CS305 hw1

Name: 胡玉斌
Student Id: 11712121

理论作业

> 1. Compare packet switch and circuit switch under the following scenario. Suppose you would like to deliver a message of x bit. There are k links from the source to destination. The propagation delay of each link is d second, the transmission rate is b bit/second. The circuit setup time under circuit switch is s second. Under packet switch network, when the packet length is p bit, the queue delay in every node can be neglected. Please calculate the condition, under which the delay of packet switch is smaller than that of the circuit switch.

Solution：

For packet switch, it has four kinds of delay: processing delay, queuing delay, transmission delay, propagation delay.

$$d_{packet} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

When $t = \frac{x}{b}$, it transmites all the packets;
To arrival the destination, the last packet needs $k-1$ routers to forward. Each forward need $\frac{p}{b}$

So that:

$$d_{packet} = \frac{x}{b} + (k-1)\frac{p}{b} + kd$$

For circuit switch, it has four kinds of delay: processing delay, queuing delay, transmission delay, propagation delay.

$$d_{circuit} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

When $t = s$, the circuit setup;
When $t = s + \frac{x}{b}$, it transmites all the packets;
When $t = s + \frac{x}{b} + kd$, the message arrives at the distination.

So that:

$$d_{circuit} = s + \frac{x}{b} + kd$$

In conclusion:

When the delay of packet switch is smaller than that of the circuit switch,

$$d_{packet} = \frac{x}{b} + (k-1)\frac{p}{b} + kd < d_{circuit} = s + \frac{x}{b} + kd$$

$$(k-1)\frac{p}{b} < s$$

> 2. Calculate the overall delay of transmitting a 1000KB file under the following circumstance. The overall delay is defined as the time from the starting point of the transmission until the arrival of the last bit to the destination. RTT is assumed to be 100ms, one packet is 1KB (1024B) size. The handshaking process costs 2RTT before transmitting the file.
>
> 1. Transmission bandwidth is 1.5Mb/s, the packets can be continuously transmitted.
> 2. Transmission bandwidth is 1.5Mb/s, but when one packet is transmitted, the next packet should wait for 1 RTT (waiting for the acknowledgement of the receiver) before being transmitted.
> 3. Transmission bandwidth is infinite, i.e. transmission delay is 0. After every 1 RTT, as many as 20 packets can be transmitted.

Solution:

1. The transmission bandwidth is 1.5Mb/s, the packet szie is 1KB.
   total time = initial handshaking + network delay
   initial handshaking = 2 * RTT = 2 * 100ms = 200ms
   1 KB = 2^{10} bytes
   1 Mb/s = 10^6 bits/s
   network delay = transmission delay + propagation delay
   = (1000KB / (1.5Mb/s)) + 100ms/2
   = (1000 * 1024 * 8 / (1.5 * 1000000)) + 50ms
   ≈ 5.461s + 50ms
   = 5461ms + 50ms
   = 5511ms
   total time = 200ms + 5511ms = 5711ms
2. The number of packets: 1000KB / 1KB = 1000
   total time = initial handshaking + network delay + (the number of packet-1) * 100ms
   initial handshaking = 2 * RTT = 2 * 100ms = 200ms
   network delay = transmission delay + propagation delay
   = (1000KB / (1.5Mb/s)) + 100ms/2
   = (1000 * 1024 * 8 / (1.5 * 1000000)) + 50ms
   ≈ 5.461s + 50ms

   = 5461ms + 50ms
   = 5511ms
   total time = 200ms + 5511ms + 99900= 105611ms
3. The number of packets: 1000KB / 1KB = 1000
   initial handshaking = 2 * RTT = 2 * 100ms = 200ms
   total time = initial handshaking + (1000/20) * 100ms = 5200ms

> 3. List six access technologies. Classify each of them as home access, enterprise access, or wide-area mobile access.

Solution:

1. Dial-up modem over telephone line: home;
2. DSL over telephone line: home or small office;
3. Cable to HFC: home;
4. Wifi (802.11): home and enterprise;
5. 100 Mbps switched Ethernet: enterprise;
6. 3G and 4G: wide-area wireless.

> 4. 1. List five nonproprietary Internet applications and the pplication-layer protocols that they use.
>
> 2. What information is used by a process running on one host to identify a process running on another host?

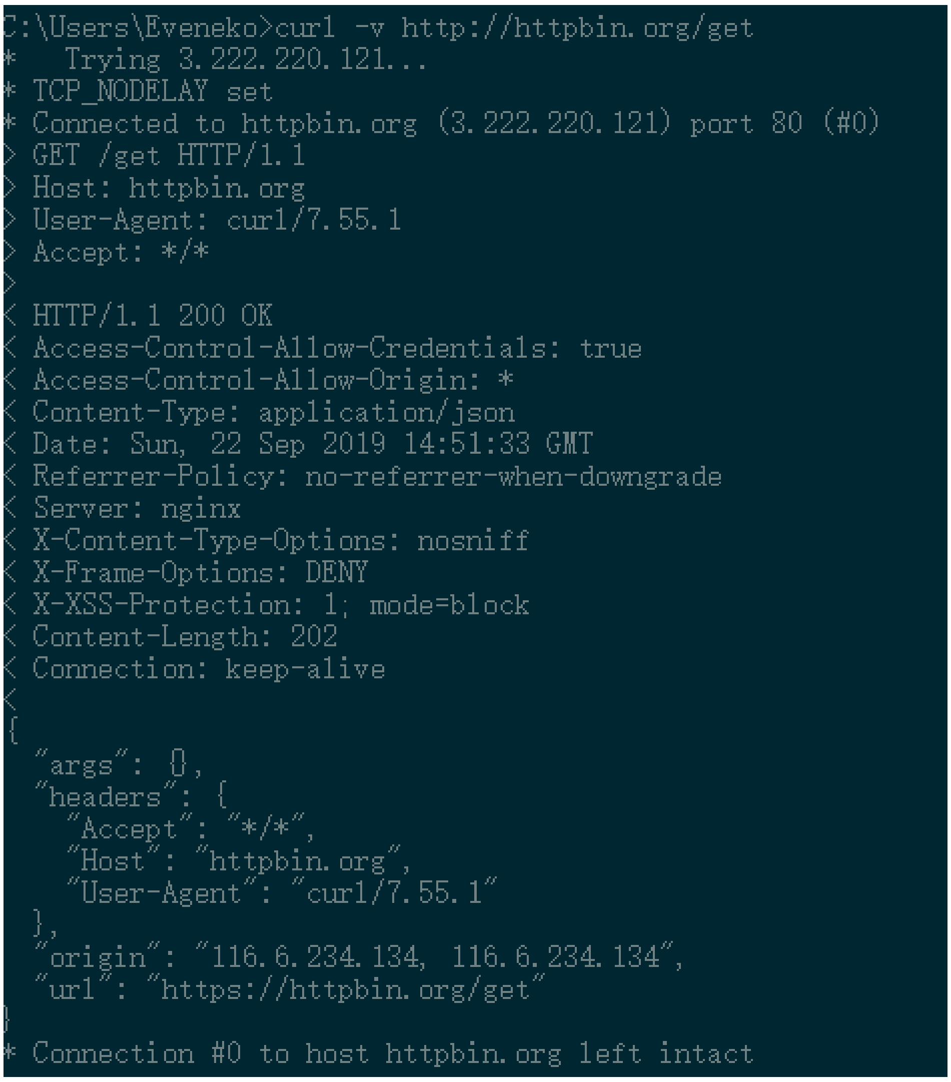
Solution:

1. 1. the Web:HTTP
   2. remote login: Telnet
   3. file transfer: Ftp
   4. Network News: NNTP
   5. email: SMTP
2. The IP address of the destination host and the port number of the destination socket.

