计算机网络

期末考查实验报告

（2022学年秋季学期）

教学班级	计科二班	专业（方向）	计算机科学与技术
学号	20337263	姓名	俞泽斌

一、实验题目

重新启动一台配置好上互联网的计算机后，
（1） 运行 Wireshark 软件准备分析上网行为，启动抓取网络数据。
（2） 输入一个 网址 http://www.tsinghua.edu.cn， 一直到整个网页显示。
请记录并分析：
（1）应用层协议的运行的协议。依据网络数据说明相关协议的基本运行情况及其作用。
（2）按实际测试，网页自动跳转运行 https 协议。请分析其应用层与传输层协议的消息交换过程。

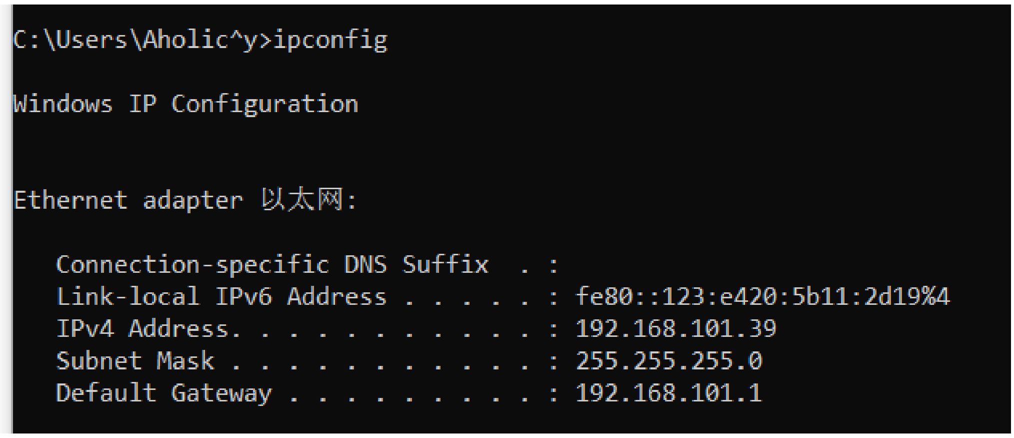
二、实验过程

1、基础设置

一开始其实应该进行DHCP和ARP协议的运行，但是我们的机器已经联网成功，所以本次实验中不涉及，主要也就是通过操作系统生成一个DHCP请求报文，然后封装到UDP报文中，放置在以太网帧中进行广播，最后通过DHCP服务器为本机分配具体的ip地址，可以通过

```
ipconfig
```

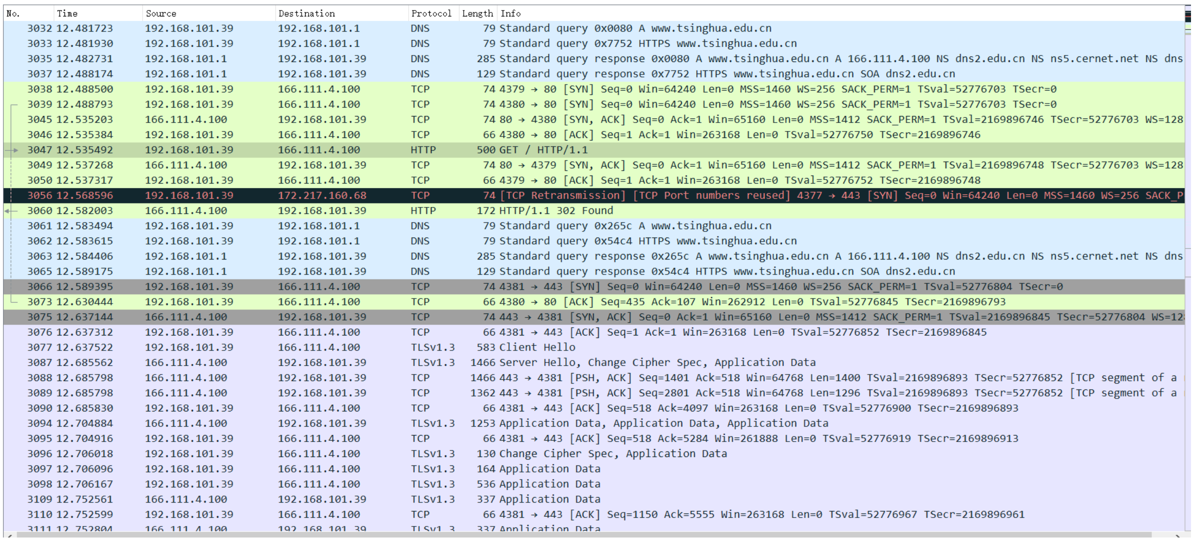
命令来查看我们电脑的ip地址



```
C:\Users\Aholic^y>ipconfig

Windows IP Configuration


Ethernet adapter 以太网:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::123:e420:5b11:2d19%4
   IPv4 Address. . . . . . . . . . . : 192.168.101.39
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.101.1
```

