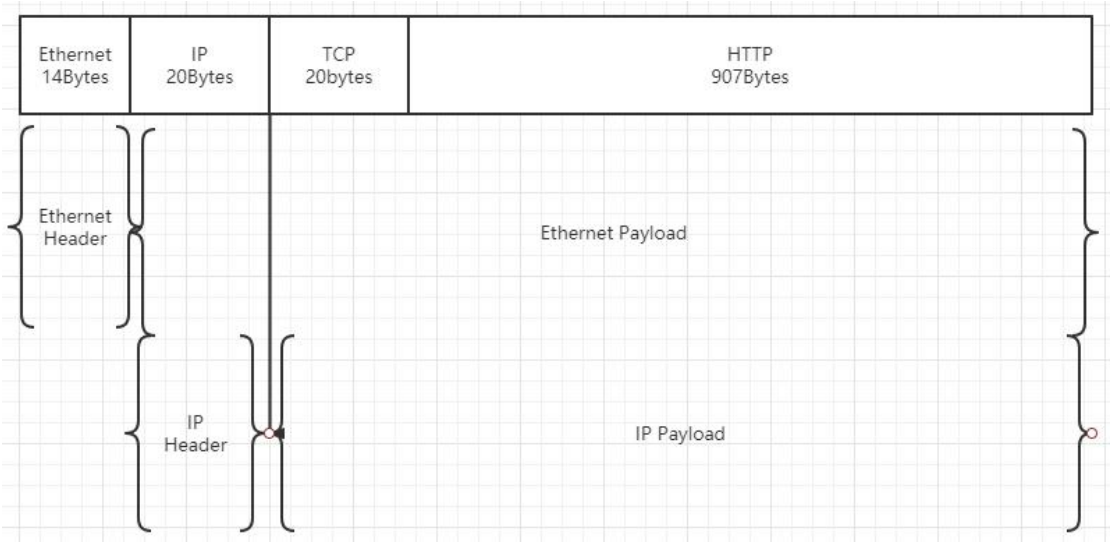
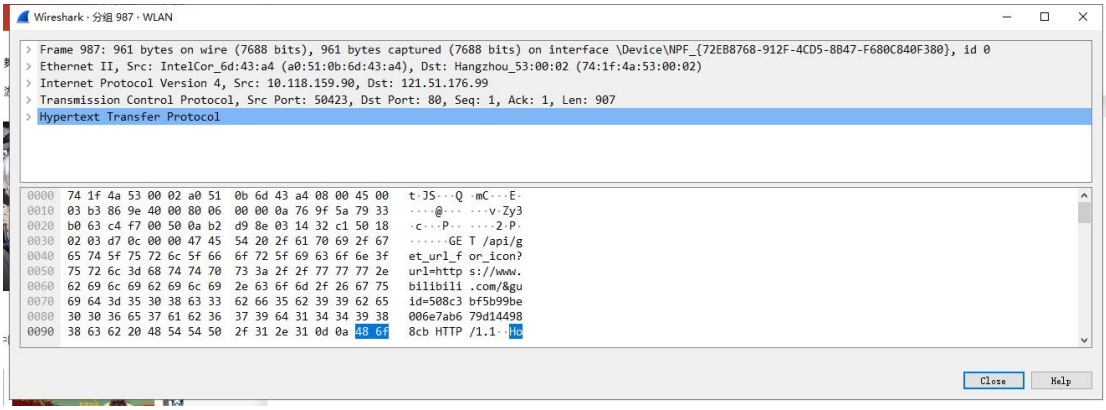
计网实验 1