实验作业

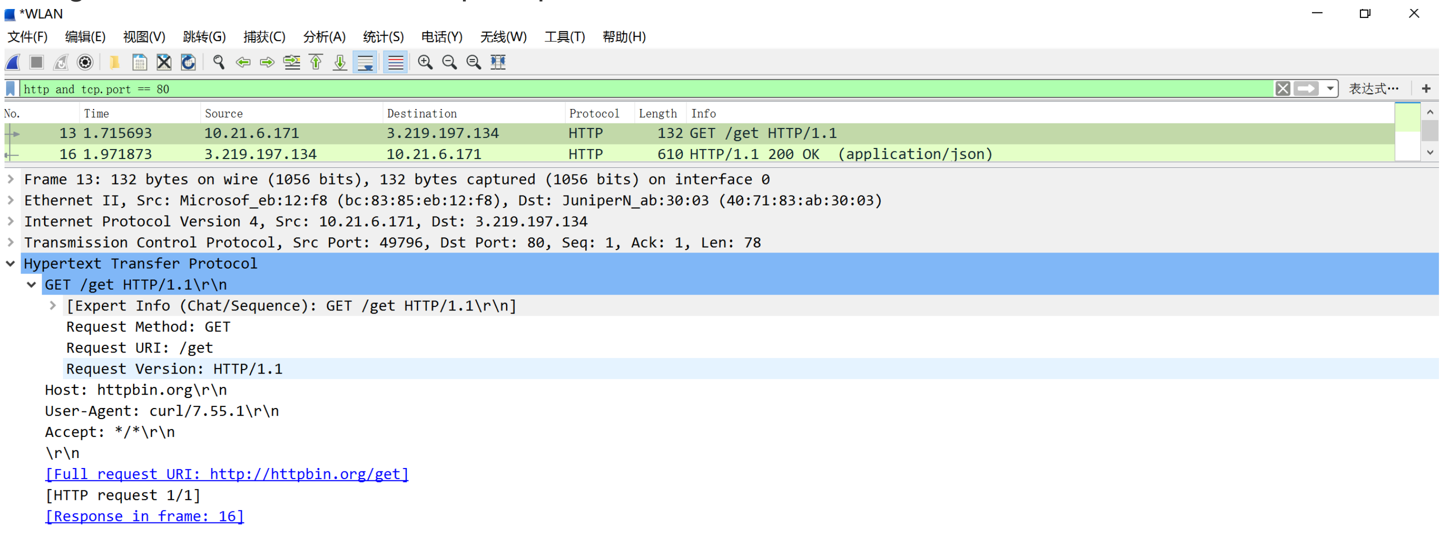
> 1. Using cURL make GET request to http://httpbin.org/get
>   – Using curl -v to inspect the interaction
>   – Using Wireshark to capture the packet cURL sent.

```
// in cmd
curl -v http://httpbin.org/get
```



```
C:\Users\Eveneko>curl -v http://httpbin.org/get
*   Trying 3.222.220.121...
* TCP_NODELAY set
* Connected to httpbin.org (3.222.220.121) port 80 (#0)
> GET /get HTTP/1.1
> Host: httpbin.org
> User-Agent: curl/7.55.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Access-Control-Allow-Credentials: true
< Access-Control-Allow-Origin: *
< Content-Type: application/json
< Date: Sun, 22 Sep 2019 14:51:33 GMT
< Referrer-Policy: no-referrer-when-downgrade
< Server: nginx
< X-Content-Type-Options: nosniff
< X-Frame-Options: DENY
< X-XSS-Protection: 1; mode=block
< Content-Length: 202
< Connection: keep-alive
<
{
  "args": {},
  "headers": {
    "Accept": "*/*",
    "Host": "httpbin.org",
    "User-Agent": "curl/7.55.1"
  },
  "origin": "116.6.234.134, 116.6.234.134",
  "url": "https://httpbin.org/get"
}
* Connection #0 to host httpbin.org left intact
```

Using Wireshark to catch the request packet



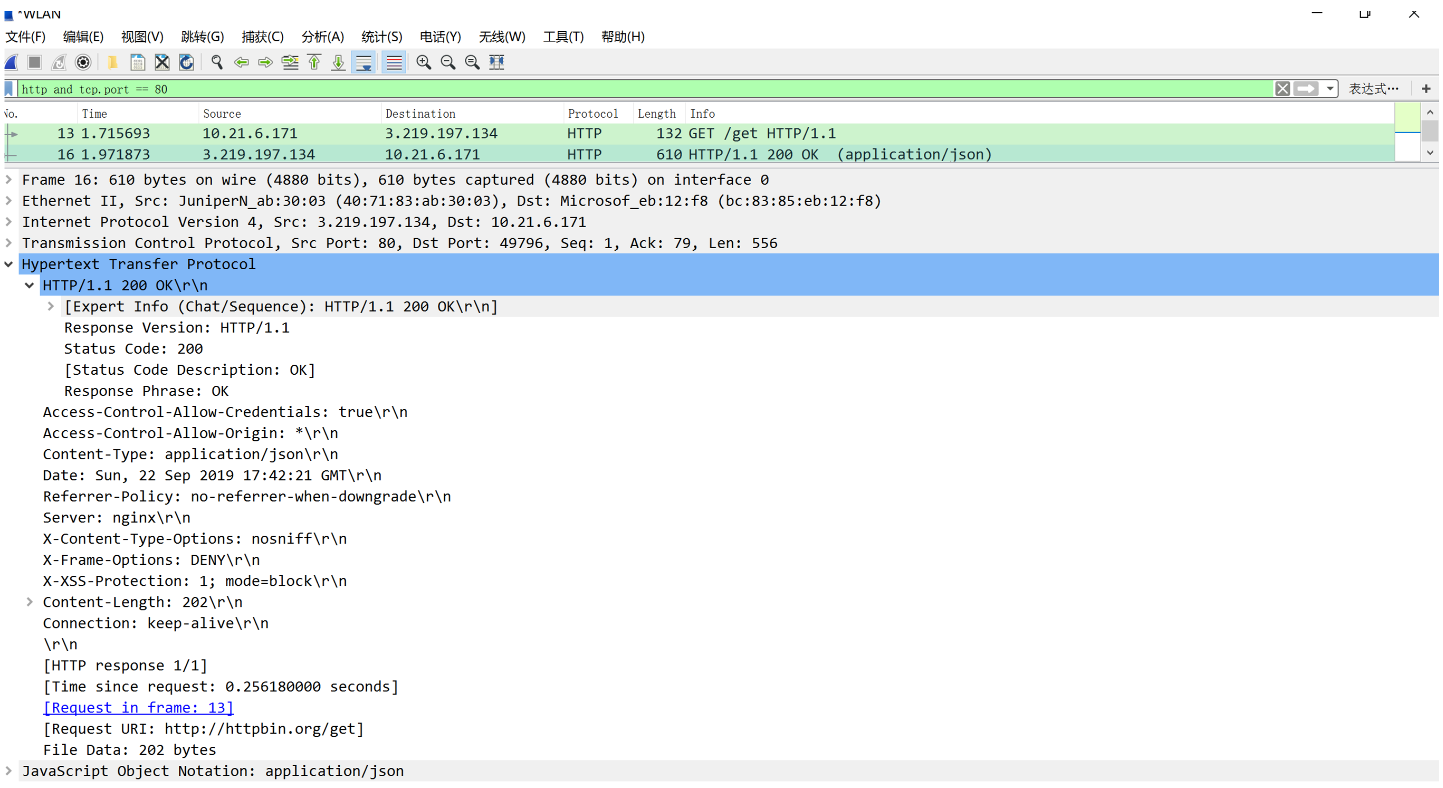
*WLAN
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
表达式…

```
http and tcp.port == 80

No.   Time       Source         Destination    Protocol  Length Info
   13 1.715693   10.21.6.171    3.219.197.134  HTTP       132 GET /get HTTP/1.1
   16 1.971873   3.219.197.134  10.21.6.171    HTTP       610 HTTP/1.1 200 OK  (application/json)

> Frame 13: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits) on interface 0
> Ethernet II, Src: Microsof_eb:12:f8 (bc:83:85:eb:12:f8), Dst: JuniperN_ab:30:03 (40:71:83:ab:30:03)
> Internet Protocol Version 4, Src: 10.21.6.171, Dst: 3.219.197.134
> Transmission Control Protocol, Src Port: 49796, Dst Port: 80, Seq: 1, Ack: 1, Len: 78
v Hypertext Transfer Protocol
  v GET /get HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /get HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /get
      Request Version: HTTP/1.1
    Host: httpbin.org\r\n
    User-Agent: curl/7.55.1\r\n
    Accept: */*\r\n
    \r\n
    [Full request URI: http://httpbin.org/get]
    [HTTP request 1/1]
    [Response in frame: 16]
```