可以看到此时电脑的ip地址为192.168.101.39，即DHCP、ARP等协议已经被执行并记录在了机器中，得到了具体的ip地址。

可以将此时的ip地址作为wireshark的过滤条件，即`ip.addr ==192.168.101.39`并开始抓包

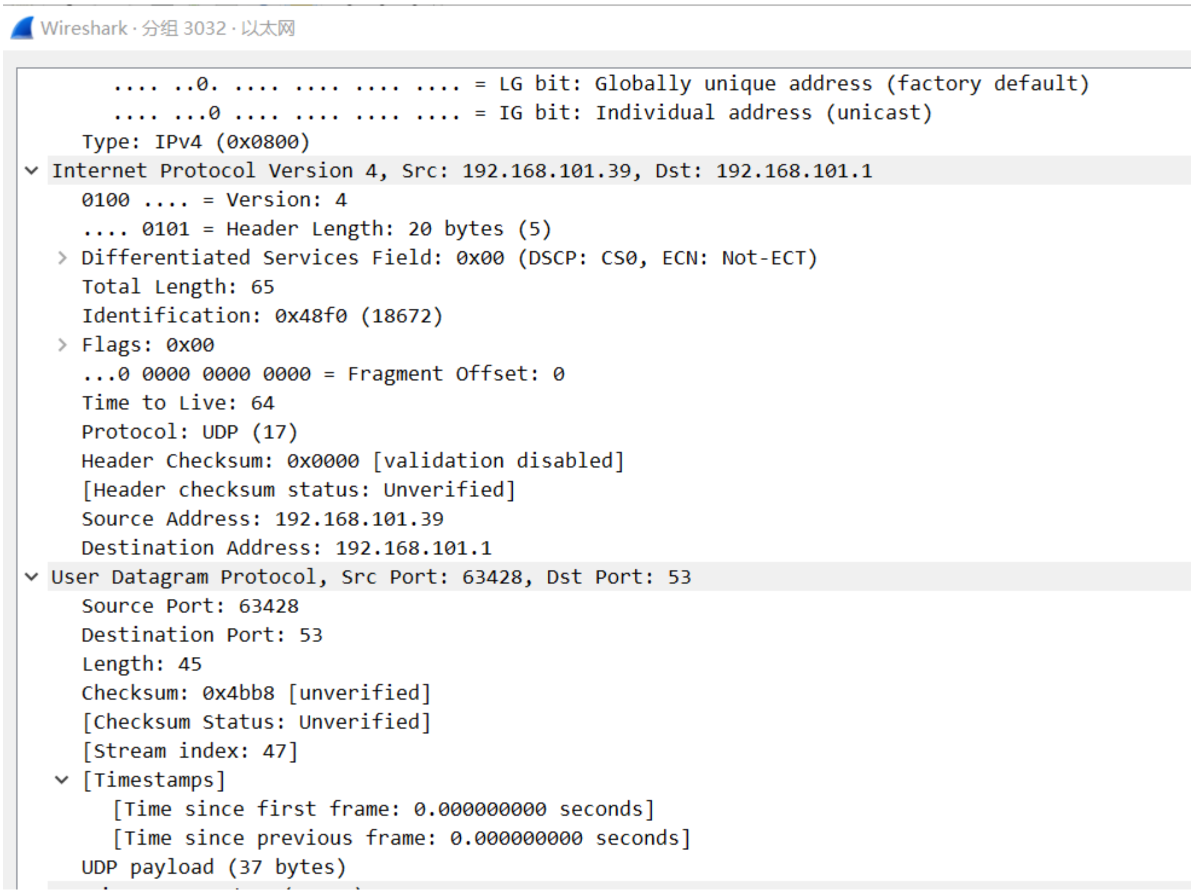


```
No.  Time      Source          Destination     Protocol Length Info
3032 12.481723 192.168.101.39  192.168.101.1   DNS      79 Standard query 0x0080 A www.tsinghua.edu.cn
3033 12.481930 192.168.101.39  192.168.101.1   DNS      79 Standard query 0x7752 HTTPS www.tsinghua.edu.cn
3035 12.482731 192.168.101.1   192.168.101.39  DNS     285 Standard query response 0x0080 A www.tsinghua.edu.cn A 166.111.4.100 NS dns2.edu.cn NS ns5.cernet.net NS dns
3037 12.488174 192.168.101.1   192.168.101.39  DNS     129 Standard query response 0x7752 HTTPS www.tsinghua.edu.cn SOA dns2.edu.cn
3038 12.488500 192.168.101.39  166.111.4.100   TCP      74 4379 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=52776703 TSecr=0
3039 12.488793 192.168.101.39  166.111.4.100   TCP      74 4380 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=52776703 TSecr=0
3045 12.535203 166.111.4.100   192.168.101.39  TCP      74 80 → 4380 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1412 SACK_PERM=1 TSval=2169896746 TSecr=52776703 WS=128
3046 12.535384 192.168.101.39  166.111.4.100   TCP      66 4380 → 80 [ACK] Seq=1 Ack=1 Win=263168 Len=0 TSval=52776750 TSecr=2169896746
3047 12.535492 192.168.101.39  166.111.4.100   HTTP    500 GET / HTTP/1.1
3049 12.537268 166.111.4.100   192.168.101.39  TCP      74 80 → 4379 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1412 SACK_PERM=1 TSval=2169896748 TSecr=52776703 WS=128
3050 12.537317 192.168.101.39  166.111.4.100   TCP      66 4379 → 80 [ACK] Seq=1 Ack=1 Win=263168 Len=0 TSval=52776752 TSecr=2169896748
3056 12.568596 192.168.101.39  172.217.160.68  TCP      74 [TCP Retransmission] [TCP Port numbers reused] 4377 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_P
3060 12.582003 166.111.4.100   192.168.101.39  HTTP    172 HTTP/1.1 302 Found
3061 12.583494 192.168.101.39  192.168.101.1   DNS      79 Standard query 0x265c A www.tsinghua.edu.cn
3062 12.583615 192.168.101.39  192.168.101.1   DNS      79 Standard query 0x54c4 HTTPS www.tsinghua.edu.cn
3063 12.584406 192.168.101.1   192.168.101.39  DNS     285 Standard query response 0x265c A www.tsinghua.edu.cn A 166.111.4.100 NS dns2.edu.cn NS ns5.cernet.net NS dns
3065 12.589175 192.168.101.1   192.168.101.39  DNS     129 Standard query response 0x54c4 HTTPS www.tsinghua.edu.cn SOA dns2.edu.cn
3066 12.589395 192.168.101.39  166.111.4.100   TCP      74 4381 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=52776804 TSecr=0
3073 12.630444 192.168.101.39  166.111.4.100   TCP      66 4380 → 80 [ACK] Seq=435 Ack=107 Win=262912 Len=0 TSval=52776845 TSecr=2169896793
3075 12.637144 166.111.4.100   192.168.101.39  TCP      74 443 → 4381 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1412 SACK_PERM=1 TSval=2169896845 TSecr=52776804 WS=12
3076 12.637312 192.168.101.39  166.111.4.100   TCP      66 4381 → 443 [ACK] Seq=1 Ack=1 Win=263168 Len=0 TSval=52776852 TSecr=2169896845
3077 12.637522 192.168.101.39  166.111.4.100   TLSv1.3 583 Client Hello
3087 12.685562 166.111.4.100   192.168.101.39  TLSv1.3 1466 Server Hello, Change Cipher Spec, Application Data
3088 12.685798 166.111.4.100   192.168.101.39  TCP    1466 443 → 4381 [PSH, ACK] Seq=1401 Ack=518 Win=64768 Len=1400 TSval=2169896893 TSecr=52776852 [TCP segment of a
3089 12.685798 166.111.4.100   192.168.101.39  TCP    1362 443 → 4381 [PSH, ACK] Seq=2801 Ack=518 Win=64768 Len=1296 TSval=2169896893 TSecr=52776852 [TCP segment of a
3090 12.685830 192.168.101.39  166.111.4.100   TCP      66 4381 → 443 [ACK] Seq=518 Ack=4097 Win=263168 Len=0 TSval=52776900 TSecr=2169896893
3094 12.704884 166.111.4.100   192.168.101.39  TLSv1.3 1253 Application Data, Application Data, Application Data
3095 12.704916 192.168.101.39  166.111.4.100   TCP      66 4381 → 443 [ACK] Seq=518 Ack=5284 Win=261888 Len=0 TSval=52776919 TSecr=2169896913
3096 12.706018 192.168.101.39  166.111.4.100   TLSv1.3 130 Change Cipher Spec, Application Data
3097 12.706096 192.168.101.39  166.111.4.100   TLSv1.3 164 Application Data
3098 12.706167 192.168.101.39  166.111.4.100   TLSv1.3 536 Application Data
3109 12.752561 166.111.4.100   192.168.101.39  TLSv1.3 337 Application Data
3110 12.752599 192.168.101.39  166.111.4.100   TCP      66 4381 → 443 [ACK] Seq=1150 Ack=5555 Win=263168 Len=0 TSval=52776967 TSecr=2169896961
3111 12.752804 166.111.4.100   192.168.101.39  TLSv1.3 337 Application Data
```

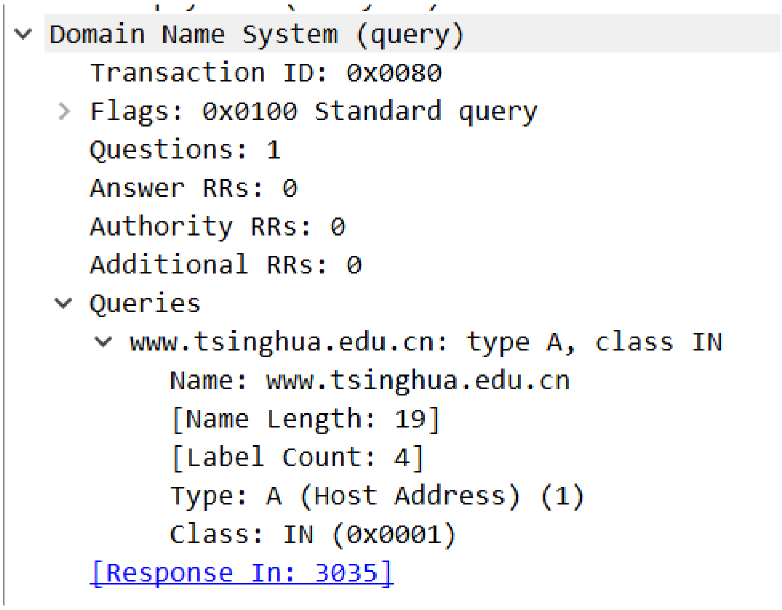
此时得到的数据包的数量是非常庞大的，开始对上述的数据包进行分类的分析

2、DNS分析

首先来看序号为3032-3037的四行DNS协议过程，上面是本机192.168.101.39向本地的DNS服务器发送了两个请求，分别请求此时ipv4地址和ipv6地址，然后本地DNS服务器向本机返回地址，来看具体报文



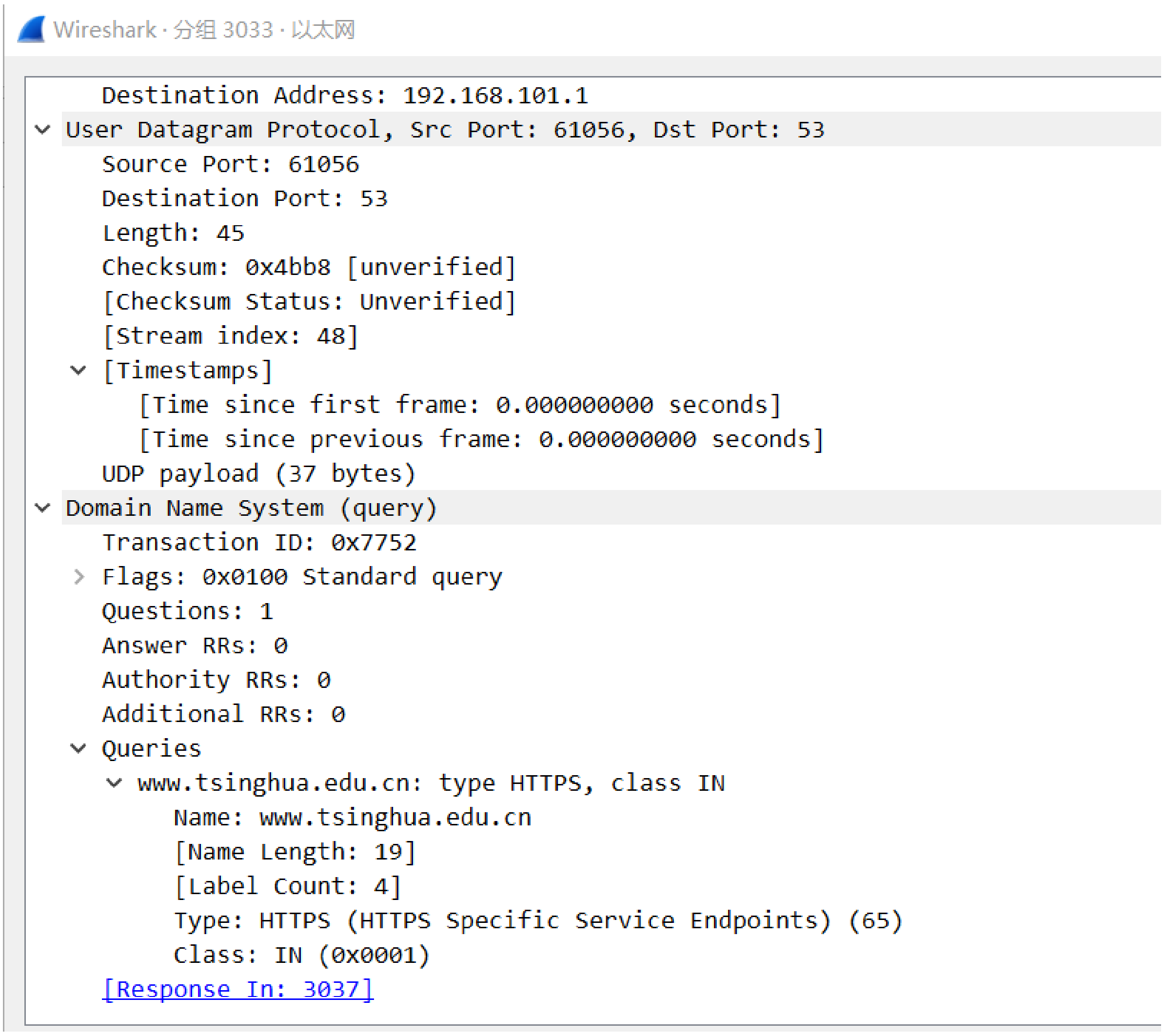
```
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
∨ Internet Protocol Version 4, Src: 192.168.101.39, Dst: 192.168.101.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 65
    Identification: 0x48f0 (18672)
  > Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.101.39
    Destination Address: 192.168.101.1
∨ User Datagram Protocol, Src Port: 63428, Dst Port: 53
    Source Port: 63428
    Destination Port: 53
    Length: 45
    Checksum: 0x4bb8 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 47]
  ∨ [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (37 bytes)
```



```
Domain Name System (query)
    Transaction ID: 0x0080
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
      v www.tsinghua.edu.cn: type A, class IN
            Name: www.tsinghua.edu.cn
            [Name Length: 19]
            [Label Count: 4]
            Type: A (Host Address) (1)
            Class: IN (0x0001)
    [Response In: 3035]
```

