

# CS305-2022Spring Lab3 Report

Name: Yitong WANG [11910104@mail.sustech.edu.cn](mailto:11910104@mail.sustech.edu.cn)

Student ID: 11910104

Lab Time: Thursday 10:20 a.m. to 12:10 p.m.

Lab Teacher: Qing WANG [wangq9@mail.sustech.edu.cn](mailto:wangq9@mail.sustech.edu.cn)

Lab SA:

- Siyu LIU [11912935@mail.sustech.edu.cn](mailto:11912935@mail.sustech.edu.cn)
- Xingying ZHENG [11912039@mail.sustech.edu.cn](mailto:11912039@mail.sustech.edu.cn)

## Practice 1: Different Status Coded

### Q1-1: Handling More Circumstances of Different Status Code

Source code(Consider using different interfaces to trigger different response and status code):

```

import socket

ok200 = [b'HTTP/1.0 200 OK\r\n',
        b'Connection: close',
        b'Content-Type:text/html; charset=utf-8\r\n',
        b'\r\n',
        b'<html><body>Hello World!<body></html>\r\n',
        b'\r\n']

noContent204 = [b'HTTP/1.0 204 No Content\r\n',
                b'Connection: close',
                b'Content-Type:text/html; charset=utf-8\r\n',
                b'\r\n',
                b'<html><body>204 No Content<body></html>\r\n',
                b'\r\n']

badRequest400 = [b'HTTP/1.0 400 Bad Request\r\n',
                 b'Connection: close',
                 b'Content-Type:text/html; charset=utf-8\r\n',
                 b'\r\n',
                 b'<html><body>400 Bad Request<body></html>\r\n',
                 b'\r\n']

notFound404 = [b'HTTP/1.0 404 Not Found\r\n',
               b'Connection: close',
               b'Content-Type:text/html; charset=utf-8\r\n',
               b'\r\n',
               b'<html><body>404 Not Found<body></html>\r\n',
               b'\r\n']

intern500 = [b'HTTP/1.0 500 Internal server error\r\n',
             b'Connection: close',
             b'Content-Type:text/html; charset=utf-8\r\n',
             b'\r\n',
             b'<html><body>500 Internal server error<body></html>\r\n',
             b'\r\n']

service503 = [b'HTTP/1.0 503 Service unavailable\r\n',
              b'Connection: close',
              b'Content-Type:text/html; charset=utf-8\r\n',
              b'\r\n',
              b'<html><body>503 Service unavailable<body></html>\r\n',
              b'\r\n']

def web():
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.bind(('127.0.0.1', 8080))
    sock.listen(10)
    while True:

```

```

conn, address = sock.accept()
data = conn.recv(2048).decode().split('\r\n')
print(data[0])
print(data[0].split(' '))
res = notFound404

s = data[0].split(' ')[1]
if s == '/':
    res = ok200
elif s == '/bad-request':
    res = badRequest400
elif s == '/no-content':
    res = noContent204
elif s == '/internal':
    res = intern500
elif s == '/service':
    res = service503

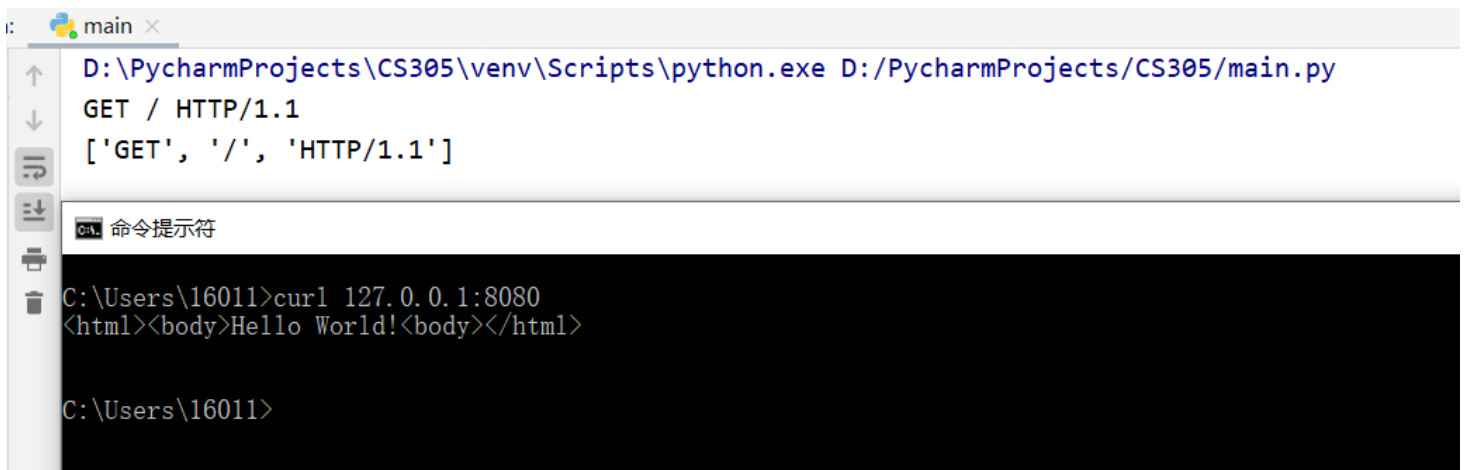
for line in res:
    conn.send(line)
conn.close()

if __name__ == "__main__":
    try:
        web()
    except KeyboardInterrupt:
        pass

