컴퓨터 네트워크(월4, 수3)

Assignment: Wireshark

이성원 교수님

컴퓨터정보공학부

2018202076

이연걸

2022-04-03

서론:

본 과제는 HTTP, UDP Protocol, TCP/IP 계층 그리고 DNS 대한 이해가 필요하다. 브라우저에 주소를 입력해 네트워크 작업을 시작하면 Wireshark를 사용해 패킷을 분석한다. 첫번째 과제에서는 cmd창에 ipconfig 를 입력하여 TCP/IP 설정, DNS 설정, IP주소 등을 받아온다. 두번째 과제에서는 HTTP Protocal, HTTP Header에는 어떤 정보가 포함되며 Request와 Response는 담고 있는 정보에서 어떤 차이가 있는지, TCP는 어떻게 사용되며 어떤 특징이 있는지 확인한다. 세번째 과제에서는 DNS에 대해 다루는데 nslookup이라는 네트워크 관리 명령어로 각 도메인들의 IP와 Name Server를 통한 도메인 검색, DNS query와 response 패킷 분석을 수행한다.

전체적으로 네트워크 통신시에 발생하는 패킷이 무엇이며, 패킷이 어떤 것을 담고 있으며, 패킷을 어떻게 분석할 것인가에 대한 과제이며 HTTP와 DNS에 대해 중점적으로 다룬다.

본문:

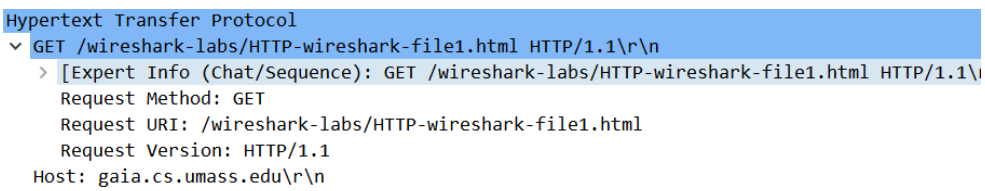
- ### Question #1:

  cmd창을 열어 ipconfig를 입력해 현재 IP를 찾는다.

```
C:₩Users₩광운MOOC센터>ipconfig

Windows IP 구성


무선 LAN 어댑터 로컬 영역 연결* 1:

   미디어 상태 . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :

이더넷 어댑터 VMware Network Adapter VMnet1:

   연결별 DNS 접미사. . . . :
   링크-로컬 IPv6 주소 . . . . : fe80::b13a:9c2f:665f:ff23%15
   IPv4 주소 . . . . . : 192.168.255.1
   서브넷 마스크 . . . . . : 255.255.255.0
   기본 게이트웨이 . . . . . :

이더넷 어댑터 VMware Network Adapter VMnet8:

   연결별 DNS 접미사. . . . :
   링크-로컬 IPv6 주소 . . . . : fe80::4fd:6fce:af67:1082%8
   IPv4 주소 . . . . . : 192.168.241.1
   서브넷 마스크 . . . . . : 255.255.255.0
   기본 게이트웨이 . . . . . :

무선 LAN 어댑터 Wi-Fi:

   연결별 DNS 접미사. . . . :
   링크-로컬 IPv6 주소 . . . . : fe80::ec2d:6b82:e344:7dd7%9
   IPv4 주소 . . . . . : 172.30.1.53
   서브넷 마스크 . . . . . : 255.255.255.0
   기본 게이트웨이 . . . . . : 172.30.1.254

이더넷 어댑터 Bluetooth 네트워크 연결:

   미디어 상태 . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :
```

- **Question #2**

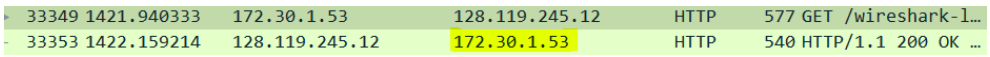
1. Request Version에서 HTTP 1.1을 사용함을 알 수 있다.



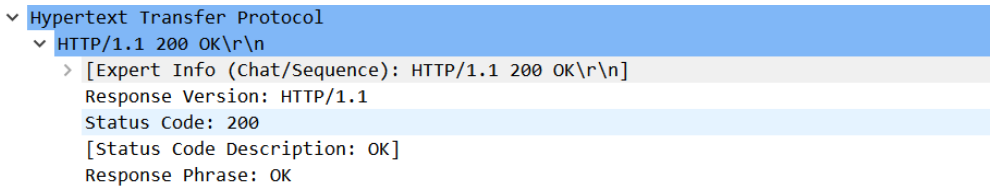
2. Accept-Language Header는 어떤 언어를 클라이언트가 이해할 수 있고 더 친숙한지, 지역 설정 중 어느 것이 더 선호되는지를 알려주는데 이를 통해 서버가 받을 수 있는 language를 알 수 있다.

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,ima
Accept-Encoding: gzip, deflate\r\n
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7\r\n
\r\n
[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
```

3. 172.30.1.53이다.



4. HTTP 응답 요청이 성공했음을 알려주는 200

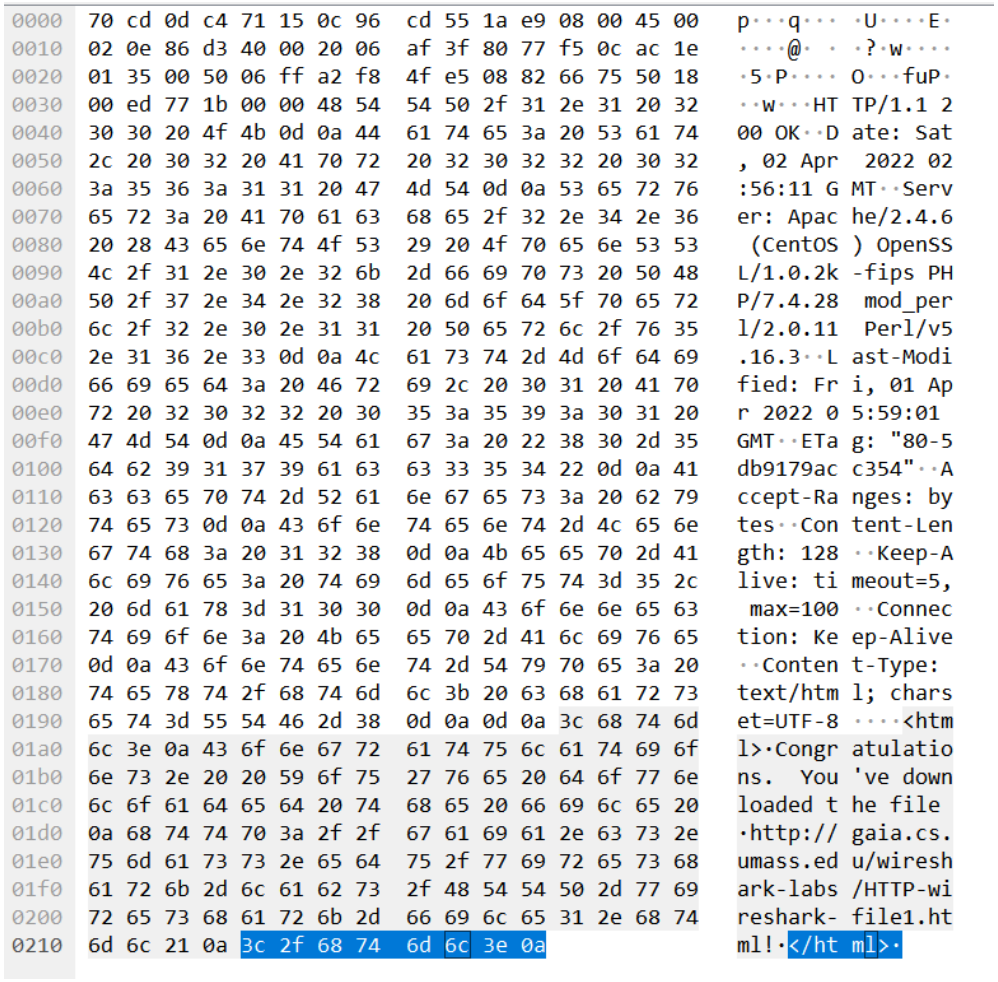


5. Last-Modified는 서버가 알고있는 파일의 가장 마지막 수정날짜와 시간이다.