学号：19020011038　　姓名：岳宇轩

实验环境：Windows　　指导教师：洪峰

实验结果：





Source Address 指示的是我的 IP 地址，地址为 10.118.159.10

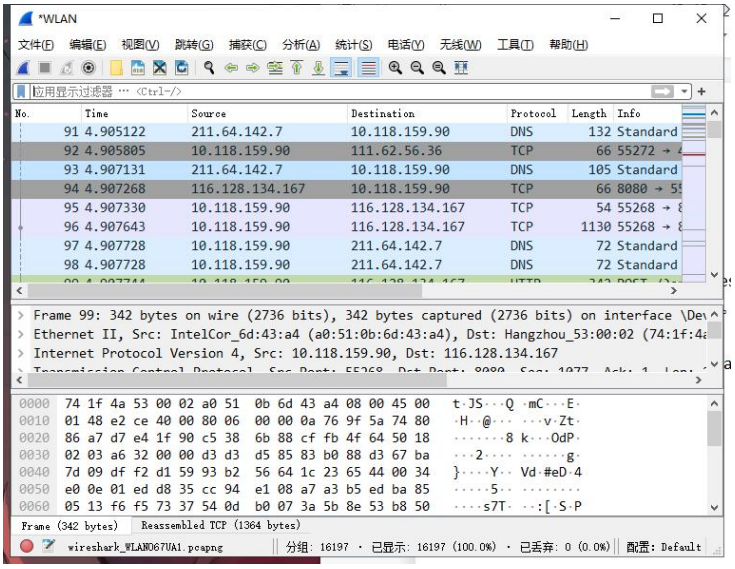
应用层的协议名称是 Hypertext Transfer Protocol，缩写为 HTTP，中文称超文本传输协议。

根据 Host: vedio.browser.qq.com 推测 wireshark 捕获的这个包是 QQ 浏览器这个应用程序发送的。

具体过程如下：

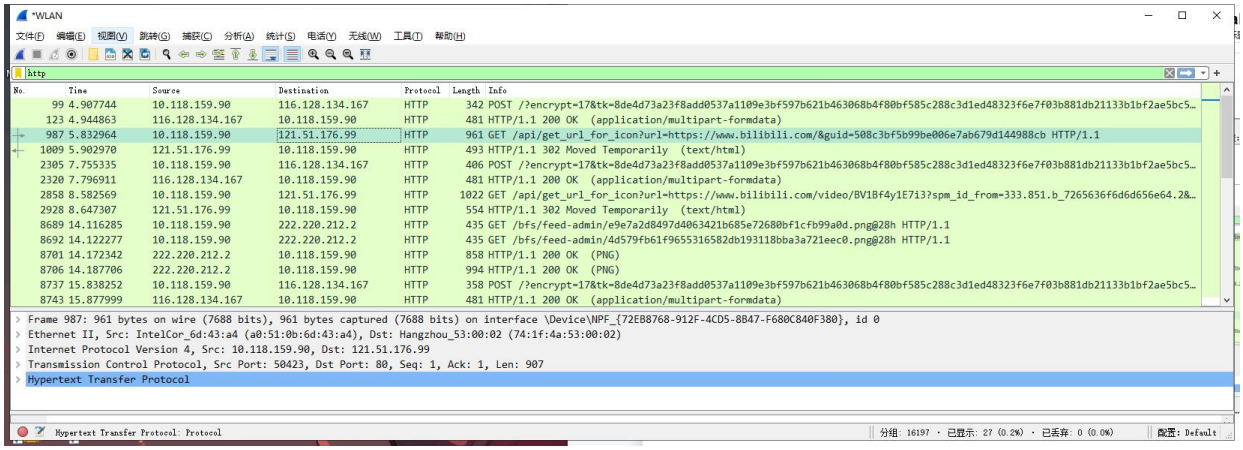
第一步：

下载 wireshark，打开 wireshark，在 capture options 中取消勾选 capture packets in promiscuous mode。点击开始捕获，随便打开一个网页进行浏览，然后停止捕获。