上面是第一个请求的报文，采用ipv4的协议，源地址为本机地址192.168.101.39，目标地址为本地DNS服务器地址192.168.101.1，因为DNS协议是运行在UDP协议上的，所以这里的Protocol是UDP，并且我们可以看到DNS 查询的地址即为本地的路由器，这是因为家用的路由器往往承担的不只是路由的功能，还会承担本地的 DNS 服务器，DHCP 服务器等功能。 并且在queries项中也包含了我们所需要查询的域名https://www.tsinghua.edu.cn/

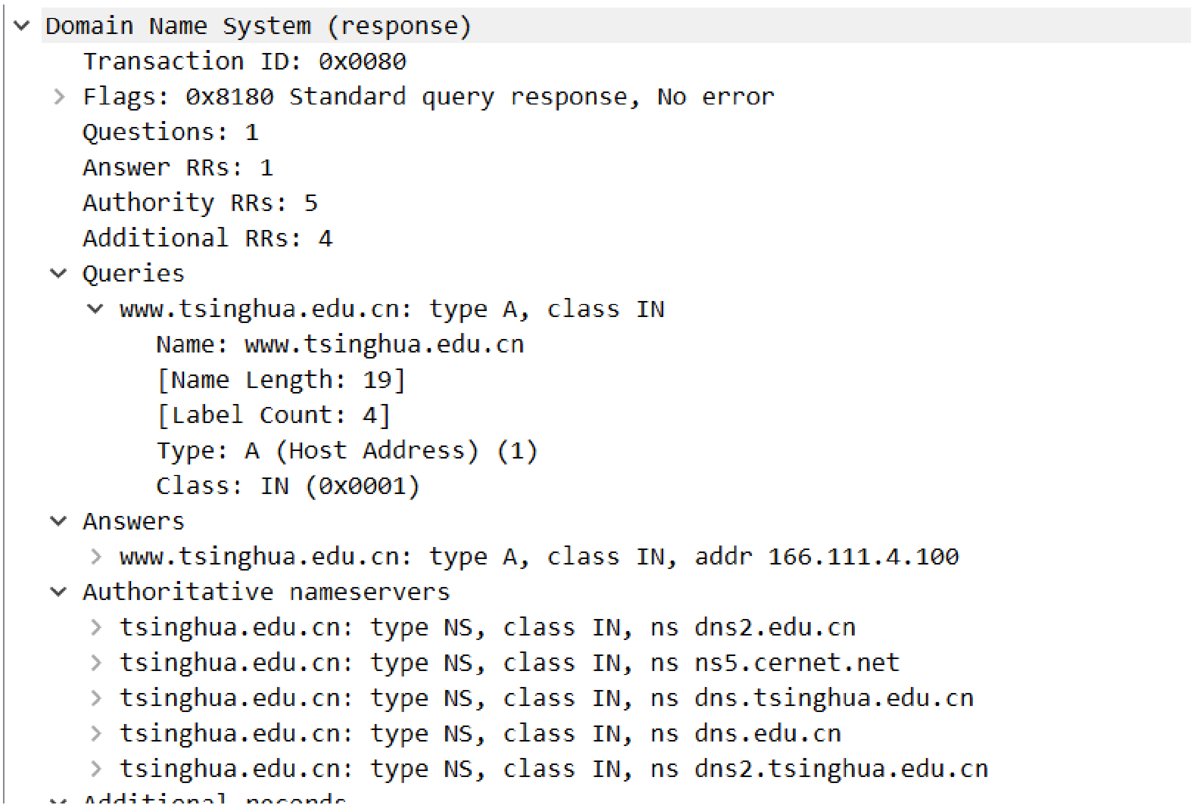
第二个报文如下，基本与上述情况相同，唯一区别是请求的是https的类型，即ipv6地址



Wireshark · 分组 3033 · 以太网

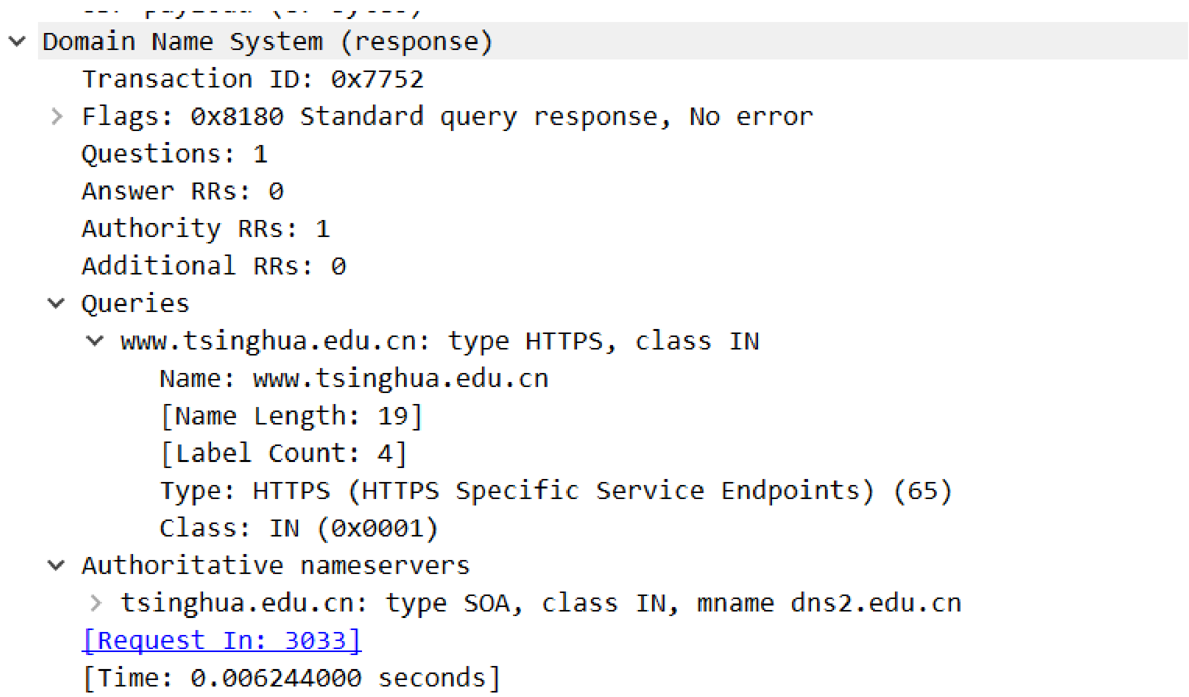
```
    Destination Address: 192.168.101.1
v User Datagram Protocol, Src Port: 61056, Dst Port: 53
    Source Port: 61056
    Destination Port: 53
    Length: 45
    Checksum: 0x4bb8 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 48]
  v [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (37 bytes)
v Domain Name System (query)
    Transaction ID: 0x7752
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
      v www.tsinghua.edu.cn: type HTTPS, class IN
            Name: www.tsinghua.edu.cn
            [Name Length: 19]
            [Label Count: 4]
            Type: HTTPS (HTTPS Specific Service Endpoints) (65)
            Class: IN (0x0001)
    [Response In: 3037]
```