```

Running Screenshots:

- 200 OK:



```

main x
D:\PycharmProjects\CS305\venv\Scripts\python.exe D:/PycharmProjects/CS305/main.py
GET / HTTP/1.1
['GET', '/', 'HTTP/1.1']
C:\Users\16011>curl 127.0.0.1:8080
<html><body>Hello World!<body></html>
C:\Users\16011>

```

- 204 No Content:



The screenshot shows a web browser window with a single tab titled 'main'. The address bar contains the URL 'D:\PycharmProjects\CS305\venv\Scripts\python.exe D:'. The page content displays the HTTP method and path: 'GET /no-content HTTP/1.1' and a list containing these elements: ['GET', '/no-content', 'HTTP/1.1']. Below the browser window, a Windows command prompt is open, showing the user's current directory as 'C:\Users\16011'. The user has entered the command 'curl 127.0.0.1:8080/no-content' twice, and the prompt is currently waiting for the next input.

```
D:\PycharmProjects\CS305\venv\Scripts\python.exe D:
GET /no-content HTTP/1.1
['GET', '/no-content', 'HTTP/1.1']

命令提示符
C:\Users\16011>curl 127.0.0.1:8080/no-content
C:\Users\16011>curl 127.0.0.1:8080/no-content
C:\Users\16011>_
```

- 400 Bad Request:



The screenshot shows a web browser window with a single tab titled 'main'. The address bar contains the URL 'D:\PycharmProjects\CS305\venv\Scripts\python.exe'. The page content displays the HTTP method and path for two requests: 'GET /no-content HTTP/1.1' and 'GET /bad-request HTTP/1.1', each followed by a list containing these elements: ['GET', '/no-content', 'HTTP/1.1'] and ['GET', '/bad-request', 'HTTP/1.1']. Below the browser window, a Windows command prompt is open, showing the user's current directory as 'C:\Users\16011'. The user has entered the command 'curl 127.0.0.1:8080/no-content' twice and 'curl 127.0.0.1:8080/bad-request' once. The output for the last command is an HTML response: '<html><body>400 Bad Request<body></html>'. The prompt is currently waiting for the next input.

```
D:\PycharmProjects\CS305\venv\Scripts\python.exe
GET /no-content HTTP/1.1
['GET', '/no-content', 'HTTP/1.1']
GET /bad-request HTTP/1.1
['GET', '/bad-request', 'HTTP/1.1']

命令提示符
C:\Users\16011>curl 127.0.0.1:8080/no-content
C:\Users\16011>curl 127.0.0.1:8080/no-content
C:\Users\16011>curl 127.0.0.1:8080/bad-request
<html><body>400 Bad Request<body></html>
C:\Users\16011>
```

- 404 Not Found:

```
C:\Program Files\Python\Python38\Scripts\python.exe C:\Program Files\Python\Python38\Scripts\python.exe  
GET /no-content HTTP/1.1  
['GET', '/no-content', 'HTTP/1.1']  
GET /bad-request HTTP/1.1  
['GET', '/bad-request', 'HTTP/1.1']  
GET /not-exist HTTP/1.1  
['GET', '/not-exist', 'HTTP/1.1']
```

命令提示符

```
C:\Users\16011>curl 127.0.0.1:8080/no-content  
C:\Users\16011>curl 127.0.0.1:8080/no-content  
C:\Users\16011>curl 127.0.0.1:8080/bad-request  
<html><body>400 Bad Request<body></html>  
  
C:\Users\16011>curl 127.0.0.1:8080/not-exist  
<html><body>404 Not Found<body></html>  
  
C:\Users\16011>
```

- 500 Internal Server Error:

```
GET /no-content HTTP/1.1
['GET', '/no-content', 'HTTP/1.1']
GET /bad-request HTTP/1.1
['GET', '/bad-request', 'HTTP/1.1']
GET /not-exist HTTP/1.1
['GET', '/not-exist', 'HTTP/1.1']
GET /internal HTTP/1.1
['GET', '/internal', 'HTTP/1.1']
GET /internal HTTP/1.1
['GET', '/internal', 'HTTP/1.1']
```

C:\> 命令提示符

```
C:\Users\16011>curl 127.0.0.1:8080/internal
<html><body>500 Internal server error<body></html>

C:\Users\16011>_
```

- 503 Service unavailable:

```
GET /service HTTP/1.1
['GET', '/service', 'HTTP/1.1']
```

C:\> 命令提示符

```
C:\Users\16011>curl 127.0.0.1:8080/internal
<html><body>500 Internal server error<body></html>

C:\Users\16011>curl 127.0.0.1:8080/service
<html><body>503 Service unavailable<body></html>

C:\Users\16011>_
```

