

Server – Client Program by Socket

2022년 04월 01일 (금)
국민대학교
최건희 (20172268)

1

Summary

2

Code Review

2.1 Client



2.2 Server



3

WireShark Capture



1 Summary

2 Code Review

2.1 Client



2.2 Server



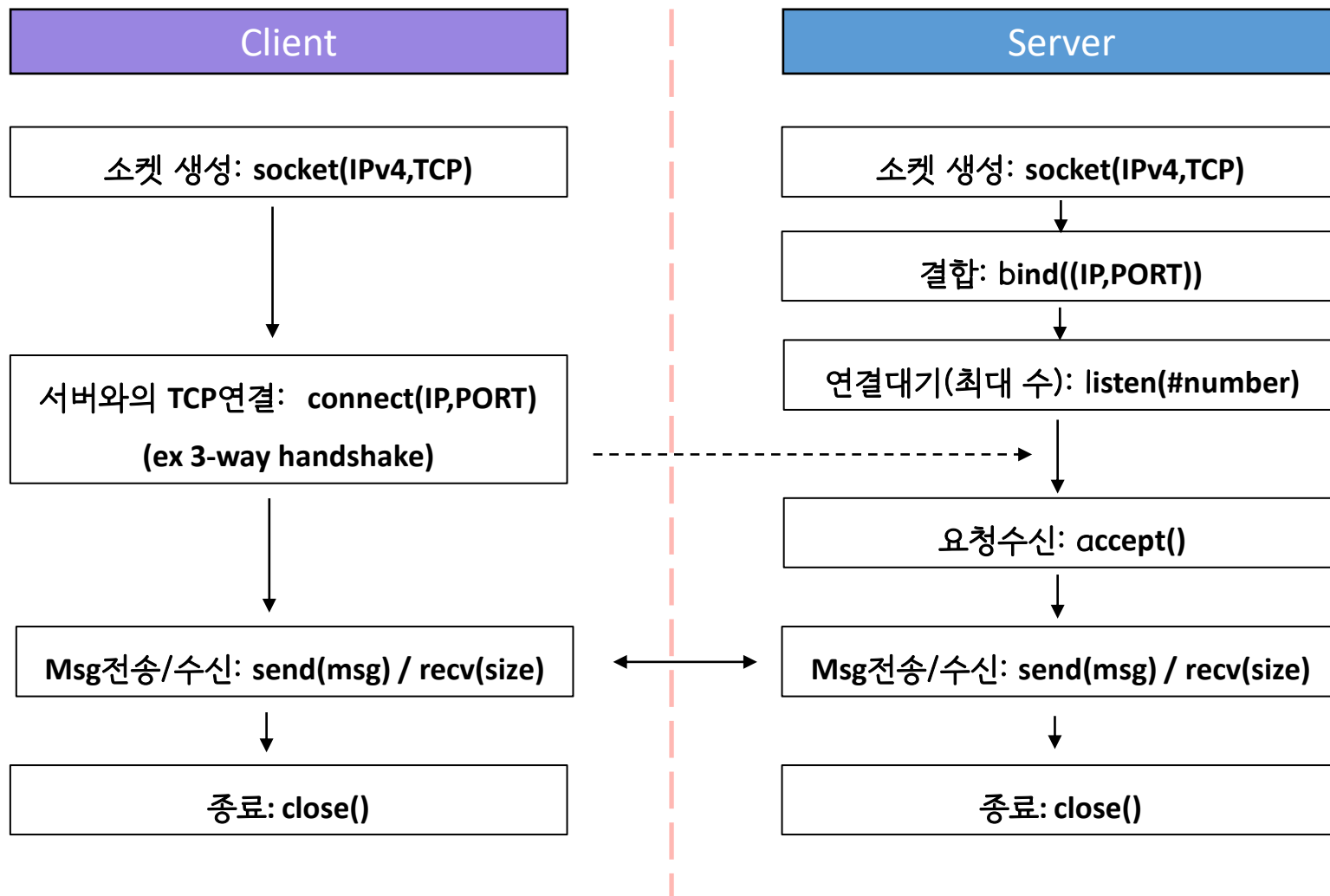
3 WireShark Capture

■ 소켓 통신을 활용하여 Server, Client 프로그램 작성

❖ 주요 내용

- 다양한 Method에 대한 다양한 응답코드
- HTTP 명령어 수행에 따른 실제 파일 생성, 수정, 내용전송
- HTTP 응답 header lines & data 구현
- Wireshark를 이용한 캡처 결과