第二步：

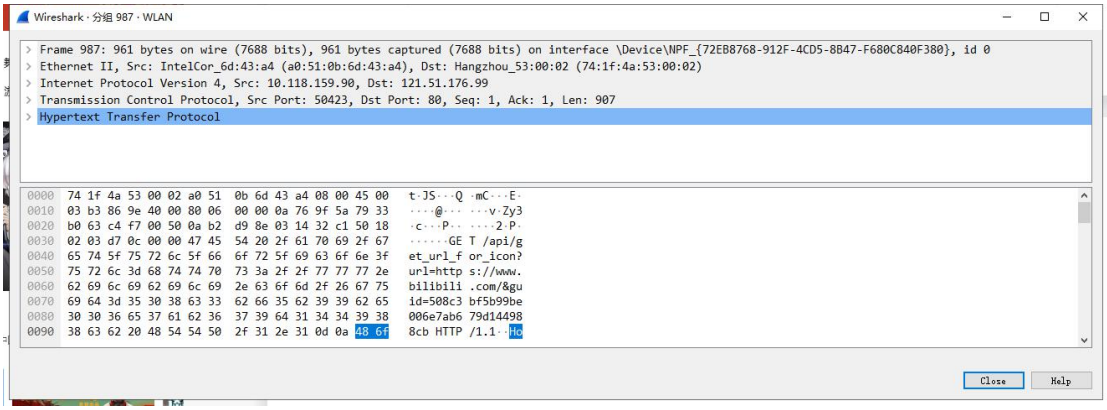
在 wireshark 中筛选出 http 协议的 get 请求。



http 是应用层用来获取率 URL 的，和许多互联网应用程序一样，它在 TCP/IP 和网络层的上面，数据链路层和物理层大多视网络不同而不同。一般来说，如果是有线连接，则组成 Ethernet，如果是无线连接，则组成 802.11。

体系结构及传输过程图如下：

对上图的 NO987 号分组进行进一步探究。



注意，在上图从上向下的顺序正是体系结构的倒序

首先分析 Frame

```
Frame 987: 961 bytes on wire (7688 bits), 961 bytes captured (7688 bits) on interface \Device\NPF_{72EB8768-912F-4CD5-8B47-F680C840F380}, id 0
  Interface id: 0 (\Device\NPF_{72EB8768-912F-4CD5-8B47-F680C840F380})
  Encapsulation type: Ethernet (1)
  Arrival Time: Oct  3, 2021 13:58:32.656194000 中国标准时间
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1633240712.656194000 seconds
  [Time delta from previous captured frame: 0.000256000 seconds]
  [Time delta from previous displayed frame: 0.888101000 seconds]
  [Time since reference or first frame: 5.832964000 seconds]
  Frame Number: 987
  Frame Length: 961 bytes (7688 bits)
  Capture Length: 961 bytes (7688 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp:http]
  [Coloring Rule Name: HTTP]
  [Coloring Rule String: http || tcp.port == 80 || http2]
```

Frame 是对这个包的信息的一个总览，比如这个包的捕获时间是 2021.10.03 13:58:32，共 7688bits。

其次分析 Ethernet（数据链路层和物理层）

```
Ethernet II, Src: IntelCor_6d:43:a4 (a0:51:0b:6d:43:a4), Dst: Hangzhou_53:00:02 (74:1f:4a:53:00:02)
  Destination: Hangzhou_53:00:02 (74:1f:4a:53:00:02)
      Address: Hangzhou_53:00:02 (74:1f:4a:53:00:02)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: IntelCor_6d:43:a4 (a0:51:0b:6d:43:a4)
      Address: IntelCor_6d:43:a4 (a0:51:0b:6d:43:a4)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

我是用的是无线网，按照上面说的，这里应该是 802.11 才对。为什么会出现这种情况呢？是因为我在使用 wireshark 时在 capture option 里选择的是用 Ethernet 的格式，所以在 802.11 header 被转成了 pseudo-Ethernet header.

Destination 显示了这个包要被送往的地方，根据拼音看出来是杭州，Source 应该指示的是数据内容。

接下来是运输层 IP

```
Internet Protocol Version 4, Src: 10.118.159.90, Dst: 121.51.176.99
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 947
    Identification: 0x869e (34462)
  > Flags: 0x40, Don't fragment
    Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.118.159.90
    Destination Address: 121.51.176.99
```

根据我的分析，**Source Address 指示的是我的 IP 地址，地址为 10.118.159.10**，Destination Address 指示的是服务器 IP 地址，为 121.51.176.99。

接下来是传输控制层 TCP

```
Transmission Control Protocol, Src Port: 50423, Dst Port: 80, Seq: 1, Ack: 1, Len: 907
    Source Port: 50423
    Destination Port: 80
    [Stream index: 23]
    [TCP Segment Len: 907]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 179493262
    [Next Sequence Number: 908    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 51655361
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 515
    [Calculated window size: 131840]
    [Window size scaling factor: 256]
    Checksum: 0xd70c [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
    TCP payload (907 bytes)
```

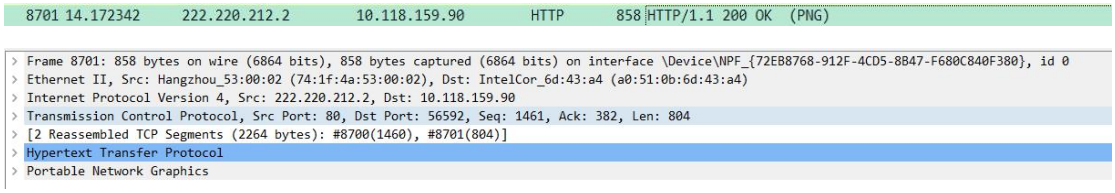
Source Port 指示客户端的端口号是 50423，Destination Port 指示服务器端口号是 80。

最后是应用层 http

