

# 信息安全原理

3200105872 庄毅非 软工2002

## 一、实验目的：

学习使用wireshark分析计算机进行网络连接时所发送和收到的数据包

实验中使用的系统为macos monterey 12.3.1

## 二、实验步骤

1. 安装wireshark并监听本机网络
2. 分析计算机在连接<http://www.cs.zju.edu.cn>的时候所发送和接收的数据包

## 三、实验过程

3→4: DNS解析

5→8: 3次握手

9→10: http请求

11: 4次挥手

1. 首先，启动wireshark，并使其监听 Wifi: en0
2. 在清空本地dns缓存之后，在浏览器地址栏请求<http://www.cs.zju.edu.cn>
3. 可以看到，在建立tcp连接之前，计算机首先向dns解析服务器询问上述连接服务器的ip地址，第66帧是本地发出的查询请求，第67帧是dns服务器返回的查询结果。

No.	Time	Source	Destination	Protocol	Length	Info
28	1.071295	10.186.94.176	10.10.0.21	DNS	77	Standard query 0x5e47 A www.cs.zju.edu.cn
29	1.081501	10.10.0.21	10.186.94.176	DNS	128	Standard query response 0x5e47 A www.cs.zju.edu.cn A 10.203.4.16 NS dns1.zju.edu.cn A 10.10.0.8

图1 发出的两个dns请求

4. 查看第66帧的查询内容，可以看到为type A，表示查询的dns是一个IPv4地址，之后的class IN表示表示查询范围是在互联网上查询，查询域名为<http://www.cs.zju.edu.cn>。

```
▼ Queries
  > www.cs.zju.edu.cn: type A, class IN
```

图2 28帧查询信息

5. 查看第67帧的dns响应，发现其给出了<http://www.cs.zju.edu.cn>对应的ip地址为10.203.4.16

```
▼ Answers
  ▼ www.cs.zju.edu.cn: type A, class IN, addr 10.203.4.16
    Name: www.cs.zju.edu.cn
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 86400 (1 day)
    Data length: 4
    Address: 10.203.4.16
```

图3 29帧dns查询结果

使用curl -vvv 进行检测，发现所查询的ip地址符合上述结果，并且还给出了连接的端口为80

```
> curl -vvv http://www.cs.zju.edu.cn/
* Trying 10.203.4.16:80...
* Connected to www.cs.zju.edu.cn (10.203.4.16) port 80 (#0)
```

图4 curl查询域名对应的ip地址

5. 在获取了服务器ip地址之后，进行三次握手

30	1.083130	10.186.94.176	10.203.4.16	TCP	78	60020 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=3057608885 TSecr=0 SACK_PERM=1
31	1.098665	10.203.4.16	10.186.94.176	TCP	74	80 → 60020 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1837236634 TSecr=305760885
32	1.098727	10.186.94.176	10.203.4.16	TCP	66	60020 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=3057608893 TSecr=1837236634

图5 三次握手涉及的tcp请求

6. 首先，本机发送一个tcp请求进行第一次握手，分析第一个数据包格式。

可以看到链路层（Ethernet）中包含了本机mac地址（可以使用ifconfig获得）为 14:7d:da:d1:de:69，服务端mac地址 2c:21:72:60:6f:c2，其最后一个字段 Type 提示了服务端请求所使用的网络层协议为IPv4。

在网络层中我们可以看到使用的ip协议为IPv4，请求没有分片，数据包寿命为64，并且告知服务端传输层使用的协议为tcp6。

在传输层我们可以看到传输的目标端口为80，flag为syn，这是建立第一次握手的标识，表示客户端向服务器发送的同步请求。

```

✓ Ethernet II, Src: Apple_d1:de:69 (14:7d:da:d1:de:69), Dst: JuniperN_60:6f:c2 (2c:21:72:60:6f:c2)
  > Destination: JuniperN_60:6f:c2 (2c:21:72:60:6f:c2)
  > Source: Apple_d1:de:69 (14:7d:da:d1:de:69)
  Type: IPv4 (0x0800)
✓ Internet Protocol Version 4, Src: 10.186.94.176, Dst: 10.203.4.16
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 64
  Identification: 0x0000 (0)
  > Flags: 0x40, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 64
  Protocol: TCP (6)
  Header Checksum: 0xc273 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.186.94.176
  Destination Address: 10.203.4.16
✓ Transmission Control Protocol, Src Port: 60020, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 60020
  Destination Port: 80
  [Stream index: 3]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 2495423317
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1011 .... = Header Length: 44 bytes (11)
  > Flags: 0x002 (SYN)
  Window: 65535
  [Calculated window size: 65535]
  Checksum: 0xfcea [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > Options: (24 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP),
  > [Timestamps]

```

图6 tcp第一次握手

7. 之后，客户端向本地回复一个tcp响应。链路层和网络层和上述分析相似。在传输层，我们看到服务器在收到第一次请求之后回复了一个 (ack,syn) 包。

```

> Frame 31: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface en0, id 0
√ Ethernet II, Src: JuniperN_60:6f:c2 (2c:21:72:60:6f:c2), Dst: Apple_d1:de:69 (14:7d:da:d1:de:69)
  > Destination: Apple_d1:de:69 (14:7d:da:d1:de:69)
  > Source: JuniperN_60:6f:c2 (2c:21:72:60:6f:c2)
  Type: IPv4 (0x0800)
√ Internet Protocol Version 4, Src: 10.203.4.16, Dst: 10.186.94.176
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 60
  Identification: 0x0000 (0)
  > Flags: 0x40, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 62
  Protocol: TCP (6)
  Header Checksum: 0xc477 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.203.4.16
  Destination Address: 10.186.94.176
√ Transmission Control Protocol, Src Port: 80, Dst Port: 60020, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 60020
  [Stream index: 3]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 2199593075
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 2495423318
  1010 .... = Header Length: 40 bytes (10)
  > Flags: 0x012 (SYN, ACK)
  Window: 28960
  [Calculated window size: 28960]
  Checksum: 0x8a12 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
  > [Timestamps]
  > [SEQ/ACK analysis]

