTL减1。此时，TTL变为0，所以该路由器会将此数据包丢掉，并送回一个「ICMP time exceeded」消息（包括发IP包的源地址，IP包的所有内容及路由器的IP地址），tracert 收到这个消息后，便知道这个路由器存在于这个路径上，接着tracert 再送出另一个TTL是2 的数据包，发现第2 个路由器。 tracert 每次将送出的数据包的TTL 加1来发现另一个路由器，这个重复的动作一直持续到某个数据包抵达目的地。当数据包到达目的地后，该主机则不会送回ICMP time exceeded消息，一旦到达目的地，会收到reply。

1. DNS分析和网页请求过程分析
2. **总体过程概述**

首先在浏览器输入www.baidu.com的URL地址，利用wireshark软件进行抓包，通过协议筛选框筛选所需要的协议报文，

该过程具体的事件顺序如下：

1. 浏览器获取输入的域名www.baidu.com   
    (2) 浏览器向DNS请求解析www.baidu.com的IP地址   
    (3) 域名系统DNS解析出百度服务器的IP地址   
    (4) 浏览器与该服务器建立TCP连接(默认端口号80)   
    (5) 浏览器发出HTTP请求，请求百度首页   
    (6) 服务器通过HTTP响应把首页文件发送给浏览器   
    (7) TCP连接释放   
    (8) 浏览器将首页文件进行解析，并将Web页显示给用户。

其中涉及到的协议如下：

(1) 应用层：HTTP(WWW访问协议)，DNS(域名解析服务)   
DNS解析域名为目的IP，通过IP找到服务器路径，客户端向服务器发起HTTP会话，然后通过运输层TCP协议封装数据包，在TCP协议基础上进行传输。

(2) 传输层：TCP(为HTTP提供可靠的数据传输)，UDP(DNS使用UDP传输)   
HTTP会话会被分成报文段，添加源、目的端口；TCP协议进行主要工作。

(3)网络层：IP(IP数据数据包传输和路由选择)，ICMP(提供网络传输过程中的差错检测)，ARP(将本机的默认网关IP地址映射成物理MAC地址) 为数据包选择路由，IP协议进行主要工作，相邻结点的可靠传输，ARP协议将IP地址转成MAC地址。

1、客户端浏览器通过DNS解析到www.baidu.com的IP地址14.215.177.38，通过这个IP地址找到客户端到服务器的路径。客户端浏览器发起一个HTTP会话到14.215.177.38，然后通过TCP进行封装数据包，输入到网络层。   
 2、在客户端的传输层，把HTTP会话请求分成报文段，添加源和目的端口，如服务器使用80端口监听客户端的请求，客户端由系统随机选择一个端口如5000，与服务器进行交换，服务器把相应的请求返回给客户端的5000端口。然后使用IP层的IP地址查找目的端。   
 3、客户端的网络层不用关心应用层或者传输层的东西，主要做的是通过查找路由表确定如何到达服务器，期间可能经过多个路由器，这些都是由路由器来完成的工作，无非就是通过查找路由表决定通过那个路径到达服务器。   
 4、客户端的链路层，包通过链路层发送到路由器，通过邻居协议查找给定IP地址的MAC地址，然后发送ARP请求查找目的地址，如果得到回应后就可以使用ARP的请求应答交换的IP数据包现在就可以传输了，然后发送IP数据包到达服务器的地址。