```
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.28 mod_perl/2.0.11 Perl/v5.
Last-Modified: Fri, 01 Apr 2022 05:59:01 GMT\r\n
ETag: "80-5db9179acc354"\r\n
```

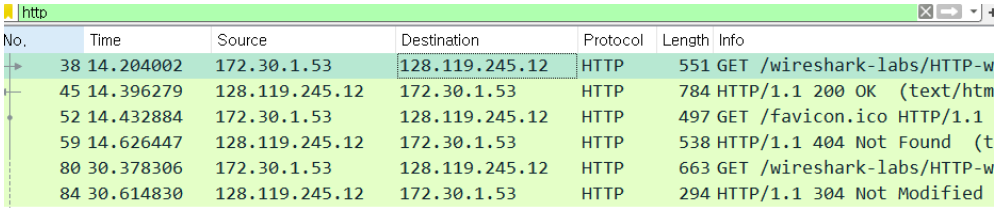
6. Content-Length는 byte 단위를 가지는 본문의 크기를 나타내므로 128 bytes이다.

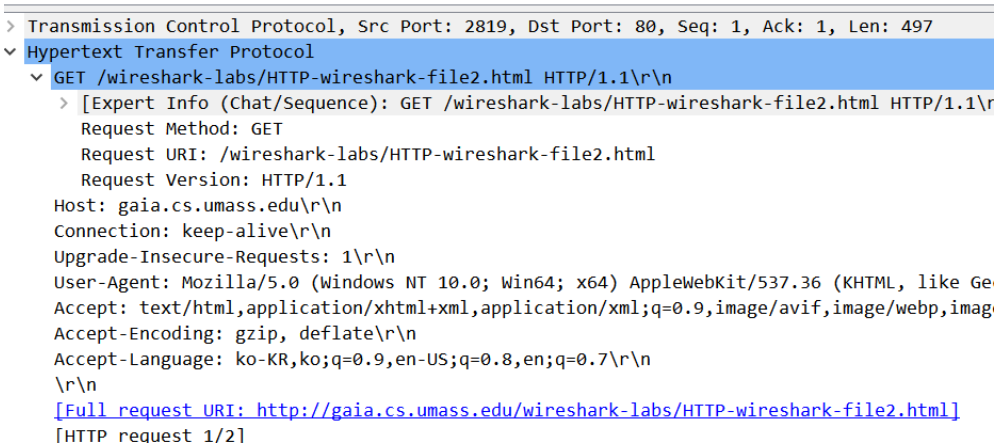
7. raw packet data를 조사해봐도 packet-listing window에 나타나지 않은 header는 존재하지 않았다.



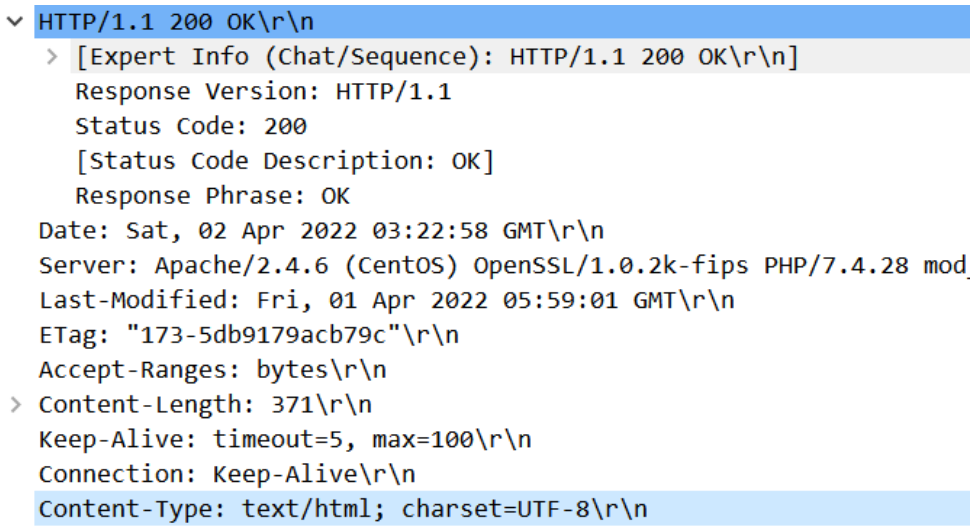
```
0000  70 cd 0d c4 71 15 0c 96  cd 55 1a e9 08 00 45 00   p···q··· ·U····E·
0010  02 0e 86 d3 40 00 20 06  af 3f 80 77 f5 0c ac 1e   ····@· · ·?·w····
0020  01 35 00 50 06 ff a2 f8  4f e5 08 82 66 75 50 18   ·5·P···· O···fuP·
0030  00 ed 77 1b 00 00 48 54  54 50 2f 31 2e 31 20 32   ··w···HT TP/1.1 2
0040  30 30 20 4f 4b 0d 0a 44  61 74 65 3a 20 53 61 74   00 OK··D ate: Sat
0050  2c 20 30 32 20 41 70 72  20 32 30 32 32 20 30 32   , 02 Apr  2022 02
0060  3a 35 36 3a 31 31 20 47  4d 54 0d 0a 53 65 72 76   :56:11 G MT··Serv
0070  65 72 3a 20 41 70 61 63  68 65 2f 32 2e 34 2e 36   er: Apac he/2.4.6
0080  20 28 43 65 6e 74 4f 53  29 20 4f 70 65 6e 53 53    (CentOS ) OpenSS
0090  4c 2f 31 2e 30 2e 32 6b  2d 66 69 70 73 20 50 48   L/1.0.2k -fips PH
00a0  50 2f 37 2e 34 2e 32 38  20 6d 6f 64 5f 70 65 72   P/7.4.28  mod_per
00b0  6c 2f 32 2e 30 2e 31 31  20 50 65 72 6c 2f 76 35   l/2.0.11  Perl/v5
00c0  2e 31 36 2e 33 0d 0a 4c  61 73 74 2d 4d 6f 64 69   .16.3··L ast-Modi
00d0  66 69 65 64 3a 20 46 72  69 2c 20 30 31 20 41 70   fied: Fr i, 01 Ap
00e0  72 20 32 30 32 32 20 30  35 3a 35 39 3a 30 31 20   r 2022 0 5:59:01
00f0  47 4d 54 0d 0a 45 54 61  67 3a 20 22 38 30 2d 35   GMT··ETa g: "80-5
0100  64 62 39 31 37 39 61 63  63 33 35 34 22 0d 0a 41   db9179ac c354"··A
0110  63 63 65 70 74 2d 52 61  6e 67 65 73 3a 20 62 79   ccept-Ra nges: by
0120  74 65 73 0d 0a 43 6f 6e  74 65 6e 74 2d 4c 65 6e   tes··Con tent-Len
0130  67 74 68 3a 20 31 32 38  0d 0a 4b 65 65 70 2d 41   gth: 128 ··Keep-A
0140  6c 69 76 65 3a 20 74 69  6d 65 6f 75 74 3d 35 2c   live: ti meout=5,
0150  20 6d 61 78 3d 31 30 30  0d 0a 43 6f 6e 6e 65 63    max=100 ··Connec
0160  74 69 6f 6e 3a 20 4b 65  65 70 2d 41 6c 69 76 65   tion: Ke ep-Alive
0170  0d 0a 43 6f 6e 74 65 6e  74 2d 54 79 70 65 3a 20   ··Conten t-Type:
0180  74 65 78 74 2f 68 74 6d  6c 3b 20 63 68 61 72 73   text/htm l; chars
0190  65 74 3d 55 54 46 2d 38  0d 0a 0d 0a 3c 68 74 6d   et=UTF-8 ····<htm
01a0  6c 3e 0a 43 6f 6e 67 72  61 74 75 6c 61 74 69 6f   l>·Congr atulatio
01b0  6e 73 2e 20 20 59 6f 75  27 76 65 20 64 6f 77 6e   ns.  You 've down
01c0  6c 6f 61 64 65 64 20 74  68 65 20 66 69 6c 65 20   loaded t he file
01d0  0a 68 74 74 70 3a 2f 2f  67 61 69 61 2e 63 73 2e   ·http:// gaia.cs.
01e0  75 6d 61 73 73 2e 65 64  75 2f 77 69 72 65 73 68   umass.ed u/wiresh
01f0  61 72 6b 2d 6c 61 62 73  2f 48 54 54 50 2d 77 69   ark-labs /HTTP-wi
0200  72 65 73 68 61 72 6b 2d  66 69 6c 65 31 2e 68 74   reshark- file1.ht
0210  6d 6c 21 0a 3c 2f 68 74  6d 6c 3e 0a               ml!·</ht ml>·
```

8. 첫번째 HTTP GET Request에서는 IF-MODIFIED-SINCE가 보이지 않는다.





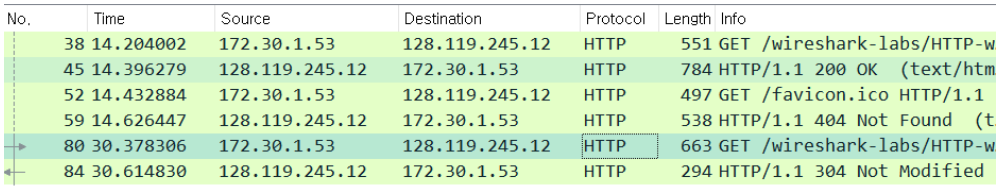
9. "HTTP/1.1 200 OK" 에서 제시된 링크로의 HTTP GET 요청이 정상적으로 수행되었음을 알 수 있고, Content-Length에서 본문의 크기가 371bytes라고 명시되어 있는데 이로서 서버가 확실히 가지고 있던 파일의 content를 return 해준 것을 알 수 있다.



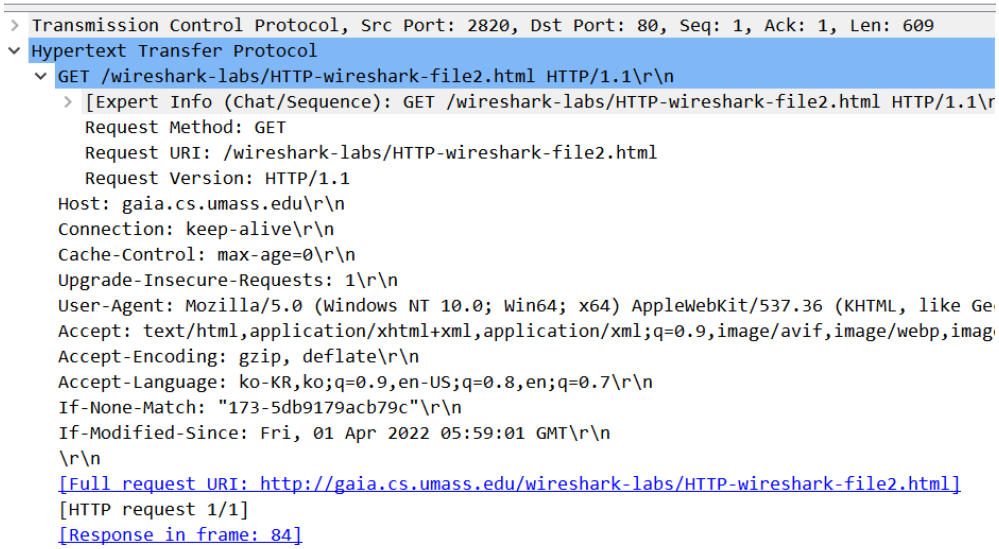
10. /favicon.ico 에 대한 GET요청을 무시하므로 663길이의 GET요청을 봤을 때

"If-Modified-Since: Fri, 01 Apr 2022 05:59:01 GMT"가 header에 포함된 것을 볼

수 있다.



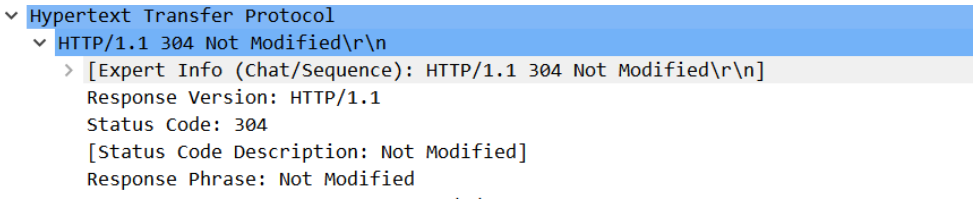
No.	Time	Source	Destination	Protocol	Length	Info
38	14.204002	172.30.1.53	128.119.245.12	HTTP	551	GET /wireshark-labs/HTTP-w
45	14.396279	128.119.245.12	172.30.1.53	HTTP	784	HTTP/1.1 200 OK (text/htm
52	14.432884	172.30.1.53	128.119.245.12	HTTP	497	GET /favicon.ico HTTP/1.1
59	14.626447	128.119.245.12	172.30.1.53	HTTP	538	HTTP/1.1 404 Not Found (t
80	30.378306	172.30.1.53	128.119.245.12	HTTP	663	GET /wireshark-labs/HTTP-w
84	30.614830	128.119.245.12	172.30.1.53	HTTP	294	HTTP/1.1 304 Not Modified



```
> Transmission Control Protocol, Src Port: 2820, Dst Port: 80, Seq: 1, Ack: 1, Len: 609
v Hypertext Transfer Protocol
  v GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r
      Request Method: GET
      Request URI: /wireshark-labs/HTTP-wireshark-file2.html
      Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7\r\n
    If-None-Match: "173-5db9179acb79c"\r\n
    If-Modified-Since: Fri, 01 Apr 2022 05:59:01 GMT\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
    [HTTP request 1/1]
    [Response in frame: 84]
```

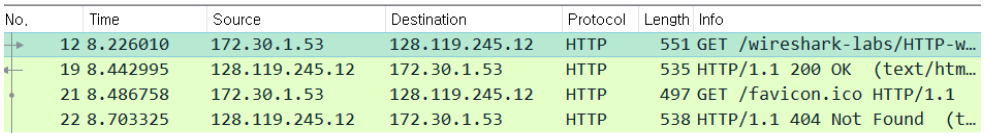
11. 두번째 HTTP GET response의 HTTP status code는 304 Not Modified이다.

이는 요청된 리소스를 서버가 재전송할 필요가 없음을 나타낸다. 이때 브라우저는 파일 content를 캐시에서 가져오기 때문에 문서가 수정되지 않는다. 따라서 서버는 파일 content를 return하지 않는다.



