



컴퓨터 네트워크

ASS1

수업 명 : 컴퓨터 네트워크
과제 이름 : ASS1
담당 교수님 : 이혁준 교수님
소 속 : 컴퓨터정보공학부
학 번 : 2019202005
이 름 : 남종식
제출 일자 : 2023/04/08
강의 시간 : 화 목 5,6교시

서론

이번 컴퓨터 네트워크 첫번째 과제를 통하여 강의 시간에 배운 패킷에 대해서 wireshark를 이용해 실제로 살펴보고 분석하는 과정을 진행할 것입니다.

강의시간에 컴퓨터 네트워크, 데이터 통신 등에 대해 배우고 있지만 실제로 눈으로 확인하며 배울 수 있는 시간이 없어 아쉬웠지만, wireshark를 통해 TCP/IP 프로토콜을 통하여 주고받는 내용을 실제로 확인하면서 이론적으로만 배웠던 프로토콜을 보다 더 쉽게 이해할 수 있을 것입니다.

HTTP와 DNS의 문제풀이를 통해 wireshark의 이용법을 익히고 네트워크를 분석하는 능력을 기를 것입니다.

본문

Question#1

```
C:\Users\Wjongs>ipconfig

Windows IP 구성

무선 LAN 어댑터 로컬 영역 연결* 1:

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

무선 LAN 어댑터 로컬 영역 연결* 2:

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

이더넷 어댑터 VMware Network Adapter VMnet1:

    연결별 DNS 접미사 . . . . . :
    링크-로컬 IPv6 주소 . . . . . : fe80::a35e:9f85:6afc:4458%5
    IPv4 주소 . . . . . : 192.168.83.1
    서브넷 마스크 . . . . . : 255.255.255.0
    기본 게이트웨이 . . . . . :

이더넷 어댑터 VMware Network Adapter VMnet8:

    연결별 DNS 접미사 . . . . . :
    링크-로컬 IPv6 주소 . . . . . : fe80::5870:f64c:84e9:e636%19
    IPv4 주소 . . . . . : 192.168.65.1
    서브넷 마스크 . . . . . : 255.255.255.0
    기본 게이트웨이 . . . . . :

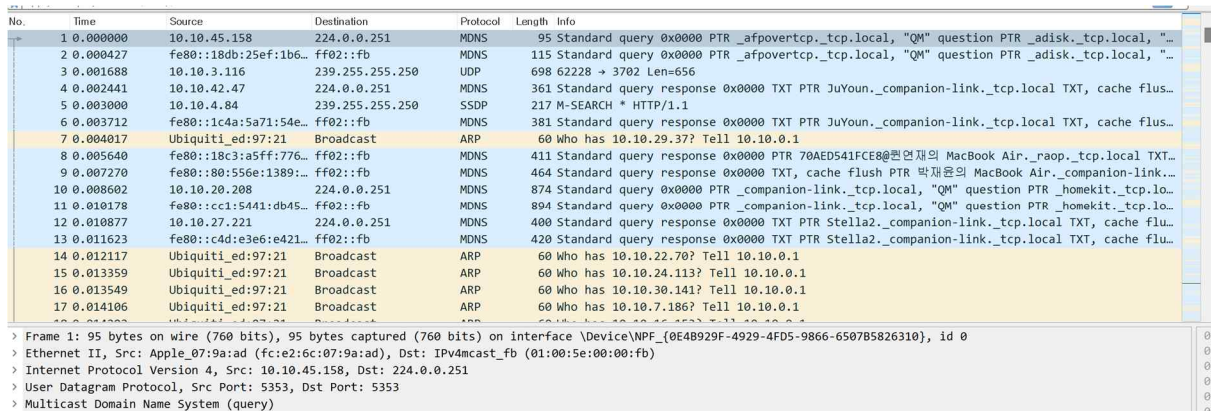
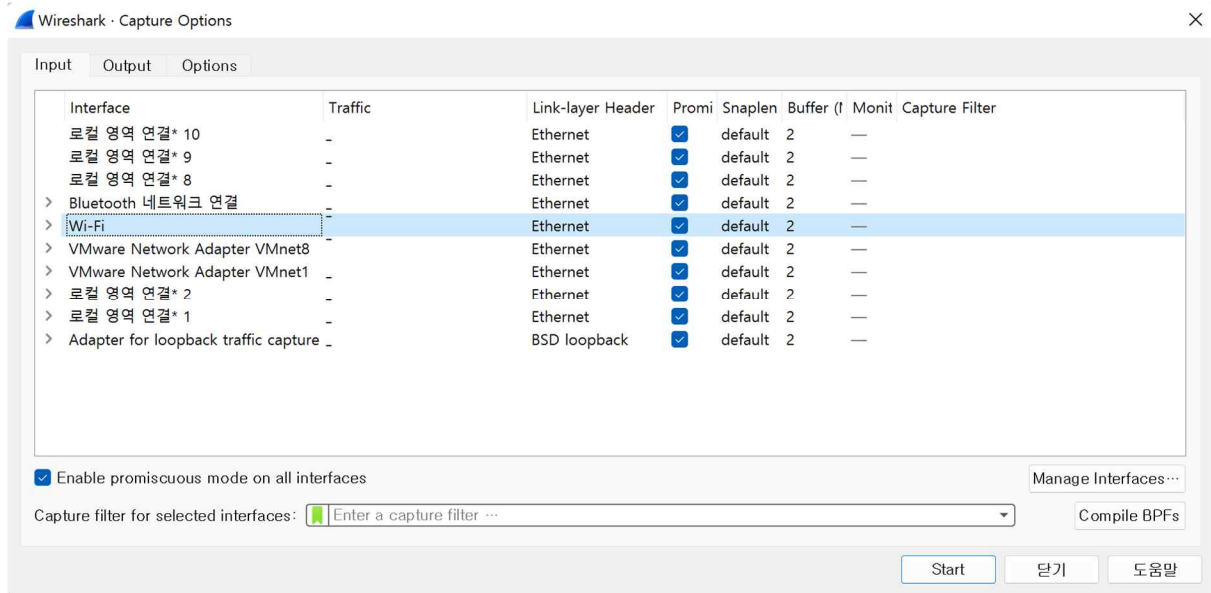
무선 LAN 어댑터 Wi-Fi:

    연결별 DNS 접미사 . . . . . :
    링크-로컬 IPv6 주소 . . . . . : fe80::88bf:3ec9:b7ff:9f7a%4
    IPv4 주소 . . . . . : 10.10.3.220
    서브넷 마스크 . . . . . : 255.255.0.0
    기본 게이트웨이 . . . . . : 10.10.0.1

이더넷 어댑터 Bluetooth 네트워크 연결:

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

Ipconfig는 DNS 서버 주소, 주소, 어댑터 타입을 포함하는 현재 TCP/IP 정보를 보여주기 위해 사용됨

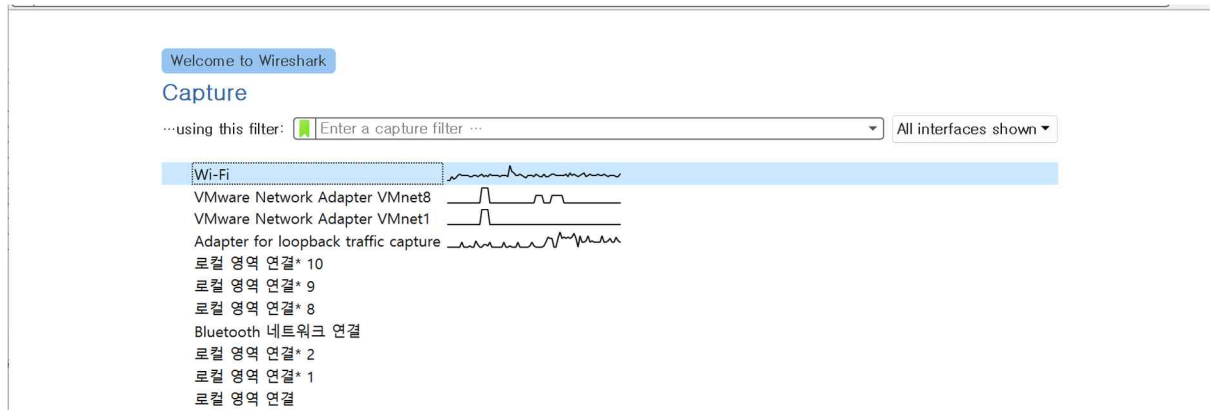


PC에서 동작중인 wifi의 Wireshark 화면 캡처

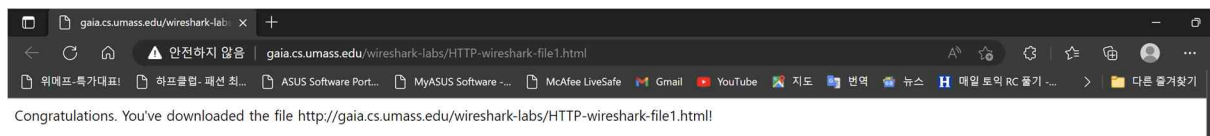
Question#2

Wireshark_HTTP_v7.0.pdf의 19개 문제 풀이

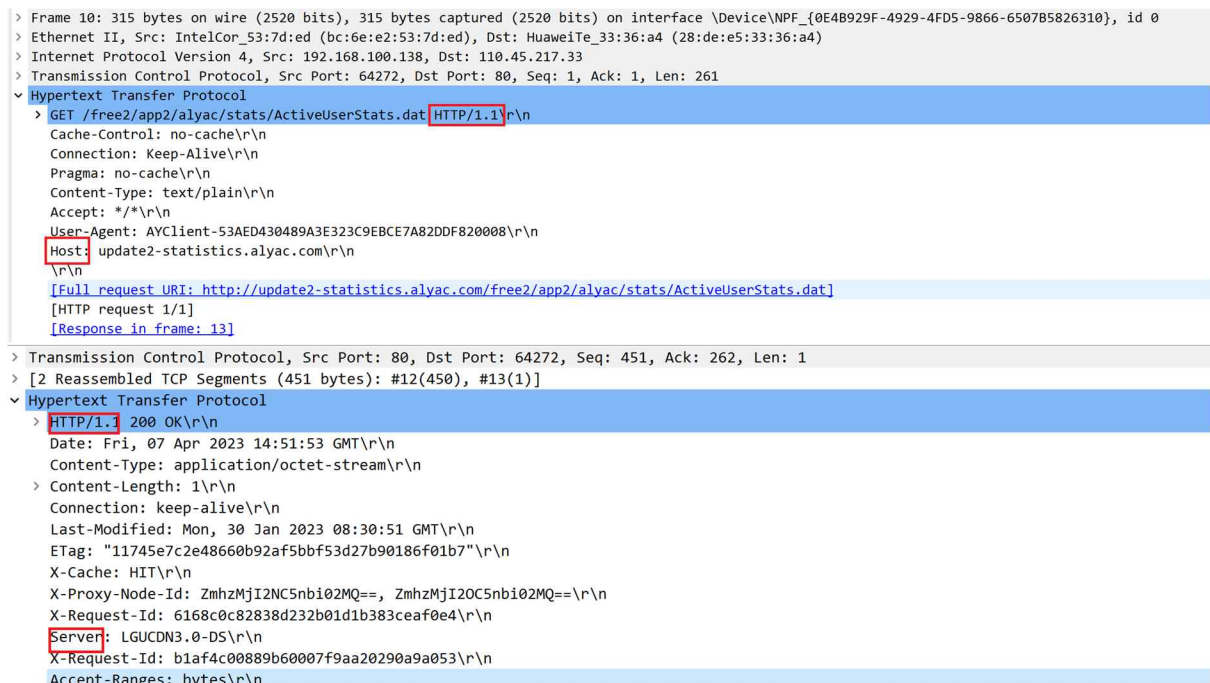
1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?



Wireshark 실행 후 wifi를 선택하여 패킷 캡처를 진행했습니다.



주어진 문제에서 알려준 웹 브라우저 <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html> 를 실행 후 캡처한 http의 패킷입니다.



->브라우저와 서버 모두 HTTP version 1.1을 사용합니다.

2. What languages (if any) does your browser indicate that it can accept to the server?