**2、DNS分析**

如图：

ARP

图 13

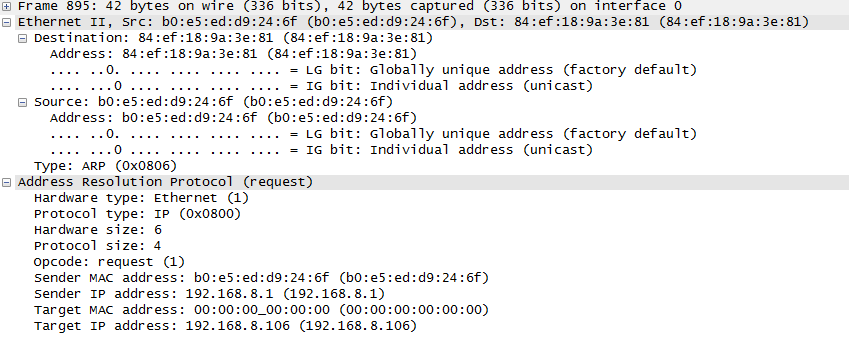


图 14

在这个实验中，首先关注的是域名的解析过程，也就是通过DNS来获得www.baidu.com的IP地址的过程。这一部分主要关注DNS和ARP类型的协议。可以看到序号895报文先发送了ARP请求，那就从ARP开始分析。

ARP是处于链路层的协议，报文中只有以太网帧的头部和ARP主体。其次，ARP的主要功能是通过IP地址来获取目的主机的MAC地址，因此为了让目的主机接收该以太网帧，ARP的目的MAC地址填写的是全1。而任何主机在收到目的MAC为全1的报文时都会收取，这也就是我们平时说的广播的实现过程，这个过程体现的是链路层的广播，另外网络层也有广播，实现方式也类似。在ARP报文内部，有关注到有一个Target MAC address为全0，与以太网帧的目的MAC全1不同.

由我的主机发出“Who has 192.168.43.1 Tell 192.168.43.238”的询问请求，注意到192.168.43.1是我主机的默认网关，因此可以推断出在ARP过程是向默认网关发ARP包，之后再由网关继续ARP从而最终得到DNS服务器的MAC地址。