Using Wireshark to catch the response packet


*WLAN
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
表达式…

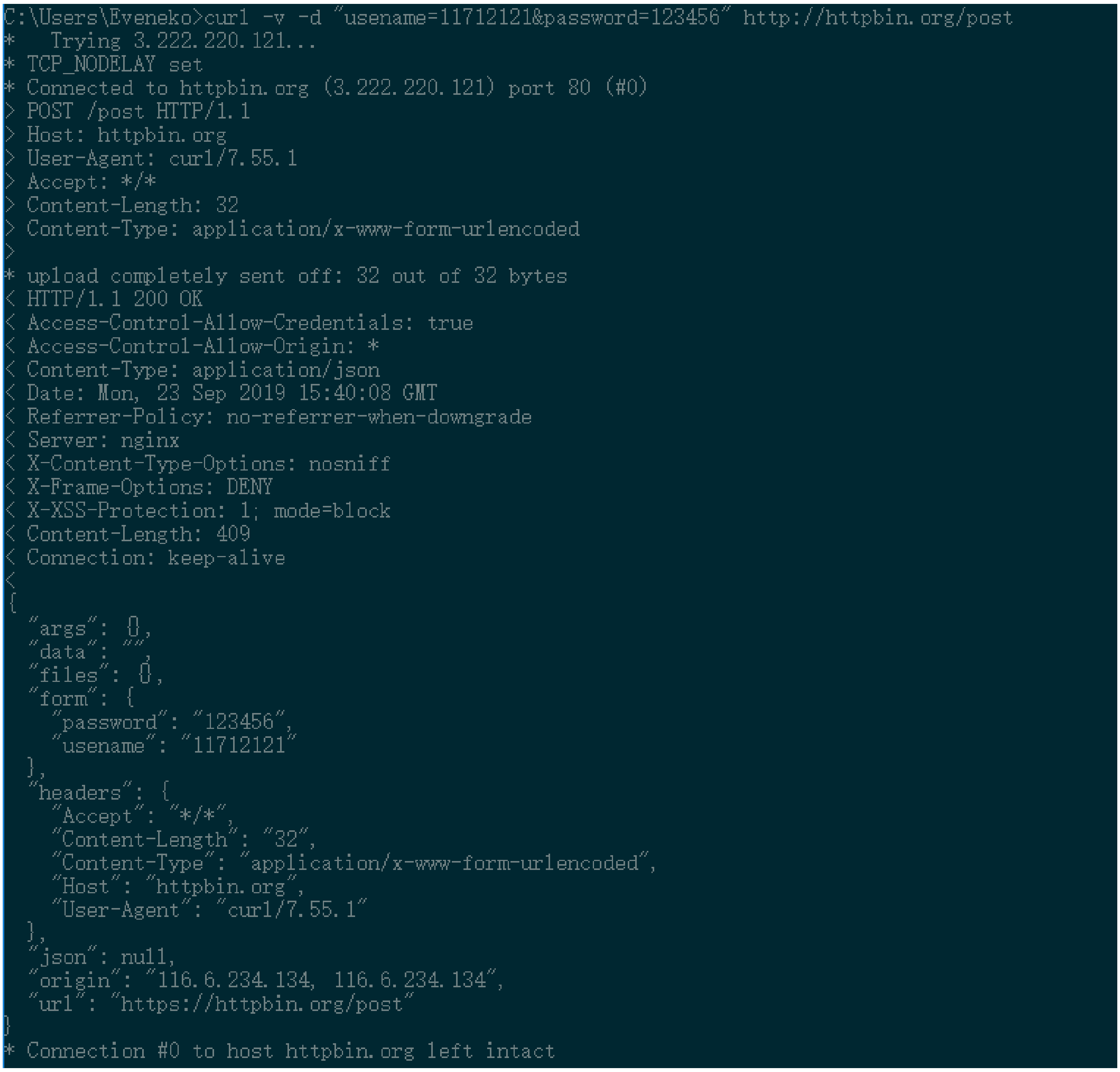
```
http and tcp.port == 80

No.  Time      Source         Destination    Protocol  Length Info
  13 1.715693  10.21.6.171    3.219.197.134  HTTP       132 GET /get HTTP/1.1
  16 1.971873  3.219.197.134  10.21.6.171    HTTP       610 HTTP/1.1 200 OK  (application/json)

> Frame 16: 610 bytes on wire (4880 bits), 610 bytes captured (4880 bits) on interface 0
> Ethernet II, Src: JuniperN_ab:30:03 (40:71:83:ab:30:03), Dst: Microsof_eb:12:f8 (bc:83:85:eb:12:f8)
> Internet Protocol Version 4, Src: 3.219.197.134, Dst: 10.21.6.171
> Transmission Control Protocol, Src Port: 80, Dst Port: 49796, Seq: 1, Ack: 79, Len: 556
v Hypertext Transfer Protocol
  v HTTP/1.1 200 OK\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Access-Control-Allow-Credentials: true\r\n
    Access-Control-Allow-Origin: *\r\n
    Content-Type: application/json\r\n
    Date: Sun, 22 Sep 2019 17:42:21 GMT\r\n
    Referrer-Policy: no-referrer-when-downgrade\r\n
    Server: nginx\r\n
    X-Content-Type-Options: nosniff\r\n
    X-Frame-Options: DENY\r\n
    X-XSS-Protection: 1; mode=block\r\n
  > Content-Length: 202\r\n
    Connection: keep-alive\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.256180000 seconds]
    [Request in frame: 13]
    [Request URI: http://httpbin.org/get]
    File Data: 202 bytes
> JavaScript Object Notation: application/json
```

