컴퓨터네트워크 프로젝트 과제

미디어학과 201821133 한규정

*2장 숙제에서 사용하였던 웹사이트가 https인 관계로 국내 http 웹사이트(외교부: http://www.mofa.go.kr)를 선택하여 사용하였습니다.

1.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	121.136.235.9	168.126.63.1	DNS	71	Standard query 0x1ef7 A wpad.kornet
2	0.004262	168.126.63.1	121.136.235.9	DNS	146	Standard query response 0x1ef7 No such name A wpad.kornet SOA a.root-servers.net

: 1번째 패킷은 클라이언트가 DNS를 요청하는 것이고, 2번째는 DNS가 요청에 응답하는 것입니다.

1-1) DNS request 메시지 요약

```
> Frame 1: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_{60175DEC-369C-4D75-9C48-12A25ECC3DA3}, id 0
> Ethernet II, Src: ASRockIn_2a:e8:fd (bc:5f:f4:2a:e8:fd), Dst: Ubiquoss_fa:16:3f (00:07:70:fa:16:3f)
> Internet Protocol Version 4, Src: 121.136.235.9, Dst: 168.126.63.1
> User Datagram Protocol, Src Port: 49964, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x1ef7
  v Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v wpad.kornet: type A, class IN
        Name: wpad.kornet
        [Name Length: 11]
        [Label Count: 2]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 2]
```

: DNS request message의 3번째 줄을 보면 IPv4를 사용하고, 클라이언트(src)가 DNS(dst)에 요청하였음을 알 수 있습니다. 또한 4번째 줄의 User Datagram Protocol, 즉 UDP를 의미하며 전송 계층에서 UDP를 사용하였음을 알 수 있습니다. 바로 우측의 포트 번호를 보면 클라이언트(src)의 경우 임의의 번호(60154)가 할당되었고 목적지에는 DNS(dst)의 포트인 53을 사용한 것을 알 수 있습니다.

지금부터는 Domain Name System 하단부에 대한 설명입니다. Flags에서 Response값이 0이므로 요청 패킷임을 알 수 있습니다. 또한 Recursion desired값이 1이므로 재귀를 사용하였음을 알 수 있습니다. Questions 값이 1이므로 1개의 요청을 포함하는 것을 알 수 있습니다. Queries(Question section)의 Name은 host name을 가리키고, Type은 쿼리의 유형으로 host address인 A를 확인할 수 있습니다. Class는 네트워크의 타입으로 IN은 인터넷을 뜻합니다.

1-2) DNS response 메시지 요약

```
> Frame 2: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits) on interface \Device\NPF_{60175DEC-369C-4D75-9C48-12A25ECC3DA3}, id 0
> Ethernet II, Src: Ubiquoss_fa:16:3f (00:07:70:fa:16:3f), Dst: ASRockIn_2a:e8:fd (bc:5f:f4:2a:e8:fd)
> Internet Protocol Version 4, Src: 168.126.63.1, Dst: 121.136.235.9
> User Datagram Protocol, Src Port: 53, Dst Port: 49964
v Domain Name System (response)
    Transaction ID: 0x1ef7
  v Flags: 0x8183 Standard query response, No such name
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0011 = Reply code: No such name (3)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 0
  v Queries
    v wpad.kornet: type A, class IN
        Name: wpad.kornet
        [Name Length: 11]
        [Label Count: 2]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  > Authoritative nameservers
    [Request In: 1]
    [Time: 0.004262000 seconds]
```

: DNS request 메시지와 반대로 DNS response 메세지에서는 서버가 클라이언트에게 응답을 보냅니다. 3번째 줄을 보면 바뀐 IP address를 통해 이를 알 수 있으며 마찬가지로 IPv4를 사용하고, UDP를 사용함을 알 수 있습니다. 포트 번호를 보면 DNS의 포트넘버인 53인 source 포트이고 클라이언트 포트넘버(49964)가 목적지(dst)로 나타납니다. Flags에서 Authoritative가 0이므로 공식 서버에서의 응답이 아님을 알 수 있고, Recursion 부분에서 응답 시 재귀가 가능함을 알 수 있습니다. Queries의 Name, Type, Class는 Request와 동일합니다.

2.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	121.136.235.9	116.67.79.26	TCP	66	1959 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.031866	116.67.79.26	121.136.235.9	TCP	62	80 → 1959 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 WS=1024
3	0.031942	121.136.235.9	116.67.79.26	TCP	54	1959 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0