```
v Hypertext Transfer Protocol
  v HTTP/1.1 304 Not Modified\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]
      Response Version: HTTP/1.1
      Status Code: 304
      [Status Code Description: Not Modified]
      Response Phrase: Not Modified
```

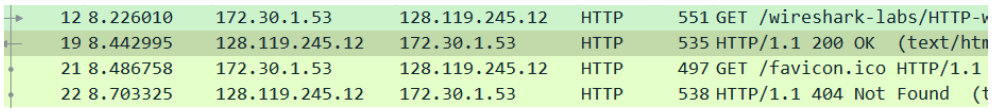
12. 1번의 GET request가 있었다. Bill 또는 Right에 대한 정보가 GET Message에 포함된 패킷의 번호는 12번이다.

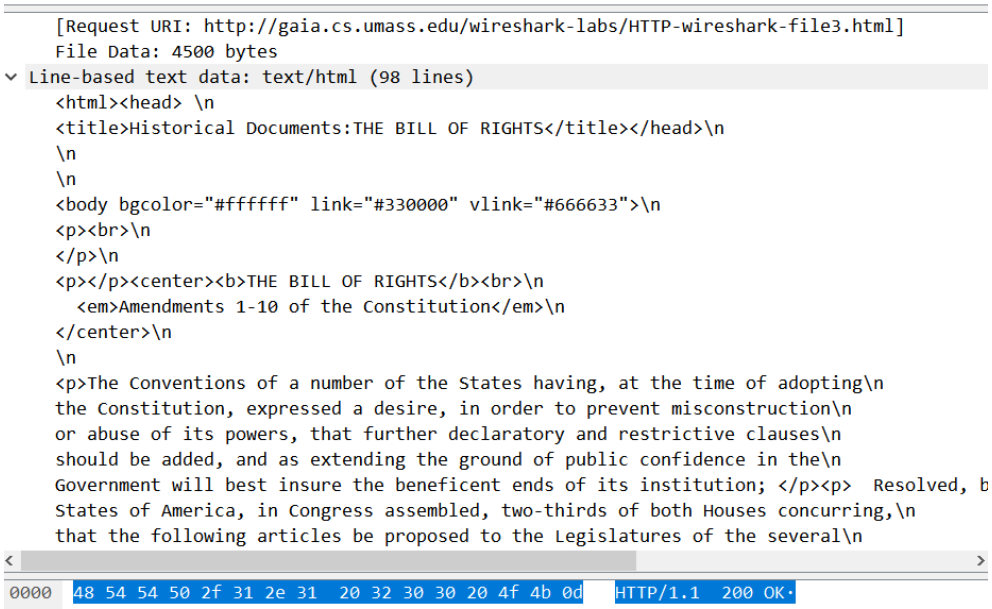


No.	Time	Source	Destination	Protocol	Length	Info
12	8.226010	172.30.1.53	128.119.245.12	HTTP	551	GET /wireshark-labs/HTTP-w…
19	8.442995	128.119.245.12	172.30.1.53	HTTP	535	HTTP/1.1 200 OK (text/htm…
21	8.486758	172.30.1.53	128.119.245.12	HTTP	497	GET /favicon.ico HTTP/1.1
22	8.703325	128.119.245.12	172.30.1.53	HTTP	538	HTTP/1.1 404 Not Found (t…

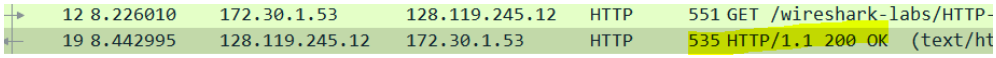
13. 제공된 request에 대한 response 정보를 갖고 있으며 HTTP status 200으로 작

업을 종료한 19번 패킷이다.

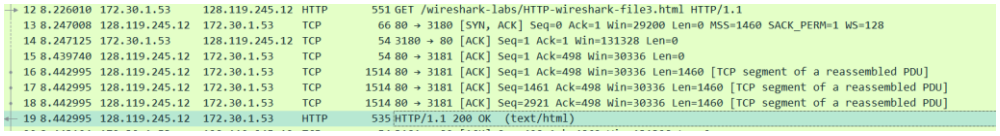




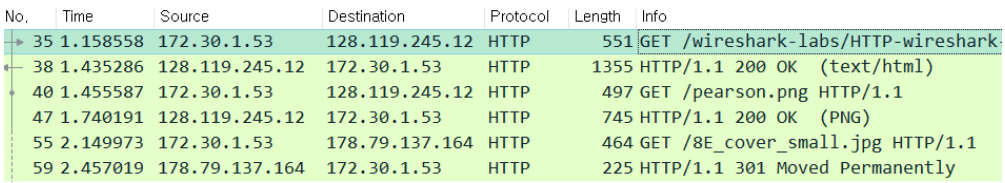
14. HTTP status code : 200, Response Phrase : OK



15. TCP에서 한 개의 패킷이 가질 수 있는 최대 바이트 MTU는 1500이다. 따라서 4500바이트를 보내기 위해서는 총 3개의 TCP segment가 필요하다. 패킷 번호는 16 ,17, 18



16. 본문 요청, 본문에 포함된 첫번째 이미지, 두번째 이미지 총 3개의 GET Request를 보냈다.



첫 번째 GET Request는 다음과 같고

[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html]

두 번째 GET Request

[Full request URI: http://gaia.cs.umass.edu/pearson.png]

세 번째 GET Request

[Full request URI: http://kurose.cslash.net/8E_cover_small.jpg]

17. 두개의 Image는 순차적으로 다운로드 되었다. 첫 번째 사진에서 볼 수 있듯 두 다운로드 모두 TCP connection을 사용하고, 두 번째 사진의 시간을 보면 첫번째 Image의 HTTP요청에 대한 응답을 수신한 후 두번째 Image의 HTTP요청을 보내 그 응답을 수신한다.

첫번째 문제에서 HTTP1.1을 사용함을 알 수 있었다. HTTP1.1은 Persistent HTTP로 파일을 다운로드할 때 TCP connection을 하나만 열어 해당 connection으로만 다운로드 한다.

```
HTTP      497 GET /pearson.png HTTP/1.1
DNS        77 Standard query 0xa6d2 A kurose.cslash.net
DNS        77 Standard query 0xa6d2 A kurose.cslash.net
DNS        73 Standard query 0x8869 A buymeacoff.ee
DNS       426 Standard query response 0x8869 A buymeacoff.ee A 172.67.134.37 A 104.21.6.12 NS ruth.ns
TCP      1514 80 → 3461 [ACK] Seq=1302 Ack=941 Win=31360 Len=1460 [TCP segment of a reassembled PDU]
TCP      1514 80 → 3461 [ACK] Seq=2762 Ack=941 Win=31360 Len=1460 [TCP segment of a reassembled PDU]
HTTP      745 HTTP/1.1 200 OK  (PNG)
TCP        54 3461 → 80 [ACK] Seq=941 Ack=4913 Win=131328 Len=0
DNS       427 Standard query response 0xa6d2 A kurose.cslash.net CNAME cslash.net A 178.79.137.164 NS
TCP        66 3464 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
DNS       427 Standard query response 0xa6d2 A kurose.cslash.net CNAME cslash.net A 178.79.137.164 NS
TCP        66 3465 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
TCP        66 80 → 3464 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
TCP        54 3464 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
HTTP      464 GET /8E_cover_small.jpg HTTP/1.1
TCP        66 80 → 3465 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
TCP        54 3465 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
TCP        54 80 → 3464 [ACK] Seq=1 Ack=411 Win=64128 Len=0
HTTP      225 HTTP/1.1 301 Moved Permanently
```

```
 40 1.455587    172.30.1.53     128.119.245.12  HTTP    497 GET /pearson.png HTTP/1.1
 47 1.740191    128.119.245.12  172.30.1.53     HTTP    745 HTTP/1.1 200 OK  (PNG)
 55 2.149973    172.30.1.53     178.79.137.164  HTTP    464 GET /8E_cover_small.jpg HTTP/1.1
 59 2.457019    178.79.137.164  172.30.1.53     HTTP    225 HTTP/1.1 301 Moved Permanently
```

18. 401 Unauthorized. 해당 리소스에 credential이 없기 때문에 GET Reqeust가 적용되지 않았기 때문이다. 인증이 가능하다는 의미도 내포하고있다.

```
No.  Time      Source          Destination     Protocol  Length  Info
  58 1.787718  172.30.1.16     128.119.245.12  HTTP         566  GET /wireshark-labs/protec
  63 2.006924  128.119.245.12  172.30.1.16     HTTP         771  HTTP/1.1 401 Unauthorized
```

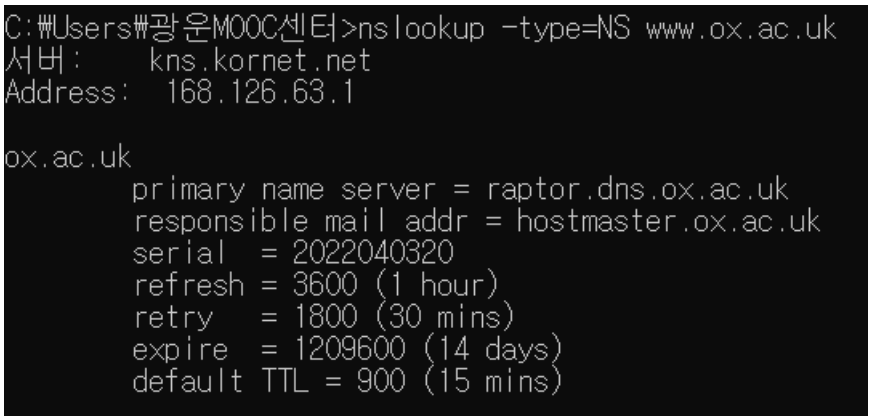
19. Authorization Field가 추가되어 있다.