```
Hypertext Transfer Protocol
  > GET /api/get_url_for_icon?url=https://www.bilibili.com/&guid=508c3bf5b99be006e7ab679d144988cb HTTP/1.1\r\n
    Host: video.browser.qq.com\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.25 Safari/537.36 Core/1.70.3877.400 QQBrowser/10…
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9\r\n
  > [truncated]Cookie: RK=T9AxH/XfGc; ptcz=387b44c00f3a957cc303de5c9cf238f27207713849be402a99de1fa46cc8fa76; luin=o1459987672; lskey=000100001983462b95854fd…
    Q-Guid: 508c3bf5b99be006e7ab679d144988cb\r\n
    Q-Ua2: PR=PC&CO=WBK&QV=3&PL=WIN&PB=GE&PPVN=10.8.0.4506&COVC=047000&CHID=43665&RL=2560*1080&MO=QB&VE=GA&BIT=64&OS=10.0.19042\r\n
    \r\n
    [Full request URI: http://video.browser.qq.com/api/get_url_for_icon?url=https://www.bilibili.com/&guid=508c3bf5b99be006e7ab679d144988cb]
    [HTTP request 1/1]
    [Response in frame: 1009]
```

应用层的协议名称是 Hypertext Transfer Protocol，缩写为 HTTP，中文称超文本传输协议。根据上图信息，首先得知这是一个 GET 请求，HOST 指示了发送这个包的应用，**根据 Host: vedio.browser.qq.com 推测 wireshark 捕获的这个包是 QQ 浏览器这个应用程序发送的。**

再分析一个分组



可以看到，比 http get 请求的分组多出了一个模块：[2 reassembled TCP segments ...]”.

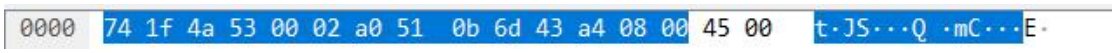
通常，在一系列的分组到达主机之后 web 响应便被送向网络。标记为 HTTP 的数据包是最后一个数据包，在 web 响应中，一个目录列表也被加入。每一个数据包具有 TCP 协议，即使数据包携带 HTTP 响应的一部分。当完整的 http 信息被解释时只有最后一个数据包显示为具有 http 协议，它列出了参与组成 http 响应的所有数据包。

第三步：

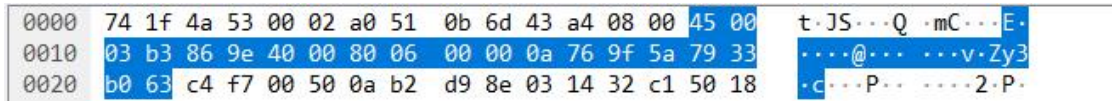
绘制一个 HTTP GET 数据包的图，该图显示 TCP、IP 和以太网协议头的位置和大小（以字节为单位）。在这张图上，显示以太网报头的范围和 IP 通过网络发送到以太网的以太网有效负载。要显示协议层的嵌套结构，注意 IP 报头和 IP 有效负载的范围。

查看各部分的 size：

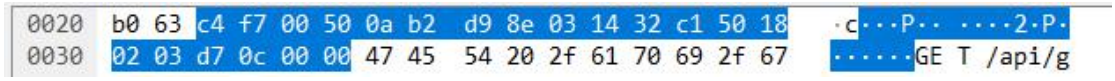
Ethernet



IP



TCP



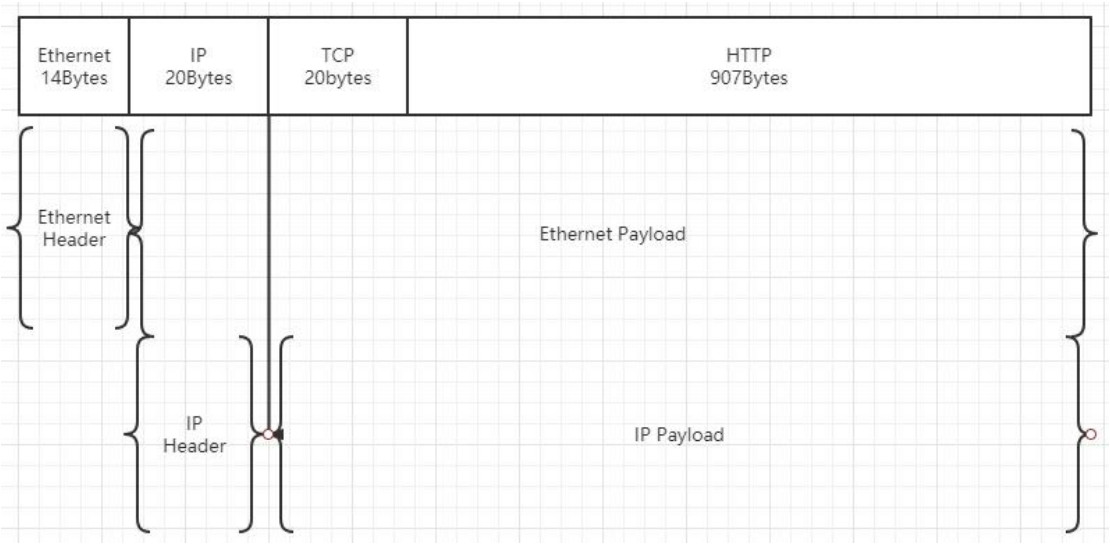
HTTP