```
> Frame 3535: 653 bytes on wire (5224 bits), 653 bytes captured (5224 bit S
> Ethernet II, Src: IntelCor_53:7d:ed (bc:6e:e2:53:7d:ed), Dst: Ubiquiti_
> Internet Protocol Version 4, Src: 10.10.3.220, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 61281, Dst Port: 80, Seq: 1, A
v Hypertext Transfer Protocol
  > GET /wireshark-labs/HTTP-wireshark-file1.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
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/v
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: ko,en;q=0.9,en-US;q=0.8\r\n
    If-None-Match: "80-5f8a49afd670f"\r\n
```

->language : ko, en을 사용하는 것을 알 수 있습니다.

3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?

```
> Frame 3535: 653 bytes on wire (5224 bits), 653 bytes captured (5224 bit
> Ethernet II, Src: IntelCor_53:7d:ed (bc:6e:e2:53:7d:ed), Dst: Ubiquiti_
> Internet Protocol Version 4, Src: 10.10.3.220, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 61281, Dst Port: 80, Seq: 1, A
v Hypertext Transfer Protocol
  > GET /wireshark-labs/HTTP-wireshark-file1.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
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/v
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: ko,en;q=0.9,en-US;q=0.8\r\n
    If-None-Match: "80-5f8a49afd670f"\r\n
```

->내 pc : 10.10.3.220

gaia.cs.umass.edu :128.119.245.12

4. What is the status code returned from the server to your browser?

```
Transmission Control Protocol, Src Port: 80, Dst Port: 65036, Seq: 1, Ack: 515, Len: 486
Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
Date: Fri, 07 Apr 2023 15:13:22 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
Last-Modified: Fri, 07 Apr 2023 05:59:01 GMT\r\n
ETag: "80-5f8b8b8d0fbab"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 128\r\n
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=UTF-8\r\n
\r\n
[HTTP response 1/2]
```

-> 200 OK

5. When was the HTML file that you are retrieving last modified at the server?

```
Transmission Control Protocol, Src Port: 80, Dst Port: 65036, Seq: 1, Ack: 515, Len: 486
Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
Date: Fri, 07 Apr 2023 15:13:22 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
Last-Modified: Fri, 07 Apr 2023 05:59:01 GMT\r\n
ETag: "80-5f8b8b8d0fbab"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 128\r\n
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=UTF-8\r\n
\r\n
[HTTP response 1/2]
```

-> FRI, 07, Apr 2023 05:59:01 GMT

6. How many bytes of content are being returned to your browser?

```
Transmission Control Protocol, Src Port: 80, Dst Port: 65036, Seq: 1, Ack: 515, Len: 486
Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
Date: Fri, 07 Apr 2023 15:13:22 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
Last-Modified: Fri, 07 Apr 2023 05:59:01 GMT\r\n
ETag: "80-5f8b8b8d0fbab"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 128\r\n
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=UTF-8\r\n
\r\n
[HTTP response 1/2]
```

-> 크기는 128 bytes며 제 브라우저에 반환되고 있습니다.

7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one

-> 없습니다. 패킷 창에 나오지 않는 헤더는 보이지 않습니다.

8. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?

```
> Frame 96: 542 bytes on wire (4336 bits), 542 bytes captured (4336 bits) on interface \Device\NPF_{0E4B929F-4929-4FD5-9866-6507B5826310}
> Ethernet II, Src: IntelCor_53:7d:ed (bc:6e:e2:53:7d:ed), Dst: HuaweiTe_33:36:a4 (28:de:e5:33:36:a4)
> Internet Protocol Version 4, Src: 192.168.100.138, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 49218, Dst Port: 80, Seq: 1, Ack: 1, Len: 488
▼ Hypertext Transfer Protocol
  > GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36 Edg/112.0.176.62\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: ko,en;q=0.9,en-US;q=0.8\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: 117]
```

-> first HTTP GET request에서는 "IF-MODIFIED-SINCE"를 찾을 수 없습니다.

9. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

```
[HTTP response 1/1]
[Time since request: 0.211549000 seconds]
[Request in frame: 96]
[Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
File Data: 371 bytes
▼ Line-based text data: text/html (10 lines)
  \n
  <html>\n
  \n
  Congratulations again! Now you've downloaded the file lab2-2.html. <br>\n
  This file's last modification date will not change. <p>\n
  Thus if you download this multiple times on your browser, a complete copy <br>\n
  will only be sent once by the server due to the inclusion of the IF-MODIFIED-SINCE<br>\n
  field in your browser's HTTP GET request to the server.\n
  \n
  </html>\n
```

-> 네 서버는 파일의 내용을 반환하고 있습니다. 이는 lined-based text data에서 볼 수 있습니다.

10. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE:" line in the HTTP GET? If so, what information follows the "IF-MODIFIED-SINCE:" header?