接下来的两个DNS类型的报文即为本地DNS服务器对于上面两个请求报文的响应



```
v Domain Name System (response)
    Transaction ID: 0x0080
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 5
    Additional RRs: 4
  v Queries
    v www.tsinghua.edu.cn: type A, class IN
        Name: www.tsinghua.edu.cn
        [Name Length: 19]
        [Label Count: 4]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  v Answers
    > www.tsinghua.edu.cn: type A, class IN, addr 166.111.4.100
  v Authoritative nameservers
    > tsinghua.edu.cn: type NS, class IN, ns dns2.edu.cn
    > tsinghua.edu.cn: type NS, class IN, ns ns5.cernet.net
    > tsinghua.edu.cn: type NS, class IN, ns dns.tsinghua.edu.cn
    > tsinghua.edu.cn: type NS, class IN, ns dns.edu.cn
    > tsinghua.edu.cn: type NS, class IN, ns dns2.tsinghua.edu.cn
  v Additional records
```

此时可以看到本地DNS服务器对于主机所发送的请求作出了回应，给出了具体的ip地址166.111.4.100



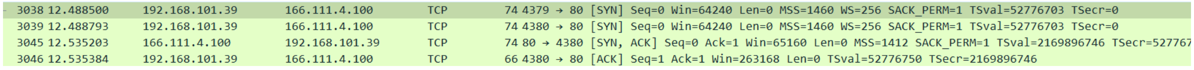
```
v Domain Name System (response)
    Transaction ID: 0x7752
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 0
  v Queries
    v www.tsinghua.edu.cn: type HTTPS, class IN
        Name: www.tsinghua.edu.cn
        [Name Length: 19]
        [Label Count: 4]
        Type: HTTPS (HTTPS Specific Service Endpoints) (65)
        Class: IN (0x0001)
  v Authoritative nameservers
    > tsinghua.edu.cn: type SOA, class IN, mname dns2.edu.cn
    [Request In: 3033]
    [Time: 0.006244000 seconds]
```

第二个响应报文也对第二个请求作出了回应，至此，通过DNS解析域名得到ip地址的部分完成

2、HTTP协议

1）tcp握手

http协议是基于tcp协议的，所以在wireshark中我们看到的首先是tcp协议所得到的数据包，三次握手，具体如下



```
3038 12.488500   192.168.101.39   166.111.4.100    TCP   74 4379 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=52776703 TSecr=0
3039 12.488793   192.168.101.39   166.111.4.100    TCP   74 4380 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=52776703 TSecr=0
3045 12.535203   166.111.4.100    192.168.101.39   TCP   74 80 → 4380 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1412 SACK_PERM=1 TSval=2169896746 TSecr=527767
3046 12.535384   192.168.101.39   166.111.4.100    TCP   66 4380 → 80 [ACK] Seq=1 Ack=1 Win=263168 Len=0 TSval=52776750 TSecr=2169896746
```

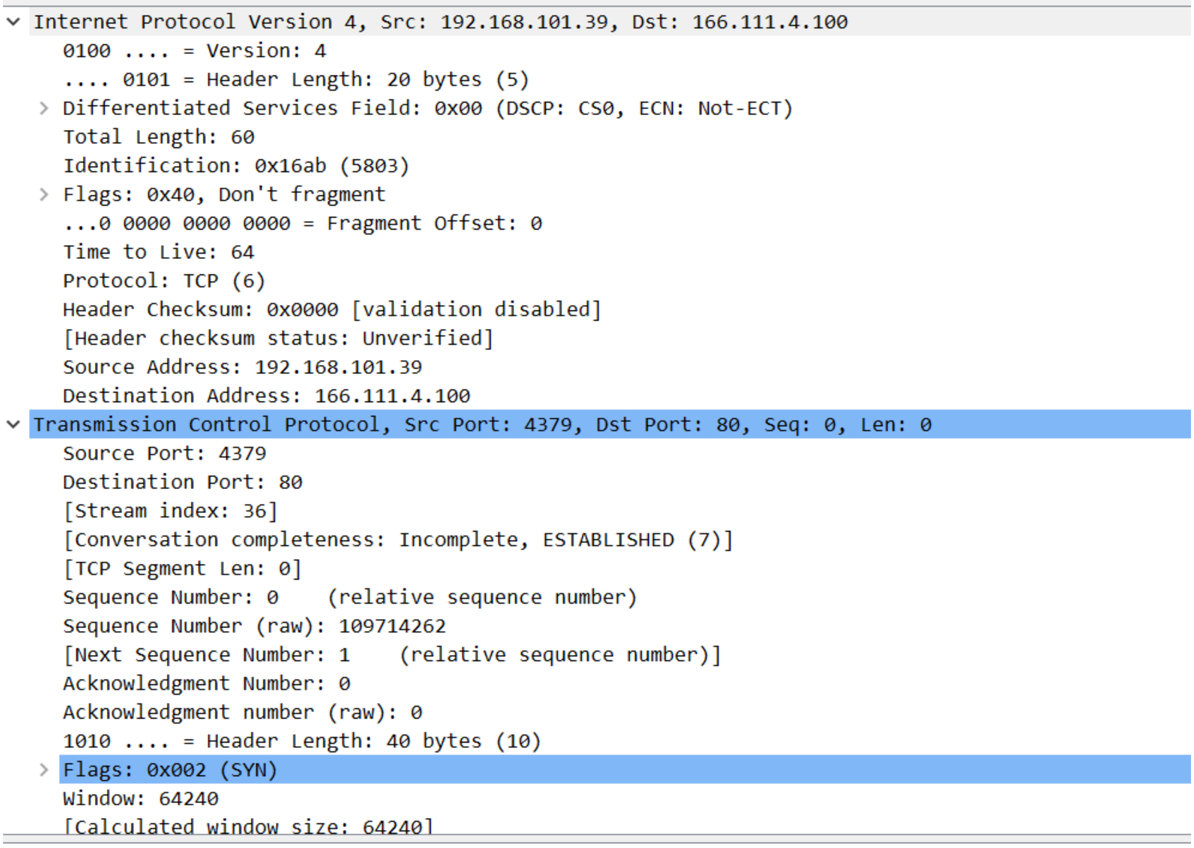
序号为3038-3046，分别为SYN，SYN，SYNACK，ACK类型，与基础的TCP三次握手情况有点区别，基础的TCP三次握手即首先客户端先向服务器端发送一个TCP报文，标记位为SYN，表示“请求建立新连接”；

第二阶段是服务器端收到来自客户端的TCP报文返回一段报文标志位为SYN和ACK，表示“确认客户端的报文Seq序号有效，服务器能正常接收客户端发送的数据，并同意创建新连接”

第三阶段客户端接收到来自服务器确认收到数据的TCP报文后，明确了从客户端到服务器的数据传输是正常的，并返回一段TCP报文，标志位为ACK，表示“确认收到服务器端同意连接的信号”

而这里可能是序号为3038的SYN数据包未收到回应超时，进行了重传， 重传后也符合上面流程

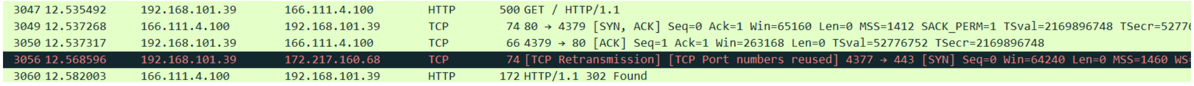
具体可以来看一下第一个报文



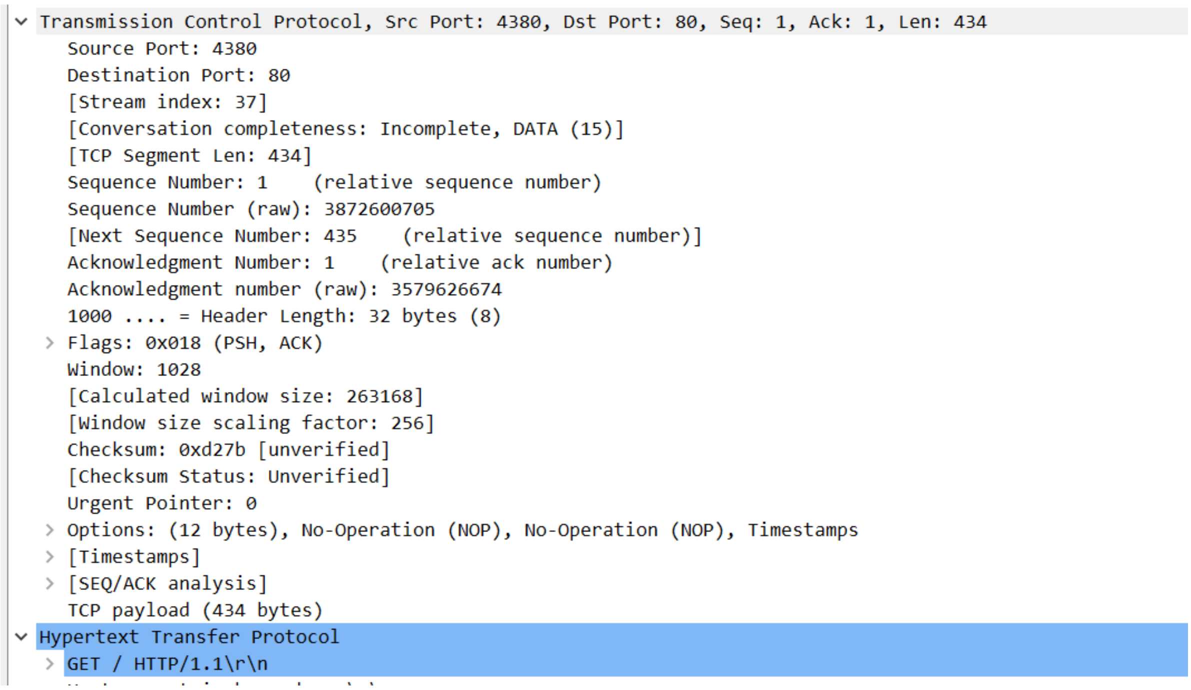
```
∨ Internet Protocol Version 4, Src: 192.168.101.39, Dst: 166.111.4.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0x16ab (5803)
  > Flags: 0x40, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.101.39
    Destination Address: 166.111.4.100
∨ Transmission Control Protocol, Src Port: 4379, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 4379
    Destination Port: 80
    [Stream index: 36]
    [Conversation completeness: Incomplete, ESTABLISHED (7)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 109714262
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1010 .... = Header Length: 40 bytes (10)
  > Flags: 0x002 (SYN)
    Window: 64240
    [Calculated window size: 64240]
```