```
Hypertext Transfer Protocol
  GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1\r\n
  Host: gaia.cs.umass.edu\r\n
  Connection: keep-alive\r\n
  Cache-Control: max-age=0\r\n
  Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n
    Credentials: wireshark-students:network
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like G
```

- **Question #3**

1. nslookup 명령어는 DNS서버에 query를 보내 도메인의 정보를 가져온다. Asia에 있는 Web서버의 IP 주소를 가져오기 위해서 naver의 주소를 넣었다.

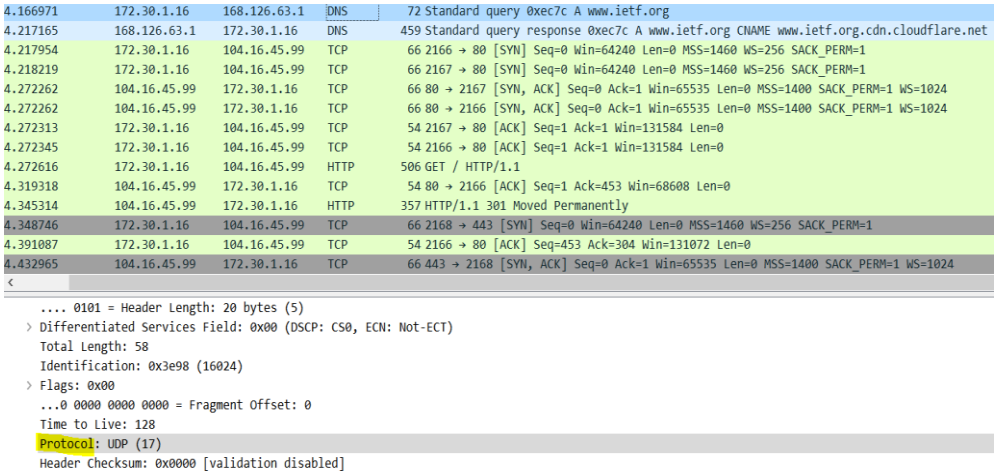
2. 유럽 대학(여기서는 Oxford Univ) 홈페이지에 대한 Authoritative DNS server는 Name Server이므로 Name Server Record가 필요하다. 때문에 -type=NS를 옵션에 추가해주었다.



3. 문제 2에서 얻은 Name Server raptor.dns.ox.ac.uk를 통해 kr.mail.yahoo.com의 ip를 조회했지만 제대로 응답을 받아올 수 없었다.

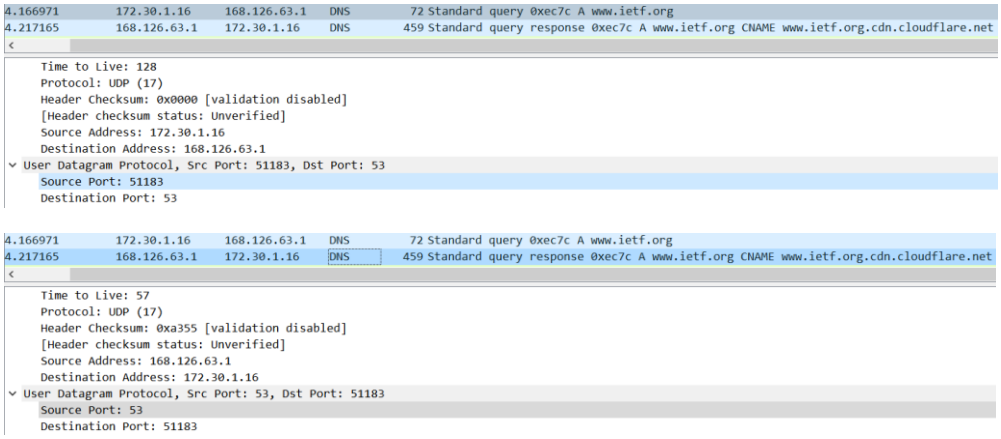
```
C:₩Users₩광운MOOC센터>nslookup kr.mail.yahoo.com raptor.dns.ox.ac.uk
DNS request timed out.
    timeout was 2 seconds.
서버:    UnKnown
Address:  163.1.2.192

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** UnKnown에 대한 요청이 제한 시간을 초과했습니다.
```

4. 해당 사진의 최상단을 확인하면 DNS query와 response message를 볼 수 있다. 또 최하단을 확인하면 UDP Protocol을 사용했음을 확인할 수 있다.

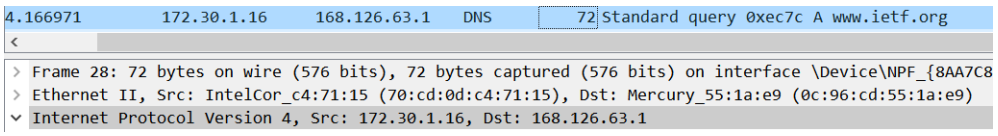


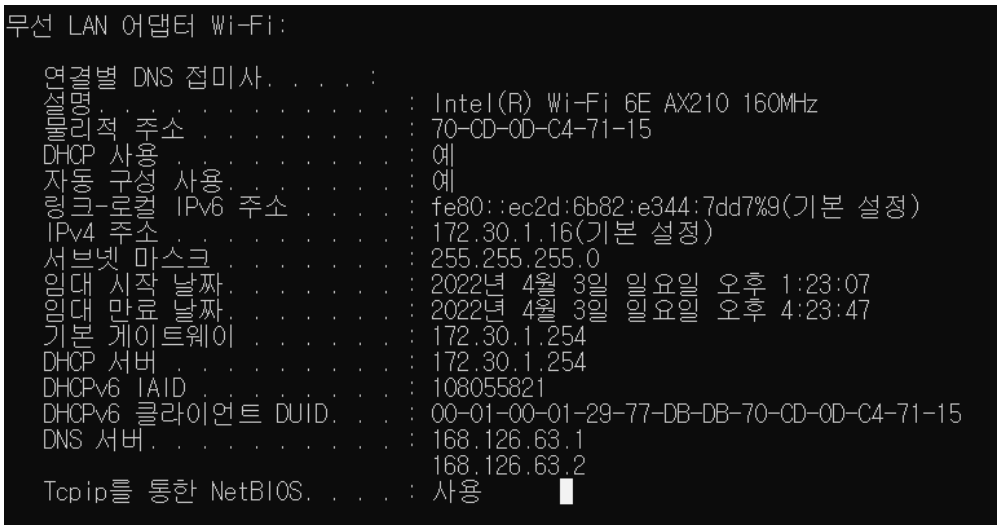
5. 첫번째 사진은 DNS query에 대한 destination port이다. 여기서는 53번 포트가 사용되고 있다.

두번째 사진은 DNS response에 대한 source port이다. source port, 즉 51183 port에서 destination port 53으로 DNS request를 보냈음으로 response는 그 반대로 53번 port가 source port가 된다.



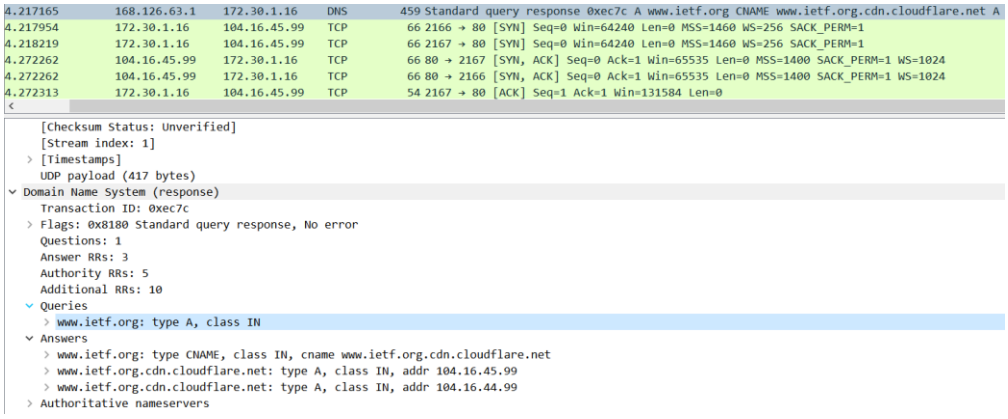
6. 첫번째 사진을 보면 DNS query message가 Dst: 168.126.63.1로 보내지는 것을 볼 수 있다. 두번째 사진은 cmd창에 ipconfig /all 명령어를 입력한 후 현재 사용하고 있는 무선 LAN 어댑터 Wi-Fi 부분이다. DNS서버가 168.126.61.1 그리고 168.126.63.2로 표시되어 있는데 DNS query message가 보내지는 IP address와 동일하다. DNS query message가 내 컴퓨터의 DNS를 사용했음을 알 수 있다.





7. DNS query의 Type 은 A이다. DNS query message에는 answers가 포함되어 있지 않다. DNS query message의 Field는 Header와 query가 전부인 반면 DNS response message의 Field는 Header와 Question, Answer, Authority, Additional 등의 정보가 포함되어 있다.

8. DNS response message에는 3개의 answer가 포함되어 있다.



첫 번째 answer는 Name, Type, Class, Time to live, Data length, CNAME을 가지고 있고

두 번째 answer는 Name, Type, Class, Time to live, Data length, Address를 가지고 있고

세 번째 answer는 두번째 answer와 동일하게 Name, Type, Class, Time to live, Data length, Address를 가지고 있다.