```
▼ 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\n]
    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 Gecko) Chrome/112.0.0.0 Safari/537.36 Edg/112.0.176.62\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: ko,en;q=0.9,en-US;q=0.8\r\n
  If-None-Match: "173-5f8b8b8d0eff3"\r\n
  If-Modified-Since: Fri, 07 Apr 2023 05:59:01 GMT\r\n
  \r\n
```

-> IF-MODIFIED-SINCE를 확인할 수 있습니다.

-> Information으로써는 FRI, 07 APR 2023 05:59:01 GMT를 확인할 수 있습니다.

이는 이전 요청에서 이 파일을 마지막으로 수정한 날짜를 나타냅니다.

11. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

56	4.030566	192.168.100.138	128.119.245.12	HTTP	654 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
58	4.302475	128.119.245.12	192.168.100.138	HTTP	294 HTTP/1.1 304 Not Modified

->처음에 파일의 정보를 가져올 때는 캐시에 저장된 내용이 없어 서버에서 정보를 가져왔고 이 과정 사이에 캐시에 정보가 저장되게 됩니다. 그 후 호스트가 request하면 캐시에 있는 정보가 반환되어 서버는 반환하지 않습니다. 브라우저가 파일 내용을 캐시에서 가져왔기 때문에 서버는 파일의 내용을 반환하지 않습니다. 따라서 304 not Modified를 반환합니다.

12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?

198	11.857329	192.168.100.138	128.119.245.12	HTTP	542 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
226	12.063702	128.119.245.12	192.168.100.138	HTTP	775 HTTP/1.1 200 OK (text/html)

-> HTTP GET request messages는 한번 보냈으며 packet number는 198입니다.

13. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

198	11.857329	192.168.100.138	128.119.245.12	HTTP	542 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
226	12.063702	128.119.245.12	192.168.100.138	HTTP	775 HTTP/1.1 200 OK (text/html)

-> packet number는 226입니다.

14. What is the status code and phrase in the response?

198	11.857329	192.168.100.138	128.119.245.12	HTTP	542 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
226	12.063702	128.119.245.12	192.168.100.138	HTTP	775 HTTP/1.1 200 OK (text/html)

-> status code는 200이며 phrase in the response는 OK입니다.

15. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

223	12.063702	128.119.245.12	192.168.100.138	TCP	1434 80 → 50326 [ACK] Seq=1 Ack=489 Win=30336 Len=1380 [TCP segment of a reassembled PDU]
224	12.063702	128.119.245.12	192.168.100.138	TCP	1434 80 → 50326 [ACK] Seq=1381 Ack=489 Win=30336 Len=1380 [TCP segment of a reassembled PDU]
225	12.063702	128.119.245.12	192.168.100.138	TCP	1434 80 → 50326 [ACK] Seq=2761 Ack=489 Win=30336 Len=1380 [TCP segment of a reassembled PDU]

-> the single HTTP response and the text of the Bill of Rights를 전달하기 위해서 총 3개의 data-containing TCP segments가 필요합니다.

16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

31	8.463947	192.168.100.138	114.108.156.62	HTTP	315 GET /free2/app2/alyac/stats/ActiveUserStats.dat HTTP/1.1
34	8.474825	114.108.156.62	192.168.100.138	HTTP	55 HTTP/1.1 200 OK
63	11.611960	192.168.100.138	128.119.245.12	HTTP	654 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
70	11.819663	128.119.245.12	192.168.100.138	HTTP	294 HTTP/1.1 304 Not Modified
475	30.369804	192.168.100.138	128.119.245.12	HTTP	542 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
505	30.583622	128.119.245.12	192.168.100.138	HTTP	1355 HTTP/1.1 200 OK (text/html)
515	30.593698	192.168.100.138	128.119.245.12	HTTP	488 GET /pearson.png HTTP/1.1
599	30.937841	192.168.100.138	178.79.137.164	HTTP	455 GET /8E_cover_small.jpg HTTP/1.1
614	31.605301	128.119.245.12	192.168.100.138	HTTP	905 HTTP/1.1 200 OK (PNG)
619	31.662319	178.79.137.164	192.168.100.138	HTTP	225 HTTP/1.1 301 Moved Permanently

-> 총 3개의 HTTP Get request message를 보내는 것을 확인할 수 있다.

요청한 주소: 128.119.245.12, 128.119.245.12, 178.79.137.164

17. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

-> 이미지 두개를 순차적으로 다운로드 받게 됩니다. pearson.png, 8E_cover_small.png 순서로 다운로드 받습니다.

18. What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?

35	4.801170	192.168.100.138	128.119.245.12	HTTP	558 GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
42	5.057074	128.119.245.12	192.168.100.138	HTTP	771 HTTP/1.1 401 Unauthorized (text/html)

-> HTTP/1.1 401 Unauthorized

-> status code: 401

-> phrase: Unauthorized

19. When your browser's sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

```
> Transmission Control Protocol, Src Port: 51362, Dst Port: 80, Seq: 1, Ack: 1, Len: 577
▼ Hypertext Transfer Protocol
  ▼ GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /wireshark-labs/protected_pages/HTTP-wireshark-file5.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
      ▼ Authorization: Basic 64Ko7KKF7IudOjEyMzQ1Njc4\r\n
        Credentials: 남종식:12345678
      Upgrade-Insecure-Requests: 1\r\n
      User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36 Edg/112.0
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
      Accept-Encoding: gzip, deflate\r\n
      Accept-Language: ko,en-gb;q=0.9,en-us;q=0.8\r\n
```

-> Authorization을 새로운 필드라고 할 수 있으며 제가 처음에 직접 입력했던 사용자와 암호가 encoding되어 있습니다.

Wireshark_DNS_v7.0.pdf의 23개 문제 풀이

1. Run nslookup to obtain the IP address of a Web server in Asia. What is the IP address of that server?

```
C:\Users\Wjongs>nslookup www.naver.com
서버: ns.dacom.co.kr
Address: 164.124.101.2

권한 없는 응답:
이름: www.naver.com.nheos.com
Addresses: 223.130.195.200
           223.130.195.95
Aliases: www.naver.com
```

->서버의 주소는 223.130.195.200/223.130.195.95입니다. 네이버의 ip주소 조회를 통해 불러왔습니다.

2. Run nslookup to determine the authoritative DNS servers for a university in Europe.

```
C:\Users\Wjongs>nslookup -type=NS www.ox.ac.uk
DNS request timed out.
    timeout was 2 seconds.
서버: UnKnown
Address: 164.124.101.2

ox.ac.uk
    primary name server = raptor.dns.ox.ac.uk
    responsible mail addr = hostmaster.ox.ac.uk
    serial = 2023040664
    refresh = 3600 (1 hour)
    retry = 1800 (30 mins)
    expire = 1209600 (14 days)
    default TTL = 900 (15 mins)
```

->유럽의 옥스퍼드 대학의 authoritative DNS server는 raptor.dns.ox.ac.uk입니다.

3. Run nslookup so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Yahoo! mail. What is its IP address?