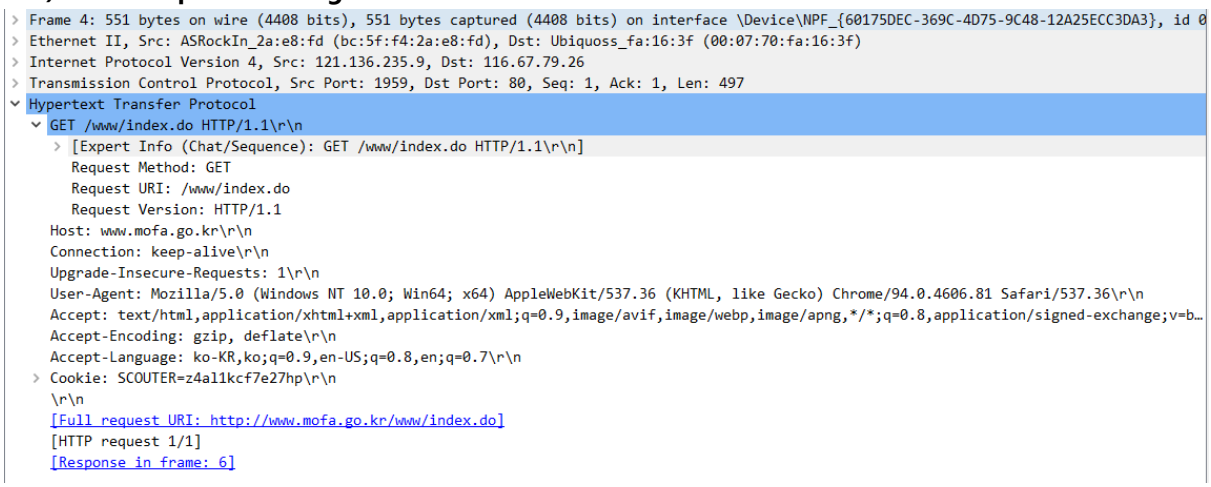
: 가장 처음 TCP가 연결되는 3 Way-handshake 과정입니다. 1번 패킷에서 클라이언트에서 서버로(1599->80) TCP 연결을 요청하고 연결 준비된 상태를 알립니다(SYN). 2번 패킷에서 서버에서 클라이언트로(80->1599) 연결 요청에 응답하는 ACK을 보내고, 연결이 준비된 상태를 알리는 SYN을 전송합니다. 3번 패킷에서는 클라이언트가 서버의 상태 수신이 완료되었음을 응답하는 ACK를 보냅니다. 이러한 과정이 정상 완료되었으므로 데이터 전송이 가능합니다.

3.

4	0.032160	121.136.235.9	116.67.79.26	HTTP	551	GET /www/index.do HTTP/1.1
5	0.043023	116.67.79.26	121.136.235.9	TCP	60	80 → 1959 [ACK] Seq=1 Ack=498 Win=16384 Len=0
6	0.044576	116.67.79.26	121.136.235.9	HTTP	555	HTTP/1.1 301 Moved Permanently (text/html)
7	0.091870	121.136.235.9	116.67.79.26	TCP	54	1959 → 80 [ACK] Seq=498 Ack=502 Win=262144 Len=0

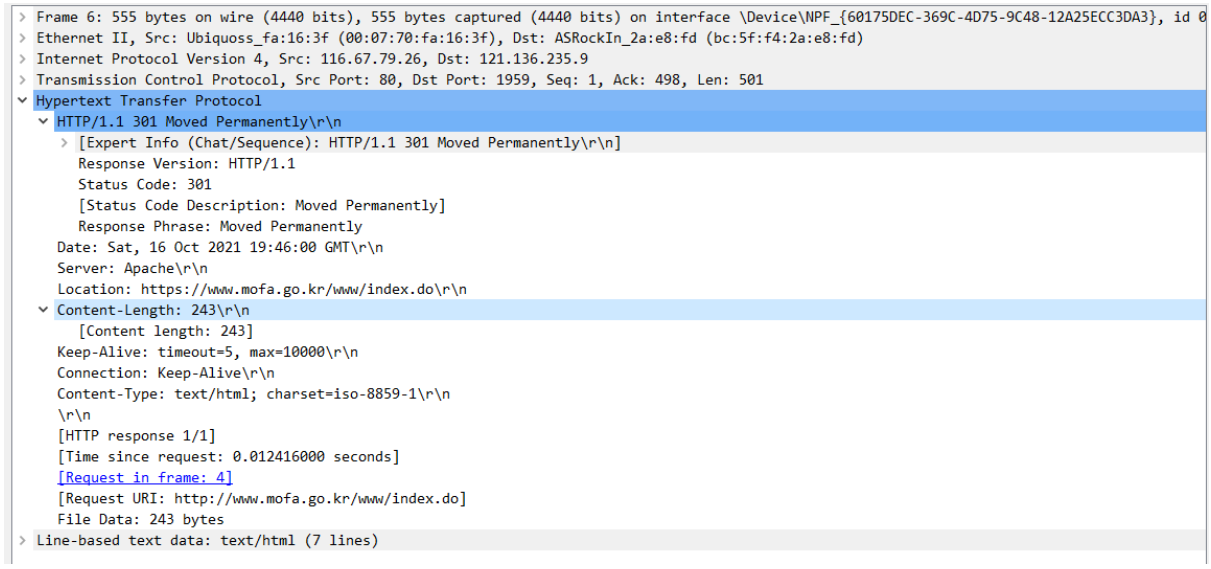
: 가장 먼저 등장하는 HTTP request와 response 메시지입니다. 4번 패킷은 클라이언트가 서버에 웹사이트에 접속했을 때 보여질 html파일 정보를 요청하는 request message입니다. 6번 패킷은 서버가 클라이언트의 요청에 응답하여 파일을 전송하는 response message을 의미하고, 301 Moved Permarently(리다이렉트 상태)는 새로운 주소로 옮겨졌음을 의미합니다. 또한 7번 패킷을 보면 클라이언트가 정상적으로 파일을 전송 받았음을 알 수 있습니다.

