

# 互联网协议实验报告

张磊 2017K8009922027

## 一、实验题目

互联网协议实验

## 二、实验内容

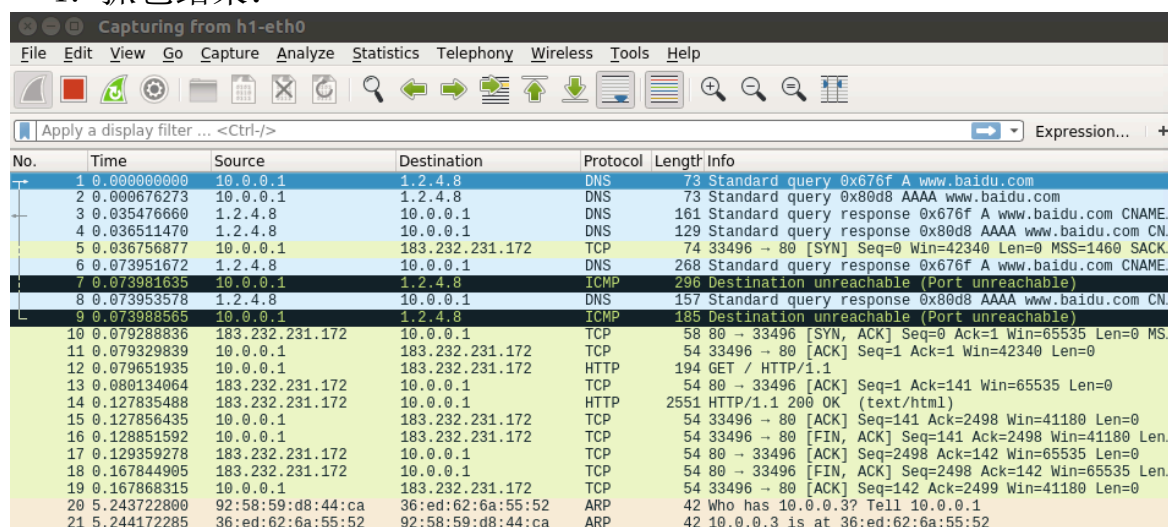
1. 在节点 h1 上开启 Wireshark 抓包，用 wget 下载 [www.baidu.com](http://www.baidu.com) 页面；
2. 调研说明 Wireshark 抓到的几种协议：ARP, DNS, TCP, HTTP；
3. 调研解释 h1 下载 baidu 页面的整个过程：几种协议的运行机制；

## 三、实验流程

1. 终端运行 `sudo mn -nat` 指令，将 hosts 连接到互联网；
2. 启动 mininet 程序后，运行 `xterm h1` 指令，打开控制 h1 的终端；
3. 在 h1 终端中运行 `echo "nameserver 1.2.4.8" > /etc/resolv.conf`；
4. 在 h1 终端中运行 `wireshark &`，启动 wireshark 抓包程序；
5. 在 GUI 界面中选择 h1-eth0，开始抓包；
6. 在 h1 终端中运行 `wget www.baidu.com` 下载百度主页；
7. 调研分析获取到的几种互联网协议；