Socket API



1

Summary



2

Code Review

2.1 Client



2.2 Server



3

WireShark Capture

❑ CODE

❖ 수행 환경: Colaboratory

➤ HTTP Request Message: 구현 내용 예시

Request line	POST /5.txt HTTP/1.0
Header lines	Host: www.easy.com\r\n data\r\n (<i>if input data</i>) \r\n

❑ CODE

- ❖ 수행 환경: Colaboratory
- 통신을 위해서는 Server IP(도착지), Port 번호(http:80)가 필요하다
- 소켓 생성시 IPv4(AF_INET), TCP(SOCK_STREAM) 통신임을 정의해준다.

```
from socket import *  
  
serverName = '1.248.223.228'  
serverPort = 80  
clientSocket = socket(AF_INET, SOCK_STREAM) ✓ 소켓 생성  
  
clientSocket.connect((serverName, serverPort)) ✓ 연결 요청
```


■ CODE

❖ 수행 환경: Colaboratory

- 클라이언트는 서버에게 HTTP프로토콜 양식에 맞는 요청을 보낸다.
- 서버로부터 받은 메시지를 출력해주고 TCP연결을 종료한다.

```
clientSocket.sendall('GET /1.txt HTTP/1.0\r\nHost: www.easy.com\r\n\r\n'.encode()) ✓ HTTP Request  
  
modifiedSentence = clientSocket.recv(1024) ✓ 응답메시지 수신  
  
print(modifiedSentence.decode())  
  
clientSocket.close() ✓ 연결 종료
```

1

Summary



2

Code Review

2.1 Client



2.2 Server



3

WireShark Capture

■ CODE

❖ 수행 환경: Local PC(이더넷 IPv4주소: 1.248.223.228)

➤ HTTP Response Message: 구현 내용 예시

Status line	HTTP/1.0 200 OK
Header lines	Date: Thu, 31 Mar 2022 23:50:25 KST Server: Easyboan Local PC Last-Modified: Thu Mar 31 23:50:01 2022 Content-Type: text/txt
Data	{'aaa', 'bbb', 'ccc'}

■ CODE

- ❖ 수행 환경: Local PC(이더넷 IPv4주소: 1.248.223.228)
- IPv4, TCP 통신을 위한 소켓 생성: `socket(AF_INET, SOCK_STREAM)`
- 통신을 위해 IP, PORT 결합 및 최대 연결의 수 지정 후 대기, 및 수신

```
from socket import *
from datetime import datetime
import os.path
import time

IP = '1.248.223.228'
server_name = 'Easyboan Local PC'
serverPort = 80
serverSocket = socket(AF_INET, SOCK_STREAM) ✓ 소켓 생성
serverSocket.bind((IP, serverPort)) ✓ IP, PORT 결합
serverSocket.listen(1) ✓ 최대 연결 수 1로 지정 후 연결 요청 대기
#연결의 최대 수
print('The sever is ready to receive!!!')

connectionSocket, addr = serverSocket.accept() ✓ 연결 요청 수신 후 연결
```

■ CODE

❖ HTTP 요청 메시지 수신

- 연결된 Client로부터 받은 HTTP Request message를 처리해준다.
- HTTP Request의 형태를 띄지 않는(길이가 짧은 메시지)에 대해 오류코드 전송

```
while True:

    sentence = connectionSocket.recv(1024).decode() ✓ 요청메시지 수신
    if not sentence:
        break
    sentence_data = sentence.split()
    if(len(sentence_data) < 4): ✓ 요청오류 Check!
        response_data = "{0} 400 Bad Request\r\nDate: {1}\r\nServer: {2}\r\n\r\n".format("HTTP/1.0", datetime.now().strftime('%a, %d %b %Y %H:%M:%S KST'), server_name)
        connectionSocket.send(response_data.encode())
        break

    request_method = sentence_data[0] ✓ HTTP 요청 처리를 위해 각각 변수 지정
    url = sentence_data[1]
    version = sentence_data[2]
    host = sentence_data[4]
    if(len(sentence_data) > 5):
        data = sentence_data[5]
```

■ CODE

- ❖ 수신된 메시지 보여주기
- 서버에 Client로부터 받은 메시지가 무엇인지 보여준다.
- 서버에 지정된 폴더 안에 url이 들어있는지 확인을 위해 경로를 지정한다.

```
print("I received ", addr, sentence)
print("This is method:", request_method)
print("This is url:", url)
print("This is version:", version)
print("This is host:", host)

file = "C:/Users/easyboan/Desktop/Network_middle"+url
```