3-1) HTTP request message



: Request Line을 보면 Request Method로 GET(페이지 불러오는 기능)을 사용하였고, HTTP version은 1.1임을 알 수 있다. 헤더 라인은 추가 정보를 전달하는데, Host는 호스트 도메인을 표시하고, Connection 부분을 보면 keep-alive 값으로, 서버의 정해진 시간만큼 연결 상태를 유지함을 알 수 있다. Upgrade-Insecure-Request 값이 1이므로, 안전하지 않은 웹사이트를 보안 URL로 교체한 것처럼 처리합니다. User-Agent를 통해 브라우저가 Chrome이고, 운영체제가 Win64인 것을 알 수 있습니다. Accept를 통해 html파일을 허용함을 알 수 있고, Accept-Language는 언어에 대한 정보(여기서는 KR(한국어))를, Accept-Encoding은 브라우저가 받아들이는 압축 알고리즘이 전송됨을 나타낸다. 상태가 없는 특성을 가지는 HTTP이므로 Cookie에는 정보가 없음을 알 수 있다.

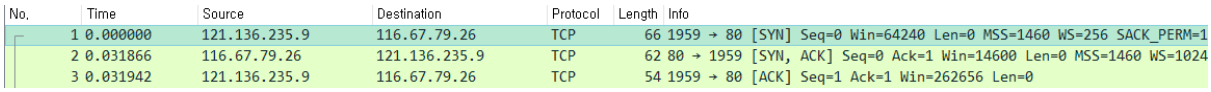
3-2) HTTP response message



: Status Code가 302(Redirection)에 해당하고, HTTP 1.1을 사용함을 알 수 있습니다. 헤더 라인의 Date를 보면 응답 날짜가 2021/10/16이고 응답 시간이 19:46:00임을 알 수 있습니다. Server는 Apache를 이용함을 알 수 있고, 302 리다이렉트 리소스이므로 Location이 제공되어 이동하는 페이지를 알 수 있습니다. Content-Length를 통해 전달 개체의 바이트 길이가 243임을 알 수 있고, Content-type은 text/html임을 알 수 있습니다.

4.

: HTTP 이전과 이후에서 데이터 전송이 생깁니다.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	121.136.235.9	116.67.79.26	TCP	66	1959 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.031866	116.67.79.26	121.136.235.9	TCP	62	80 → 1959 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 WS=1024
3	0.031942	121.136.235.9	116.67.79.26	TCP	54	1959 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0

위는 HTTP 이전의 TCP connection setup 과정입니다. (3-way handshaking)

1) 클라이언트 -> 서버 요청

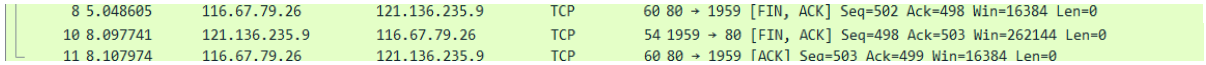
- sequence number가 0입니다.

2) 서버 -> 클라이언트 응답 + 상태 전송

- 1)의 sequence number에 1을 더한 1을 Ack로 갖고, sequence number는 0입니다.

3) 클라이언트 -> 서버 확인

- 2)의 sequence number에 1을 더한 1을 Ack로 갖고, sequence number는 이전 Ack인 1을 갖습니다.



8	5.048605	116.67.79.26	121.136.235.9	TCP	60	80 → 1959 [FIN, ACK] Seq=502 Ack=498 Win=16384 Len=0
10	8.097741	121.136.235.9	116.67.79.26	TCP	54	1959 → 80 [FIN, ACK] Seq=498 Ack=503 Win=262144 Len=0
11	8.107974	116.67.79.26	121.136.235.9	TCP	60	80 → 1959 [ACK] Seq=503 Ack=499 Win=16384 Len=0

위는 HTTP 이후의 TCP connection finish 과정입니다. (4-way handshaking)

1) 클라이언트 -> 서버 [FIN, ACK] 연결종료 요청

- sequence number로 502를 갖고 Ack는 498을 갖습니다.

2) 서버 -> 클라이언트 [FIN, ACK] 응답

- 1)의 sequence number에 1을 더한 503을 Ack로 갖고, sequence number는 1)의 Ack인 498입니다.

3) 클라이언트 -> 서버 [ACK] 확인

- 2)의 sequence number에 1을 더한 499를 Ack로 갖고, sequence number는 2)의 Ack인 503입니다.

5.

: HTTP 1.0은 Non-persistent Connection으로 매번 연결을 끊고 새로 연결하는 방식이고, HTTP 1.1 부터는 Keep-Alive 기능이 존재하는 Persistent Connection 방식입니다. 본 프로젝트에서 요약한 HTTP message에 따르면 HTTP 1.1을 사용하므로 Persistent connection입니다.