## 四、实验结果

### 1. 抓包结果：



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.0.1	1.2.4.8	DNS	73	Standard query 0x676f A www.baidu.com
2	0.000676273	10.0.0.1	1.2.4.8	DNS	73	Standard query 0x80d8 AAAA www.baidu.com
3	0.035476660	1.2.4.8	10.0.0.1	DNS	161	Standard query response 0x676f A www.baidu.com CNAME...
4	0.036511470	1.2.4.8	10.0.0.1	DNS	129	Standard query response 0x80d8 AAAA www.baidu.com CN...
5	0.036756877	10.0.0.1	183.232.231.172	TCP	74	33496 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK...
6	0.073951672	1.2.4.8	10.0.0.1	DNS	268	Standard query response 0x676f A www.baidu.com CNAME...
7	0.073981635	10.0.0.1	1.2.4.8	ICMP	296	Destination unreachable (Port unreachable)
8	0.073953578	1.2.4.8	10.0.0.1	DNS	157	Standard query response 0x80d8 AAAA www.baidu.com CN...
9	0.073985565	10.0.0.1	1.2.4.8	ICMP	185	Destination unreachable (Port unreachable)
10	0.079288836	183.232.231.172	10.0.0.1	TCP	58	80 → 33496 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MS...
11	0.079329839	10.0.0.1	183.232.231.172	TCP	54	33496 → 80 [ACK] Seq=1 Ack=1 Win=42340 Len=0
12	0.079651935	10.0.0.1	183.232.231.172	HTTP	194	GET / HTTP/1.1
13	0.080134064	183.232.231.172	10.0.0.1	TCP	54	80 → 33496 [ACK] Seq=1 Ack=141 Win=65535 Len=0
14	0.127835488	183.232.231.172	10.0.0.1	HTTP	2551	HTTP/1.1 200 OK (text/html)
15	0.127856435	10.0.0.1	183.232.231.172	TCP	54	33496 → 80 [ACK] Seq=141 Ack=2498 Win=41180 Len=0
16	0.128851592	10.0.0.1	183.232.231.172	TCP	54	33496 → 80 [FIN, ACK] Seq=141 Ack=2498 Win=41180 Len...
17	0.129359278	183.232.231.172	10.0.0.1	TCP	54	80 → 33496 [ACK] Seq=2498 Ack=142 Win=65535 Len=0
18	0.167844905	183.232.231.172	10.0.0.1	TCP	54	80 → 33496 [FIN, ACK] Seq=2498 Ack=142 Win=65535 Len...
19	0.167868315	10.0.0.1	183.232.231.172	TCP	54	33496 → 80 [ACK] Seq=142 Ack=2499 Win=41180 Len=0
20	5.243722800	92:58:59:d8:44:ca	36:ed:62:6a:55:52	ARP	42	Who has 10.0.0.3? Tell 10.0.0.1
21	5.244172285	36:ed:62:6a:55:52	92:58:59:d8:44:ca	ARP	42	10.0.0.3 is at 36:ed:62:6a:55:52

## 2. ARP 协议层次: Ethernet < ARP

```

20 5.243722800    92:58:59:d8:44:ca    36:ed:62:6a:55:52    ARP    42 Who has 10.0.0.3? Tell 10.0.0.1
21 5.244172285    36:ed:62:6a:55:52    92:58:59:d8:44:ca    ARP    42 10.0.0.3 is at 36:ed:62:6a:55:52

```

---

```

▶ Frame 21: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▶ Ethernet II, Src: 36:ed:62:6a:55:52 (36:ed:62:6a:55:52), Dst: 92:58:59:d8:44:ca (92:58:59:d8:44:ca)
  ▶ Destination: 92:58:59:d8:44:ca (92:58:59:d8:44:ca)
  ▶ Source: 36:ed:62:6a:55:52 (36:ed:62:6a:55:52)
    Type: ARP (0x0806)
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: 36:ed:62:6a:55:52 (36:ed:62:6a:55:52)
  Sender IP address: 10.0.0.3
  Target MAC address: 92:58:59:d8:44:ca (92:58:59:d8:44:ca)
  Target IP address: 10.0.0.1

```

### 3. DNS 协议层次: Ethernet < IP < UDP < DNS

1	0.000000000	10.0.0.1	1.2.4.8	DNS	73 Standard query 0x676f A www.baidu.com
2	0.000676273	10.0.0.1	1.2.4.8	DNS	73 Standard query 0x80d8 AAAA www.baidu.com
3	0.035476660	1.2.4.8	10.0.0.1	DNS	161 Standard query response 0x676f A www.baidu.com CNA...
4	0.036511470	1.2.4.8	10.0.0.1	DNS	129 Standard query response 0x80d8 AAAA www.baidu.com ...
5	0.036756877	10.0.0.1	183.232.231.172	TCP	74 33496 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SA=...
6	0.073951672	1.2.4.8	10.0.0.1	DNS	268 Standard query response 0x676f A www.baidu.com CNA...
7	0.073981635	10.0.0.1	1.2.4.8	ICMP	156 Destination unreachable (Port unreachable)
8	0.073953578	1.2.4.8	10.0.0.1	DNS	157 Standard query response 0x80d8 AAAA www.baidu.com ...
9	0.073988565	10.0.0.1	1.2.4.8	ICMP	185 Destination unreachable (Port unreachable)
10	0.079288836	183.232.231.172	10.0.0.1	TCP	58 80 → 33496 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 ...
11	0.079329839	10.0.0.1	183.232.231.172	TCP	54 33496 → 80 [ACK] Seq=1 Ack=1 Win=42340 Len=0
12	0.079651935	10.0.0.1	183.232.231.172	HTTP	194 GET / HTTP/1.1
13	0.080134064	183.232.231.172	10.0.0.1	TCP	54 80 → 33496 [ACK] Seq=1 Ack=141 Win=65535 Len=0
14	0.127835488	183.232.231.172	10.0.0.1	HTTP	2551 HTTP/1.1 200 OK (text/html)
15	0.127856435	10.0.0.1	183.232.231.172	TCP	54 33496 → 80 [ACK] Seq=141 Ack=2498 Win=41180 Len=0
16	0.128851592	10.0.0.1	183.232.231.172	TCP	54 33496 → 80 [FIN, ACK] Seq=141 Ack=2498 Win=41180 L...
17	0.129359278	183.232.231.172	10.0.0.1	TCP	54 80 → 33496 [ACK] Seq=2498 Ack=142 Win=65535 Len=0