```
▾ Queries
  › www.ietf.org: type A, class IN
▾ Answers
  ▾ www.ietf.org: type CNAME, class IN, cname www.ietf.org.cdn.cloudflare.net
      Name: www.ietf.org
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
      Time to live: 73 (1 minute, 13 seconds)
      Data length: 33
      CNAME: www.ietf.org.cdn.cloudflare.net
  ▾ www.ietf.org.cdn.cloudflare.net: type A, class IN, addr 104.16.45.99
      Name: www.ietf.org.cdn.cloudflare.net
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 113 (1 minute, 53 seconds)
      Data length: 4
      Address: 104.16.45.99
  ▾ www.ietf.org.cdn.cloudflare.net: type A, class IN, addr 104.16.44.99
      Name: www.ietf.org.cdn.cloudflare.net
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 113 (1 minute, 53 seconds)
      Data length: 4
      Address: 104.16.44.99
```

9. TCP 통신은 3 way handshake라는 방식으로 연결된다. 3번의 요청과 응답이 교차한 후에 Client와 Server가 연결된다.

첫째로 Client에서 Server로 SYN 데이터를 보낸다. 이때 SYN데이터 안에는 연결 요청을 위한 정보들이 들어있다.

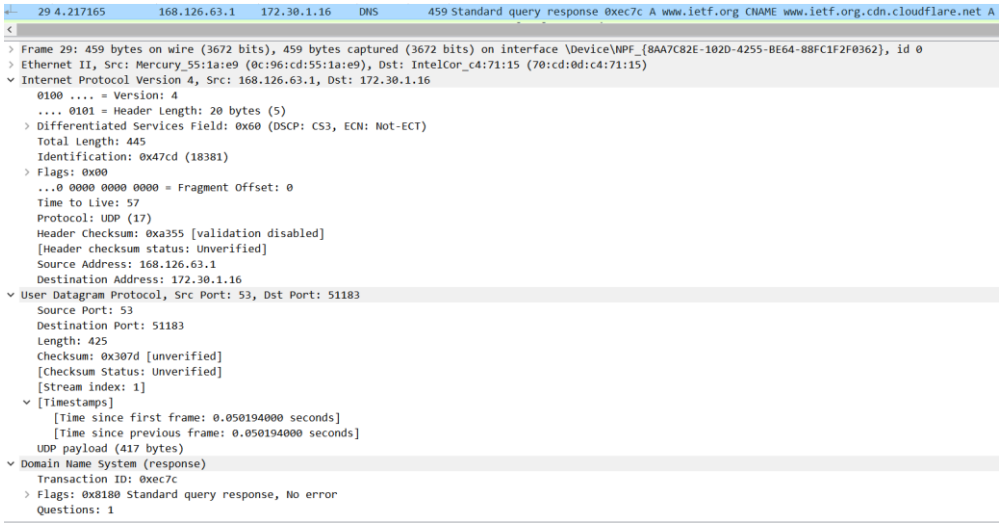
둘째로 Server의 포트가 SYN데이터를 받고, 그 요청을 수신했다는 대답(ACK)과 Client가 port를 열게끔 SYN 데이터를 함께 보낸다.

셋째로 Client가 Server로부터 ACK, SYN 데이터를 받고 Server에 수신 확인이라는 ACK를 전송한다. 다음의 사진은 두번째까지의 과정을 나타낸 것이다.

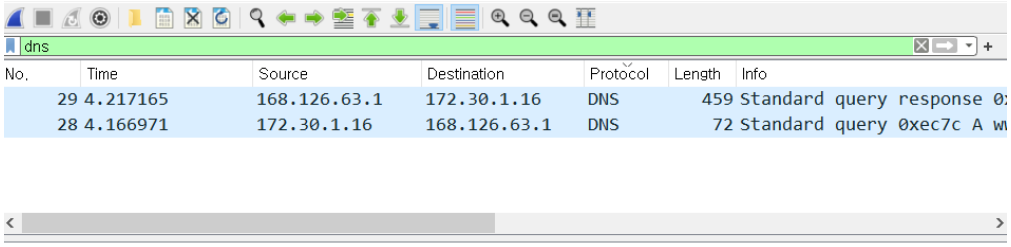
29	4.217165	168.126.63.1	172.30.1.16	DNS	459 Standard query response 0xec7c A www.ietf.org CNAME www.ietf.org.cdn.cloudflare.net
30	4.217954	172.30.1.16	104.16.45.99	TCP	66 2166 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
31	4.218219	172.30.1.16	104.16.45.99	TCP	66 2167 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
32	4.272262	104.16.45.99	172.30.1.16	TCP	66 80 → 2167 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM=1 WS=1024

TCP SYN 첫번째 패킷의 Destination IP는 104.16.45.99이다.

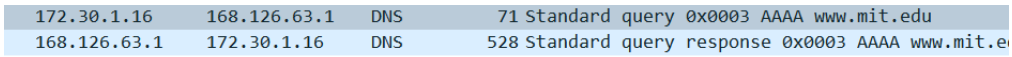
DNS response message에 해당하는 패킷 그 어디에도 104.16.45.99에 상응하는 IP는 존재하지 않는다.

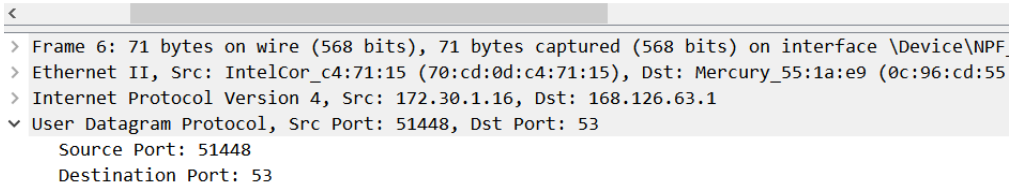


10. 추가적인 DNS query는 존재하지 않는다.

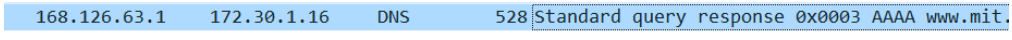


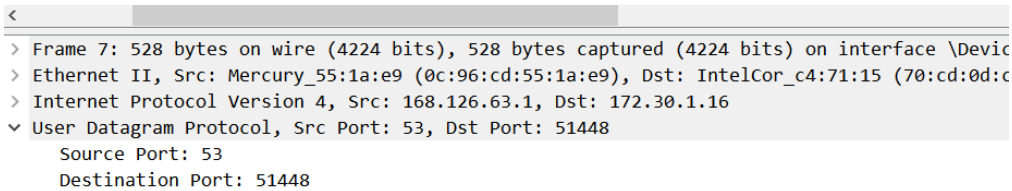
11. 사진에서 볼 수 있듯, DNS query message에 대한 destination port는 53이다.





두번째 사진에서 볼 수 있듯, DNS response message에 대한 source port는 53이다. 내 컴퓨터의 송신포트 51448을 사용해 수신포트 53으로 데이터를 보냈기 때문에 DNS response는 그 반대로 송신포트(source port) 53을 사용한다.





12. 첫번째 사진은 DNS query message가 172.30.1.16에서 168.126.31.1로 보내지는

DNS통신을 보여주고 있다. DNS query message는 168.126.31.1로 보내지는데 이는 ipconfig /all 명령어로 확인한 내 컴퓨터의 default local DNS server와 같다.

```
172.30.1.16     168.126.63.1    DNS        71 Standard query 0x0003 AAAA www.mit.edu
168.126.63.1    172.30.1.16     DNS       528 Standard query response 0x0003 AAAA www.
```

```
> Frame 6: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Devic
> Ethernet II, Src: IntelCor_c4:71:15 (70:cd:0d:c4:71:15), Dst: Mercury_55:1a:e9 (0c:96:
> Internet Protocol Version 4, Src: 172.30.1.16, Dst: 168.126.63.1
```

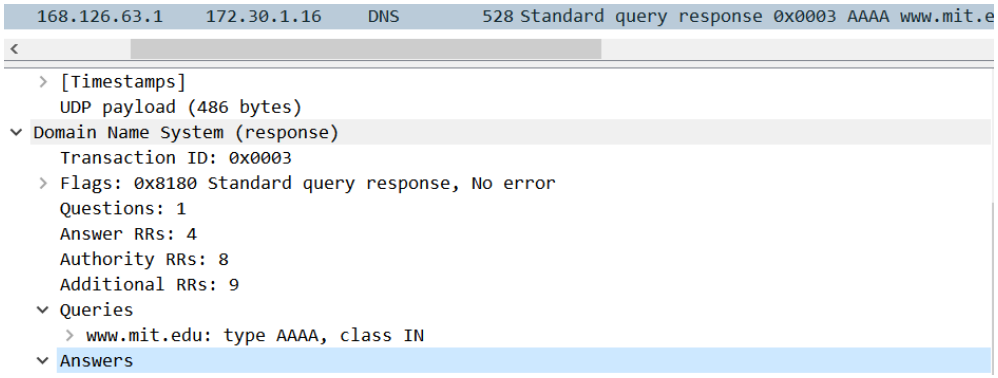
```
무선 LAN 어댑터 Wi-Fi:

   연결별 DNS 접미사. . . . :
   설명. . . . . . . . . . . : Intel(R) Wi-Fi 6E AX210 160MHz
   물리적 주소 . . . . . . . : 70-CD-0D-C4-71-15
   DHCP 사용 . . . . . . . . : 예
   자동 구성 사용. . . . . . : 예
   링크-로컬 IPv6 주소 . . . . : fe80::ec2d:6b82:e344:7dd7%9(기본 설정)
   IPv4 주소 . . . . . . . . : 172.30.1.16(기본 설정)
   서브넷 마스크 . . . . . . : 255.255.255.0
   임대 시작 날짜. . . . . . : 2022년 4월 3일 일요일 오후 1:23:07
   임대 만료 날짜. . . . . . : 2022년 4월 3일 일요일 오후 5:54:18
   기본 게이트웨이 . . . . . : 172.30.1.254
   DHCP 서버 . . . . . . . . : 172.30.1.254
   DHCPv6 IAID . . . . . . . : 108055821
   DHCPv6 클라이언트 DUID. . . : 00-01-00-01-29-77-DB-DB-70-CD-0D-C4-71-15
   DNS 서버. . . . . . . . . : 168.126.63.1
                               168.126.63.2
   Tcpip를 통한 NetBIOS. . . . : 사용
```

13. DNS query message의 Type은 AAAA이다. answer는 포함하고 있지 않다. 이는 위에서 설명한 적 있듯, DNS query message는 Answer영역이 포함되지 않기 때문이다.

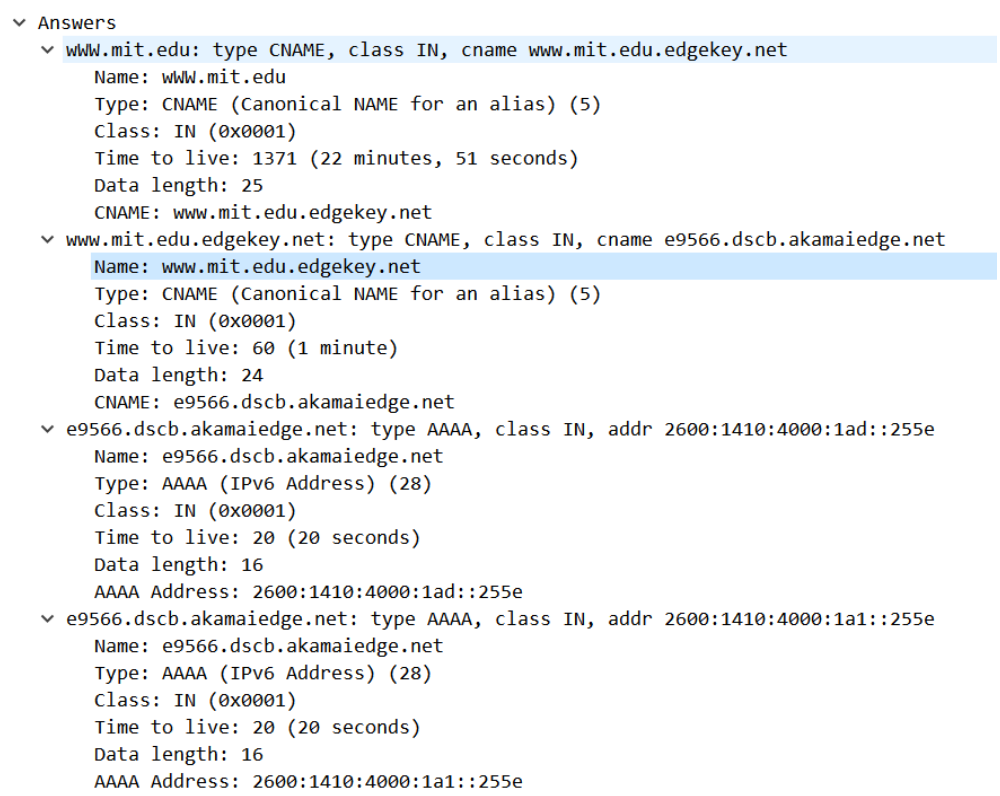
```
172.30.1.16     168.126.63.1    DNS        71 Standard query 0x0003 AAAA www.mit.edu
168.126.63.1    172.30.1.16     DNS       528 Standard query response 0x0003 AAAA www.mit
```