筛选出dns报文，如图：

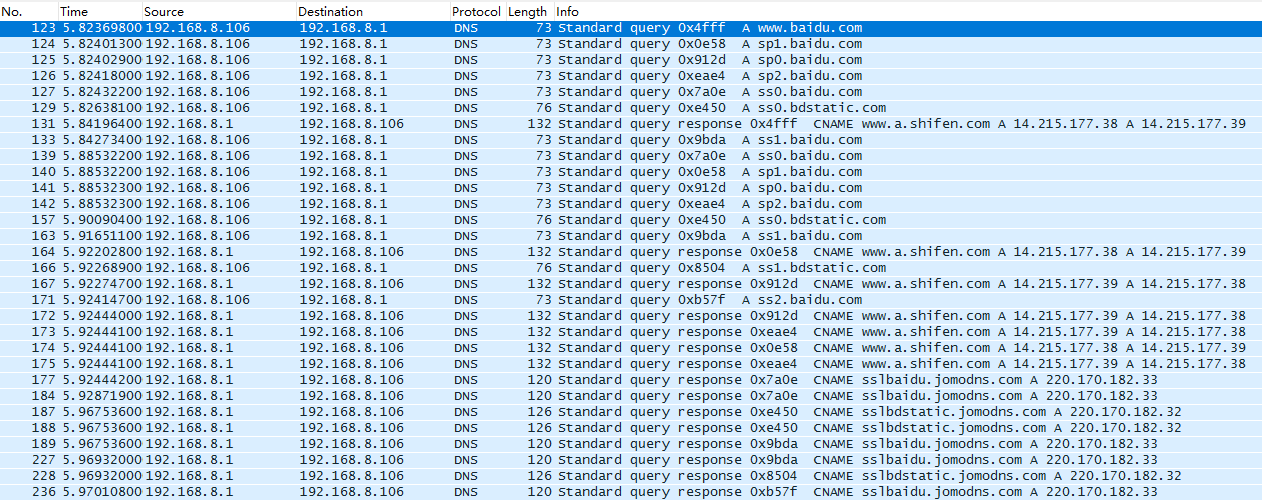


图 15

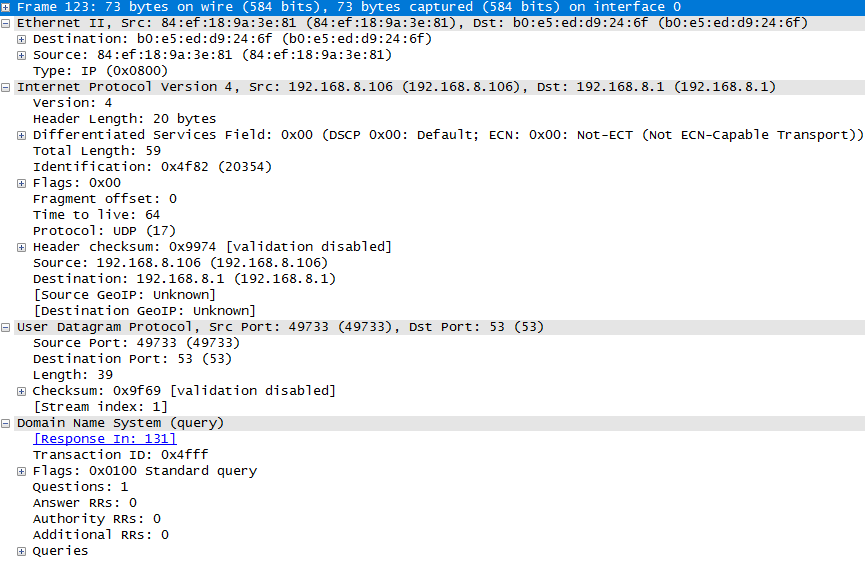


图 16

DNS是应用层协议，因此会经过层层打包才经过物理层发出，接下来我按照自顶向下的顺序进行分析。

首先是在应用层，因为是进行域名的解析，所以发出的报文是类型A的的报文。对照上一个实验中对DNS协议的分析，可以得到该DNS的标识为0x4fff、标志0x0100、问题数1、回答等信息。

在传输层，运用的是UDP的传输方式，本机端口分配了49733号端口，目的端口为53。这里值得注意的是，53号端口一般就是用于DNS服务器的服务端口。至于为什么用了UDP没有用TCP是因为UDP相对来说速度更快效率更高，另外不需要与服务器之间建立连接，降低了服务器的负载。即便是失败了丢包，也可以重传，影响不大。UDP的长度够为39字节，检验和为0x9f69这里用于检测收到的报文是否有错。

在网络层，进行的是IPv4协议，其中目的地址为192.168.8.1。

DNS具体分析如下：

1、在浏览器中输入www.baidu.com域名，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、如果hosts里没有这个域名的映射，则查找本地DNS解析器缓存，是否有这个网址映射关系，如果有，直接返回，完成域名解析。

3、如果hosts与本地DNS解析器缓存都没有相应的网址映射关系，首先会找TCP/IP参数中设置的首选DNS服务器，在此我们叫它本地DNS服务器，此服务器收到查询时，如果要查询的域名，包含在本地配置区域资源中，则返回解析结果给客户机，完成域名解析，此解析具有权威性。

4、如果要查询的域名，不由本地DNS服务器区域解析，但该服务器已缓存了此网址映射关系，则调用这个IP地址映射，完成域名解析，此解析不具有权威性。

5、如果本地DNS服务器本地区域文件与缓存解析都失效，则根据本地DNS服务器的设置（是否设置转发器）进行查询，如果未用转发模式，本地DNS就把请求发至13台根DNS，根DNS服务器收到请求后会判断这个域名(.com)是谁来授权管理，并会返回一个负责该顶级域名服务器的一个IP。本地DNS服务器收到IP信息后，将会联系负责.com域的这台服务器。这台负责.com域的服务器收到请求后，如果自己无法解析，它就会找一个管理.com域的下一级DNS服务器地址(baidu.com)给本地DNS服务器。当本地DNS服务器收到这个地址后，就会找baidu.com域服务器，重复上面的动作，进行查询，直至找到www.baidu.com主机。