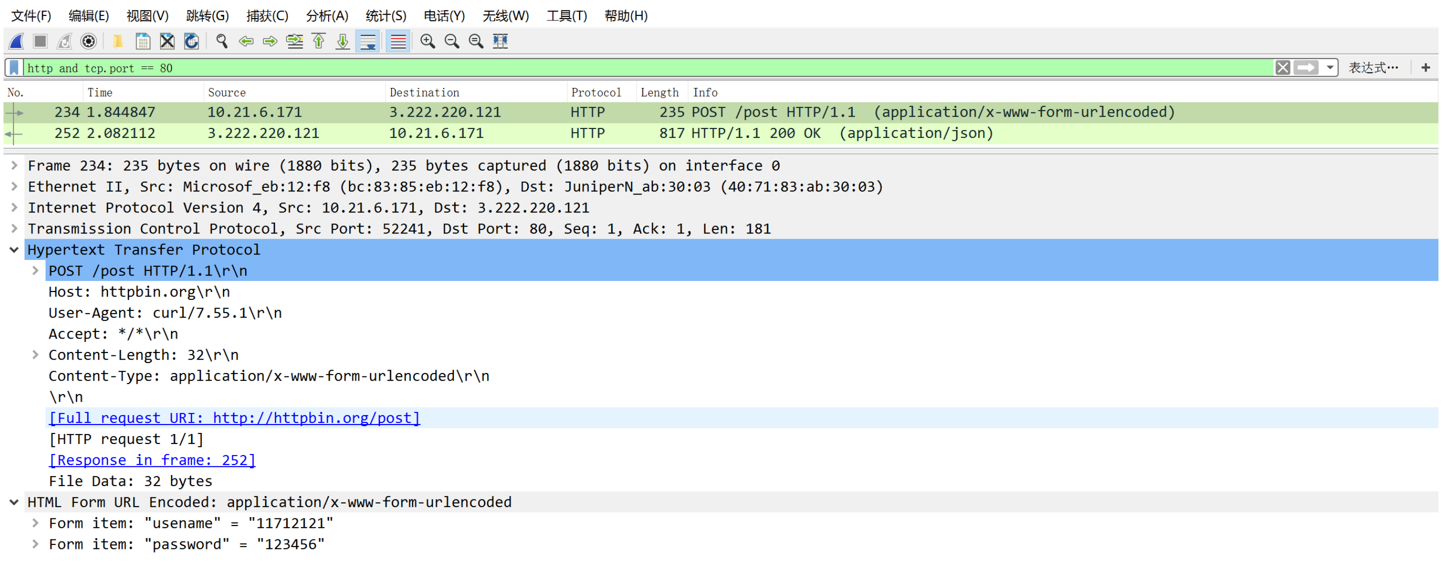
> 2.Using cURL make POST request to http://httpbin.org/post
> – Using curl -v to inspect the interaction
> – Using Wireshark to capture the packet cURL sent.

```
// in cmd
curl -v -d "usename=11712121&password=123456" http://httpbin.org/post
```



```
C:\Users\Eveneko>curl -v -d "usename=11712121&password=123456" http://httpbin.org/post
*   Trying 3.222.220.121...
* TCP_NODELAY set
* Connected to httpbin.org (3.222.220.121) port 80 (#0)
> POST /post HTTP/1.1
> Host: httpbin.org
> User-Agent: curl/7.55.1
> Accept: */*
> Content-Length: 32
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 32 out of 32 bytes
< HTTP/1.1 200 OK
< Access-Control-Allow-Credentials: true
< Access-Control-Allow-Origin: *
< Content-Type: application/json
< Date: Mon, 23 Sep 2019 15:40:08 GMT
< Referrer-Policy: no-referrer-when-downgrade
< Server: nginx
< X-Content-Type-Options: nosniff
< X-Frame-Options: DENY
< X-XSS-Protection: 1; mode=block
< Content-Length: 409
< Connection: keep-alive
<
{
  "args": {},
  "data": "",
  "files": {},
  "form": {
    "password": "123456",
    "usename": "11712121"
  },
  "headers": {
    "Accept": "*/*",
    "Content-Length": "32",
    "Content-Type": "application/x-www-form-urlencoded",
    "Host": "httpbin.org",
    "User-Agent": "curl/7.55.1"
  },
  "json": null,
  "origin": "116.6.234.134, 116.6.234.134",
  "url": "https://httpbin.org/post"
}
* Connection #0 to host httpbin.org left intact
```

Using Wireshark to catch the request packet



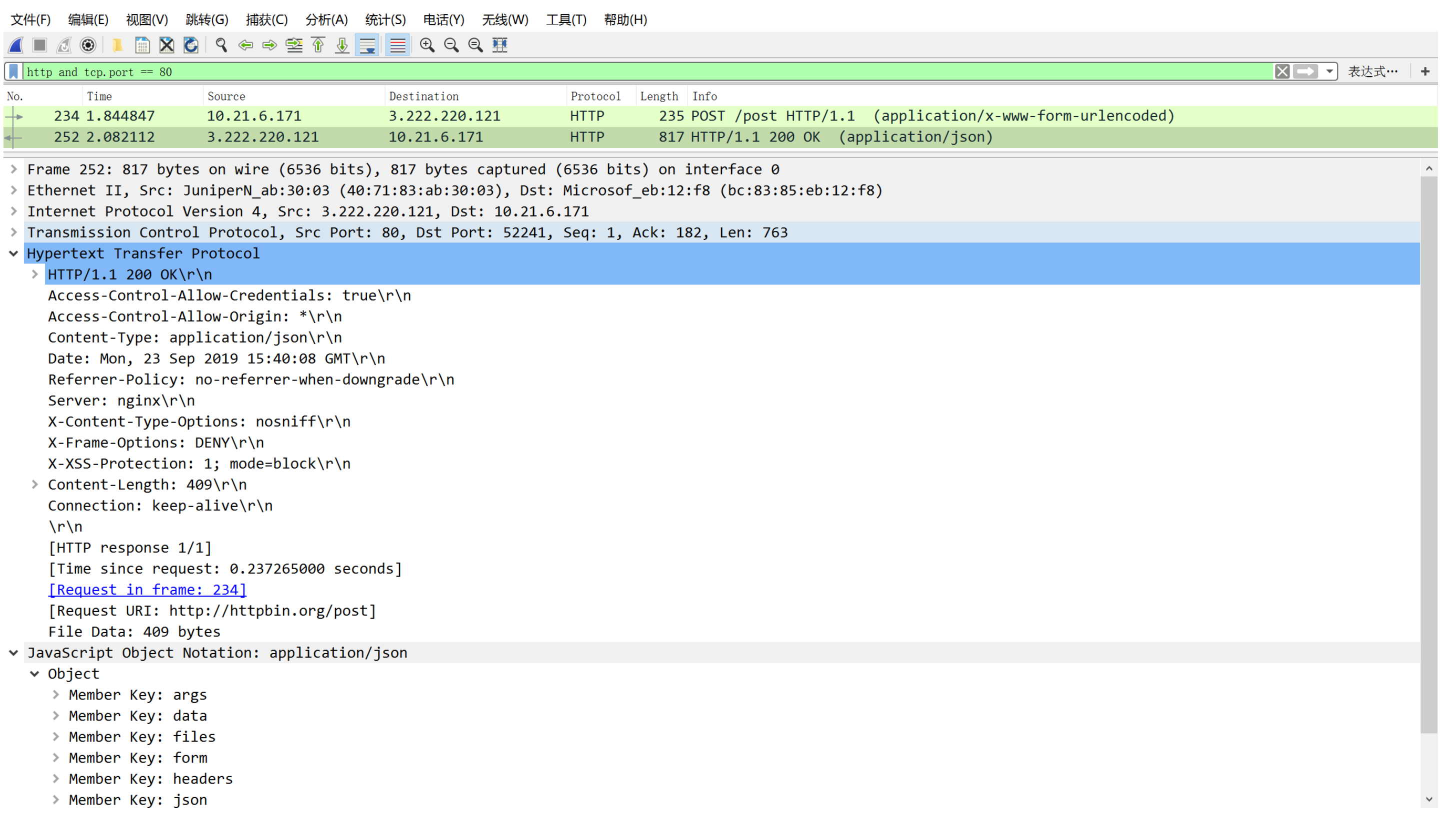
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
表达式…

```
http and tcp.port == 80

No.   Time        Source          Destination     Protocol  Length  Info
  234 1.844847    10.21.6.171     3.222.220.121   HTTP      235 POST /post HTTP/1.1  (application/x-www-form-urlencoded)
  252 2.082112    3.222.220.121   10.21.6.171     HTTP      817 HTTP/1.1 200 OK  (application/json)

> Frame 234: 235 bytes on wire (1880 bits), 235 bytes captured (1880 bits) on interface 0
> Ethernet II, Src: Microsof_eb:12:f8 (bc:83:85:eb:12:f8), Dst: JuniperN_ab:30:03 (40:71:83:ab:30:03)
> Internet Protocol Version 4, Src: 10.21.6.171, Dst: 3.222.220.121
> Transmission Control Protocol, Src Port: 52241, Dst Port: 80, Seq: 1, Ack: 1, Len: 181
v Hypertext Transfer Protocol
  > POST /post HTTP/1.1\r\n
    Host: httpbin.org\r\n
    User-Agent: curl/7.55.1\r\n
    Accept: */*\r\n
  > Content-Length: 32\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    \r\n
    [Full request URI: http://httpbin.org/post]
    [HTTP request 1/1]
    [Response in frame: 252]
    File Data: 32 bytes
v HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "usename" = "11712121"
  > Form item: "password" = "123456"
```