## Q1-2: Use Wireshark to Capture and Analyze the Packets

- 200 OK:

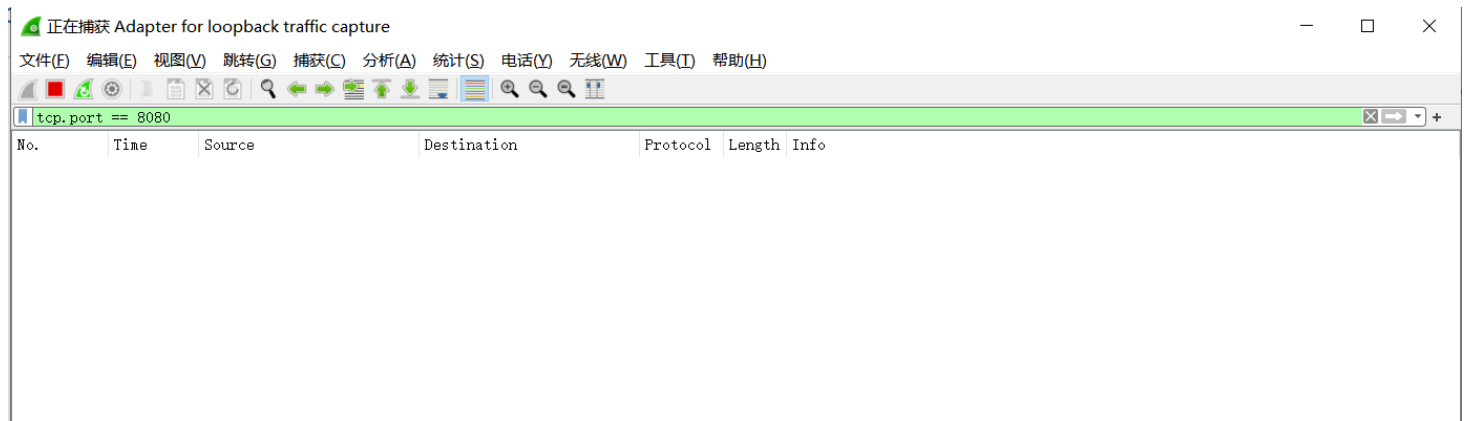
First run the python script and enable the localhost server:

```
python3 main.py
```

Then open Wireshark and set the display filter:

```
tcp.port == 8080
```

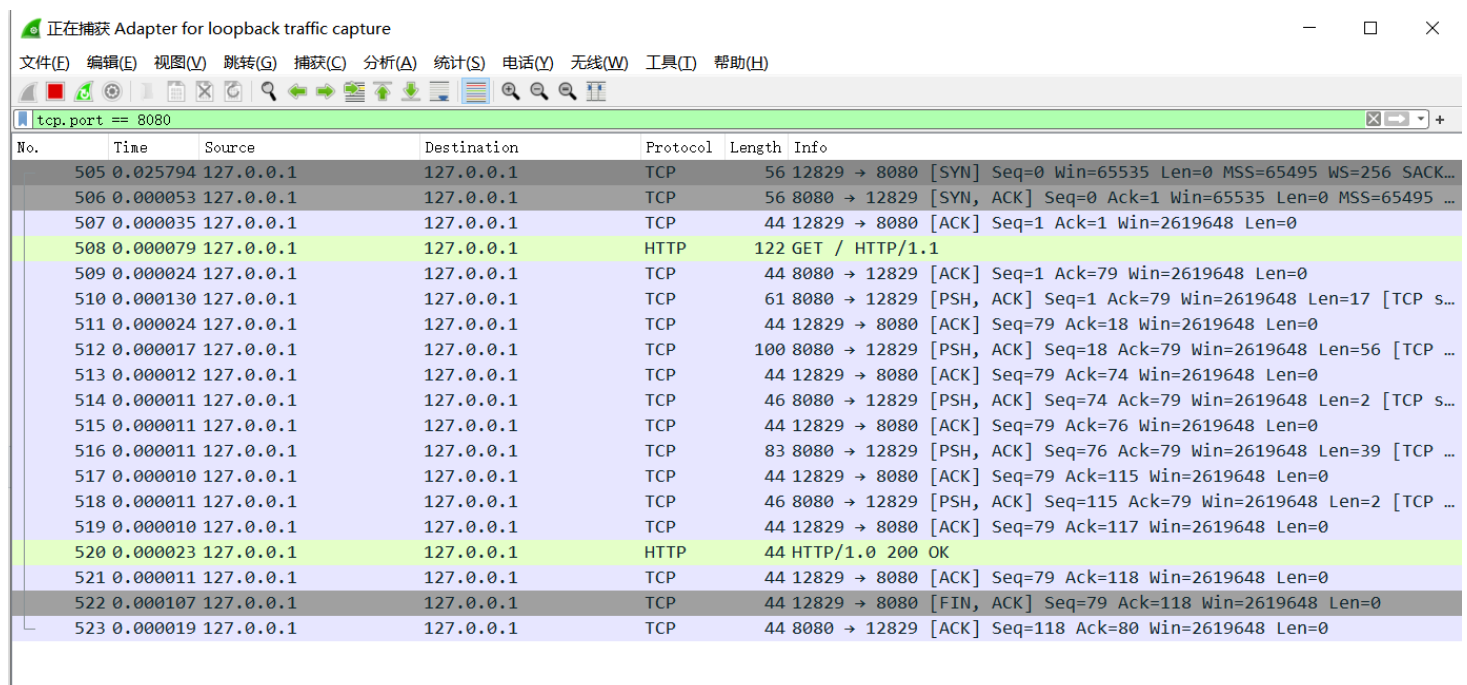
Start capturing packets:



Send request to the server by typing the following in the command line:

```
curl 127.0.0.1:8080/
```

Now we can get the packets from the Wireshark:



Then we select this packet:

170	0.000021	127.0.0.1	127.0.0.1	HTTP	44	HTTP/1.0 200 OK
-----	----------	-----------	-----------	------	----	-----------------

Source IP and Destination IP, Source Port and Destination Port:

Wireshark · 分组 170 · Adapter for loopback traffic capture

> Flags: 0x40, Don't fragment  
...0 0000 0000 0000 = Fragment Offset: 0  
Time to Live: 128  
Protocol: TCP (6)  
Header Checksum: 0x0000 [validation disabled]  
[Header checksum status: Unverified]  
Source Address: 127.0.0.1  
Destination Address: 127.0.0.1

Transmission Control Protocol, Src Port: 8080, Dst Port: 5866, Seq: 117, Ack: 79, Len: 0  
Source Port: 8080  
Destination Port: 5866  
[Stream index: 16]  
[Conversation completeness: Complete, WITH\_DATA (31)]  
[TCP Segment Len: 0]  
Sequence Number: 117 (relative sequence number)  
Sequence Number (raw): 2431088563  
[Next Sequence Number: 118 (relative sequence number)]  
Acknowledgment Number: 79 (relative ack number)  
Acknowledgment number (raw): 3331799891

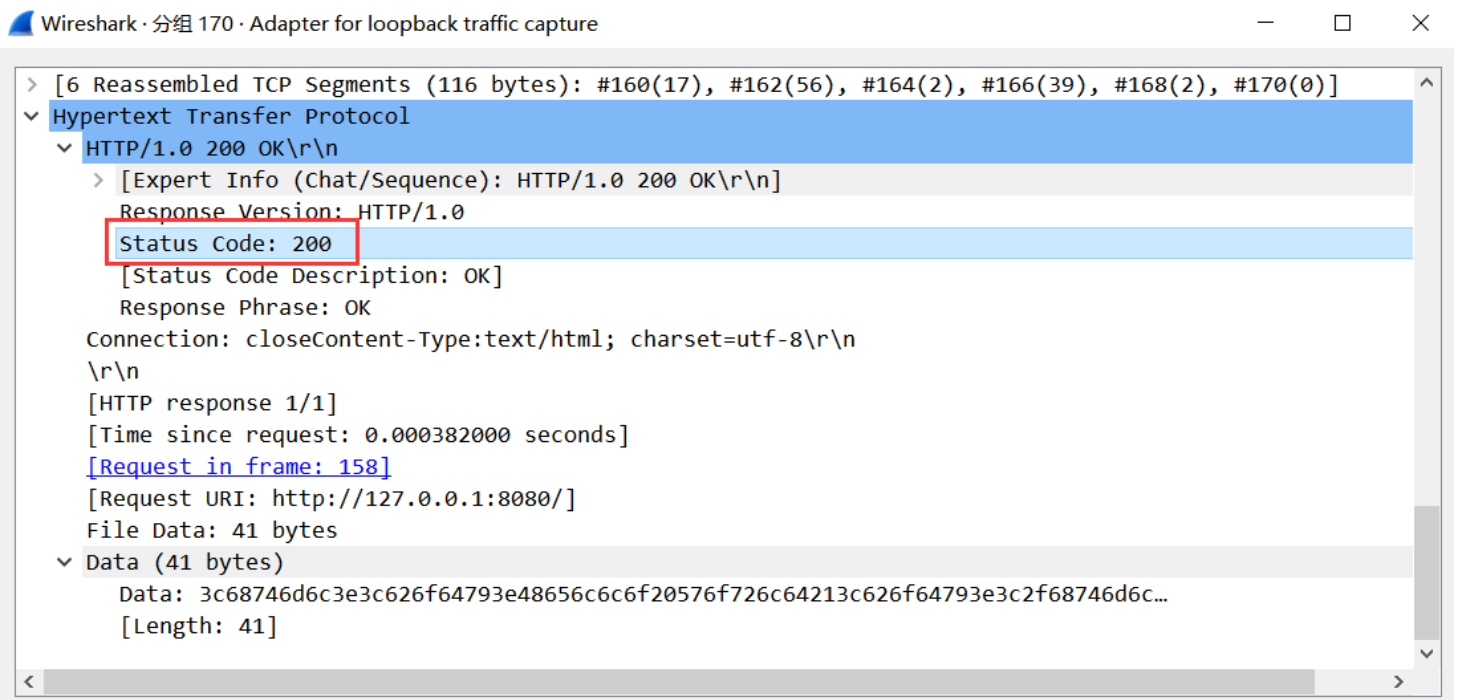
0000	02 00 00 00 45 00 00 28	4a e9 40 00 80 06 00 00	....E..( J.@.....
0010	7f 00 00 01 7f 00 00 01	1f 90 16 ea 90 e7 77 b3	.....w.
0020	c6 97 3b 53 50 11 27 f9	48 d8 00 00	..;SP.' H...