```
> Frame 6: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NP
> Ethernet II, Src: IntelCor_c4:71:15 (70:cd:0d:c4:71:15), Dst: Mercury_55:1a:e9 (0c:96:cd:
> Internet Protocol Version 4, Src: 172.30.1.16, Dst: 168.126.63.1
v User Datagram Protocol, Src Port: 51448, Dst Port: 53
    Source Port: 51448
    Destination Port: 53
    Length: 37
    Checksum: 0x94e4 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 2]
  > [Timestamps]
    UDP payload (29 bytes)
v Domain Name System (query)
    Transaction ID: 0x0003
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    > www.mit.edu: type AAAA, class IN
    [Response In: 7]
```

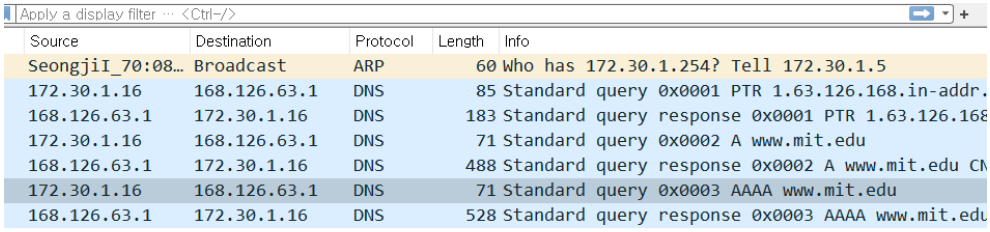
14. 첫번째 사진에서 확인할 수 있듯이 DNS response message는 4개의 answer를 포함하고 있다.

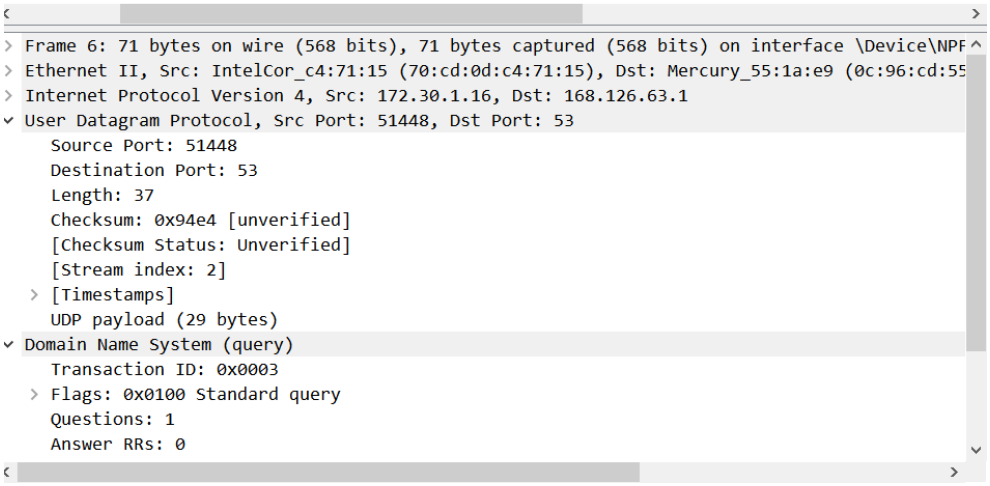


네 가지 Answer모두 Name, Type, Class, Time to live를 포함하고 있다. 그리고 첫번째와 두번째 Answer는 Data length, CNAME이 추가로 포함되어 있고, 세번째와 네번째 Answer는 Data length와 AAAA Address가 추가로 포함되어 있다.

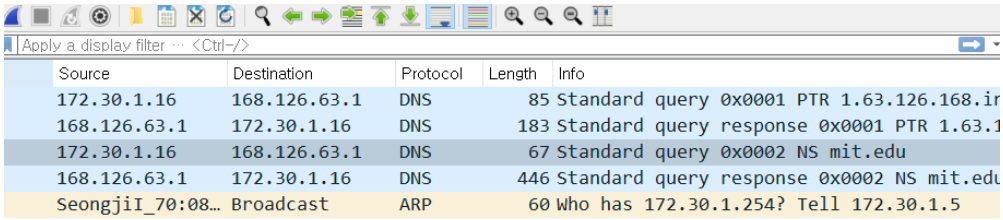


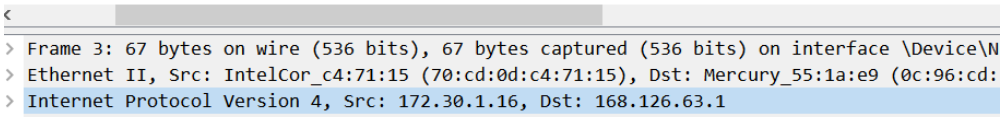
15.





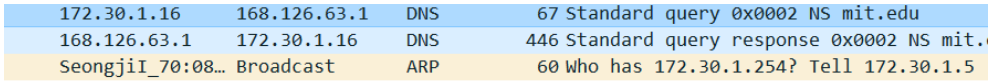
16. 168.126.63.1 로 DNS query message가 보내지는데 이는 내 컴퓨터의 local DNS server와 동일하다

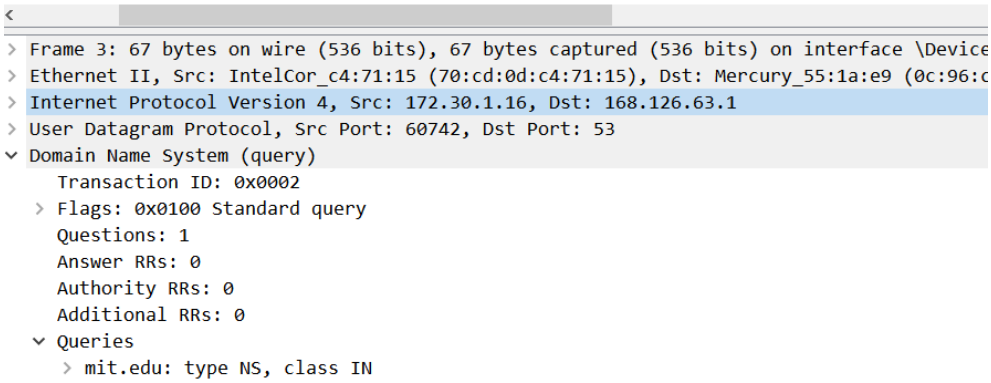




```
무선 LAN 어댑터 Wi-Fi:

   연결별 DNS 접미사. . . . :
   설명. . . . . . . . . . . : Intel(R) Wi-Fi 6E AX210 160MHz
   물리적 주소 . . . . . . . : 70-CD-0D-C4-71-15
   DHCP 사용 . . . . . . . . : 예
   자동 구성 사용. . . . . . : 예
   링크-로컬 IPv6 주소 . . . . : fe80::ec2d:6b82:e344:7dd7%9(기본 설정)
   IPv4 주소 . . . . . . . . : 172.30.1.16(기본 설정)
   서브넷 마스크 . . . . . . : 255.255.255.0
   임대 시작 날짜. . . . . . : 2022년 4월 3일 일요일 오후 1:23:07
   임대 만료 날짜. . . . . . : 2022년 4월 3일 일요일 오후 5:54:18
   기본 게이트웨이 . . . . . : 172.30.1.254
   DHCP 서버 . . . . . . . . : 172.30.1.254
   DHCPv6 IAID . . . . . . . : 108055821
   DHCPv6 클라이언트 DUID. . . : 00-01-00-01-29-77-DB-DB-70-CD-0D-C4-71-15
   DNS 서버. . . . . . . . . : 168.126.63.1
                               168.126.63.2
   Tcpip를 통한 NetBIOS. . . . : 사용
```

17. DNS query message의 Type 은 NS이다. nslookup 명령어 사용시 -type=NS를 사용해 도메인의 네임서버를 확인하였으므로 Type이 NS로 바뀌었다. 그리고 위에서 설명했듯이 DNS query message에는 answer가 존재하지 않는다.





18. 첫번째 사진의 DNS response message를 보면 MIT nameserver의 Answer영역에 8개의 answer가포함되어 있고,

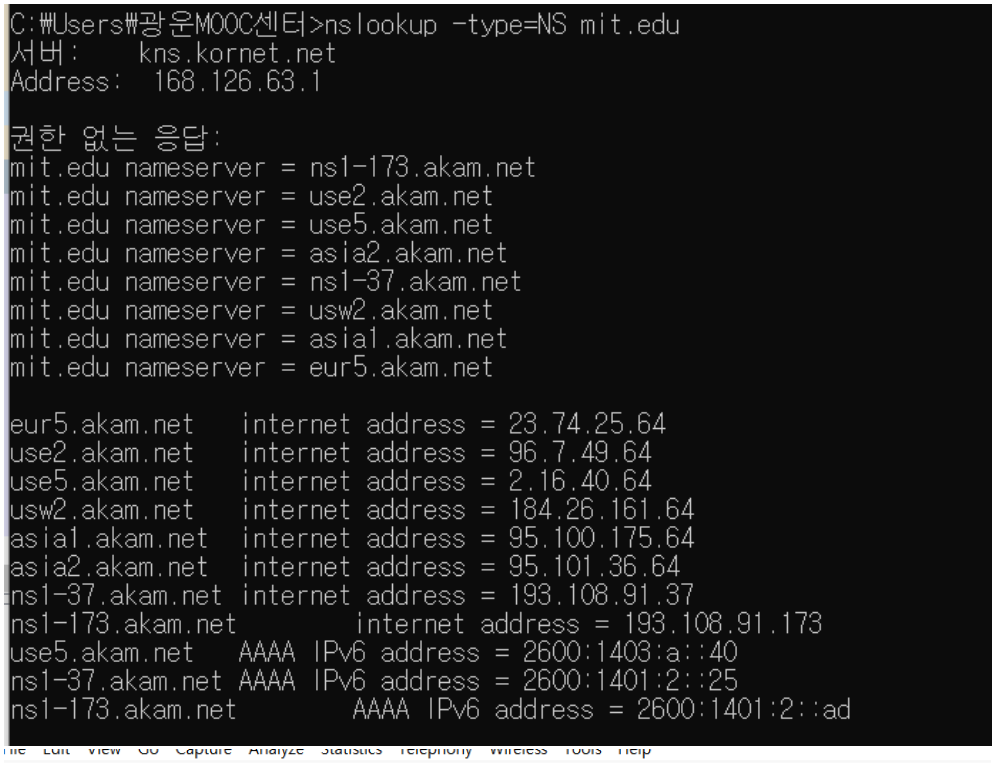
```
    168.126.63.1    172.30.1.16     DNS         446 Standard query response 0x0002 NS mit.edu
    SeongjiI_70:08… Broadcast       ARP          60 Who has 172.30.1.254? Tell 172.30.1.5
```

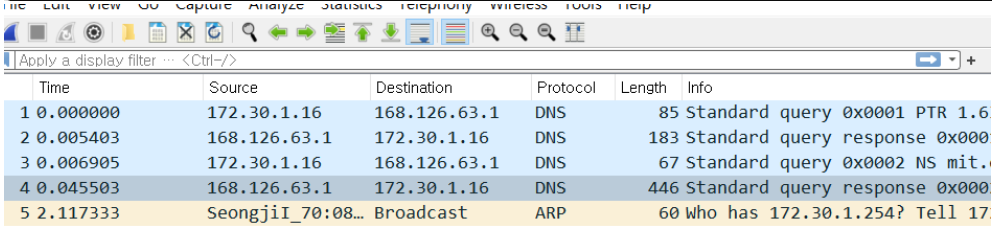
```
> Internet Protocol Version 4, Src: 168.126.63.1, Dst: 172.30.1.16
> User Datagram Protocol, Src Port: 53, Dst Port: 60742
v Domain Name System (response)
    Transaction ID: 0x0002
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 8
    Authority RRs: 0
    Additional RRs: 11
  v Queries
    > mit.edu: type NS, class IN
  v Answers
    > mit.edu: type NS, class IN, ns ns1-173.akam.net
    > mit.edu: type NS, class IN, ns use2.akam.net
    > mit.edu: type NS, class IN, ns use5.akam.net
    > mit.edu: type NS, class IN, ns asia2.akam.net
    > mit.edu: type NS, class IN, ns ns1-37.akam.net
    > mit.edu: type NS, class IN, ns usw2.akam.net
    > mit.edu: type NS, class IN, ns asia1.akam.net
    > mit.edu: type NS, class IN, ns eur5.akam.net
```

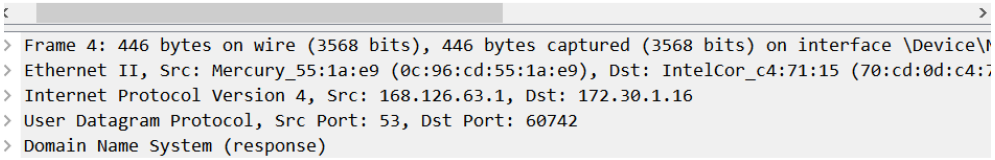
두번째 사진에는 MIT nameserver들의 IP Address가 Additional records영역에 포함되어 수신된 것을 알 수 있다. 이는 NS record를 사용해 DNS 목록을 확인했기 때문이다.