▶ Frame 3: 161 bytes on wire (1288 bits), 161 bytes captured (1288 bits) on interface 0  
 ▶ Ethernet II, Src: 36:ed:62:6a:55:52 (36:ed:62:6a:55:52), Dst: 92:58:59:d8:44:ca (92:58:59:d8:44:ca)  
 ▶ Internet Protocol Version 4, Src: 1.2.4.8, Dst: 10.0.0.1  
 ▶ User Datagram Protocol, Src Port: 53, Dst Port: 35470  
 ▶ Domain Name System (response)

4. TCP 协议层次: Ethernet < IP < TCP

15	0.127856435	10.0.0.1	183.232.231.172	TCP	54	33496 → 80	[ACK] Seq=141 Ack=2498 Win=41180 Len=0
16	0.128851592	10.0.0.1	183.232.231.172	TCP	54	33496 → 80	[FIN, ACK] Seq=141 Ack=2498 Win=41180 Len=0
17	0.129359278	183.232.231.172	10.0.0.1	TCP	54	80 → 33496	[ACK] Seq=2498 Ack=142 Win=65535 Len=0
18	0.167844905	183.232.231.172	10.0.0.1	TCP	54	80 → 33496	[FIN, ACK] Seq=2498 Ack=142 Win=65535 Len=0
19	0.167868315	10.0.0.1	183.232.231.172	TCP	54	33496 → 80	[ACK] Seq=142 Ack=2499 Win=41180 Len=0
20	5.243722800	92:58:59:d8:44:ca	36:ed:62:6a:55:52	ARP	42	Who has 10.0.0.3? Tell 10.0.0.1	

▶ Frame 17: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0  
 ▶ Ethernet II, Src: 36:ed:62:6a:55:52 (36:ed:62:6a:55:52), Dst: 92:58:59:d8:44:ca (92:58:59:d8:44:ca)  
 ▶ Internet Protocol Version 4, Src: 183.232.231.172, Dst: 10.0.0.1  
 ▼ Transmission Control Protocol, Src Port: 80, Dst Port: 33496, Seq: 2498, Ack: 142, Len: 0  
     Source Port: 80  
     Destination Port: 33496  
     [Stream index: 0]  
     [TCP Segment Len: 0]  
     Sequence number: 2498 (relative sequence number)  
     [Next sequence number: 2498 (relative sequence number)]  
     Acknowledgment number: 142 (relative ack number)  
     0101 .... = Header Length: 20 bytes (5)  
     ▶ Flags: 0x010 (ACK)  
     Window size value: 65535  
     [Calculated window size: 65535]  
     [Window size scaling factor: -2 (no window scaling used)]  
     Checksum: 0xb5b8 [unverified]  
     [Checksum Status: Unverified]  
     Urgent pointer: 0  
     ▶ [SEQ/ACK analysis]  
     ▶ [Timestamps]