Frame (44 bytes) Reassembled TCP (116 bytes)

Close Help

Response Status Code:





So for the request whose status code is 200 OK, we get the information in the packets using Wireshark:

Source IP: 127.0.0.1  
Destination IP: 127.0.0.1  
Source Port: 8080  
Destination Port: 5866  
Response Status Code: 200

- 204 No Content:

```

Protocol: TCP (6)
Header Checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]
Source Address: 127.0.0.1
Destination Address: 127.0.0.1
Transmission Control Protocol, Src Port: 8080, Dst Port: 2358, Seq: 82, Ack: 89, Len: 2
Source Port: 8080
Destination Port: 2358
[Stream index: 6]
[Conversation completeness: Complete, WITH_DATA (47)]
[TCP Segment Len: 2]
Sequence Number: 82 (relative sequence number)

```

```

TCP segment data (2 bytes)
> [3 Reassembled TCP Segments (83 bytes): #52(25), #54(56), #56(2)]
v Hypertext Transfer Protocol
  v HTTP/1.0 204 No Content\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.0 204 No Content\r\n]
      Response Version: HTTP/1.0
      Status Code: 204
      [Status Code Description: No Content]
      Response Phrase: No Content
      Connection: closeContent-Type:text/html; charset=utf-8\r\n