Using Wireshark to catch the response packet


文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
表达式…
+

```
http and tcp.port == 80

No.   Time        Source          Destination     Protocol  Length  Info
  234 1.844847    10.21.6.171     3.222.220.121   HTTP         235 POST /post HTTP/1.1  (application/x-www-form-urlencoded)
  252 2.082112    3.222.220.121   10.21.6.171     HTTP         817 HTTP/1.1 200 OK  (application/json)

> Frame 252: 817 bytes on wire (6536 bits), 817 bytes captured (6536 bits) on interface 0
> Ethernet II, Src: JuniperN_ab:30:03 (40:71:83:ab:30:03), Dst: Microsof_eb:12:f8 (bc:83:85:eb:12:f8)
> Internet Protocol Version 4, Src: 3.222.220.121, Dst: 10.21.6.171
> Transmission Control Protocol, Src Port: 80, Dst Port: 52241, Seq: 1, Ack: 182, Len: 763
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Access-Control-Allow-Credentials: true\r\n
    Access-Control-Allow-Origin: *\r\n
    Content-Type: application/json\r\n
    Date: Mon, 23 Sep 2019 15:40:08 GMT\r\n
    Referrer-Policy: no-referrer-when-downgrade\r\n
    Server: nginx\r\n
    X-Content-Type-Options: nosniff\r\n
    X-Frame-Options: DENY\r\n
    X-XSS-Protection: 1; mode=block\r\n
  > Content-Length: 409\r\n
    Connection: keep-alive\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.237265000 seconds]
    [Request in frame: 234]
    [Request URI: http://httpbin.org/post]
    File Data: 409 bytes
v JavaScript Object Notation: application/json
  v Object
    > Member Key: args
    > Member Key: data
    > Member Key: files
    > Member Key: form
    > Member Key: headers
    > Member Key: json
```

> 3. Write your report

> What did you get via cURL?

The client, curl, sends a HTTP request. The request contains a method (like GET, POST, HEAD etc), a number of request headers and sometimes a request body. The HTTP server responds with a status line (indicating if things went well), response headers and most often also a response body. And with -v argument, we can inspect HTTP transaction in detail.

> What are the meaning of fields in your request and response headers?

request header

```
(3.222.220.121) port 80 (#0)
> GET /get HTTP/1.1 # 请求类型:GET 协议：HTTP协议1.1版本
> Host: httpbin.org # 指定请求的服务器的域名和端口号
> User-Agent: curl/7.55.1   # 发出请求的用户信息：curl请求
> Accept: */*   # 指定client可以接受内容的类型：任意
```

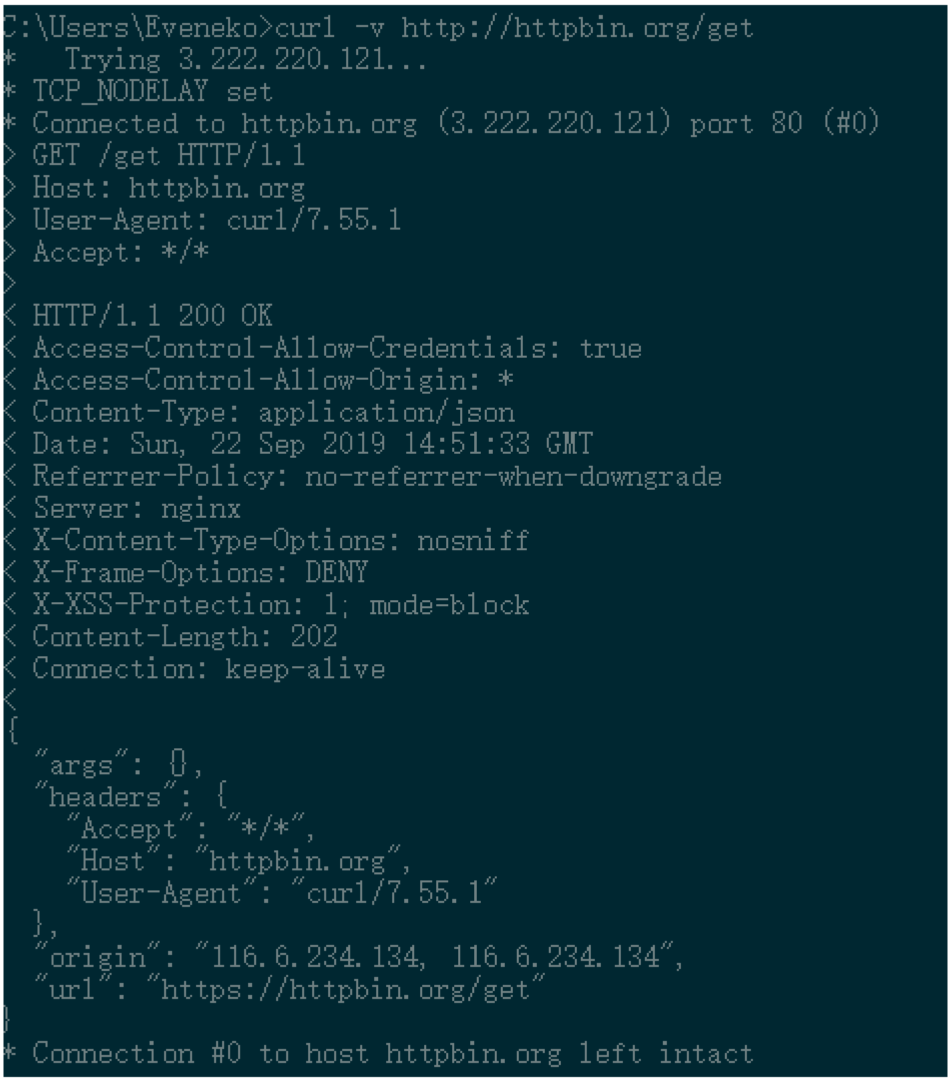
response header

```
< HTTP/1.1 200 OK
< Access-Control-Allow-Credentials: true    #  允许将对请求的响应暴露
< Access-Control-Allow-Origin: *    # 该资源是否被所有域共享
< Content-Type: application/json    # 返回的MIME类型
< Date: Sun, 22 Sep 2019 17:31:00 GMT   # 服务器消息发出时间
< Referrer-Policy: no-referrer-when-downgrade   # 检测访问来源信息
< Server: nginx # web服务器
< X-Content-Type-Options: nosniff   # 用来禁用浏览器内容嗅探行为，script 和 styleSheet 元素会拒绝包含错误的 MIME
< X-Frame-Options: DENY # 不允许在frame中展示
< X-XSS-Protection: 1; mode=block   # 启用XSS过滤
< Content-Length: 202   # 响应体的长度
< Connection: keep-alive    # 表示是否需要保持连接：是
```