```

C:\Users\Wjongs>nslookup raptor.dns.ox.ac.uk mail.yahoo.com
DNS request timed out.
    timeout was 2 seconds.
서버:      Unknown
Address: 119.161.5.248

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.

```

->ip address는 119.161.5.248임을 확인했습니다.

4. Locate the DNS query and response messages. Are then sent over UDP or TCP?

59462	64.559068	192.168.94.201	192.168.94.54	DNS	72 Standard query 0x0273 A www.ietf.org	
> Frame 59337: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface \Device\NPF_{0E48929F-4929-4FD5-9866-6507B5826310}, 1						
> Ethernet II, Src: IntelCor_53:7d:ed (bc:6e:e2:53:7d:ed), Dst: be:31:3c:b2:cd:15 (be:31:3c:b2:cd:15)						
> Internet Protocol Version 4, Src: 192.168.94.201, Dst: 192.168.94.54						
> User Datagram Protocol, Src Port: 56309, Dst Port: 53						
Source Port: 56309						
Destination Port: 53						
Length: 38						
Checksum: 0x3e88 [unverified]						
[Checksum Status: Unverified]						
[Stream index: 125]						
> [Timestamps]						
UDP payload (30 bytes)						
> Domain Name System (query)						
						0000 be 31 3c b2 cd 15 bc 6e 0010 00 3a c1 ab 00 00 80 11 0020 5e 36 db f5 00 35 00 26 0030 00 00 00 00 00 03 77 0040 6f 72 67 00 00 01 00 01

->query와 response는 UDP를 통해 전달됩니다.

5. What is the destination port for the DNS query message? What is the source port of DNS response message?

41545	12.341326	172.30.1.1	168.126.63.1	DNS	74 Standard query 0x32e9 A www.google.com
41546	12.341771	172.30.1.1	168.126.63.1	DNS	74 Standard query 0x0729 HTTPS www.google.com
41713	12.385078	168.126.63.1	172.30.1.1	DNS	99 Standard query response 0x0729 HTTPS www.google.com HTTPS
41714	12.385078	168.126.63.1	172.30.1.1	DNS	90 Standard query response 0x32e9 A www.google.com A 142.250.76.132
42568	12.582275	172.30.1.1	168.126.63.1	DNS	75 Standard query 0x88a3 A chat.openai.com
42569	12.582762	172.30.1.1	168.126.63.1	DNS	75 Standard query 0x9940 HTTPS chat.openai.com
42776	12.629524	168.126.63.1	172.30.1.1	DNS	155 Standard query response 0x88a3 A chat.openai.com CNAME chat.openai.com.cdn.cloudflare.net
42777	12.629524	168.126.63.1	172.30.1.1	DNS	202 Standard query response 0x9940 HTTPS chat.openai.com CNAME chat.openai.com.cdn.cloudflare.net
44995	13.089605	172.30.1.1	168.126.63.1	DNS	86 Standard query 0x3f42 A encrypted-tbn0.gstatic.com
44996	13.089755	172.30.1.1	168.126.63.1	DNS	86 Standard query 0xb8db HTTPS encrypted-tbn0.gstatic.com
45192	13.143552	168.126.63.1	172.30.1.1	DNS	102 Standard query response 0x3f42 A encrypted-tbn0.gstatic.com A 172.217.25.174
45193	13.143552	168.126.63.1	172.30.1.1	DNS	146 Standard query response 0xb8db HTTPS encrypted-tbn0.gstatic.com SOA ns1.google.com
46242	13.348898	172.30.1.1	168.126.63.1	DNS	74 Standard query 0xdf9e A lh3.google.com
46243	13.349302	172.30.1.1	168.126.63.1	DNS	74 Standard query 0xed1e HTTPS lh3.google.com
46452	13.393226	168.126.63.1	172.30.1.1	DNS	154 Standard query response 0xed1e HTTPS lh3.google.com CNAME lh2.l.google.com SOA ns1.google.com

> Frame 41545: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{0E4B929F-4929-4FD5-9866-6507B5826310}, interface 0
> Ethernet II, Src: IntelCor_53:7d:ed (bc:6e:e2:53:7d:ed), Dst: Mercury_c5:5b:15 (b4:a9:4f:c5:5b:15)
> Internet Protocol Version 4, Src: 172.30.1.1, Dst: 168.126.63.1
 v User Datagram Protocol, Src Port: 62578, Dst Port: 53
 Source Port: 62578
 Destination Port: 53
 Length: 40
 Checksum: 0x94d8 [unverified]
 [Checksum Status: Unverified]
 [Stream index: 8]
 > [Timestamps]
 UDP payload (32 bytes)

41545	12.341326	172.30.1.1	168.126.63.1	DNS	74 Standard query 0x32e9 A www.google.com
41546	12.341771	172.30.1.1	168.126.63.1	DNS	74 Standard query 0x0729 HTTPS www.google.com
41713	12.385078	168.126.63.1	172.30.1.1	DNS	99 Standard query response 0x0729 HTTPS www.google.com HTTPS
41714	12.385078	168.126.63.1	172.30.1.1	DNS	90 Standard query response 0x32e9 A www.google.com A 142.250.76.132
42568	12.582275	172.30.1.1	168.126.63.1	DNS	75 Standard query 0x88a3 A chat.openai.com
42569	12.582762	172.30.1.1	168.126.63.1	DNS	75 Standard query 0x9940 HTTPS chat.openai.com
42776	12.629524	168.126.63.1	172.30.1.1	DNS	155 Standard query response 0x88a3 A chat.openai.com CNAME chat.openai.com.cdn.cloudflare.net
42777	12.629524	168.126.63.1	172.30.1.1	DNS	202 Standard query response 0x9940 HTTPS chat.openai.com CNAME chat.openai.com.cdn.cloudflare.net
44995	13.089605	172.30.1.1	168.126.63.1	DNS	86 Standard query 0x3f42 A encrypted-tbn0.gstatic.com
44996	13.089755	172.30.1.1	168.126.63.1	DNS	86 Standard query 0xb8db HTTPS encrypted-tbn0.gstatic.com
45192	13.143552	168.126.63.1	172.30.1.1	DNS	102 Standard query response 0x3f42 A encrypted-tbn0.gstatic.com A 172.217.25.174
45193	13.143552	168.126.63.1	172.30.1.1	DNS	146 Standard query response 0xb8db HTTPS encrypted-tbn0.gstatic.com SOA ns1.google.com
46242	13.348898	172.30.1.1	168.126.63.1	DNS	74 Standard query 0xdf9e A lh3.google.com
46243	13.349302	172.30.1.1	168.126.63.1	DNS	74 Standard query 0xed1e HTTPS lh3.google.com
46452	13.393226	168.126.63.1	172.30.1.1	DNS	154 Standard query response 0xed1e HTTPS lh3.google.com CNAME lh2.l.google.com SOA ns1.google.com

> Frame 41714: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device\NPF_{0E4B929F-4929-4FD5-9866-6507B5826310}, interface 0
> Ethernet II, Src: Mercury_c5:5b:15 (b4:a9:4f:c5:5b:15), Dst: IntelCor_53:7d:ed (bc:6e:e2:53:7d:ed)
> Internet Protocol Version 4, Src: 168.126.63.1, Dst: 172.30.1.1
 v User Datagram Protocol, Src Port: 53, Dst Port: 62578
 Source Port: 53
 Destination Port: 62578
 Length: 56
 Checksum: 0x97b3 [unverified]
 [Checksum Status: Unverified]
 [Stream index: 8]
 > [Timestamps]
 UDP payload (48 bytes)

->DNS query message의 destination port와 DNS response message의 source port가 서로 53으로 똑같다는 것을 확인할 수 있습니다.

6. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?