```

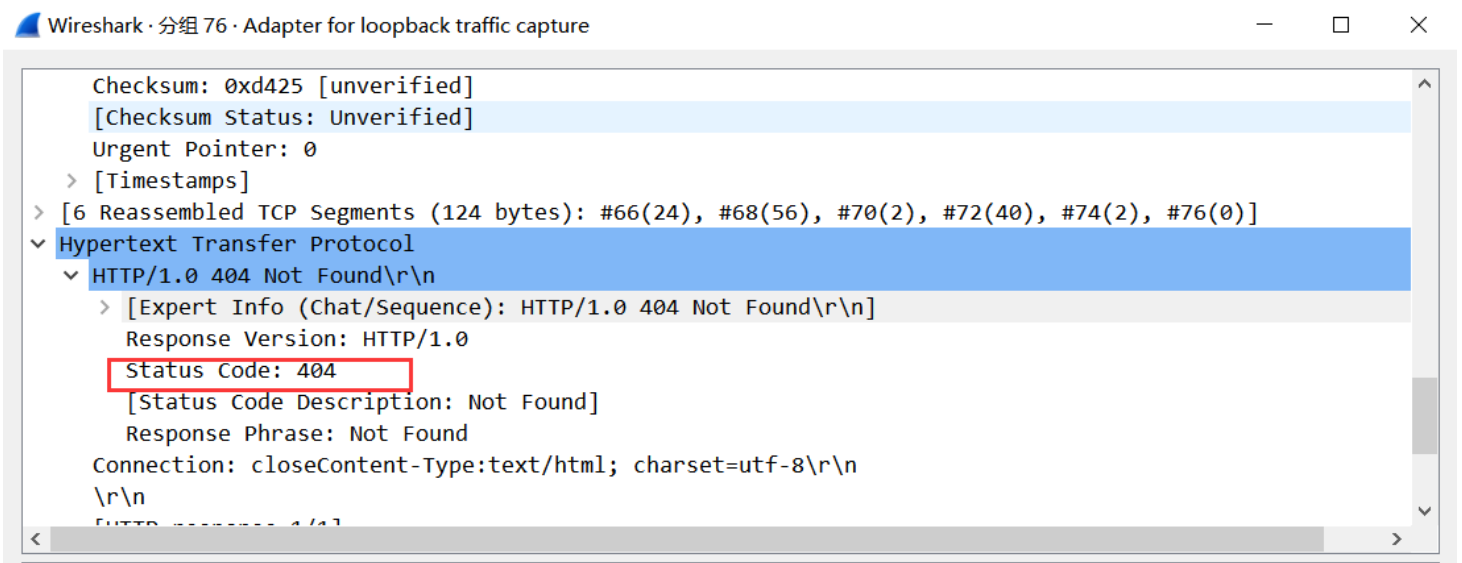
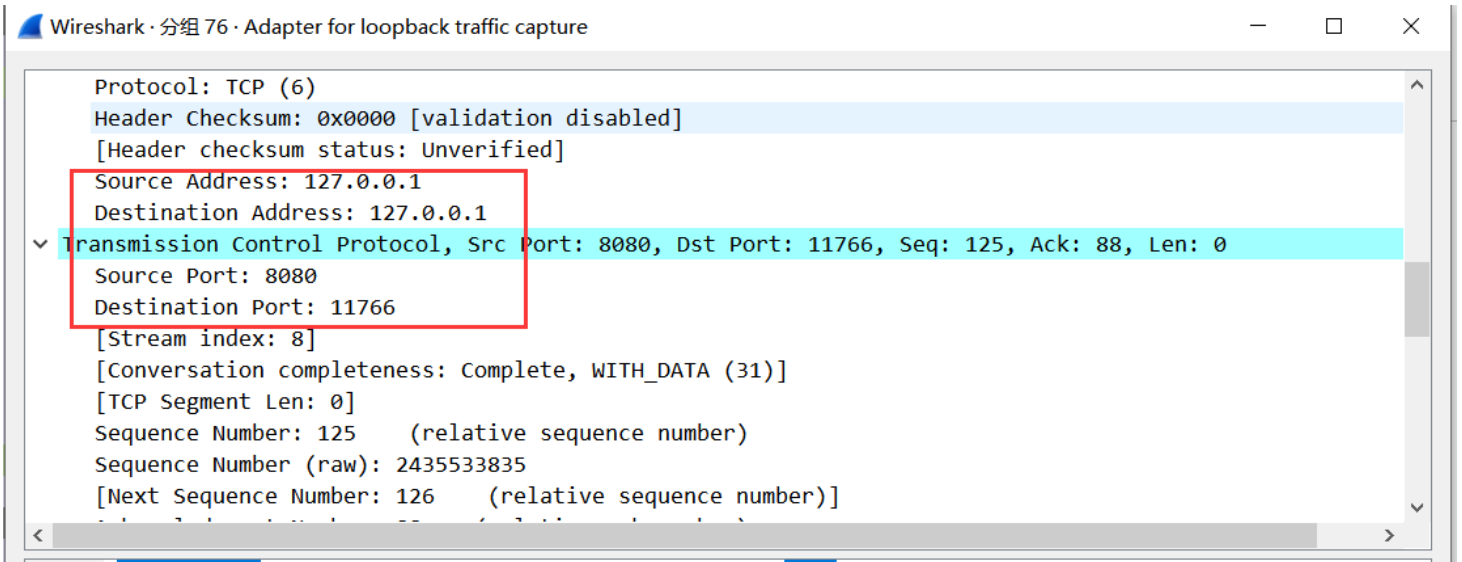
- 400 Bad Request:

```

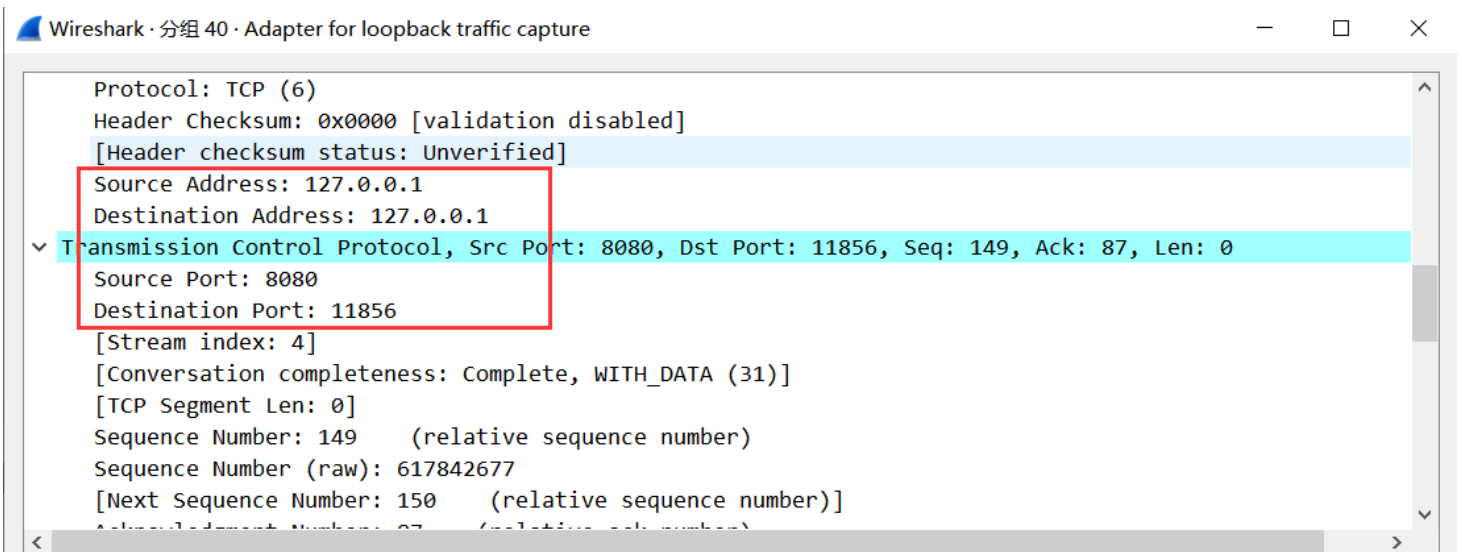
Identification: 0x546d (21613)
> Flags: 0x40, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: TCP (6)
Header Checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]
Source Address: 127.0.0.1
Destination Address: 127.0.0.1
v Transmission Control Protocol, Src Port: 8080, Dst Port: 11672, Seq: 129, Ack: 90, Len: 0
  Source Port: 8080
  Destination Port: 11672
  [Stream index: 6]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 129 (relative sequence number)