■ CODE

❖ GET & HEAD에 대한 처리

- 요청된 파일의 존재유무를 확인하고 응답한다.
- 파일 존재 시, GET의 경우에만 object를 전송해준다.

```
if(request_method == 'GET' or request_method == 'HEAD'):  
  
    if(os.path.exists(file)): ✓ File 존재 Check!  
        atime = time.ctime(os.path.getatime(file)) ✓ 최근 수정일 저장  
        if(request_method == 'HEAD'):  
            response_data = "{0} 200 OK#r#nDate: {1}#r#nServer: {2}#r#n#r#n".format(version, datetime.now().strftime('%a, %d %b %Y %H:%M:%S KST'),server_name)  
        else:  
            f = open(file, 'r')  
            fileType = os.path.splitext(file)[1]  
            lines = f.readlines()  
            response_data = "{0} 200 OK#r#nDate: {1}#r#nServer: {2}#r#nLast-Modified: {3}#r#nContent-Type: text/{4}#r#n#r#ndata: {5}" .format(version, datetime.now(),  
                                                                    ✓ GET의 경우 data(object)함께 전송  
                                                                    lines, fileType)  
            f.close()  
        else: ✓ File 없을 시 오류코드 전송  
            response_data = "{0} 404 Not Found#r#nDate: {1}#r#nServer: {2}#r#n#r#n".format(version, datetime.now().strftime('%a, %d %b %Y %H:%M:%S KST'),server_name)
```


■ CODE

❖ POST에 대한 처리

- 허가된 호스트인지 확인하고 그렇지 않으면 오류코드를 전송한다.
- 파일이 존재한다면 수정, 존재하지 않는다면 생성을 통해 명령을 수행한다.

```
elif(request_method == 'POST'):
    if(host == "www.easy.com"): ✓ 허가된 호스트 name Check!
        if(os.path.exists(file)):
            atime = time.ctime(os.path.getatime(file))
            f = open(file, 'a') ✓ 파일 존재 시 수정
            f.write(data)
            fileTYPE = os.path.splitext(file)[1]
            response_data = "{0} 200 OK\r\nDate: {1}\r\nServer: {2}\r\nLast-Modified: {3}\r\nContent-Type: text/{4}\r\n\r\n".format(version, datetime.now(),
            f.close()
        else:
            f = open(file, 'w') ✓ 파일 생성
            f.write(data)
            fileTYPE = os.path.splitext(file)[1]
            atime = time.ctime(os.path.getatime(file))
            response_data = "{0} 200 OK\r\nDate: {1}\r\nServer: {2}\r\nLast-Modified: {3}\r\nContent-Type: text/{4}\r\n\r\n".format(version, datetime.now(),
            f.close()
    else: ✓ 허가되지 않은 호스트에 대한 오류 코드 전송
        response_data = "{0} 401 Authorization Required\r\nDate: {1}\r\nServer: {2}\r\n\r\n".format(version, datetime.now().strftime('%a, %d %b %Y %H:%M:%S
```

■ CODE

❖ PUT에 대한 처리

- 허가된 호스트인지 확인하고 그렇지 않으면 오류코드를 전송한다.
- 파일 존재 유/무와 관련 없이 요청에 의해 파일을 새로 쓴다.

```
elif(request_method == 'PUT'):
    if(host == "www.easy.com"):
        f = open(file, 'w')    ✓ 파일 생성 및 갱신 (수정의 의미가 아님)
        f.write(data)
        fileType = os.path.splitext(file)[1]
        atime = time.ctime(os.path.getatime(file))
        response_data = "{0} 200 OK\r\nDate: {1}\r\nServer: {2}\r\nLast-Modified: {3}\r\nContent-Type: text{4}\r\n\r\n".format(version, datetime.now().strftime('%a, %d %b %Y %H:%M:%S KST'), server_name, atime, fileType)
        f.close()
    else:
        response_data = "{0} 401 Authorization Required\r\nDate: {1}\r\nServer: {2}\r\n\r\n".format(version, datetime.now().strftime('%a, %d %b %Y %H:%M:%S KST'), server_name)
else:
    response_data = "{0} 405 Method Not Allowed\r\nDate: {1}\r\nServer: {2}\r\n\r\n".format(version, datetime.now().strftime('%a, %d %b %Y %H:%M:%S KST'), server_name)
    ✓ GET / HEAD / POST / PUT Method가 아닐 경우 현재 서버에서 처리하지 않음
connectionSocket.send(response_data.encode())
```