```
v Answers
  > mit.edu: type NS, class IN, ns ns1-173.akam.net
  > mit.edu: type NS, class IN, ns use2.akam.net
  > mit.edu: type NS, class IN, ns use5.akam.net
  > mit.edu: type NS, class IN, ns asia2.akam.net
  > mit.edu: type NS, class IN, ns ns1-37.akam.net
  > mit.edu: type NS, class IN, ns usw2.akam.net
  > mit.edu: type NS, class IN, ns asia1.akam.net
  > mit.edu: type NS, class IN, ns eur5.akam.net
v Additional records
  > eur5.akam.net: type A, class IN, addr 23.74.25.64
  > use2.akam.net: type A, class IN, addr 96.7.49.64
  > use5.akam.net: type A, class IN, addr 2.16.40.64
  > usw2.akam.net: type A, class IN, addr 184.26.161.64
  > asia1.akam.net: type A, class IN, addr 95.100.175.64
  > asia2.akam.net: type A, class IN, addr 95.101.36.64
  > ns1-37.akam.net: type A, class IN, addr 193.108.91.37
  > ns1-173.akam.net: type A, class IN, addr 193.108.91.173
  > use5.akam.net: type AAAA, class IN, addr 2600:1403:a::40
  > ns1-37.akam.net: type AAAA, class IN, addr 2600:1401:2::25
  > ns1-173.akam.net: type AAAA, class IN, addr 2600:1401:2::ad
```

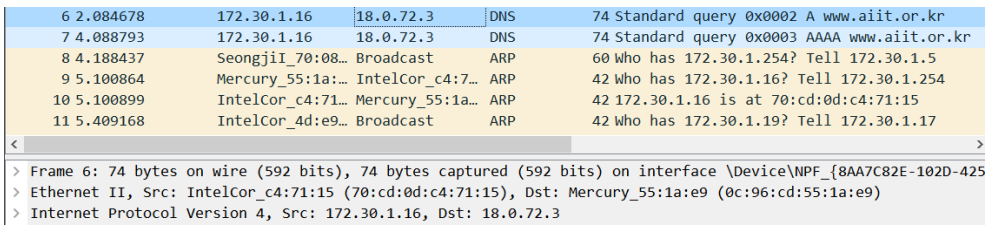
19.





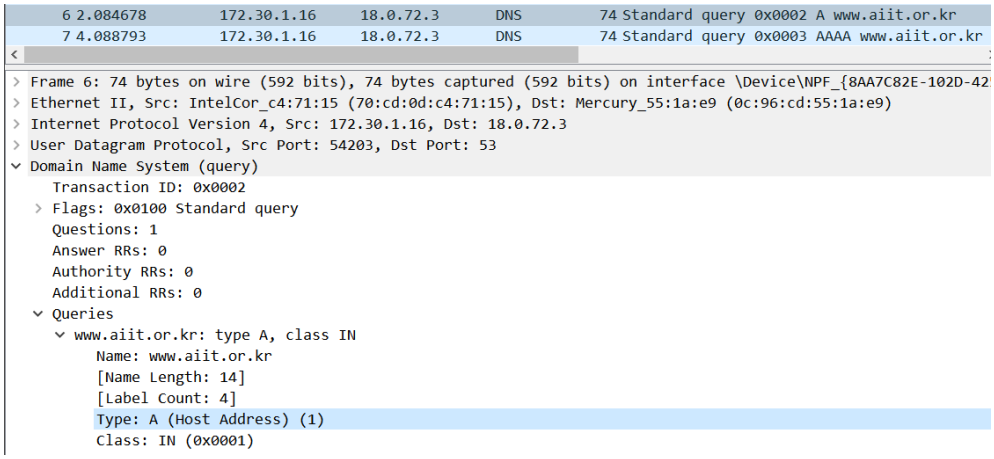


20. DNS query message가 18.0.72.3으로 보내진다. 이는 내 컴퓨터의 default local DNS server인 168.126.63.2와 다른데 이는 문제에서 제시된 명령어인 nslookup을 실행할 때 입력한 네임서버(bitsy.mit.edu)로 도메인(www.aiit.or.kr)의 ip을 조회하기 때문이다. 그렇기 때문에 DNS query message는 Name server의 IP로 전송되고 두번째 사진에서 Name server의 IP와 DNS query message가 전송된 IP가 동일 한 것을 알 수 있다.



```
C:₩Users₩광운MOOC센터>nslookup bitsy.mit.edu
서버:    kns.kornet.net
Address:  168.126.63.1

권한 없는 응답:
이름:    bitsy.mit.edu
Address:  18.0.72.3
```

21. DNS query message의 Type은 A이다. nslookup 명령어를 실행할 때 따로 타입을 지정해주지 않았다. 그리고 DNS query message에는 answer영역이 포함되지 않는다.

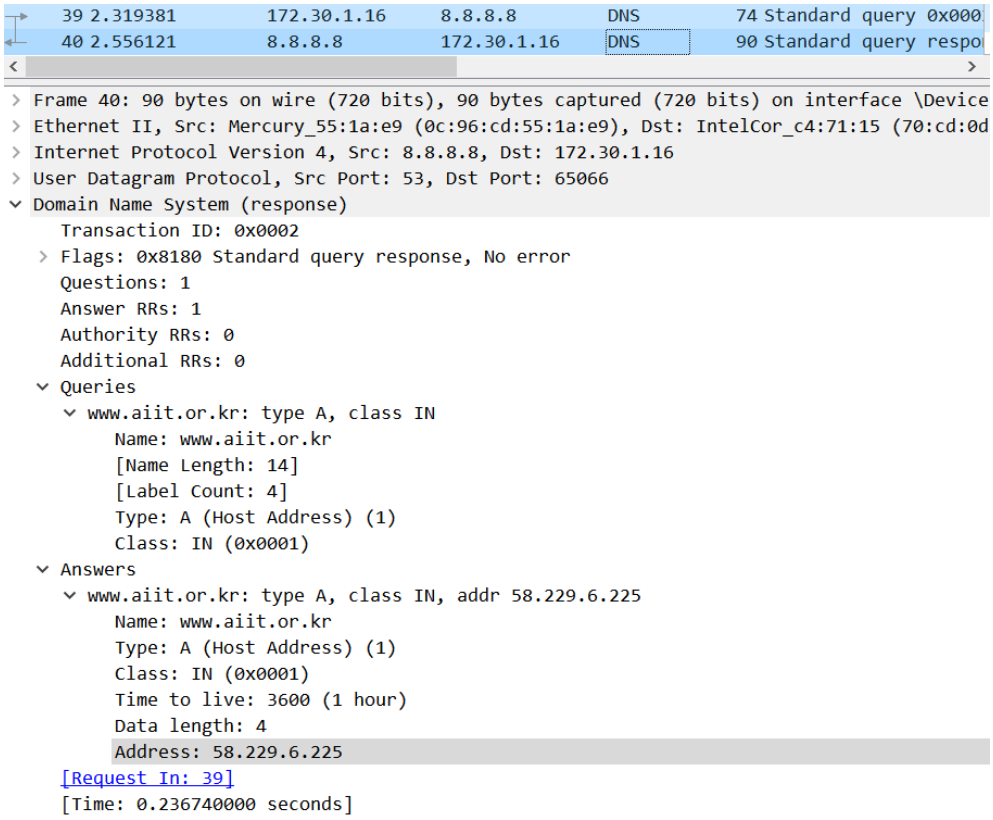


22. 지정한 Name server(bitsy.mit.edu)를 사용해 지정한 Domain(www.aiit.or.kr)에 DNS qeury를 보내고 그에 대한 응답을 받아야 하는데 Name server오류로 응답이 오지 않는다. local default DNS server, dns.google, kns.kornet.net 등 다른 Name server를 사용해 동작을 확인해 보면 DNS response를 잘 받아오지만 bitsy.mit.edu Name server를 사용하면 response를 받아오지 않아 answer의 개수, 각각의 answer들이 포함하고 있는 것들을 확인할 수 없다. 이는 첫번째 사진 CMD 캡처와 두번째 사진(DNS query는 있지만 DNS response가 오지 않은 캡처)를 통해 확인할 수 있다.

dns.google를 사용해 www.aiit.or.kr에 DNS query를 보낸 뒤 DNS response를 확인하면 Answer 1개를 가지고 있고, Name, Type, Class, Time to live, Data length, Address 필드를 가지고 있다. 이는 세번째 사진으로 확인할 수 있다.

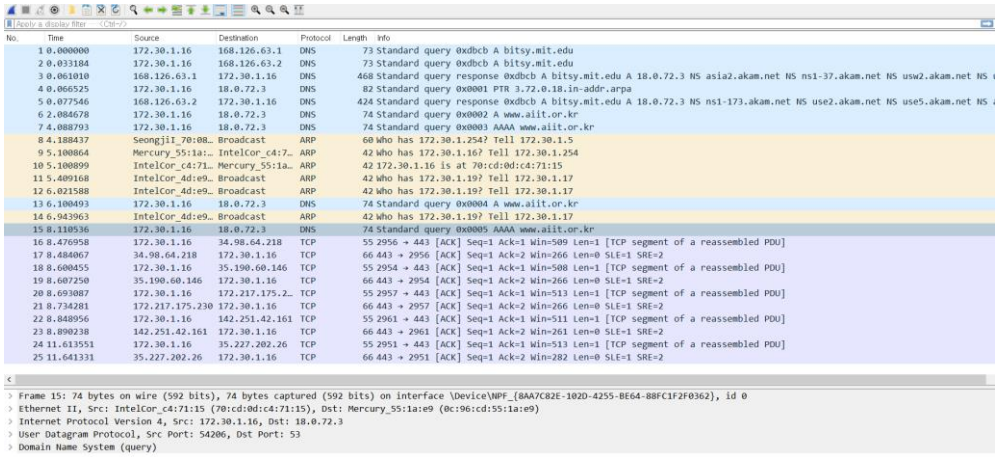
```
C:₩Users₩광운MOOC센터>nslookup www.aiit.or.kr
서버:    dns.google
Address:  8.8.8.8