6、如果用的是转发模式，此DNS服务器就会把请求转发至上一级DNS服务器，由上一级服务器进行解析，上一级服务器如果不能解析，或找根DNS或把转请求转至上上级，以此循环。不管是本地DNS服务器用是是转发，还是根提示，最后都是把结果返回给本地DNS服务器，由此DNS服务器再返回给客户机。

总的来说，分为以下几步：

1. 本机向本地DNS请求www.baidu.com;
2. 本地DNS向根域请求www.baidu.com，根域返回com.域的服务器IP，向com.域请求www.baidu.com，com.域返回baidu.com域的服务器IP;
3. 向baidu.com请求www.baidu.com，返回cname www.a.shifen.com和a.shifen.com域的服务器IP;
4. 向root域请求www.a.shifen.com;
5. 向com.域请求www.a.shifen.com;
6. 向shifen.com请求;
7. 向a.shifen.com请求;
8. 拿到www.a.shifen.com的IP;
9. 本地DNS返回本机www.baidu.com cname www.a.shifen.com以及www.a.shifen.com的IP。

**3、网页请求过程分析**

利用wireshark软件筛选出HTTP报文，结果如下：

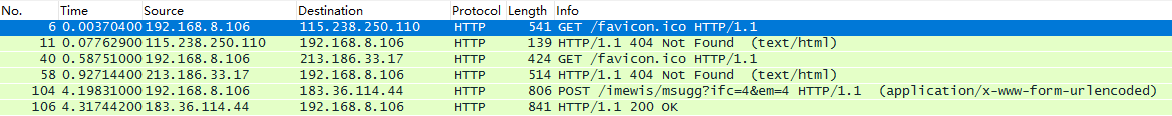


图 17

**3.1 请求报文**

如图：

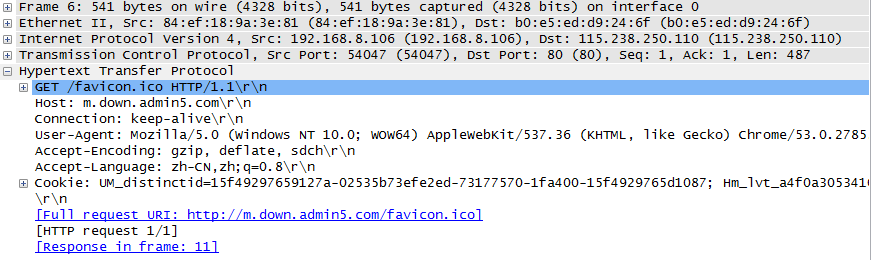


图 18

这是一个方法字段为GET的请求报文，所用到的协议的版本为HTTP/1.1。

User-Agent首部描述了提出请求的Web浏览器类型及客户机器的信息（图中显示了我的操作系统的信息）。接下来是一系列的Accpet首部，包括Accept（接受）、Accept-Language（接受语言）、Accept-Encoding（接受编码）等。它们告诉Web服务器客户Web浏览器准备处理的数据类型。Web服务器可以将数据转变为不同的语言和格式。

Connection及Keep-Alive首部描述了有关TCP连接的信息，通过此连接发送HTTP请求和响应。它表明在发送请求之后连接是否保持活动状态及保持多久。大多数HTTP1.1连接是持久的（persistent）,意思是在每次请求后不关闭TCP连接，而是保持该连接以接受从同一台服务器发来的多个请求，如果是非持久的就为Connection：close。这里无cookie，说明无web缓存。