❑ CODE

❖ 연결 종료

➤ Client와의 통신을 종료한다.

```
#while문 종료
```

```
connectionSocket.close()  
serverSocket.close()
```

```
print('-end system-')
```

1

Summary

2

Code Review

2.1 Client



2.2 Server



3

WireShark Capture

1. 존재하는 파일에 대한 GET 요청

➤ 요청 메시지

```
clientSocket.sendall('GET /1.txt HTTP/1.0\r\nHost: www.easy.com\r\n\r\n'.encode())
```

➤ 응답 Message

```
HTTP/1.0 200 OK
Date: Fri, 01 Apr 2022 19:44:14 KST
Server: Easyboan Local PC
Last-Modified: Thu Mar 31 22:41:46 2022
Content-Type: text/txt
```

```
data: ['aaa\r\n', 'bbb\r\n', 'ccc']
```

➤ Wire Shark

```
Hypertext Transfer Protocol
> HTTP/1.0 200 OK\r\n
Date: Fri, 01 Apr 2022 19:44:14 KST\r\n
Server: Easyboan Local PC\r\n
Last-Modified: Thu Mar 31 22:41:46 2022\r\n
Content-Type: text/txt\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.071199000 seconds]
[Request in frame: 10685]
[Request URI: http://www.easy.com/1.txt]
File Data: 31 bytes
```

→ 10685	21.212267	35.236.147.188	1.248.223.228	HTTP	97 GET /1.txt HTTP/1.0
- 10767	21.283466	1.248.223.228	35.236.147.188	HTTP	54 HTTP/1.0 200 OK (text/txt)

❑ 2. 존재하지 않는 파일에 대한 GET 요청

➤ 요청 메시지

```
clientSocket.sendall('GET /2.txt HTTP/1.0\r\nHost: www.easy.com\r\n\r\n'.encode())
```

➤ 응답 Message

```
HTTP/1.0 404 Not Found
Date: Fri, 01 Apr 2022 19:54:52 KST
Server: Easyboan Local PC
```

➤ Wire Shark

```
Hypertext Transfer Protocol
> HTTP/1.0 404 Not Found\r\n
Date: Fri, 01 Apr 2022 19:54:52 KST\r\n
Server: Easyboan Local PC\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.071859000 seconds]
[Request in frame: 9283]
[Request URI: http://www.easy.com/2.txt]
```

9283	17.542205	35.236.147.188	1.248.223.228	HTTP	97 GET /2.txt HTTP/1.0
9367	17.614064	1.248.223.228	35.236.147.188	HTTP	54 HTTP/1.0 404 Not Found

3. 존재하는 파일에 대한 HEAD 요청

요청 메시지

```
clientSocket.sendall('HEAD /1.txt HTTP/1.0\r\nHost: www.easy.com\r\n\r\n'.encode())
```

응답 Message

```
HTTP/1.0 200 OK
Date: Fri, 01 Apr 2022 20:19:35 KST
Server: Easyboan Local PC
```

Wire Shark

```
✓ Hypertext Transfer Protocol
  > HTTP/1.0 200 OK\r\n
    Date: Fri, 01 Apr 2022 20:20:08 KST\r\n
    Server: Easyboan Local PC\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.001721000 seconds]
    [Request in frame: 2433]
    [Request URI: http://www.easy.com/1.txt]
```

2433	4.865666	35.236.147.188	1.248.223.228	HTTP	98	HEAD /1.txt HTTP/1.0
2434	4.867387	1.248.223.228	35.236.147.188	HTTP	137	HTTP/1.0 200 OK

4. 존재하는 파일에 대한 POST 요청 (File 수정)

요청 메시지

```
clientSocket.sendall('POST /1.txt HTTP/1.0\r\nHost: www.easy.com\r\n\r\n'.encode())
```

응답 Message

HTTP/1.0 200 OK
Date: Fri, 01 Apr 2022 20:22:28 KST
Server: Easyboan Local PC
Last-Modified: Thu Mar 31 22:41:46 2022
Content-Type: text/txt

✓ 수정일
✓ 지난 접근

1 - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V)

aaa
bbb
ccc

→
✓ 파일 수정

1 - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) D

aaa
bbb
cccl_WANA_APPLE

Wire Shark

✓ Hypertext Transfer Protocol
> HTTP/1.0 200 OK\r\n
Date: Fri, 01 Apr 2022 20:22:28 KST\r\n
Server: Easyboan Local PC\r\n
Last-Modified: Thu Mar 31 22:41:46 2022\r\n
Content-Type: text/txt\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.071489000 seconds]
[\[Request in frame: 2140\]](#)
[Request URI: http://www.easy.com/1.txt]
File Data: 1 bytes