> Is the packet captured by Wireshark capture correspond to the cURL request?

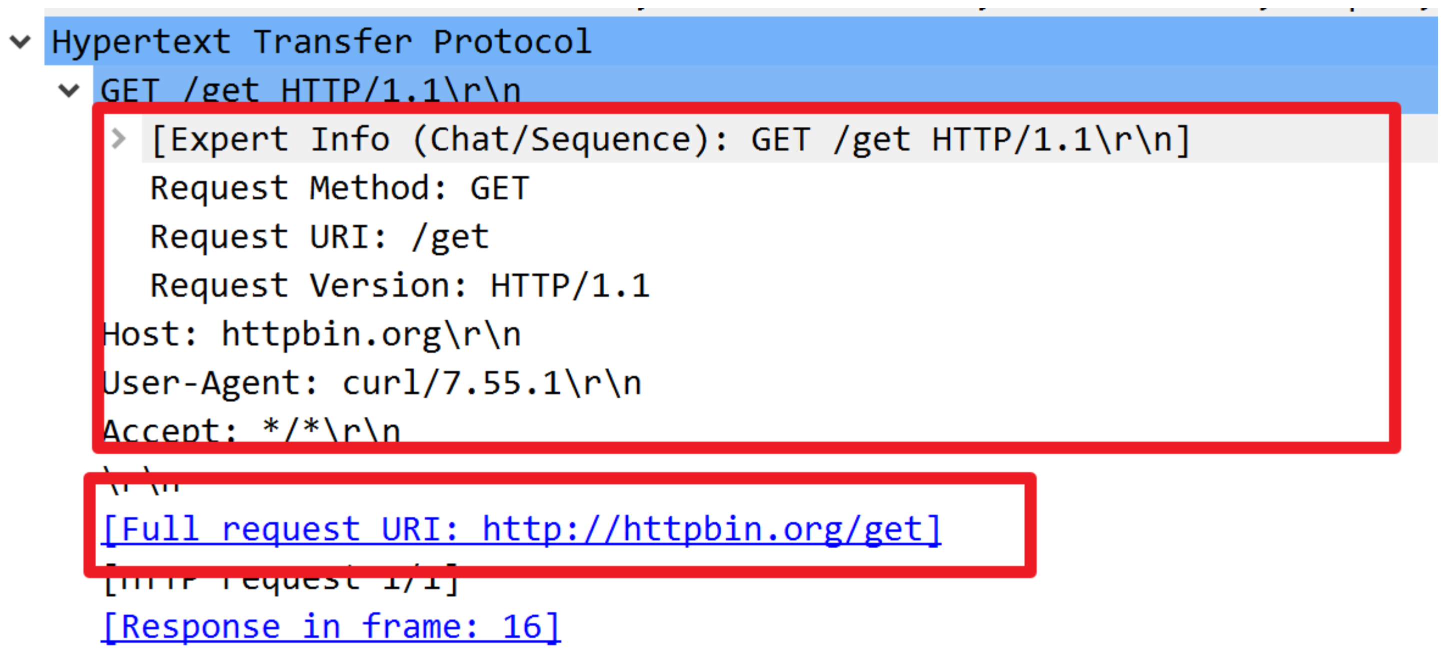
Yes.

In cmd:

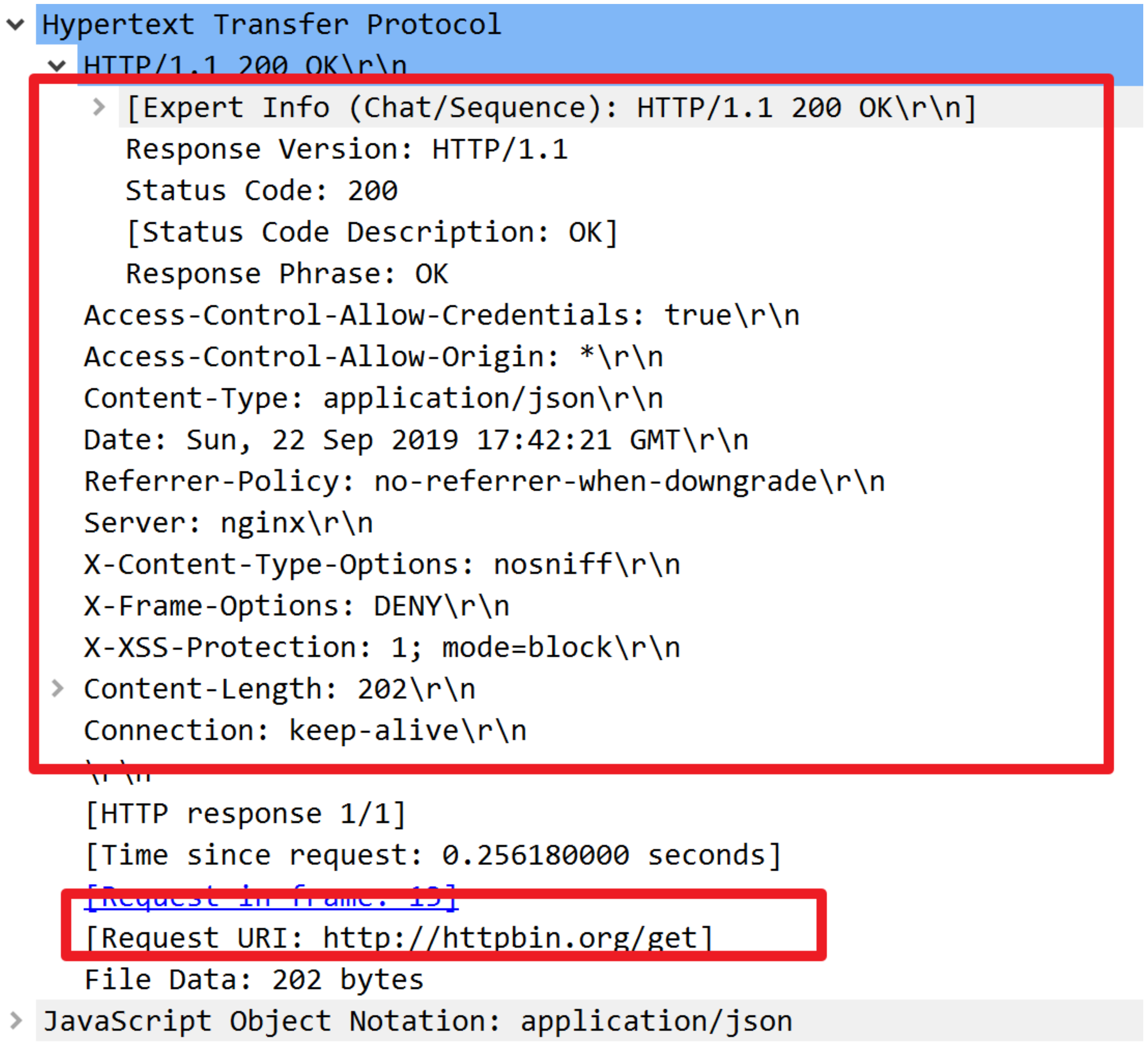


```
C:\Users\Eveneko>curl -v http://httpbin.org/get
*   Trying 3.222.220.121...
* TCP_NODELAY set
* Connected to httpbin.org (3.222.220.121) port 80 (#0)
> GET /get HTTP/1.1
> Host: httpbin.org
> User-Agent: curl/7.55.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Access-Control-Allow-Credentials: true
< Access-Control-Allow-Origin: *
< Content-Type: application/json
< Date: Sun, 22 Sep 2019 14:51:33 GMT
< Referrer-Policy: no-referrer-when-downgrade
< Server: nginx
< X-Content-Type-Options: nosniff
< X-Frame-Options: DENY
< X-XSS-Protection: 1; mode=block
< Content-Length: 202
< Connection: keep-alive
<
{
  "args": {},
  "headers": {
    "Accept": "*/*",
    "Host": "httpbin.org",
    "User-Agent": "curl/7.55.1"
  },
  "origin": "116.6.234.134, 116.6.234.134",
  "url": "https://httpbin.org/get"
}
* Connection #0 to host httpbin.org left intact
```

In wireshake:



```
⌄ Hypertext Transfer Protocol
  ⌄ GET /get HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /get HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /get
      Request Version: HTTP/1.1
    Host: httpbin.org\r\n
    User-Agent: curl/7.55.1\r\n
    Accept: */*\r\n
    \r\n
    [Full request URI: http://httpbin.org/get]
    [HTTP request 1/1]
    [Response in frame: 16]
```

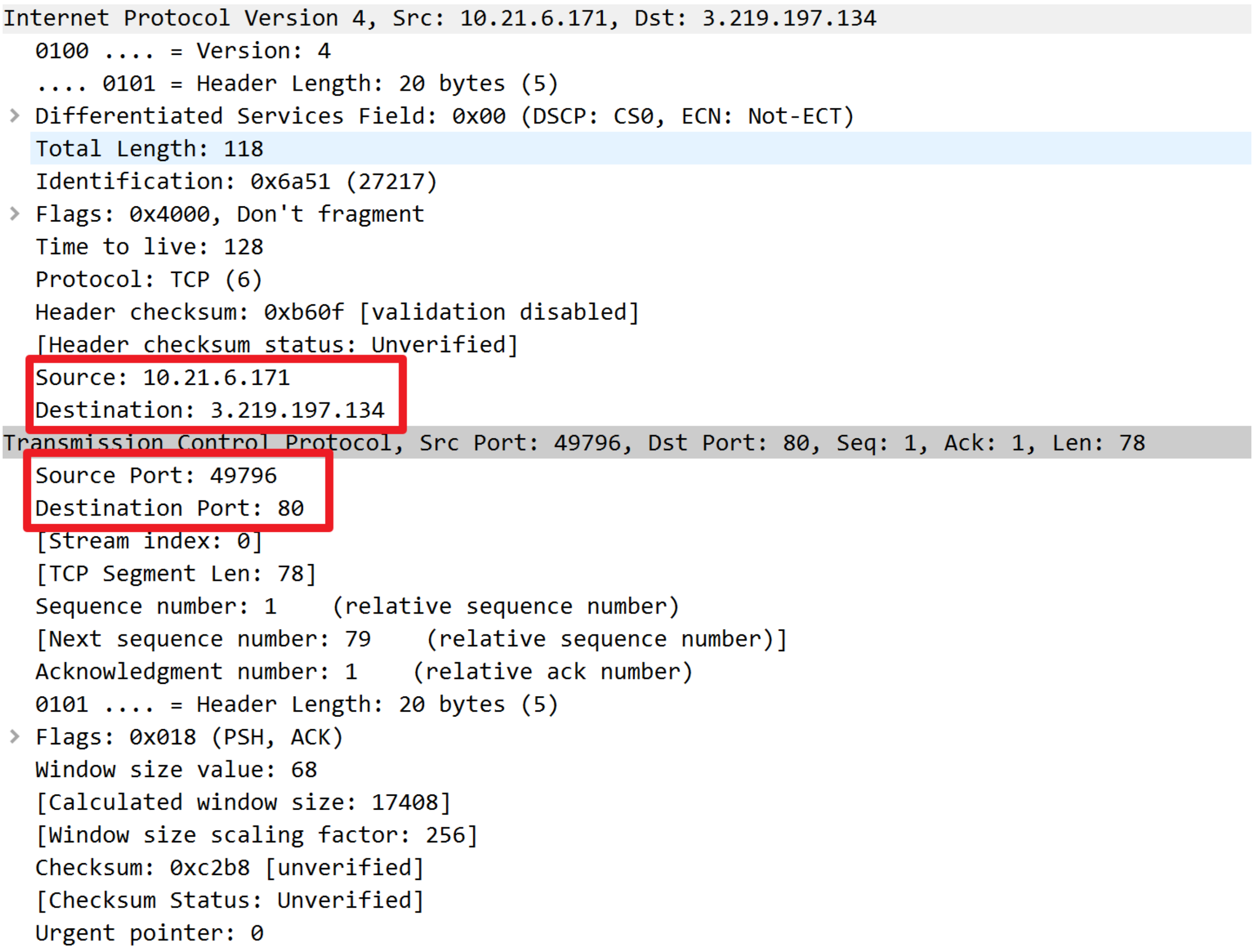


```
⌄ Hypertext Transfer Protocol
  ⌄ HTTP/1.1 200 OK\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Access-Control-Allow-Credentials: true\r\n
    Access-Control-Allow-Origin: *\r\n
    Content-Type: application/json\r\n
    Date: Sun, 22 Sep 2019 17:42:21 GMT\r\n
    Referrer-Policy: no-referrer-when-downgrade\r\n
    Server: nginx\r\n
    X-Content-Type-Options: nosniff\r\n
    X-Frame-Options: DENY\r\n
    X-XSS-Protection: 1; mode=block\r\n
  > Content-Length: 202\r\n
    Connection: keep-alive\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.256180000 seconds]
    [Request in frame: 13]
    [Request URI: http://httpbin.org/get]
    File Data: 202 bytes
> JavaScript Object Notation: application/json
```

We can see the same imformation between cmd and wireshake about cURL request and its response.