可以看到此时的协议为TCP，源地址为客户端即本机192.168.101.39，目标地址为166.111.4.100（通过DNS服务器解析出的地址），标志为SYN，端口为80，符合我们上面的叙述。

2）重定向



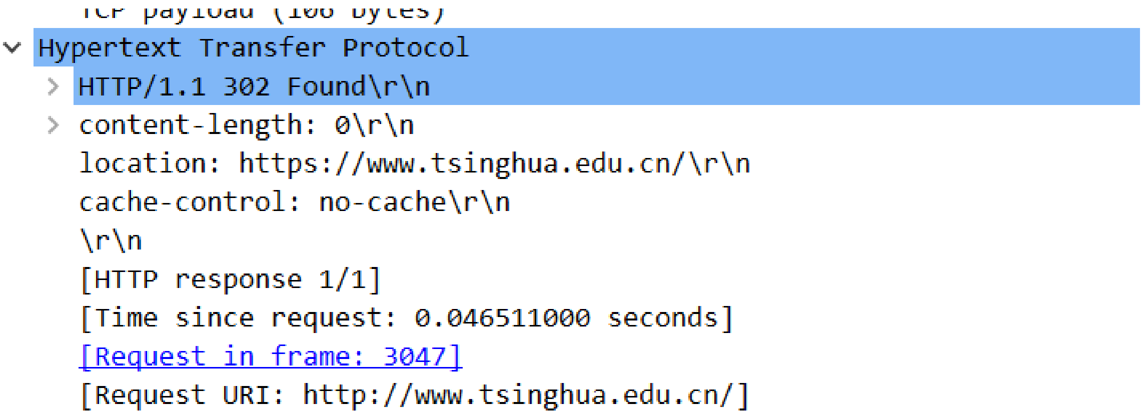
```
3047 12.535492   192.168.101.39   166.111.4.100    HTTP    500 GET / HTTP/1.1
3049 12.537268   166.111.4.100    192.168.101.39   TCP      74 80 → 4379 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1412 SACK_PERM=1 TSval=2169896748 TSecr=5277
3050 12.537317   192.168.101.39   166.111.4.100    TCP      66 4379 → 80 [ACK] Seq=1 Ack=1 Win=263168 Len=0 TSval=52776752 TSecr=2169896748
3056 12.568596   192.168.101.39   172.217.160.68   TCP      74 [TCP Retransmission] [TCP Port numbers reused] 4377 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
3060 12.582003   166.111.4.100    192.168.101.39   HTTP    172 HTTP/1.1 302 Found
```



```
v Transmission Control Protocol, Src Port: 4380, Dst Port: 80, Seq: 1, Ack: 1, Len: 434
    Source Port: 4380
    Destination Port: 80
    [Stream index: 37]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 434]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 3872600705
    [Next Sequence Number: 435    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3579626674
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 1028
    [Calculated window size: 263168]
    [Window size scaling factor: 256]
    Checksum: 0xd27b [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (434 bytes)
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
```

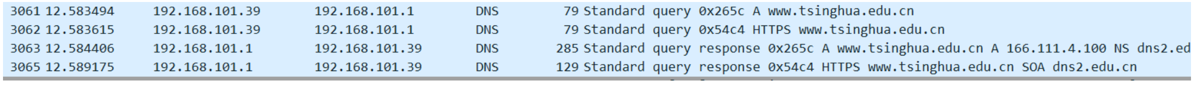
然后开始进入http协议的get请求，请求的地址即为上面所请求的 www.tsinghua.edu.cn ，且从目标 IP 地址为166.111.4.100 即为上面 DNS 查询的 A 记录的响应，传输层的 TCP 协议目标的端口为 80，也是标准的 HTTP 协议的服务端口，同时运行的 HTTP 协议版本为 HTTP 1.1。 发送http请求后另一方也返回了tcp协议的SYN，ACK包，符合TCP的协议要求

对于这个get请求，作出的回应为序号为3060的HTTP报文，响应为302 found，一般意味着重定向，即资源存在，但是位置不再原先位置 ，具体报文如下



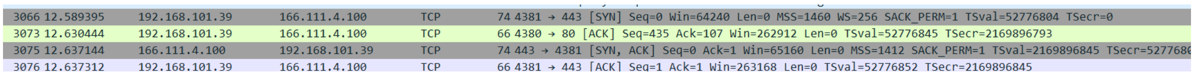
```
v Hypertext Transfer Protocol
  > HTTP/1.1 302 Found\r\n
  > content-length: 0\r\n
    location: https://www.tsinghua.edu.cn/\r\n
    cache-control: no-cache\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.046511000 seconds]
    [Request in frame: 3047]
    [Request URI: http://www.tsinghua.edu.cn/]
```

可以看到源地址http://www.tsinghua.edu.cn/被重定向到https://www.tsinghua.edu.cn/，由原先的http协议转到https协议，接下来本地对于这个重定向的地址进行了再一次的DNS查询，



```
3061 12.583494    192.168.101.39    192.168.101.1     DNS     79 Standard query 0x265c A www.tsinghua.edu.cn
3062 12.583615    192.168.101.39    192.168.101.1     DNS     79 Standard query 0x54c4 HTTPS www.tsinghua.edu.cn
3063 12.584406    192.168.101.1     192.168.101.39    DNS    285 Standard query response 0x265c A www.tsinghua.edu.cn A 166.111.4.100 NS dns2.ed
3065 12.589175    192.168.101.1     192.168.101.39    DNS    129 Standard query response 0x54c4 HTTPS www.tsinghua.edu.cn SOA dns2.edu.cn
```

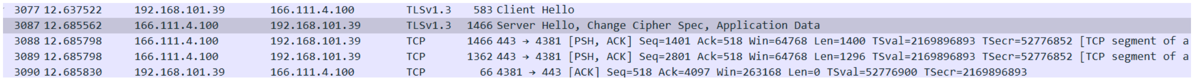
同样的，经过这一次DNS查询后也需要进行tcp的三次握手才能与新的地址建立tcp连接



```
3066 12.589395    192.168.101.39    166.111.4.100     TCP     74 4381 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=52776804 TSecr=0
3073 12.630444    192.168.101.39    166.111.4.100     TCP     66 4380 → 80 [ACK] Seq=435 Ack=107 Win=262912 Len=0 TSval=52776845 TSecr=2169896793
3075 12.637144    166.111.4.100     192.168.101.39    TCP     74 443 → 4381 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1412 SACK_PERM=1 TSval=2169896845 TSecr=527768
3076 12.637312    192.168.101.39    166.111.4.100     TCP     66 4381 → 443 [ACK] Seq=1 Ack=1 Win=263168 Len=0 TSval=52776852 TSecr=2169896845
```