```
> Frame 59337: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface \Device\NPF_{0E4B929F-4929-4FD5-9866-6507B5826310}, i
> Ethernet II, Src: IntelCor_53:7d:ed (bc:6e:e2:53:7d:ed), Dst: be:31:3c:b2:cd:15 (be:31:3c:b2:cd:15)
> Internet Protocol Version 4, Src: 192.168.94.201, Dst: 192.168.94.54
v User Datagram Protocol, Src Port: 56309, Dst Port: 53

무선 LAN 어댑터 Wi-Fi:

연결별 DNS 접미사. . . . . :
설명. . . . . : Intel(R) Wi-Fi 6E AX211 160MHz
물리적 주소. . . . . : BC-6E-E2-53-7D-ED
DHCP 사용. . . . . : 예
자동 구성 사용. . . . . : 예
IPv6 주소. . . . . : 2001:e60:9363:944:63b2:8475:53e:b32f(기본 설정)
임시 IPv6 주소. . . . . : 2001:e60:9363:944:c826:c027:4c24:a64f(기본 설정)
링크-로컬 IPv6 주소. . . . : fe80::88bf:3ec9:b7ff:9f7a%4(기본 설정)
IPv4 주소. . . . . : 192.168.94.201(기본 설정)
서브넷 마스크. . . . . : 255.255.255.0
임대 시작 날짜. . . . . : 2023년 4월 8일 토요일 오전 3:56:46
임대 만료 날짜. . . . . : 2023년 4월 8일 토요일 오전 5:26:56
기본 게이트웨이. . . . . : fe80::bc31:3cff:feb2:cd15%4
. . . . . : 192.168.94.54
DHCP 서버. . . . . : 192.168.94.54
DHCPv6 IAID. . . . . : 62680802
DHCPv6 클라이언트 DUID. . . : 00-01-00-01-2A-5A-55-0D-00-E0-4C-71-FC-52
DNS 서버. . . . . : 192.168.94.54
Tcpip를 통한 NetBIOS. . . . : 사용
```

-> 192.168.94.54로 서로 동일하다는 점을 알 수 있습니다.

7. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?

```
v Domain Name System (query)
Transaction ID: 0xe332
> Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
v Queries
> www.ietf.org: type A, class IN
[Response In: 59377]
```

-> query message는 어떠한 answer를 포함하지 않으며 이는 type A query입니다.

8. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?

```
v Domain Name System (response)
Transaction ID: 0x0273
> Flags: 0x8180 Standard query response, No error
Questions: 1
Answer RRs: 3
Authority RRs: 0
Additional RRs: 0
v Queries
> www.ietf.org: type A, class IN
v Answers
> www.ietf.org: type CNAME, class IN, cname www.ietf.org.cdn.cloudflare.net
> www.ietf.org.cdn.cloudflare.net: type A, class IN, addr 104.16.45.99
> www.ietf.org.cdn.cloudflare.net: type A, class IN, addr 104.16.44.99
[Request In: 59462]
```

-> DNS response message는 3개의 answers를 제공합니다. 여기에는 조회된 웹사이트 주소와 타입 그리고 class, TTL, data length와 주소를 포함하고 있습니다.

9. Consider the subsequent TCP SYN packet sent by your host. Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message?

```
> www.ietf.org.cdn.cloudflare.net: type A, class IN, addr 104.16.45.99
> www.ietf.org.cdn.cloudflare.net: type A, class IN, addr 104.16.44.99
```

```

v 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: 300 (5 minutes)
  Data length: 4
  Address: 104.16.44.99
v 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: 300 (5 minutes)
  Data length: 4
  Address: 104.16.45.99

```

1279	57.133629	192.168.123.113	104.16.44.99	TCP	66 64493 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1301	57.133629	192.168.123.113	104.16.44.99	TCP	66 64493 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2085	85.762604	192.168.123.113	104.16.45.99	TCP	66 64519 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

-> SYN packet의 대상 IP Address는 DNS response인 104.16.45.99와 104.16.44.99에 의해 제공된 주소에 해당됩니다.

10. This web page contains images. Before retrieving each image, does your host issue new DNS queries?

-> host는 image file에 대한 DNS query를 발행하지 않습니다.

11. What is the destination port for the DNS query message? What is the source port of DNS response message?

65948	59.679591	192.168.94.201	192.168.94.54	DNS	71 Standard query 0x159d A www.mit.edu
65949	59.679743	192.168.94.201	192.168.94.54	DNS	71 Standard query 0x5a46 AAAA www.mit.edu
65995	59.735140	192.168.94.201	192.168.94.54	DNS	71 Standard query 0x1f38 A www.mit.edu
65996	59.735346	192.168.94.201	192.168.94.54	DNS	71 Standard query 0x2c9b AAAA www.mit.edu
65997	59.735476	192.168.94.201	192.168.94.54	DNS	71 Standard query 0x8eb8 HTTPS www.mit.edu
66538	60.038409	192.168.94.201	192.168.94.54	DNS	81 Standard query 0x8a39 A edgeservices.bing.com
66539	60.038599	192.168.94.201	192.168.94.54	DNS	81 Standard query 0x2931 AAAA edgeservices.bing.com
66550	60.044132	192.168.94.201	192.168.94.54	DNS	81 Standard query 0x96c0 A edgeservices.bing.com
66551	60.044286	192.168.94.201	192.168.94.54	DNS	81 Standard query 0xa3c9 AAAA edgeservices.bing.com
66552	60.044386	192.168.94.201	192.168.94.54	DNS	81 Standard query 0xf263 HTTPS edgeservices.bing.com
66622	60.076255	192.168.94.54	192.168.94.201	DNS	310 Standard query response 0x8a39 A edgeservices.bing.com CNAME www.bing.com CNAME www-
66623	60.079729	192.168.94.54	192.168.94.201	DNS	306 Standard query response 0x2931 AAAA edgeservices.bing.com CNAME www.bing.com CNAME w
66640	60.096054	192.168.94.54	192.168.94.201	DNS	310 Standard query response 0x96c0 A edgeservices.bing.com CNAME www.bing.com CNAME ww-
66643	60.099783	192.168.94.54	192.168.94.201	DNS	306 Standard query response 0xa3c9 AAAA edgeservices.bing.com CNAME www.bing.com CNAME w
66646	60.103372	192.168.94.54	192.168.94.201	DNS	267 Standard query response 0xf263 HTTPS edgeservices.bing.com CNAME www.bing.com CNAME
66658	60.113415	192.168.94.54	192.168.94.201	DNS	524 Standard query response 0x5a46 AAAA www.mit.edu CNAME www.mit.edu.edgekey.net CNAME

> Frame 65948: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_{0E4B929F-4929-4FD5-9866-6507B5826310},	0000	be 31 3c b2 cd 1
> Ethernet II, Src: IntelCor_53:7d:ed (bc:6e:e2:53:7d:ed), Dst: be:31:3c:b2:cd:15 (be:31:3c:b2:cd:15)	0010	00 39 c3 ef 00 0
> Internet Protocol Version 4, Src: 192.168.94.201, Dst: 192.168.94.54	0020	5e 36 cd 73 00 3
> User Datagram Protocol, Src Port: 52595, Dst Port: 53	0030	00 00 00 00 0 0
	0040	64 75 00 00 01 0