  Sequence Number (seq): 3733000074
  [Timestamps]
> [6 Reassembled TCP Segments (128 bytes): #48(26), #50(56), #52(2), #54(42), #56(2), #58(0)]
v Hypertext Transfer Protocol
  v HTTP/1.0 400 Bad Request\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.0 400 Bad Request\r\n]
      Response Version: HTTP/1.0
      Status Code: 400
      [Status Code Description: Bad Request]
      Response Phrase: Bad Request
      Connection: closeContent-Type:text/html; charset=utf-8\r\n
      \r\n

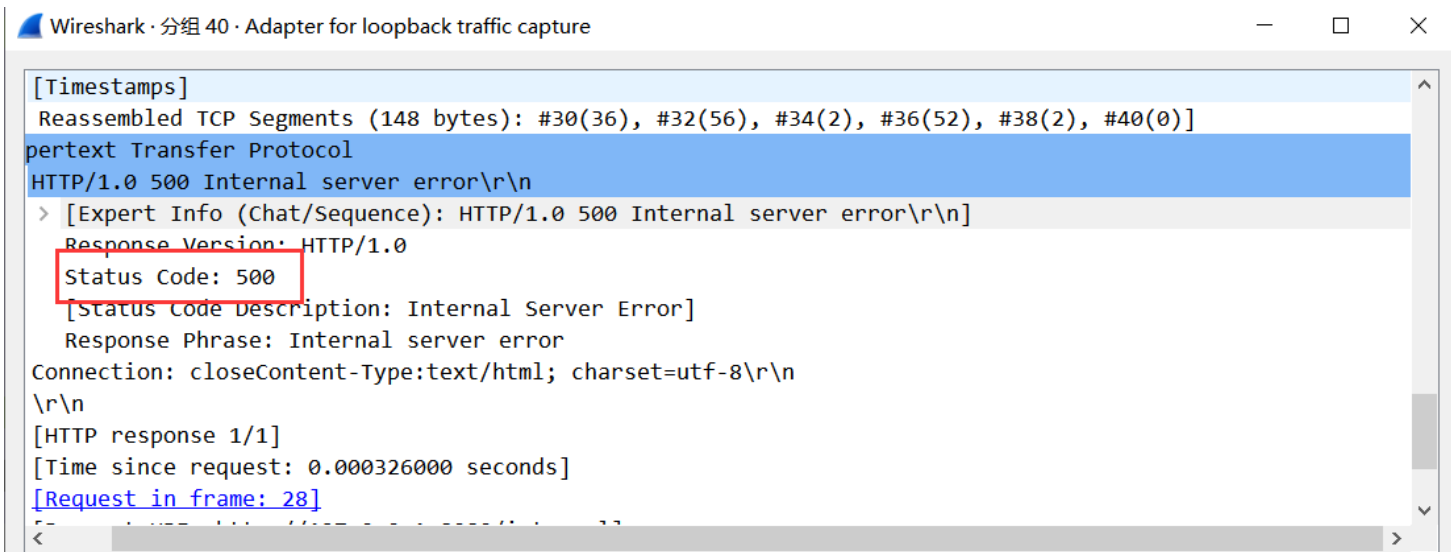
```