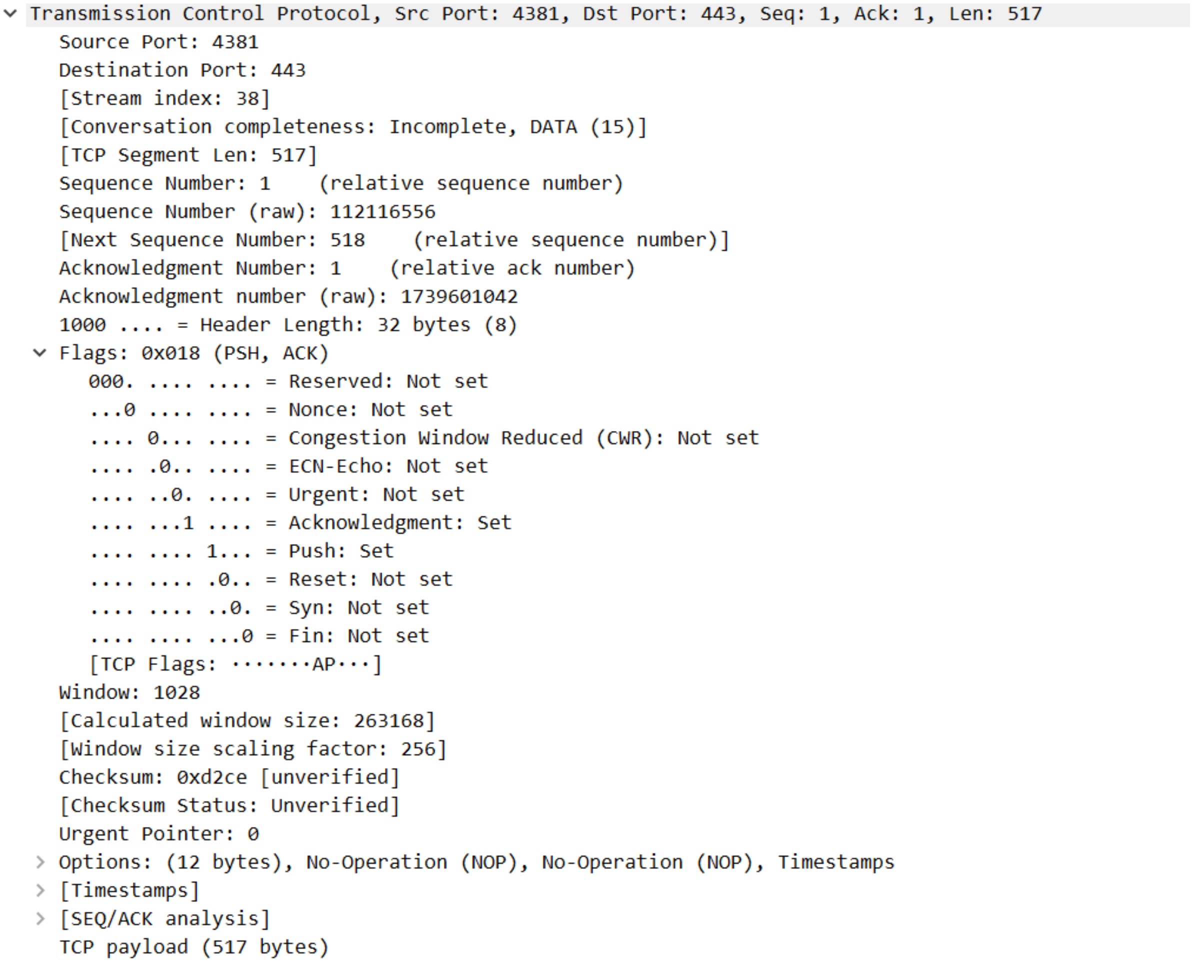
与上面的情况基本相同，就不重新再次进行介绍了，握手成功后，进入到 TLS 握手阶段

3）TLS握手

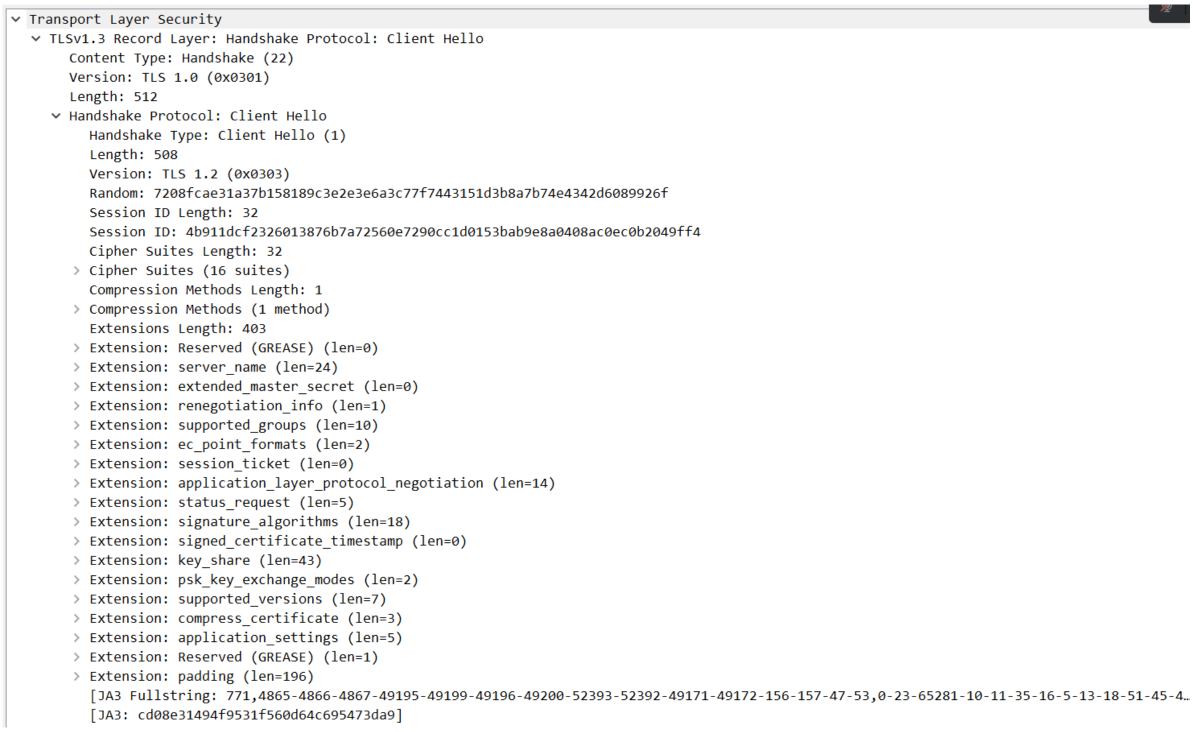


```
3077 12.637522    192.168.101.39    166.111.4.100     TLSv1.3   583 Client Hello
3087 12.685562    166.111.4.100     192.168.101.39    TLSv1.3  1466 Server Hello, Change Cipher Spec, Application Data
3088 12.685798    166.111.4.100     192.168.101.39    TCP      1466 443 → 4381 [PSH, ACK] Seq=1401 Ack=518 Win=64768 Len=1400 TSval=2169896893 TSecr=52776852 [TCP segment of a
3089 12.685798    166.111.4.100     192.168.101.39    TCP      1362 443 → 4381 [PSH, ACK] Seq=2801 Ack=518 Win=64768 Len=1296 TSval=2169896893 TSecr=52776852 [TCP segment of a
3090 12.685830    192.168.101.39    166.111.4.100     TCP        66 4381 → 443 [ACK] Seq=518 Ack=4097 Win=263168 Len=0 TSval=52776900 TSecr=2169896893
```

具体来看第一条的报文

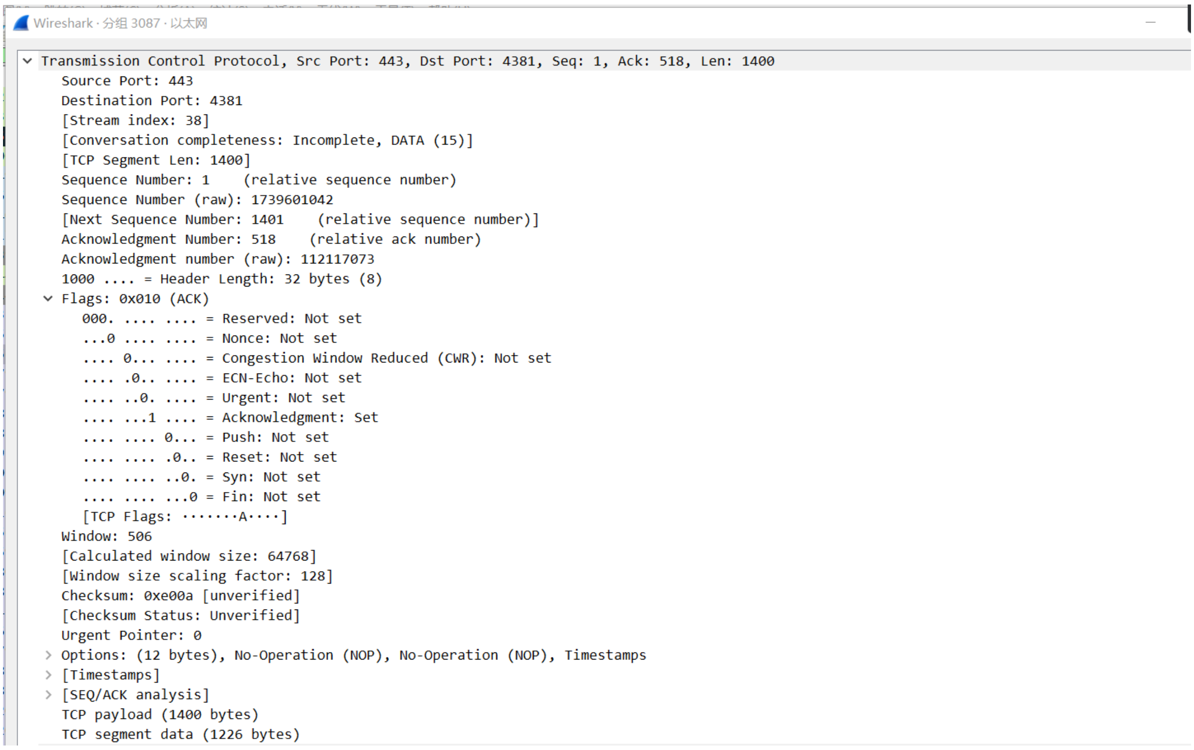


```
∨ Transmission Control Protocol, Src Port: 4381, Dst Port: 443, Seq: 1, Ack: 1, Len: 517
    Source Port: 4381
    Destination Port: 443
    [Stream index: 38]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 517]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 112116556
    [Next Sequence Number: 518    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 1739601042
    1000 .... = Header Length: 32 bytes (8)
  ∨ Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 1... = Push: Set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······AP···]
    Window: 1028
    [Calculated window size: 263168]
    [Window size scaling factor: 256]
    Checksum: 0xd2ce [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (517 bytes)
```