2140	4.228471	35.236.147.188	1.248.223.228	HTTP	112 POST /1.txt HTTP/1.0 Continuation
2166	4.299960	1.248.223.228	35.236.147.188	HTTP	54 HTTP/1.0 200 OK (text/txt)

5. 존재하지 않는 파일에 대한 POST 요청(File 생성)

요청 메시지

```
clientSocket.sendall('POST /2.txt HTTP/1.0\r\nHost: www.easy.com\r\nILOVESAMSUNG\r\n\r\n'.encode())
```

응답 Message

```
HTTP/1.0 200 OK
Date: Fri, 01 Apr 2022 20:26:18 KST
Server: Easyboan Local PC
Last-Modified: Fri Apr 1 20:26:18 2022
Content-Type: text/txt
```

✓ 일치

2 - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) !

ILOVESAMSUNG ✓ 파일 생성

Wire Shark

```
✓ Hypertext Transfer Protocol
  > HTTP/1.0 200 OK\r\n
    Date: Fri, 01 Apr 2022 20:26:18 KST\r\n
    Server: Easyboan Local PC\r\n
    Last-Modified: Fri Apr 1 20:26:18 2022\r\n
    Content-Type: text/txt\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.072406000 seconds]
    [Request in frame: 2289]
    [Request URI: http://www.easy.com/2.txt]
    File Data: 1 bytes
```

2289	4.669111	35.236.147.188	1.248.223.228	HTTP	112 POST /2.txt HTTP/1.0 Continuation
2362	4.741517	1.248.223.228	35.236.147.188	HTTP	54 HTTP/1.0 200 OK (text/txt)

❑ 6. 허가되지 않은 HOST의 PUT요청

➤ 요청 메시지

```
clientSocket.sendall('PUT /1.txt HTTP/1.0\r\nHost: www.hard.com\r\n\r\n'.encode())
```

✓ 허가되지 않은 호스트

➤ 응답 Message

```
HTTP/1.0 401 Authorization Required
Date: Fri, 01 Apr 2022 20:34:30 KST
Server: Easyboan Local PC
```

➤ Wire Shark

```
Hypertext Transfer Protocol
> HTTP/1.0 401 Authorization Required\r\n
  Date: Fri, 01 Apr 2022 20:34:30 KST\r\n
  Server: Easyboan Local PC\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.071714000 seconds]
  [Request in frame: 2545]
  [Request URI: http://www.hard.com/1.txt]
```

2545	4.895770	35.236.147.188	1.248.223.228	HTTP	111 PUT /1.txt HTTP/1.0 Continuation
2641	4.967484	1.248.223.228	35.236.147.188	HTTP	54 HTTP/1.0 401 Authorization Required

7. 서버에서 제공하지 않는 Method에 대한 처리

요청 메시지

```
clientSocket.sendall('DELETE /1.txt HTTP/1.0\r\nHost: www.easy.com\r\n\r\n'.encode())
```

✓ 제공하지 않는 Method

응답 Message

```
HTTP/1.0 405 Method Not Allowed
Date: Fri, 01 Apr 2022 20:37:49 KST
Server: Easyboan Local PC
```

Wire Shark

Hypertext Transfer Protocol

```
> HTTP/1.0 405 Method Not Allowed\r\n
  Date: Fri, 01 Apr 2022 20:37:49 KST\r\n
  Server: Easyboan Local PC\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.071251000 seconds]
  [Request in frame: 2619]
  [Request URI: http://www.easy.com/1.txt]
```

2619	5.277372	35.236.147.188	1.248.223.228	HTTP	114 DELETE /1.txt HTTP/1.0 Continuation
2686	5.348623	1.248.223.228	35.236.147.188	HTTP	54 HTTP/1.0 405 Method Not Allowed

❑ 별첨 – 서버에서 메시지 수신 시, print 문

```
PS C:\Users\easyboan\Desktop\20172268_최건희> python -u "c:\Users\easyboan\Desktop\20172268_최건희\TCPServer.py"
The sever is ready to receive!!
I received ('35.236.147.188', 60832) DELETE /1.txt HTTP/1.0
Host: www.easy.com
I_WANA_APPLE

This is method: DELETE
This is url: /1.txt
This is version: HTTP/1.0
This is host: www.easy.com
-end system-
```


Q&A

Email: easyboan@gmail.com