- 404 Not Found:

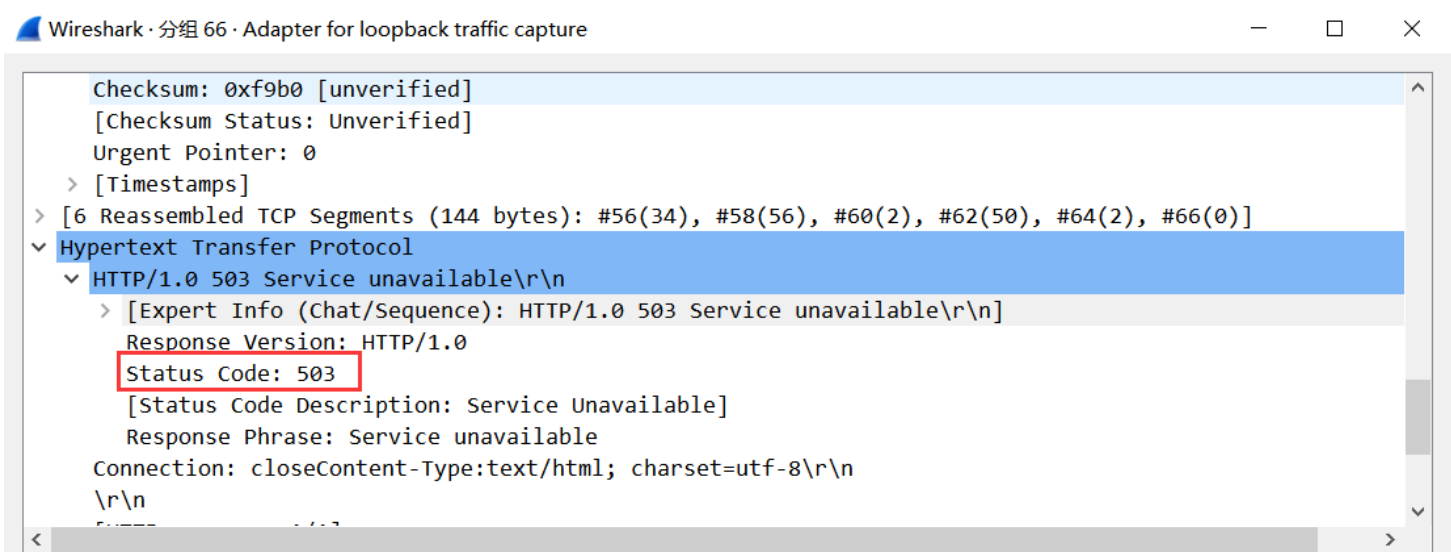
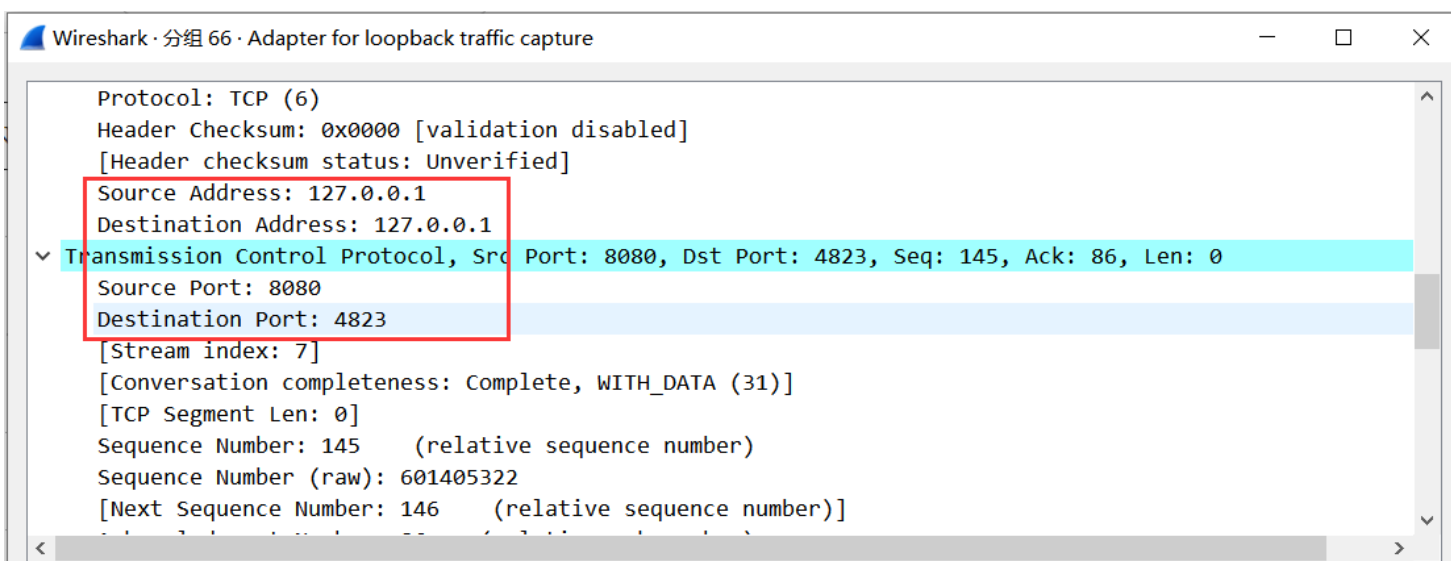


- 500 Internal Server Error:





- 503 Service Unavailable:



These running result can be seen in the following table:

Type	Source IP	Destination IP	Source Port Number	Destination Port Number	Response Status Code
200 OK	127.0.0.1	127.0.0.1	8080	5866	200
204 No Content	127.0.0.1	127.0.0.1	8080	2358	204
400 Bad Request	127.0.0.1	127.0.0.1	8080	11672	400
404 Not Found	127.0.0.1	127.0.0.1	8080	11766	404
500 Internal Server Error	127.0.0.1	127.0.0.1	8080	11856	500
503 Service unavailable	127.0.0.1	127.0.0.1	8080	4823	503