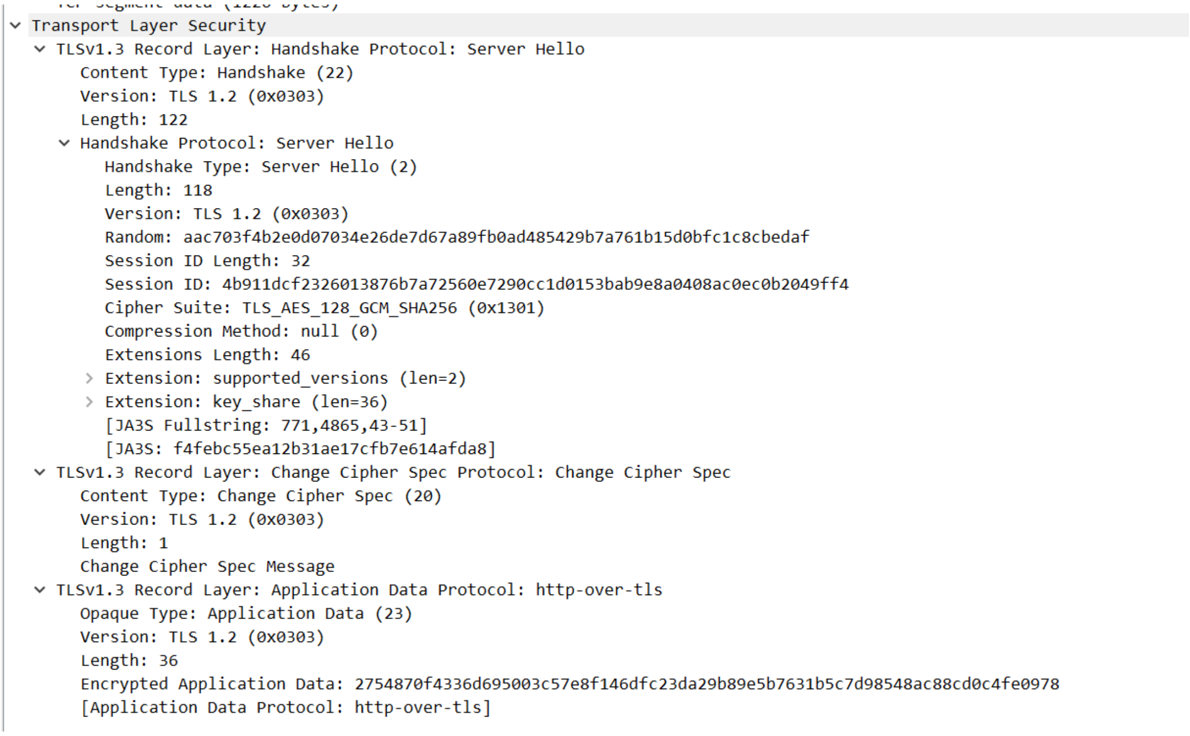
```
∨ Transport Layer Security
  ∨ TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 512
    ∨ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 508
        Version: TLS 1.2 (0x0303)
        Random: 7208fcae31a37b158189c3e2e3e6a3c77f7443151d3b8a7b74e4342d6089926f
        Session ID Length: 32
        Session ID: 4b911dcf2326013876b7a72560e7290cc1d0153bab9e8a0408ac0ec0b2049ff4
        Cipher Suites Length: 32
      > Cipher Suites (16 suites)
        Compression Methods Length: 1
      > Compression Methods (1 method)
        Extensions Length: 403
      > Extension: Reserved (GREASE) (len=0)
      > Extension: server_name (len=24)
      > Extension: extended_master_secret (len=0)
      > Extension: renegotiation_info (len=1)
      > Extension: supported_groups (len=10)
      > Extension: ec_point_formats (len=2)
      > Extension: session_ticket (len=0)
      > Extension: application_layer_protocol_negotiation (len=14)
      > Extension: status_request (len=5)
      > Extension: signature_algorithms (len=18)
      > Extension: signed_certificate_timestamp (len=0)
      > Extension: key_share (len=43)
      > Extension: psk_key_exchange_modes (len=2)
      > Extension: supported_versions (len=7)
      > Extension: compress_certificate (len=3)
      > Extension: application_settings (len=5)
      > Extension: Reserved (GREASE) (len=1)
      > Extension: padding (len=196)
        [JA3 Fullstring: 771,4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53,0-23-65281-10-11-35-16-5-13-18-51-45-4…
        [JA3: cd08e31494f9531f560d64c695473da9]
```

可以看到，传输层协议下，发送端口仍为4381，目标端口不再是之前的 80 端口，而是变更为了https协议的默认端口443 ，同时报文中多出了一项为transport layer security 部分，表明加密协议为TLSv1.3，握手协议，客户端的hello，然后把客户端所支持的版本TLS发送过来，并给出了session ID，即本地将自己支持的加密方法，支持的 TLS 版本以及其余参数打包成报文发送到服务端。服务端响应如下



Wireshark · 分组 3087 · 以太网

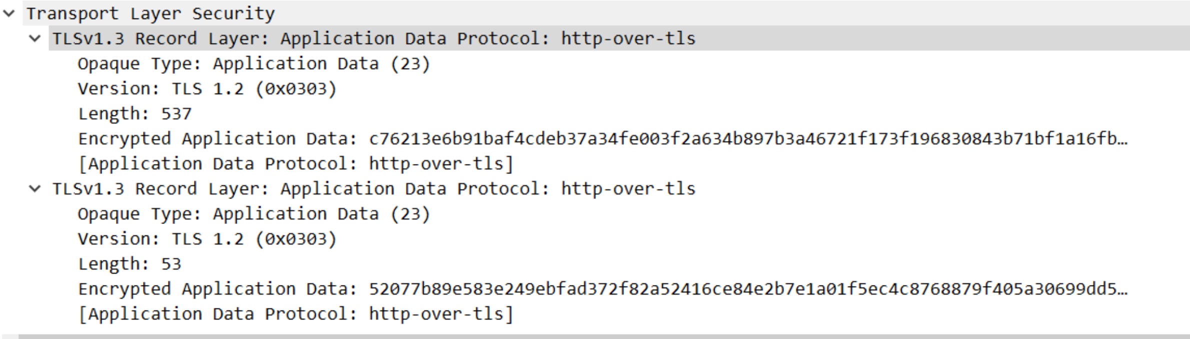
```
⌄ Transmission Control Protocol, Src Port: 443, Dst Port: 4381, Seq: 1, Ack: 518, Len: 1400
    Source Port: 443
    Destination Port: 4381
    [Stream index: 38]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 1400]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 1739601042
    [Next Sequence Number: 1401    (relative sequence number)]
    Acknowledgment Number: 518    (relative ack number)
    Acknowledgment number (raw): 112117073
    1000 .... = Header Length: 32 bytes (8)
  ⌄ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······A····]
    Window: 506
    [Calculated window size: 64768]
    [Window size scaling factor: 128]
    Checksum: 0xe00a [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  › Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  › [Timestamps]
  › [SEQ/ACK analysis]
    TCP payload (1400 bytes)
    TCP segment data (1226 bytes)
```



```
⌄ Transport Layer Security
  ⌄ TLSv1.3 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 122
    ⌄ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 118
        Version: TLS 1.2 (0x0303)
        Random: aac703f4b2e0d07034e26de7d67a89fb0ad485429b7a761b15d0bfc1c8cbedaf
        Session ID Length: 32
        Session ID: 4b911dcf2326013876b7a72560e7290cc1d0153bab9e8a0408ac0ec0b2049ff4
        Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
        Compression Method: null (0)
        Extensions Length: 46
      › Extension: supported_versions (len=2)
      › Extension: key_share (len=36)
        [JA3S Fullstring: 771,4865,43-51]
        [JA3S: f4febc55ea12b31ae17cfb7e614afda8]
  ⌄ TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.2 (0x0303)
      Length: 1
      Change Cipher Spec Message
  ⌄ TLSv1.3 Record Layer: Application Data Protocol: http-over-tls
      Opaque Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 36
      Encrypted Application Data: 2754870f4336d695003c57e8f146dfc23da29b89e5b7631b5c7d98548ac88cd0c4fe0978
      [Application Data Protocol: http-over-tls]
```