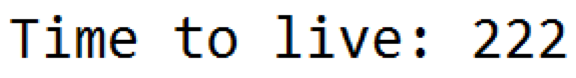
> 补充说明:

In request:
Src ip addr: 10.21.6.171,
Dsr ip addr: 3.219.197.134,
Src Port: 49796,
Dst Port: 80



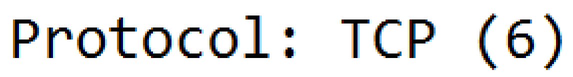
```
Internet Protocol Version 4, Src: 10.21.6.171, Dst: 3.219.197.134
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 118
  Identification: 0x6a51 (27217)
> Flags: 0x4000, Don't fragment
  Time to live: 128
  Protocol: TCP (6)
  Header checksum: 0xb60f [validation disabled]
  [Header checksum status: Unverified]
  Source: 10.21.6.171
  Destination: 3.219.197.134
Transmission Control Protocol, Src Port: 49796, Dst Port: 80, Seq: 1, Ack: 1, Len: 78
  Source Port: 49796
  Destination Port: 80
  [Stream index: 0]
  [TCP Segment Len: 78]
  Sequence number: 1    (relative sequence number)
  [Next sequence number: 79    (relative sequence number)]
  Acknowledgment number: 1    (relative ack number)
  0101 .... = Header Length: 20 bytes (5)
> Flags: 0x018 (PSH, ACK)
  Window size value: 68
  [Calculated window size: 17408]
  [Window size scaling factor: 256]
  Checksum: 0xc2b8 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
```

TTL: 222



```
Time to live: 222
```

Protocol: TCP

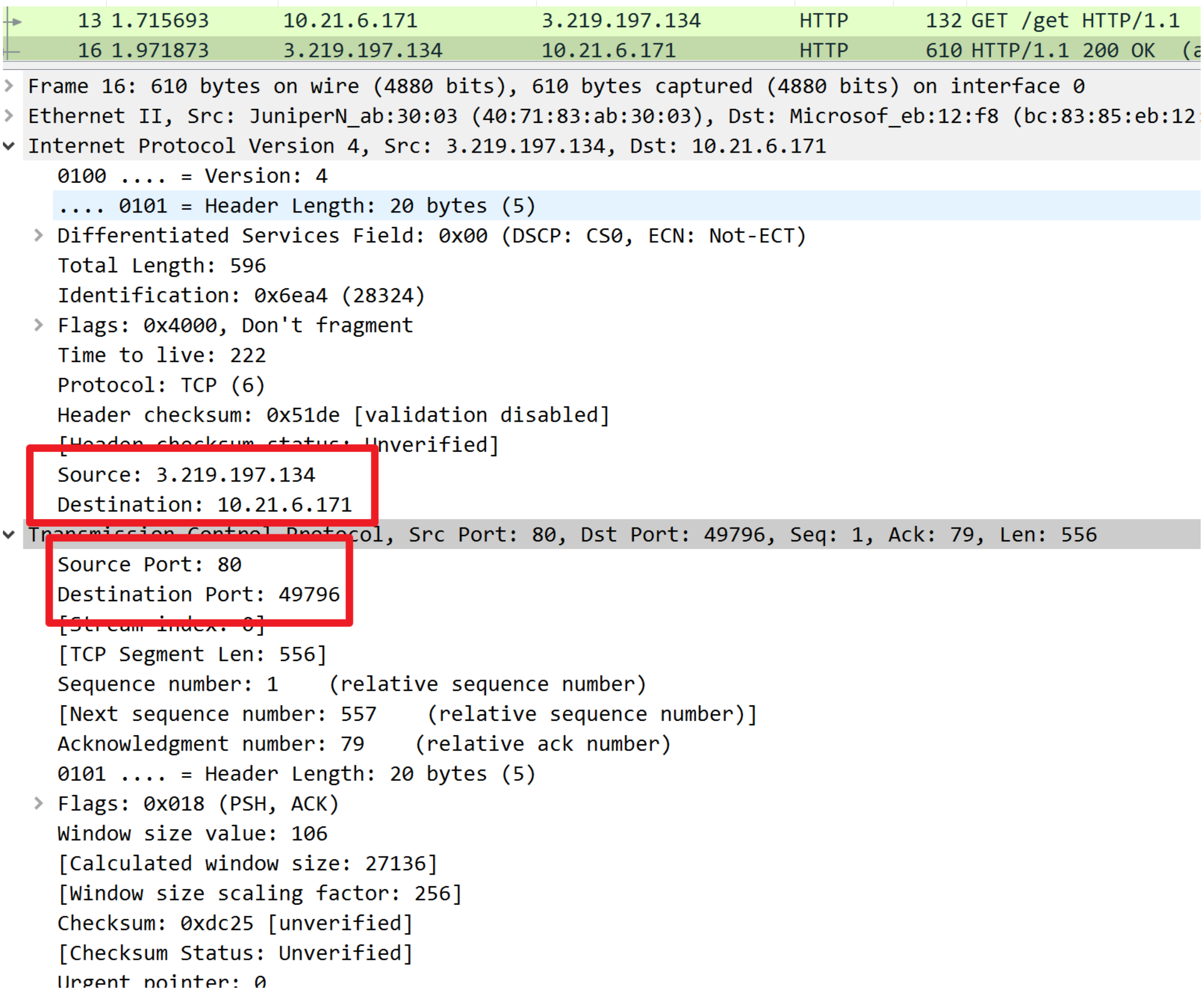


```
Protocol: TCP (6)
```

In response:
Src ip addr: 3.219.197.134,
Dsr ip addr: 10.21.6.171,

Src Port: 80,

Dst Port: 49796



```
   13 1.715693     10.21.6.171       3.219.197.134     HTTP      132 GET /get HTTP/1.1
   16 1.971873     3.219.197.134     10.21.6.171       HTTP      610 HTTP/1.1 200 OK  (a
> Frame 16: 610 bytes on wire (4880 bits), 610 bytes captured (4880 bits) on interface 0
> Ethernet II, Src: JuniperN_ab:30:03 (40:71:83:ab:30:03), Dst: Microsof_eb:12:f8 (bc:83:85:eb:12
v Internet Protocol Version 4, Src: 3.219.197.134, Dst: 10.21.6.171
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 596
    Identification: 0x6ea4 (28324)
  > Flags: 0x4000, Don't fragment
    Time to live: 222
    Protocol: TCP (6)
    Header checksum: 0x51de [validation disabled]
    [Header checksum status: Unverified]
    Source: 3.219.197.134
    Destination: 10.21.6.171
v Transmission Control Protocol, Src Port: 80, Dst Port: 49796, Seq: 1, Ack: 79, Len: 556
    Source Port: 80
    Destination Port: 49796
    [Stream index: 0]
    [TCP Segment Len: 556]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 557    (relative sequence number)]
    Acknowledgment number: 79    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window size value: 106
    [Calculated window size: 27136]
    [Window size scaling factor: 256]
    Checksum: 0xdc25 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
```