```

Source Port: 52595
Destination Port: 53
Length: 37
Checksum: 0x3e87 [unverified]
[Checksum Status: Unverified]
[Stream index: 20]
> [Timestamps]
UDP payload (29 bytes)
v Domain Name System (query)
  Transaction ID: 0x159d
  Flags: 0x0100 Standard query
  Questions: 1
  Answer RRs: 0

```

66539	60.038599	192.168.94.201	192.168.94.54	DNS	81 Standard query 0x2931 AAAA edgeservices.bing.com
66550	60.044132	192.168.94.201	192.168.94.54	DNS	81 Standard query 0x96c0 A edgeservices.bing.com
66551	60.044286	192.168.94.201	192.168.94.54	DNS	81 Standard query 0xa3c9 AAAA edgeservices.bing.com
66552	60.044386	192.168.94.201	192.168.94.54	DNS	81 Standard query 0xf263 HTTPS edgeservices.bing.com
66622	60.076255	192.168.94.54	192.168.94.201	DNS	310 Standard query response 0x8a39 A edgeservices.bing.com CNAME www.bing.com CNAME www-ww...
66623	60.079729	192.168.94.54	192.168.94.201	DNS	306 Standard query response 0x2931 AAAA edgeservices.bing.com CNAME www.bing.com CNAME www-ww...
66640	60.096054	192.168.94.54	192.168.94.201	DNS	310 Standard query response 0x96c0 A edgeservices.bing.com CNAME www.bing.com CNAME www-ww...
66643	60.099783	192.168.94.54	192.168.94.201	DNS	306 Standard query response 0xa3c9 AAAA edgeservices.bing.com CNAME www.bing.com CNAME www-ww...
66646	60.103372	192.168.94.54	192.168.94.201	DNS	267 Standard query response 0xf263 HTTPS edgeservices.bing.com CNAME www.bing.com CNAME www-ww...
66658	60.113415	192.168.94.54	192.168.94.201	DNS	524 Standard query response 0x5a46 AAAA www.mit.edu CNAME www.mit.edu.edgekey.net CNAME e956...
66700	60.136817	192.168.94.54	192.168.94.201	DNS	524 Standard query response 0x2c9b AAAA www.mit.edu CNAME www.mit.edu.edgekey.net CNAME e956...
66716	60.172389	192.168.94.54	192.168.94.201	DNS	484 Standard query response 0x159d A www.mit.edu CNAME www.mit.edu.edgekey.net CNAME e956...
66776	60.200621	192.168.94.54	192.168.94.201	DNS	484 Standard query response 0x1f38 A www.mit.edu CNAME www.mit.edu.edgekey.net CNAME e956...
66798	60.204526	192.168.94.54	192.168.94.201	DNS	208 Standard query response 0x8eb8 HTTPS www.mit.edu CNAME www.mit.edu.edgekey.net CNAME e95...
66922	60.356979	192.168.94.201	192.168.94.54	DNS	71 Standard query 0x7458 A www.mit.edu
66923	60.357164	192.168.94.201	192.168.94.54	DNS	71 Standard query 0x1022 AAAA www.mit.edu

> Frame 66716: 484 bytes on wire (3872 bits), 484 bytes captured (3872 bits) on interface \Device\NPF_{0E4B929F-4929-4FD5-9866-6507B58263} ...

> Ethernet II, Src: be:31:3c:b2:cd:15 (be:31:3c:b2:cd:15), Dst: IntelCor_53:7d:ed (bc:6e:21:53:7d:ed)

> Internet Protocol Version 4, Src: 192.168.94.54, Dst: 192.168.94.201

> User Datagram Protocol, Src Port: 53, Dst Port: 52595

Source Port: 53

Destination Port: 52595

Length: 450

Checksum: 0x0b42 [unverified]

[Checksum Status: Unverified]

[Stream index: 20]

> [Timestamps]

UDP payload (442 bytes)

> Domain Name System (response)

Transaction ID: 0x159d

> Flags: 0x8180 Standard query response, No error

Questions: 1

->DNS query message의 destination port는 53이며 DNA response message의 source port는 53입니다.

12. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?

>	Internet Protocol Version 4, Src: 192.168.94.201, Dst: 192.168.94.54
>	User Datagram Protocol, Src Port: 52595, Dst Port: 53
	Source Port: 52595
	Destination Port: 53
	Length: 37
	Checksum: 0x3e87 [unverified]
	[Checksum Status: Unverified]
	[Stream index: 20]
	> [Timestamps]
	UDP payload (29 bytes)
>	Domain Name System (query)
	Transaction ID: 0x159d
	> Flags: 0x0100 Standard query
	Questions: 1
	Answer RRs: 0

무선 LAN 어댑터 Wi-Fi:

```

연결별 DNS 접미사. . . . . :
설명. . . . . : Intel(R) Wi-Fi 6E AX211 160MHz
물리적 주소. . . . . : BC-6E-E2-53-7D-ED
DHCP 사용. . . . . : 예
자동 구성 사용. . . . . : 예
IPv6 주소. . . . . : 2001:e60:9363:944:63b2:8475:53e:b32f(기본 설정)
임시 IPv6 주소. . . . . : 2001:e60:9363:944:c826:c027:4c24:a64f(기본 설정)
링크-로컬 IPv6 주소. . . . . : fe80::88bf:3ec9:b7ff:9f7a%4(기본 설정)
IPv4 주소. . . . . : 192.168.94.201(기본 설정)
서브넷 마스크. . . . . : 255.255.255.0
임대 시작 날짜. . . . . : 2023년 4월 8일 토요일 오전 3:56:46
임대 만료 날짜. . . . . : 2023년 4월 8일 토요일 오전 5:26:56
기본 게이트웨이. . . . . : fe80::bc31:3cff:feb2:cd15%4
. . . . . : 192.168.94.54
DHCP 서버. . . . . : 192.168.94.54
DHCPv6 IAID. . . . . : 62680802
DHCPv6 클라이언트 DUID. . . . . : 00-01-00-01-2A-5A-55-0D-00-E0-4C-71-FC-52
DNS 서버. . . . . : 192.168.94.54
Tcpip를 통한 NetBIOS. . . . . : 사용

```

->DNS query message는 192.168.94.54로 전송됩니다. 저의 local DNS server의 ip주소와 일치하는 것을 확인할 수 있습니다.

13. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?

```

> [Timestamps]
  UDP payload (29 bytes)
- Domain Name System (query)
  Transaction ID: 0x159d
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  - Queries
    - www.mit.edu: type A, class IN
      Name: www.mit.edu
      [Name Length: 11]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
\[Response In: 66716\]

```

-> query는 A type이며 answers를 포함하고 있지 않습니다.

14. Examine the DNS response message. How many "answers" are provided?

What do each of these answers contain?

```

Checksum: 0xb42 [unverified]
[Checksum Status: Unverified]
[Stream index: 20]
> [Timestamps]
  UDP payload (442 bytes)
- Domain Name System (response)
  Transaction ID: 0x159d
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 3
    Authority RRs: 8
    Additional RRs: 9
  - Queries
    - www.mit.edu: type A, class IN
      Name: www.mit.edu
      [Name Length: 11]
      [Label Count: 3]

- Answers
  > www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
  > www.mit.edu.edgekey.net: type CNAME, class IN, cname e9566.dscb.akamaiedge.net
  > e9566.dscb.akamaiedge.net: type A, class IN, addr 104.74.211.78

- Answers
  - www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
    Name: www.mit.edu
    Type: CNAME (Canonical NAME for an alias) (5)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 25
    CNAME: www.mit.edu.edgekey.net
  - www.mit.edu.edgekey.net: type CNAME, class IN, cname e9566.dscb.akamaiedge.net
    Name: www.mit.edu.edgekey.net
    Type: CNAME (Canonical NAME for an alias) (5)
    Class: IN (0x0001)
    Time to live: 60 (1 minute)
    Data length: 24
    CNAME: e9566.dscb.akamaiedge.net
  - e9566.dscb.akamaiedge.net: type A, class IN, addr 104.74.211.78
    Name: e9566.dscb.akamaiedge.net
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 20 (20 seconds)
    Data length: 4
    Address: 104.74.211.78

```

-> DNS response message는 총 3개의 answers를 포함하고 있습니다. 이는 주소의 타입과 CName, class, host name, ip 주소를 포함합니다. Screenshot를 포함하였습니다.

15. Provide a screenshot.

```
Checksum: 0x0b42 [unverified]
[Checksum Status: Unverified]
[Stream index: 20]
> [Timestamps]
UDP payload (442 bytes)
Domain Name System (response)
Transaction ID: 0x159d
> Flags: 0x8180 Standard query response, No error
Questions: 1
Answer RRs: 3
Authority RRs: 8
Additional RRs: 9
v Queries
  www.mit.edu: type A, class IN
    Name: www.mit.edu
    [Name Length: 11]
    [Label Count: 3]

Answers
> www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
> www.mit.edu.edgekey.net: type CNAME, class IN, cname e9566.dscb.akamaiedge.net
> e9566.dscb.akamaiedge.net: type A, class IN, addr 104.74.211.78

Answers
v www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
  Name: www.mit.edu
  Type: CNAME (Canonical NAME for an alias) (5)
  Class: IN (0x0001)
  Time to live: 1800 (30 minutes)
  Data length: 25
  CNAME: www.mit.edu.edgekey.net
v www.mit.edu.edgekey.net: type CNAME, class IN, cname e9566.dscb.akamaiedge.net
  Name: www.mit.edu.edgekey.net
  Type: CNAME (Canonical NAME for an alias) (5)
  Class: IN (0x0001)
  Time to live: 60 (1 minute)
  Data length: 24
  CNAME: e9566.dscb.akamaiedge.net
e9566.dscb.akamaiedge.net: type A, class IN, addr 104.74.211.78
  Name: e9566.dscb.akamaiedge.net
  Type: A (Host Address) (1)
  Class: IN (0x0001)
  Time to live: 20 (20 seconds)
  Data length: 4
  Address: 104.74.211.78
```

16. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?