권한 없는 응답:
이름:    www.aiit.or.kr
Address:  58.229.6.225


C:₩Users₩광운MOOC센터>nslookup www.aiit.or.kr dns.google
서버:    dns.google
Address:  8.8.8.8

권한 없는 응답:
이름:    www.aiit.or.kr
Address:  58.229.6.225


C:₩Users₩광운MOOC센터>nslookup www.aiit.or.kr kns.kornet.net
서버:    kns.kornet.net
Address:  168.126.63.1

권한 없는 응답:
이름:    www.aiit.or.kr
Address:  58.229.6.225


C:₩Users₩광운MOOC센터>nslookup www.aiit.or.kr bitsy.mit.edu
DNS request timed out.
    timeout was 2 seconds.
서버:    UnKnown
Address:  18.0.72.3

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** UnKnown에 대한 요청이 제한 시간을 초과했습니다.

C:₩Users₩광운MOOC센터>
```

6	2.084678	172.30.1.16	18.0.72.3	DNS	74	Standard query 0x0002 A www.aiit.or.kr
7	4.088793	172.30.1.16	18.0.72.3	DNS	74	Standard query 0x0003 AAAA www.aiit.or.kr
8	4.188437	SeongjiI_70:08…	Broadcast	ARP	60	Who has 172.30.1.254? Tell 172.30.1.5
9	5.100864	Mercury_55:1a:…	IntelCor_c4:7…	ARP	42	Who has 172.30.1.16? Tell 172.30.1.254
10	5.100899	IntelCor_c4:71…	Mercury_55:1a…	ARP	42	172.30.1.16 is at 70:cd:0d:c4:71:15
11	5.409168	IntelCor_4d:e9…	Broadcast	ARP	42	Who has 172.30.1.19? Tell 172.30.1.17
12	6.021588	IntelCor_4d:e9…	Broadcast	ARP	42	Who has 172.30.1.19? Tell 172.30.1.17
13	6.100493	172.30.1.16	18.0.72.3	DNS	74	Standard query 0x0004 A www.aiit.or.kr
14	6.943963	IntelCor_4d:e9…	Broadcast	ARP	42	Who has 172.30.1.19? Tell 172.30.1.17
15	8.110536	172.30.1.16	18.0.72.3	DNS	74	Standard query 0x0005 AAAA www.aiit.or.kr
16	8.476958	172.30.1.16	34.98.64.218	TCP	55	2956 → 443 [ACK] Seq=1 Ack=1 Win=509 Len=1 [TCP segment of a reassembled PDU]
17	8.484067	34.98.64.218	172.30.1.16	TCP	66	443 → 2956 [ACK] Seq=1 Ack=2 Win=266 Len=0 SLE=1 SRE=2
18	8.600455	172.30.1.16	35.190.60.146	TCP	55	2954 → 443 [ACK] Seq=1 Ack=1 Win=508 Len=1 [TCP segment of a reassembled PDU]
19	8.607250	35.190.60.146	172.30.1.16	TCP	66	443 → 2954 [ACK] Seq=1 Ack=2 Win=266 Len=0 SLE=1 SRE=2
20	8.693087	172.30.1.16	172.217.175.2…	TCP	55	2957 → 443 [ACK] Seq=1 Ack=1 Win=513 Len=1 [TCP segment of a reassembled PDU]
21	8.734281	172.217.175.230	172.30.1.16	TCP	66	443 → 2957 [ACK] Seq=1 Ack=2 Win=266 Len=0 SLE=1 SRE=2
22	8.848956	172.30.1.16	142.251.42.161	TCP	55	2961 → 443 [ACK] Seq=1 Ack=1 Win=511 Len=1 [TCP segment of a reassembled PDU]
23	8.890238	142.251.42.161	172.30.1.16	TCP	66	443 → 2961 [ACK] Seq=1 Ack=2 Win=261 Len=0 SLE=1 SRE=2
24	11.613551	172.30.1.16	35.227.202.26	TCP	55	2951 → 443 [ACK] Seq=1 Ack=1 Win=513 Len=1 [TCP segment of a reassembled PDU]
25	11.641331	35.227.202.26	172.30.1.16	TCP	66	443 → 2951 [ACK] Seq=1 Ack=2 Win=282 Len=0 SLE=1 SRE=2


39 2.319381 172.30.1.16 8.8.8.8 DNS 74 Standard query 0x000
40 2.556121 8.8.8.8 172.30.1.16 DNS 90 Standard query respo
Frame 40: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device
Ethernet II, Src: Mercury_55:1a:e9 (0c:96:cd:55:1a:e9), Dst: IntelCor_c4:71:15 (70:cd:0d
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 172.30.1.16
User Datagram Protocol, Src Port: 53, Dst Port: 65066
Domain Name System (response)
Transaction ID: 0x0002
Flags: 0x8180 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
Queries
www.aiit.or.kr: type A, class IN
Name: www.aiit.or.kr
[Name Length: 14]
[Label Count: 4]
Type: A (Host Address) (1)
Class: IN (0x0001)
Answers
www.aiit.or.kr: type A, class IN, addr 58.229.6.225
Name: www.aiit.or.kr
Type: A (Host Address) (1)
Class: IN (0x0001)
Time to live: 3600 (1 hour)
Data length: 4
Address: 58.229.6.225
[Request In: 39]
[Time: 0.236740000 seconds]

23.

```
C:₩Users₩광운MOOC센터>nslookup www.aiit.or.kr
서버:    dns.google
Address:  8.8.8.8

권한 없는 응답:
이름:    www.aiit.or.kr
Address:  58.229.6.225


C:₩Users₩광운MOOC센터>nslookup www.aiit.or.kr dns.google
서버:    dns.google
Address:  8.8.8.8

권한 없는 응답:
이름:    www.aiit.or.kr
Address:  58.229.6.225


C:₩Users₩광운MOOC센터>nslookup www.aiit.or.kr kns.kornet.net
서버:    kns.kornet.net
Address:  168.126.63.1

권한 없는 응답:
이름:    www.aiit.or.kr
Address:  58.229.6.225


C:₩Users₩광운MOOC센터>nslookup www.aiit.or.kr bitsy.mit.edu
DNS request timed out.
    timeout was 2 seconds.
서버:    UnKnown
Address:  18.0.72.3

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** UnKnown에 대한 요청이 제한 시간을 초과했습니다.

C:₩Users₩광운MOOC센터>
```



결론 및 고찰:

HTTP Header는 Client, Server간 통신에 부가적인 정보를 전송할 수 있게 한다. 과제에서는 Request Header에 credential을 담아 파일의 사용자 인증 정보를 서버에게 알려주었고, Response Header에는 If-Modified-Since, Last-Modified 등으로 파일의 수정 여부, 최근 수정날짜를 담아 사용자에게 알려주었다.

Q2-17은 파일 다운로드 방식이 parallel 인지 serial인지 판단하는 문제였다. HTTP는 serial과 parallel 두가지 방식으로 Object를 다운로드한다. 여기서 새롭게 알게 된 점은 Image파일 자체를 다운로드 하는 것이 아니라 문서 안에 내장된 Object를 다운로드 한다는 것이다. 이때 Object란 reference URL을 말한다. HTTP는 Base HTML-file에 URL형식으로 참조된 이미지, 동영상, 각종 파일들을 TCP connection을 사용해 가져온다. HTTP1.0은 여러 개의 TCP connection을 만들어 parallel 방식으로 다운로드를 진행하고, HTTP1.1은 한 개의 TCP connection을 사용해 순차적으로 파일을 받아온다. 파일의 크기에 따라 HTTP Request Time, HTTP Response Time은 달라질 수 있기 때문에 parallel을 serial로 오해하기 쉽고 그렇기 때문에 단순한 패킷의 시간 비교보단 HTTP의 version 확인이 확실하다.

본 과제를 진행하던 중 Q3-22에서 문제가 있었다. bitsy.mit.edu 네임 서버를 사용해 www.aiit.co.kr 에 DNS query를 보내는 문제였는데, 네임 서버의 IP는 조회 가능했지만 네임서버가 response를 보내주지 않았다. local default DNS server, dns.google 등 다른 네임 서버를 사용해 query를 보낼 때는 정상적으로 response message가 도착한 것으로 볼 때, 문제에서 제공된 네임 서버 자체에 문제가 있던 것으로 보인다.

이 문제를 겪으면서 알 수 있는 다른 하나는 DNS서버가 Domain Name을 IP Address로 변환해 주는 것이 아닌, Domain Name에 매핑된 IP주소를 반환해 주는 것이다. 만약 DNS서버의 기능이 Domain Name을 받아 어떠한 알고리즘을 거쳐 IP 주소로 변환해주는 것이라면 bitsy.mit.edu역시 같은 동작이 예상되지만 DNS query에 대한 요청이 오지 않았다. 그 때문에 네임서버 자체에 문제가 있는 것으로 추론한 것이다.

최근 웹서버를 만들 때, 사용자의 로그인 정보를 DB에 저장하려면 그전에 반드시 암호화가 필요하다는 말을 자주 접하곤 했다. 전송되는 과정에서 해킹공격으로 패킷을 빼앗긴다면 사용자의 정보가 유출되기 때문인데 이번 패킷 분석을 통해 더 명료히 알 수 있었다. Q2-18에서 HTTP Authorization Header에 credential이 담겨 서버로 향하는데 이때 내가 입력한 정보가 Base64를 통해 hash되어 전송된다. 간단한 패킷 분석 툴만으로도 credential을 획득할 수 있다. 바로 이점이 HTTP 요청 전에 유저 정보에 대한 암호화가 필요한 이유다.