```
0030  02 03 d7 0c 00 00 47 45  54 20 2f 61 70 69 2f 67   ......GE T /api/g
0040  65 74 5f 75 72 6c 5f 66  6f 72 5f 69 63 6f 6e 3f   et_url_f or_icon?
0050  75 72 6c 3d 68 74 74 70  73 3a 2f 2f 77 77 77 2e   url=http s://www.
0060  62 69 6c 69 62 69 6c 69  2e 63 6f 6d 2f 26 67 75   bilibili .com/&gu
0070  69 64 3d 35 30 38 63 33  62 66 35 62 39 39 62 65   id=508c3 bf5b99be
0080  30 30 36 65 37 61 62 36  37 39 64 31 34 34 39 38   006e7ab6 79d14498
0090  38 63 62 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f   8cb HTTP /1.1··Ho
00a0  73 74 3a 20 76 69 64 65  6f 2e 62 72 6f 77 73 65   st: vide o.browse
00b0  72 2e 71 71 2e 63 6f 6d  0d 0a 55 73 65 72 2d 41   r.qq.com ··User-A
00c0  67 65 6e 74 3a 20 4d 6f  7a 69 6c 6c 61 2f 35 2e   gent: Mo zilla/5.
00d0  30 20 28 57 69 6e 64 6f  77 73 20 4e 54 20 31 30   0 (Windo ws NT 10
00e0  2e 30 3b 20 57 4f 57 36  34 29 20 41 70 70 6c 65   .0; WOW6 4) Apple
00f0  57 65 62 4b 69 74 2f 35  33 37 2e 33 36 20 28 4b   WebKit/5 37.36 (K
0100  48 54 4d 4c 2c 20 6c 69  6b 65 20 47 65 63 6b 6f   HTML, li ke Gecko
0110  29 20 43 68 72 6f 6d 65  2f 37 30 2e 30 2e 33 35   ) Chrome /70.0.35
0120  33 38 2e 32 35 20 53 61  66 61 72 69 2f 35 33 37   38.25 Sa fari/537
0130  2e 33 36 20 43 6f 72 65  2f 31 2e 37 30 2e 33 38   .36 Core /1.70.38
0140  37 37 2e 34 30 30 20 51  51 42 72 6f 77 73 65 72   77.400 Q QBrowser
0150  2f 31 30 2e 38 2e 34 35  30 36 2e 34 30 30 0d 0a   /10.8.45 06.400··
0160  41 63 63 65 70 74 2d 45  6e 63 6f 64 69 6e 67 3a   Accept-E ncoding:
0170  20 67 7a 69 70 2c 20 64  65 66 6c 61 74 65 0d 0a    gzip, d eflate··
0180  41 63 63 65 70 74 2d 4c  61 6e 67 75 61 67 65 3a   Accept-L anguage:
0190  20 7a 68 2d 43 4e 2c 7a  68 3b 71 3d 30 2e 39 0d    zh-CN,z h;q=0.9·
01a0  0a 43 6f 6f 6b 69 65 3a  20 52 4b 3d 54 39 41 78   ·Cookie:  RK=T9Ax
01b0  48 2f 58 66 47 63 3b 20  70 74 63 7a 3d 33 38 37   H/XfGc;  ptcz=387
01c0  62 34 34 63 30 30 66 33  61 39 35 37 63 63 33 30   b44c00f3 a957cc30
```

.....