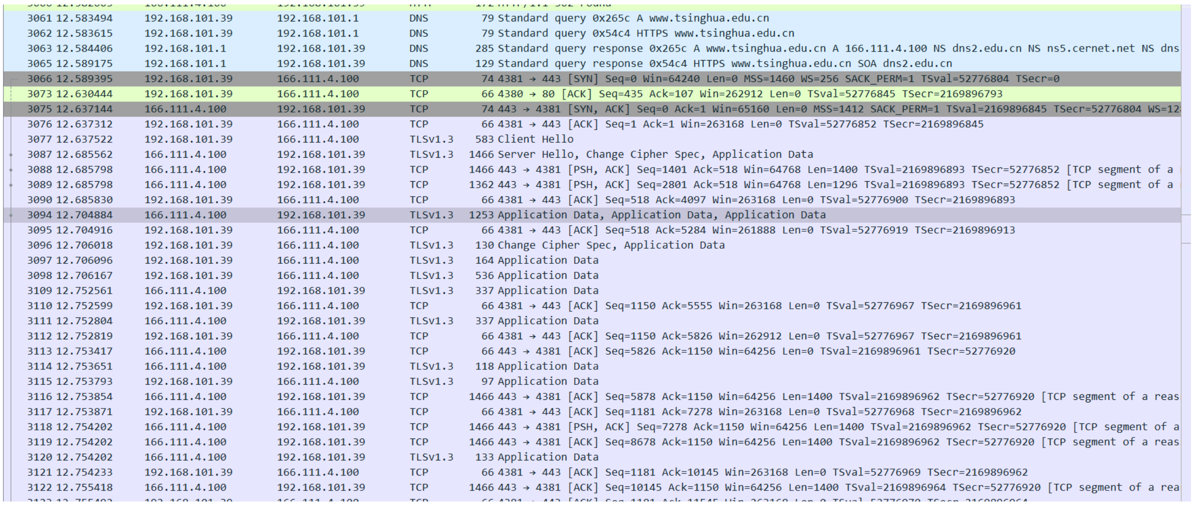
传输层协议下，发送端口为443，目标端口为4381，此时服务器在收到客户端的请求后，返回Server Hello，其中包括服务端的TLS版本，具体的session ID还有Cipher Suite（表示后面建立的加密通信），TLS 的握手到此就结束了，之后便是通过这个建立的加密连接进行通信

之后所得到的报文，都是Application Data形式的加密通信，无法进行解码了，保护了数据传输的安全性



```
Transport Layer Security
  TLSv1.3 Record Layer: Application Data Protocol: http-over-tls
    Opaque Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 537
    Encrypted Application Data: c76213e6b91baf4cdeb37a34fe003f2a634b897b3a46721f173f196830843b71bf1a16fb…
    [Application Data Protocol: http-over-tls]
  TLSv1.3 Record Layer: Application Data Protocol: http-over-tls
    Opaque Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 53
    Encrypted Application Data: 52077b89e583e249ebfad372f82a52416ce84e2b7e1a01f5ec4c8768879f405a30699dd5…
    [Application Data Protocol: http-over-tls]
```

之后便是通过这个建立的加密连接进行通信



```
3061 12.583494  192.168.101.39  192.168.101.1   DNS       79 Standard query 0x265c A www.tsinghua.edu.cn
3062 12.583615  192.168.101.39  192.168.101.1   DNS       79 Standard query 0x54c4 HTTPS www.tsinghua.edu.cn
3063 12.584406  192.168.101.1   192.168.101.39  DNS      285 Standard query response 0x265c A www.tsinghua.edu.cn A 166.111.4.100 NS dns2.edu.cn NS ns5.cernet.net NS dns
3065 12.589175  192.168.101.1   192.168.101.39  DNS      129 Standard query response 0x54c4 HTTPS www.tsinghua.edu.cn SOA dns2.edu.cn
3066 12.589395  192.168.101.39  166.111.4.100   TCP       74 4381 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=52776804 TSecr=0
3073 12.630444  192.168.101.39  166.111.4.100   TCP       66 4380 → 80 [ACK] Seq=435 Ack=107 Win=262912 Len=0 TSval=52776845 TSecr=2169896793
3075 12.637144  166.111.4.100   192.168.101.39  TCP       74 443 → 4381 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1412 SACK_PERM=1 TSval=2169896845 TSecr=52776804 WS=12
3076 12.637312  192.168.101.39  166.111.4.100   TCP       66 4381 → 443 [ACK] Seq=1 Ack=1 Win=263168 Len=0 TSval=52776852 TSecr=2169896845
3077 12.637522  192.168.101.39  166.111.4.100   TLSv1.3  583 Client Hello
3087 12.685562  166.111.4.100   192.168.101.39  TLSv1.3 1466 Server Hello, Change Cipher Spec, Application Data
3088 12.685798  166.111.4.100   192.168.101.39  TCP     1466 443 → 4381 [PSH, ACK] Seq=1401 Ack=518 Win=64768 Len=1400 TSval=2169896893 TSecr=52776852 [TCP segment of a
3089 12.685798  166.111.4.100   192.168.101.39  TCP     1362 443 → 4381 [PSH, ACK] Seq=2801 Ack=518 Win=64768 Len=1296 TSval=2169896893 TSecr=52776852 [TCP segment of a
3090 12.685830  192.168.101.39  166.111.4.100   TCP       66 4381 → 443 [ACK] Seq=518 Ack=4097 Win=263168 Len=0 TSval=52776900 TSecr=2169896893
3094 12.704884  166.111.4.100   192.168.101.39  TLSv1.3 1253 Application Data, Application Data, Application Data
3095 12.704916  192.168.101.39  166.111.4.100   TCP       66 4381 → 443 [ACK] Seq=518 Ack=5284 Win=261888 Len=0 TSval=52776919 TSecr=2169896913
3096 12.706018  192.168.101.39  166.111.4.100   TLSv1.3  130 Change Cipher Spec, Application Data
3097 12.706096  192.168.101.39  166.111.4.100   TLSv1.3  164 Application Data
3098 12.706167  192.168.101.39  166.111.4.100   TLSv1.3  536 Application Data
3109 12.752561  166.111.4.100   192.168.101.39  TLSv1.3  337 Application Data
3110 12.752599  192.168.101.39  166.111.4.100   TCP       66 4381 → 443 [ACK] Seq=1150 Ack=5555 Win=263168 Len=0 TSval=52776967 TSecr=2169896961
3111 12.752804  166.111.4.100   192.168.101.39  TLSv1.3  337 Application Data
3112 12.752819  192.168.101.39  166.111.4.100   TCP       66 4381 → 443 [ACK] Seq=1150 Ack=5826 Win=262912 Len=0 TSval=52776967 TSecr=2169896961
3113 12.753417  166.111.4.100   192.168.101.39  TCP       66 443 → 4381 [ACK] Seq=5826 Ack=1150 Win=64256 Len=0 TSval=2169896961 TSecr=52776920
3114 12.753651  166.111.4.100   192.168.101.39  TLSv1.3  118 Application Data
3115 12.753793  192.168.101.39  166.111.4.100   TLSv1.3   97 Application Data
3116 12.753854  166.111.4.100   192.168.101.39  TCP     1466 443 → 4381 [ACK] Seq=5878 Ack=1150 Win=64256 Len=1400 TSval=2169896962 TSecr=52776920 [TCP segment of a reas
3117 12.753871  192.168.101.39  166.111.4.100   TCP       66 4381 → 443 [ACK] Seq=1181 Ack=7278 Win=263168 Len=0 TSval=52776968 TSecr=2169896962
3118 12.754202  166.111.4.100   192.168.101.39  TCP     1466 443 → 4381 [PSH, ACK] Seq=7278 Ack=1150 Win=64256 Len=1400 TSval=2169896962 TSecr=52776920 [TCP segment of a
3119 12.754202  166.111.4.100   192.168.101.39  TCP     1466 443 → 4381 [ACK] Seq=8678 Ack=1150 Win=64256 Len=1400 TSval=2169896962 TSecr=52776920 [TCP segment of a reas
3120 12.754202  166.111.4.100   192.168.101.39  TLSv1.3  133 Application Data
3121 12.754233  192.168.101.39  166.111.4.100   TCP       66 4381 → 443 [ACK] Seq=1181 Ack=10145 Win=263168 Len=0 TSval=52776969 TSecr=2169896962
3122 12.755418  166.111.4.100   192.168.101.39  TCP     1466 443 → 4381 [ACK] Seq=10145 Ack=1150 Win=64256 Len=1400 TSval=2169896964 TSecr=52776920 [TCP segment of a rea
```

其中未加密的tcp协议都是发送的ACK包，只是一个响应，不涉及具体的数据，同时数据包也能看到 TCP ，便于流水线以及拥塞控制，所以ACK不需要进行加密操作，就如上图所示。

三、总结

综上，一台已经联网的计算机上 http://www.tsinghua.edu.cn的网页，主要涉及以下几方面的交互

首先是本地客户端和DNS服务器（本地路由器）之间：主机发送DNS的两次查询，分别查询ipv4和ipv6的ip地址，DNS服务器发送两次响应包含对应的ip地址

然后是本地客户端和网页服务端166.111.4.100之间：

1、tcp的三次握手

2、http get请求　 http 302found重定向

3、tcp三次握手

4、TLS握手

5、加密通信数据

具体的已经在第二部分做了具体阐述，至此实验结束。