5. HTTP 协议层次: Ethernet < IP < TCP < HTTP

13	0.080134064	183.232.231.172	10.0.0.1	TCP	54	80 → 33496	[ACK]	Seq=1 Ack=141 Win=65535 Len=0
14	0.127835488	183.232.231.172	10.0.0.1	HTTP	2551	HTTP/1.1 200 OK (text/html)		
15	0.127856435	10.0.0.1	183.232.231.172	TCP	54	33496 → 80	[ACK]	Seq=141 Ack=2498 Win=41180 Len=0
16	0.128851592	10.0.0.1	183.232.231.172	TCP	54	33496 → 80	[FIN, ACK]	Seq=141 Ack=2498 Win=41180 Len=0
17	0.129359278	183.232.231.172	10.0.0.1	TCP	54	80 → 33496	[ACK]	Seq=2498 Ack=142 Win=65535 Len=0
18	0.167844905	183.232.231.172	10.0.0.1	TCP	54	80 → 33496	[FIN, ACK]	Seq=2498 Ack=142 Win=65535 Len=0
19	0.167868315	10.0.0.1	183.232.231.172	TCP	54	33496 → 80	[ACK]	Seq=142 Ack=2499 Win=41180 Len=0
20	5.243722800	92:58:59:d8:44:ca	36:ed:62:6a:55:52	ARP	42	Who has 10.0.0.3? Tell 10.0.0.1		

▶ Frame 14: 2551 bytes on wire (20408 bits), 2551 bytes captured (20408 bits) on interface 0  
 ▶ Ethernet II, Src: 36:ed:62:6a:55:52 (36:ed:62:6a:55:52), Dst: 92:58:59:d8:44:ca (92:58:59:d8:44:ca)  
 ▶ Internet Protocol Version 4, Src: 183.232.231.172, Dst: 10.0.0.1  
 ▶ Transmission Control Protocol, Src Port: 80, Dst Port: 33496, Seq: 1, Ack: 141, Len: 2497  
 ▶ Hypertext Transfer Protocol  
 ▶ Line-based text data: text/html (2 lines)

## 五、实验分析

1. 互联网数据传输过程中在不同层次使用了不同的协议，主要协议有：ARP 协议，DNS 协议，TCP 协议，HTTP 协议；
2. DNS 协议的封装层次：Ethernet < IP < UDP < DNS；
3. HTTP 协议的封装层次：Ethernet < IP < TCP < HTTP；
4. TCP 协议承载 HTTP 协议；

## 六、调研解释

### 1. ARP 协议：

地址解析协议（英语：Address Resolution Protocol，缩写：ARP）是一个通过解析网络层地址来找寻数据链路层地址的网络传输协议。

在以太网协议中规定，同一局域网中的一台主机要和另一台主机进行直接通信，必须要知道目标主机的 MAC 地址。而在 TCP/IP 协议中，网络层和传输层只关心目标主机的 IP 地址。这就导致在以太网中使用 IP 协议时，数据链路层的以太网协议接到上层 IP 协议提供的数据中，只包含目的主机的 IP 地址。于是需要一种方法，根据目的主机的 IP 地址，获得其 MAC 地址。这就是 ARP 协议要做的事情。所谓地址解析（address resolution）就是主机在发送帧前将目标 IP 地址转换成目标 MAC 地址的过程。<sup>i</sup>

### 2. DNS 协议：

域名系统（英语：Domain Name System，缩写：DNS）是互联网的一项服务。它作为将域名和 IP 地址相互映射的一个分布式数据库，能够使人更方便地访问互联网。

举一个例子，zh.wikipedia.org 作为一个域名就和 IP 地址 198.35.26.96 相对应。DNS 就像是一个自动的电话号码簿，我们可以直接拨打 198.35.26.96 的名字 zh.wikipedia.org 来代替电话号码（IP 地址）。DNS 在我们直接调用网站的名字以后就会将像 zh.wikipedia.org 一样便于人类使用的名字转化成像 198.35.26.96 一样便于机器识别的 IP 地址。<sup>ii</sup>

### 3. TCP 协议：

传输控制协议（英语：Transmission Control Protocol，缩写：TCP）是一种面向连接的、可靠的、基于字节流的传输层通信协议。

在因特网协议族（Internet protocol suite）中，TCP 层是位于 IP 层之上，应用层之下的中间层。不同主机的应用层之间经常需要可靠的、像管道一样的连接，但是 IP 层不提供这样的流机制，而是提供不可靠的包交换。

应用层向 TCP 层发送用于网间传输的、用 8 位字节表示的数据流，然后 TCP 把数据流分割成适当长度的报文段（通常受该计算机连接的网络的数据链路层的最大传输单元（MTU）的限制）。之后 TCP 把结果包传给 IP 层，由它来透过网络将包传送给接收端实体的 TCP 层。TCP 为了保证不发生丢包，就给每个包一个序号，同时序号也保证了传送到接收端实体的包的按序接收。然后接收端实体对已成功收到的包发回一个相应的确认信息（ACK）；如果发送端实体在合理的往返时延（RTT）内未收到确认，那么对应的数据包就被假设为已丢失并