```

图7 服务端tcp响应

8. 客户端在收到上述响应之后，向服务器回复一个ack包，和客户端之间确认连接。

```

> Frame 32: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface en0, id 0
v Ethernet II, Src: Apple_d1:de:69 (14:7d:da:d1:de:69), Dst: JuniperN_60:6f:c2 (2c:21:72:60:6f:c2)
  > Destination: JuniperN_60:6f:c2 (2c:21:72:60:6f:c2)
  > Source: Apple_d1:de:69 (14:7d:da:d1:de:69)
  Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 10.186.94.176, Dst: 10.203.4.16
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 52
  Identification: 0x0000 (0)
  > Flags: 0x40, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 64
  Protocol: TCP (6)
  Header Checksum: 0xc27f [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.186.94.176
  Destination Address: 10.203.4.16
v Transmission Control Protocol, Src Port: 60020, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
  Source Port: 60020
  Destination Port: 80
  [Stream index: 3]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 2495423318
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 2199593076
  1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x010 (ACK)
  Window: 2058
  [Calculated window size: 131712]
  [Window size scaling factor: 64]
  Checksum: 0x21ed [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]

```

图8 客户端返回的ack包

9. 在上述三个包交换完毕之后，客户端和服务端之间完成tcp连接建立。客户端向服务器正式发出http/GET请求，其应用层为超文本传输协议（http）。Host表示服务器地址，User-Agent就是发送请求的程序（这里使用firefox发送请求），Accept要求响应html文件等格式的文件，Accept-encoding表示接受的编码格式为gzip, Cookie表示请求者身份。

```

v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
  Host: www.cs.zju.edu.cn\r\n
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:98.0) Gecko/20100101 Firefox/98.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-US,en;q=0.5\r\n
  Cache-Control: max-age=0\r\n
  > Cookie: JS[REDACTED]\r\n
  Upgrade-Insecure-Requests: 1\r\n
  \r\n
  [Full request URI: http://www.cs.zju.edu.cn/]
  [HTTP request 1/1]
  [Response in frame: 35]

```

图9 客户端发出的http请求

\_10. 服务器在收到上述请求之后，返回所请求的资源。可以看到返回的状态码为200，表示请求成功。Content-Encoding表示使用的压缩方式为gzip。Content-Length表示响应头空行之后的数据长度，便于客户端进行解压。

```
TCP payload (1005 bytes)
Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
Date: Thu, 07 Apr 2022 10:10:31 GMT\r\n
Server: Apache\r\n
X-Frame-Options: SAMEORIGIN\r\n
Frame-Options: SAMEORIGIN\r\n
Accept-Ranges: bytes\r\n
Vary: Accept-Encoding\r\n
Content-Encoding: gzip\r\n
> Content-Length: 742\r\n
Connection: close\r\n
Content-Type: text/html\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.004540000 seconds]
[Request in frame: 33]
[Request URI: http://www.cs.zju.edu.cn/]
Content-encoded entity body (gzip): 742 bytes -> 1936 bytes
File Data: 1936 bytes
Line-based text data: text/html (52 lines)
<!DOCTYPE html>\r\n
<html>\r\n
<head>\r\n
<meta charset="utf-8" />\r\n
<title>浙江大学计算机科学与技术学院</title>\r\n
<link rel="stylesheet" type="text/css" href="/_upload/tpl/04/b2/1202/template1202/css/entry_page.css"/>\r\n
<script src="/_upload/tpl/04/b2/1202/template1202/js/jquery-1.11.1.min.js" type="text/javascript" charset="utf-8"></script>\r\n
<script src="/_upload/tpl/04/b2/1202/template1202/js/entry_page.js" type="text/javascript" charset="utf-8"></script>\r\n
```

图10 客户端返回html文件

\_11. 随后，服务端发起4次挥手，断开连接，整个请求流程结束。

36	1.095443	10.203.4.16	10.186.94.176	TCP	66	80 -> 60020	[FIN, ACK] Seq=1006 Ack=409 Win=30080 Len=0 TSval=1837236638 TSecr=3057608893
38	1.095557	10.186.94.176	10.203.4.16	TCP	66	60020 -> 80	[ACK] Seq=409 Ack=1007 Win=130752 Len=0 TSval=3057608898 TSecr=1837236638
39	1.095756	10.186.94.176	10.203.4.16	TCP	66	60020 -> 80	[FIN, ACK] Seq=409 Ack=1007 Win=131072 Len=0 TSval=3057608898 TSecr=1837236638
40	1.098587	10.203.4.16	10.186.94.176	TCP	66	80 -> 60020	[ACK] Seq=1007 Ack=410 Win=30080 Len=0 TSval=1837236642 TSecr=3057608898

图11 4次挥手

# 四、 总结

1. 实验中，如果不先清空dns缓存，直接访问<http://www.cs.zju.edu.cn>，那么本机无需向dns服务器发送查询请求，直接访问对应的ip地址，网页加载时间更短。
2. html网页资源请求的流程整体为
  1. 本机查询dns缓存中是否有该域名，如果有，直接转到第2步，否则想dns服务器询问对应的ip地址
  2. 获取到ip地址之后，通过3次交换tcp包，建立tcp连接
  3. 客户端向服务器请求对应的资源（比如html文件）
  4. 服务器返回对应资源
  5. 在资源传输完毕之后，执行4次挥手，断开连接。