```
0230  34 37 38 61 61 62 63 63  64 64 66 37 62 65 33 64   478aabcc ddf7be3d
0240  63 30 34 31 31 36 62 30  62 32 63 30 61 32 34 61   c04116b0 b2c0a24a
0250  33 38 62 33 61 38 66 64  33 65 34 62 30 62 65 30   38b3a8fd 3e4b0be0
0260  63 61 33 64 36 61 33 62  38 65 32 64 32 30 62 3b   ca3d6a3b 8e2d20b;
0270  20 6f 5f 63 6f 6f 6b 69  65 3d 31 34 35 39 39 38    o_cooki e=145998
0280  37 36 37 32 3b 20 70 61  63 5f 75 69 64 3d 31 5f   7672; pa c_uid=1_
0290  31 34 35 39 39 38 37 36  37 32 3b 20 74 6f 6b 65   14599876 72; toke
02a0  6e 50 61 72 61 6d 73 3d  25 33 46 41 44 54 41 47   nParams= %3FADTAG
02b0  25 33 44 45 76 65 6e 74  42 6c 61 63 6b 54 6f 70   %3DEvent BlackTop
02c0  2e 62 75 74 74 6f 6e 2e  62 74 6e 61 76 2e 65 63   .button. btnav.ec
02d0  74 65 72 3b 20 65 61 73  5f 73 69 64 3d 77 31 33   ter; eas _sid=w13
02e0  36 30 33 4c 33 4b 32 69  34 56 30 72 36 75 33 68   603L3K2i 4V0r6u3h
02f0  32 39 31 68 33 4b 38 3b  20 63 66 71 71 63 6f 6d   291h3K8;  cfqqcom
0300  72 6f 75 74 65 4c 69 6e  65 3d 6d 61 69 6e 5f 61   routeLin e=main_a
0310  63 74 69 6f 6e 73 0d 0a  51 2d 47 75 69 64 3a 20   ctions·· Q-Guid:
0320  35 30 38 63 33 62 66 35  62 39 39 62 65 30 30 36   508c3bf5 b99be006
0330  65 37 61 62 36 37 39 64  31 34 34 39 38 38 63 62   e7ab679d 144988cb
0340  0d 0a 51 2d 55 61 32 3a  20 50 52 3d 50 43 26 43   ··Q-Ua2:  PR=PC&C
0350  4f 3d 57 42 4b 26 51 56  3d 33 26 50 4c 3d 57 49   O=WBK&QV =3&PL=WI
0360  4e 26 50 42 3d 47 45 26  50 50 56 4e 3d 31 30 2e   N&PB=GE& PPVN=10.
0370  38 2e 30 2e 34 35 30 36  26 43 4f 56 43 3d 30 34   8.0.4506 &COVC=04
0380  37 30 30 30 26 43 48 49  44 3d 34 33 36 36 35 26   7000&CHI D=43665&
0390  52 4c 3d 32 35 36 30 2a  31 30 38 30 26 4d 4f 3d   RL=2560* 1080&MO=
03a0  51 42 26 56 45 3d 47 41  26 42 49 54 3d 36 34 26   QB&VE=GA &BIT=64&
03b0  4f 53 3d 31 30 2e 30 2e  31 39 30 34 32 0d 0a 0d   OS=10.0. 19042···
03c0  0a                                                 ·
```

最终绘制下图：


Ethernet
14Bytes
IP
20Bytes
TCP
20bytes
HTTP
907Bytes
Ethernet
Header
Ethernet Payload
IP
Header
IP Payload