进行重传。TCP 用一个校验和函数来检验数据是否有错误，在发送和接收时都要计算校验和。<sup>iii</sup>

#### 4. HTTP 协议：

超文本传输协议（英语：HyperText Transfer Protocol，缩写：HTTP）是一种用于分布式、协作式和超媒体信息系统的应用层协议。HTTP 是万维网的数据通信的基础。

HTTP 是一个客户端（用户）和服务端（网站）之间请求和应答的标准，通常使用 TCP 协议。通常，由 HTTP 客户端发起一个请求，创建一个到服务器指定端口（默认是 80 端口）的 TCP 连接。HTTP 服务器则在那个端口监听客户端的请求。一旦收到请求，服务器会向客户端返回一个状态，比如“HTTP/1.1 200 OK”，以及返回的内容，如请求的文件、错误消息、或者其它信息。<sup>iv</sup>

#### 5. H1 下载 baidu 页面的过程：<sup>v</sup>

- (1) 输入 `wget www.baidu.com` 并回车后，`wget` 会将域名 [www.baidu.com](http://www.baidu.com) 通过 DNS 协议解析为相应的目的服务器 IP 地址；
- (2) 解析获取到目的服务器的 IP 地址后，`wget` 会选择一个大于 1024 的本地端口向目标 IP 地址的 80 端口发起 TCP 连接请求，与目的主机握手成功后，连接建立完成；
- (3) `Wget` 通过向目的服务器 IP 发出 GET 方法报文（HTTP 请求），该 GET 报文通过 TCP > IP (DNS) > MAC (ARP) > 网关 > 目的服务器；
- (4) 目的服务器收到数据帧，通过 IP > TCP > HTTP，目的主机通过 HTTP 协议从请求信息中获得我的主机想要访问的主机名，想要访问的 web 应用，以及想要访问的 web 资源，并按照 HTTP 协议格式将 web 资源封装为 HTML 形式的数据（HTTP 响应）；
- (5) 该 HTML 数据通过 TCP > IP > MAC > 网关 > 我的主机，我的主机收到数据帧，下载完毕；<sup>vi</sup>

## 七、反思总结

1. 虽然本次实验过程和内容都较为简单，几条命令就可以完成，但是却包含了大量的细节内容。比如通过 `wireshark` 抓包获取到的协议的类型，每种协议的内容，用途，以及在互联网协议中的层次，这些内容需要仔细的思考理解才能够掌握；
2. 通过调研 h1 下载 baidu 主页的过程，让我加深了理论课上的互联网数据传输过程的理解，更加清楚的掌握了互联网各个层次的协议的作用，以及数据传输的流程；
3. 调研过程中我发现，第一次打开一个网页时速度较慢，但是之后打开同一个网页速度就明显加快，根据调研结果，可以发现，在将域名通过 DNS 转换为目的主机 IP 时会耗费较多时间，造成打开网页慢的现象，之后主机将域名到 IP 的映射缓存在本地，再次打开就避免了再次解析域名浪费时间；

## 八、 参考文献

---

<sup>i</sup> ARP 协议

<https://zh.wikipedia.org/wiki/%E5%9C%B0%E5%9D%80%E8%A7%A3%E6%9E%90%E5%8D%8F%E8%AE%AE>

<sup>ii</sup> DNS 协议

<https://zh.wikipedia.org/wiki/%E5%9F%9F%E5%90%8D%E7%B3%BB%E7%BB%9F>

<sup>iii</sup> TCP 协议

<https://zh.wikipedia.org/wiki/%E4%BC%A0%E8%BE%93%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE>

<sup>iv</sup> HTTP 协议

<https://zh.wikipedia.org/wiki/%E8%B6%85%E6%96%87%E6%9C%AC%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE>

<sup>v</sup> Baidu 网页的下载过程-1

[https://blog.csdn.net/u012862311/article/details/78753232?depth\\_1-utm\\_source=distribute.pc\\_relevant.none-task-blog-BlogCommendFromBaidu-1&utm\\_source=distribute.pc\\_relevant.none-task-blog-BlogCommendFromBaidu-1](https://blog.csdn.net/u012862311/article/details/78753232?depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-1&utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-1)

<sup>vi</sup> Baidu 网页的下载过程-2

<https://blog.csdn.net/weibo1230123/article/details/82899205>