**3.2 回答报文**

如图：

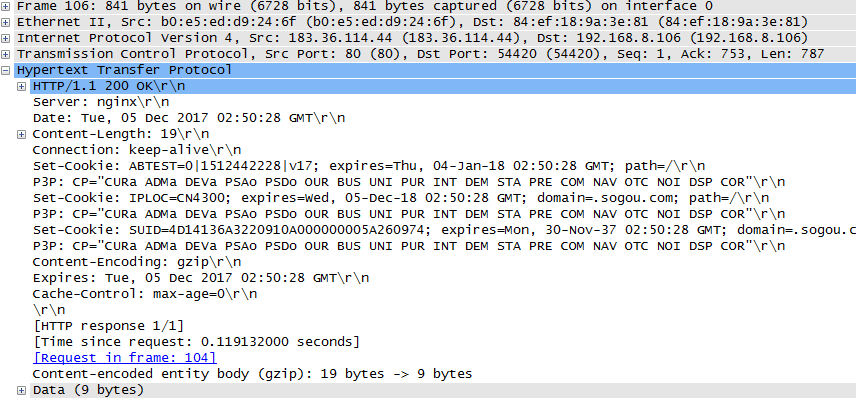


图 19

这是Web服务器对由web浏览器请求做出的响应。响应首先发送“HTTP/1.1 200 ok”，指明它开始使用HTTP1.1版本来发送网页。同样，在响应分组中，它后面也跟随着一些首部。最后，被请求的实际数据被发送。

Cache-control首部，用于描述是否将数据的副本存储或高速缓存起来，以便将来引用。一般个人的Web浏览器会高速缓存一些本机最近访问过的网页，随后对同一页面再次进行访问时，如果该网页仍存储于高速缓存中，则不再向服务器请求数据。类似地，在同一个网络中的计算机可以共享一些存在高速缓存中的页面，防止多个用户通过到其他网路的低速网路连接从网上获取相同的数据。这样的高速缓存被称为代理高速缓存（proxy cache）。

在HTTP请求中，Web服务器列出内容类型及可接受的内容编码。此例中Web服务器选择发送内容的类型是application/x-javascript。

服务器描述了一些关于自身的信息。响应分组还用Content-Length首部描述了数据的长度。此外，服务器还在Date首部中列出了数据发送的日期和时间。

### 五、实验中遇到的问题

1.一开始，首先对wireshark这款软件不了解，只知道可以用它来进行抓包实验，进而分析报文格式以及分析验证网络协议，但是不了解它如何操作，直到仔细阅读了使用手册以及上网查阅教程之后，才算是真正开始了实验。

2.在分析ICMP报文之前，坦白来说，对ICMP报文并不是很了解，甚至于连大致的报文格式也不知道，在分析抓的包的时候，对首部明细中许多信息感到很迷茫，主要是对每一部分代表的含义不是很了解，直到在网上搜到相关实验说明以及参考往届学长的实验之后，才对ICMP报文格式以及工作过程有了更深的理解。

3.当在浏览器输入百度的URL地址，分析DNS过程和HTTP过程时，遇到的问题是最多的，因为不仅也分析这两种协议，同时还要兼顾IP、ARP、TCP、UDP和以太网等，由于之前对这些协议只是停留在书本知识的简单理解，所以在分析这两个过程一开始，感到无从下手，知道查阅知识以及仔细研究报文后，才逐渐柳暗花明。

### 六、实验心得体会

之前进行的计网实验socket编程主要考察的是编程能力，而此次用wireshark抓包实验，我认为更重要的是你对这些网络协议的理解和吸收。

此次抓包实验的内容可以说是涉及到很多知识，包括应用层、运输层、网络层和链路层的各个方面，并且只是单单理解其中一部分内容是无法完成实验的，它需要将这几部分内容涉及到的知识融合起来，来共同解决一个问题。

通过此次实验，原本对以上网络协议的了解只停留在课本知识层面，用wireshark软件进行抓包后，我可以清晰地看到报文的每一部分的具体格式以及工作过程是如何通过一条条报文来体现的，这不仅加深了我对这些报文格式的理解和工作过程的熟知，同时也使我更深刻地了解不同网络协议的具体作用，使我对知识的了解上升到实践的层次。

虽然收获很多，但是在实验中我也发现了自己的不足之处，首先是对课本知识理解的不够深刻，只是停留在表面，没有往深层次去想网络协议为什么要这样规定，以及他们的作用是如何体现的；其次是个人的实验实践能力还不够，不足以靠个人独立完成实验。在接下来的学习生活中，我将针对以上问题完善自我，努力钻研，动手实践，提高自己。