```
> Internet Protocol Version 4, Src: 192.168.94.201, Dst: 192.168.94.54
v User Datagram Protocol, Src Port: 52595, Dst Port: 53
  Source Port: 52595
  Destination Port: 53
  Length: 37
  Checksum: 0x3e87 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 20]
> [Timestamps]
UDP payload (29 bytes)
Domain Name System (query)
Transaction ID: 0x159d
> Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
```


무선 LAN 어댑터 Wi-Fi:

```
연결별 DNS 접미사. . . . . :  
설명. . . . . : Intel(R) Wi-Fi 6E AX211 160MHz  
물리적 주소. . . . . : BC-6E-E2-53-7D-ED  
DHCP 사용. . . . . : 예  
자동 구성 사용. . . . . : 예  
IPv6 주소. . . . . : 2001:e60:9363:944:63b2:8475:53e:b32f(기본 설정)  
임시 IPv6 주소. . . . . : 2001:e60:9363:944:c826:c027:4c24:a64f(기본 설정)  
링크-로컬 IPv6 주소. . . . : fe80::88bf:3ec9:b7ff:9f7a%4(기본 설정)  
IPv4 주소. . . . . : 192.168.94.201(기본 설정)  
서브넷 마스크. . . . . : 255.255.255.0  
임대 시작 날짜. . . . . : 2023년 4월 8일 토요일 오전 3:56:46  
임대 만료 날짜. . . . . : 2023년 4월 8일 토요일 오전 5:26:56  
기본 게이트웨이. . . . . : fe80::bc31:3cff:feb2:cd15%4  
192.168.94.54  
DHCP 서버. . . . . : 192.168.94.54  
DHCPv6 IAID. . . . . : 62680802  
DHCPv6 클라이언트 DUID. . . : 00-01-00-01-2A-5A-55-0D-00-E0-4C-71-FC-52  
DNS 서버. . . . . : 192.168.94.54  
Tcpip를 통한 NetBIOS. . . . : 사용
```

->DNS query message와 저의 local DNS server는 192.168.94.54로 동일합니다.

17. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?

The image shows a Wireshark packet capture of a DNS query. The packet list at the top shows a standard query for mit.edu. The packet details pane on the right shows the query structure, including the transaction ID, flags, and the query section. The query section is highlighted with a red box, showing the query for mit.edu, type NS, class IN.

->Type은 NS인 것을 확인할 수 있고 answers는 포함하지 않다는 것을 알 수 있습니다.

18. Examine the DNS response message. What MIT nameservers does the response message provide? Does this response message also provide the IP addresses of the MIT namesers?

```
Answers  
> mit.edu: type NS, class IN, ns usw2.akam.net  
> mit.edu: type NS, class IN, ns eur5.akam.net  
> mit.edu: type NS, class IN, ns use2.akam.net  
> mit.edu: type NS, class IN, ns asia2.akam.net  
> mit.edu: type NS, class IN, ns use5.akam.net  
> mit.edu: type NS, class IN, ns ns1-37.akam.net  
> mit.edu: type NS, class IN, ns asia1.akam.net  
> mit.edu: type NS, class IN, ns ns1-173.akam.net
```

->Nameservers: ns1-37, sus5, usw2, ns1-137, use2, eur5, asia1, asis2

```
Answers
  mit.edu: type NS, class IN, ns usw2.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 15
    Name Server: usw2.akam.net
  mit.edu: type NS, class IN, ns eur5.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 7
    Name Server: eur5.akam.net
  mit.edu: type NS, class IN, ns use2.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 7
    Name Server: use2.akam.net
  mit.edu: type NS, class IN, ns asia2.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 8
    Name Server: asia2.akam.net
  Name Server: asia2.akam.net
  mit.edu: type NS, class IN, ns use5.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 7
    Name Server: use5.akam.net
  mit.edu: type NS, class IN, ns ns1-37.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 9
    Name Server: ns1-37.akam.net
  mit.edu: type NS, class IN, ns ns1-137.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 8
    Name Server: ns1-137.akam.net
  mit.edu: type NS, class IN, ns ns1-173.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 10
    Name Server: ns1-173.akam.net
```

additional records를 맨 아래에서 확인할 수 있다

```
Additional records
  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
  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
```

19. Provide a screenshot

```
Answers
  mit.edu: type NS, class IN, ns usw2.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 15
    Name Server: usw2.akam.net
  mit.edu: type NS, class IN, ns eur5.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 7
    Name Server: eur5.akam.net
  mit.edu: type NS, class IN, ns use2.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 7
    Name Server: use2.akam.net
  mit.edu: type NS, class IN, ns asia2.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 8
    Name Server: asia2.akam.net
  Name Server: asia2.akam.net
  mit.edu: type NS, class IN, ns use5.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 7
    Name Server: use5.akam.net
  mit.edu: type NS, class IN, ns ns1-37.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 9
    Name Server: ns1-37.akam.net
  mit.edu: type NS, class IN, ns ns1-137.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 8
    Name Server: ns1-137.akam.net
  mit.edu: type NS, class IN, ns ns1-173.akam.net
    Name: mit.edu
    Type: NS (authoritative Name Server) (2)
    Class: IN (0x0001)
    Time to live: 1800 (30 minutes)
    Data length: 10
    Name Server: ns1-173.akam.net
Additional records
  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
  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
```


20. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? If not, what does the IP address correspond to?

```
C:\Users\Wjongs>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.
```

3696	8.731234	192.168.94.201	192.168.94.54	TCP	54	54204 → 53 [ACK] Seq=1 Ack=1 Win=131328 Len=0
3697	8.731266	192.168.94.201	192.168.94.54	TCP	54	54205 → 53 [ACK] Seq=1 Ack=1 Win=131328 Len=0
3698	8.731311	192.168.94.201	192.168.94.54	TCP	56	54205 → 53 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=2 [TCP seg
3699	8.731351	192.168.94.201	192.168.94.54	DNS	86	Standard query 0xb032 AAAA www.aiit.or.kr
3700	8.731422	192.168.94.201	192.168.94.54	TCP	56	54204 → 53 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=2 [TCP seg
3701	8.731445	192.168.94.201	192.168.94.54	DNS	86	Standard query 0x8e87 A www.aiit.or.kr
3702	8.742010	192.168.94.54	192.168.94.201	TCP	54	53 → 54205 [ACK] Seq=1 Ack=3 Win=65536 Len=0
3703	8.742010	192.168.94.54	192.168.94.201	TCP	54	53 → 54205 [ACK] Seq=1 Ack=35 Win=65536 Len=0
3704	8.742010	192.168.94.54	192.168.94.201	TCP	54	53 → 54204 [ACK] Seq=1 Ack=3 Win=65536 Len=0
3705	8.742010	192.168.94.54	192.168.94.201	TCP	54	53 → 54204 [ACK] Seq=1 Ack=35 Win=65536 Len=0

> [2 Reassembled TCP Segments (34 bytes): #3700(2), #3701(32)]

▼ Domain Name System (query)

Length: 32

Transaction ID: 0x8e87

> Flags: 0x0100 Standard query

Questions: 1

Answer RRs: 0

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)

[Response In: 3750]

->DNS query message는 192.168.94.54로 전송되며 저의 local DNS server ip address와 일치하는 것을 알 수 있습니다.

무선 LAN 어댑터 Wi-Fi:

```

연결별 DNS 접미사. . . . . :
설명. . . . . : Intel(R) Wi-Fi 6E AX211 160MHz
물리적 주소. . . . . : BC-6E-E2-53-7D-ED
DHCP 사용. . . . . : 예
자동 구성 사용. . . . . : 예
IPv6 주소. . . . . : 2001:e60:9363:944:63b2:8475:53e:b32f(기본 설정)
임시 IPv6 주소. . . . . : 2001:e60:9363:944:c826:c027:4c24:a64f(기본 설정)
링크-로컬 IPv6 주소. . . . . : fe80::88bf:3ec9:b7ff:9f7a%4(기본 설정)
IPv4 주소. . . . . : 192.168.94.201(기본 설정)
서브넷 마스크. . . . . : 255.255.255.0
임대 시작 날짜. . . . . : 2023년 4월 8일 토요일 오전 3:56:46
임대 만료 날짜. . . . . : 2023년 4월 8일 토요일 오전 6:27:14
기본 게이트웨이. . . . . : fe80::bc31:3cff:feb2:cd15%4
                               192.168.94.54
DHCP 서버. . . . . : 192.168.94.54
DHCPv6 IAID. . . . . : 62680802
DHCPv6 클라이언트 DUID. . . . : 00-01-00-01-2A-5A-55-0D-00-E0-4C-71-FC-52
DNS 서버. . . . . : 192.168.94.54
Tcpip를 통한 NetBIOS. . . . . : 사용
  
```

21. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?

100 3.500748	192.168.94.201	192.168.94.54	DNS	86 Standard query 0xa450 AAAA www.aiit.or.kr	
101 3.500765	192.168.94.201	192.168.94.54	TCP	56 58482 → 53 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=2 [TCP segment of a reassembled PDU]	
102 3.500774	192.168.94.201	192.168.94.54	DNS	86 Standard query 0xd878 A www.aiit.or.kr	
103 3.500793	192.168.94.201	192.168.94.54	TCP	56 58481 → 53 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=2 [TCP segment of a reassembled PDU]	
104 3.500802	192.168.94.201	192.168.94.54	DNS	86 Standard query 0x5d06 AAAA www.aiit.or.kr	
105 3.500818	192.168.94.201	192.168.94.54	TCP	56 58484 → 53 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=2 [TCP segment of a reassembled PDU]	
106 3.500827	192.168.94.201	192.168.94.54	DNS	86 Standard query 0x632c HTTPS www.aiit.or.kr	
107 3.509419	192.168.94.54	192.168.94.201	TCP	54 53 → 58480 [ACK] Seq=1 Ack=3 Win=65536 Len=0	
108 3.509666	192.168.94.54	192.168.94.201	TCP	54 53 → 58480 [ACK] Seq=1 Ack=35 Win=65536 Len=0	
109 3.509666	192.168.94.54	192.168.94.201	TCP	54 53 → 58483 [ACK] Seq=1 Ack=3 Win=65536 Len=0	
110 3.509666	192.168.94.54	192.168.94.201	TCP	54 53 → 58483 [ACK] Seq=1 Ack=35 Win=65536 Len=0	
111 3.509666	192.168.94.54	192.168.94.201	TCP	54 53 → 58482 [ACK] Seq=1 Ack=3 Win=65536 Len=0	

Frame 100: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{0E4B929F-4929-4FD5-9866-6507B5826310}, id 0000	be 31 3c b2 cd 15 bc
Ethernet II, Src: IntelCor_53:7d:ed (bc:6e:e2:53:7d:ed), Dst: he:31:3c:b2:cd:15 (he:31:3c:b2:cd:15)	0010 00 48 f3 28 40 00 80
Internet Protocol Version 4, Src: 192.168.94.201, Dst: 192.168.94.54	0020 5e 36 e4 73 00 35 f6
Transmission Control Protocol, Src Port: 58483, Dst Port: 53, Seq: 3, Ack: 1, Len: 32	0030 02 01 3e 8b 00 00 a4
[2 Reassembled TCP Segments (34 bytes): #99(2), #100(32)]	0040 00 00 03 77 77 77 04
Domain Name System (query)	0050 72 00 00 1c 00 01
Length: 32	
Transaction ID: 0xa450	
Flags: 0x0100 Standard query	
Questions: 1	
Answer RRs: 0	
Authority RRs: 0	
Additional RRs: 0	
Queries	
[Response In: 126]	

->type은 AAAA type이며 answers는 포함하지 않는 것을 알 수 있습니다.

22. Examine the DNS response message. How many "answers" are provided? What does each of these answers contain? 23. Provide a screenshot.

100	3.500748	192.168.94.201	192.168.94.54	DNS	86 Standard query 0xa450 AAAA www.aiit.or.kr
101	3.500765	192.168.94.201	192.168.94.54	TCP	56 58482 → 53 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=2 [TCP segment of a stream ...]
102	3.500774	192.168.94.201	192.168.94.54	DNS	86 Standard query 0xd878 A www.aiit.or.kr
103	3.500793	192.168.94.201	192.168.94.54	TCP	56 58481 → 53 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=2 [TCP segment of a stream ...]
104	3.500802	192.168.94.201	192.168.94.54	DNS	86 Standard query 0x5d06 AAAA www.aiit.or.kr
105	3.500818	192.168.94.201	192.168.94.54	TCP	56 58484 → 53 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=2 [TCP segment of a stream ...]
106	3.500827	192.168.94.201	192.168.94.54	DNS	86 Standard query 0x632c HTTPS www.aiit.or.kr
107	3.509419	192.168.94.54	192.168.94.201	TCP	54 53 → 58480 [ACK] Seq=1 Ack=3 Win=65536 Len=0
108	3.509666	192.168.94.54	192.168.94.201	TCP	54 53 → 58480 [ACK] Seq=1 Ack=35 Win=65536 Len=0
109	3.509666	192.168.94.54	192.168.94.201	TCP	54 53 → 58483 [ACK] Seq=1 Ack=3 Win=65536 Len=0
110	3.509666	192.168.94.54	192.168.94.201	TCP	54 53 → 58483 [ACK] Seq=1 Ack=35 Win=65536 Len=0
111	3.509666	192.168.94.54	192.168.94.201	TCP	54 53 → 58482 [ACK] Seq=1 Ack=3 Win=65536 Len=0

Frame 100: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{0E4B929F-4929-4FD5-9866-6507B5826310}, id 0
 Ethernet II, Src: IntelCor_53:7d:ed (bc:6e:e2:53:7d:ed), Dst: be:31:3c:b2:cd:15 (be:31:3c:b2:cd:15)

Internet Protocol Version 4, Src: 192.168.94.201, Dst: 192.168.94.54

Transmission Control Protocol, Src Port: 58483, Dst Port: 53, Seq: 3, Ack: 1, Len: 32

2 Reassembled TCP Segments (34 bytes): #99(2), #100(32)]

Domain Name System (query)

Length: 32
 Transaction ID: 0xa450
 > Flags: 0x0100 Standard query
 Questions: 1
 Answer RRs: 0
 Authority RRs: 0
 Additional RRs: 0
 > Queries
[\[Response In: 126\]](#)

-> DNS response message는 없으며 answers는 포함하지 않습니다.

결론 및 고찰

이번 HW1과제에서는 컴퓨터 네트워크 시간에 배운 패킷에 대해서 자세히 알아보기 위해 wireshark를 이용하여 패킷에 대해 분석을 진행하였습니다. 이번 과제를 마무리하였다고 하여 패킷 분석을 완벽하게 할 수 있는 것은 아니지만 조금은 wireshark tool에 대해 익숙해질 수 있었습니다. host가 request를 server에 보내고 server가 host에게 respond를 보내는 과정을 익히고 이 과정에서 캐시에 정보가 저장되어 다음에 host가 데이터를 request할 때 캐시에 이러한 정보가 있다면 굳이 server까지 않고 캐시에서 정보를 가져온다는 점을 알게 되었습니다. 이러한 이유 때문에 과제를 진행하면서 캐시를 계속 삭제해주어 과제의 HTTP부분에서 원하는 값을 얻을 수 있었습니다. 그리고, cmd창에서 ipconfig명령어를 통해 host의 다양한 정보 또한 알 수 있게 되어 신기하기도 했습니다. 처음에는 wireshark tool이 익숙하지 않아서 과제를 진행하는데 어려움이 많았지만 사용법을 익힌 후에는 수월하게 진행할 수 있었습니다.

Reference

Wireshark 강의자료 pdf