Chapter 05

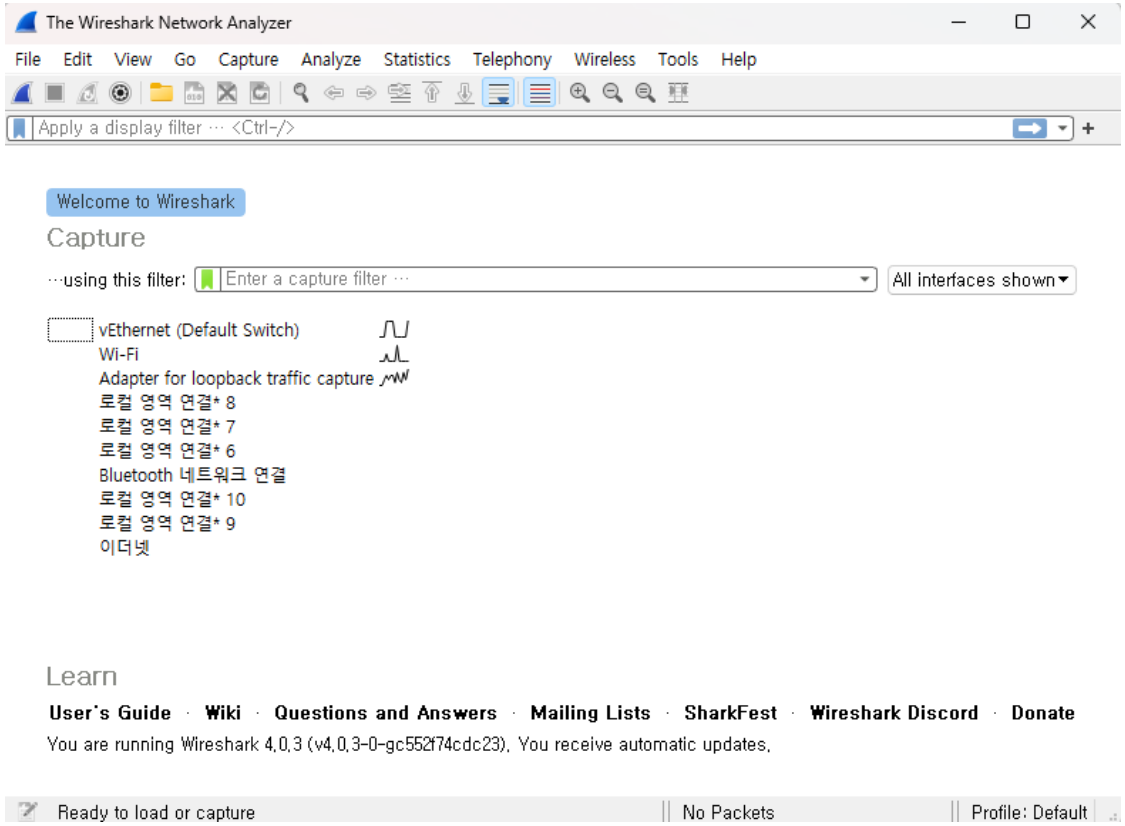
패킷 헤더 확인과 덤프 분석

1. 패킷 헤더 확인

홈페이지 접속 패킷 캡처

홈페이지 패킷을 캡처하는 순서는 다음과 같다.

- 화면에 있는 와이어샤크의 실행 아이콘을 더블클릭한다.
- 그러면 와이어샤크가 실행되며 시작 화면이 나타난다.

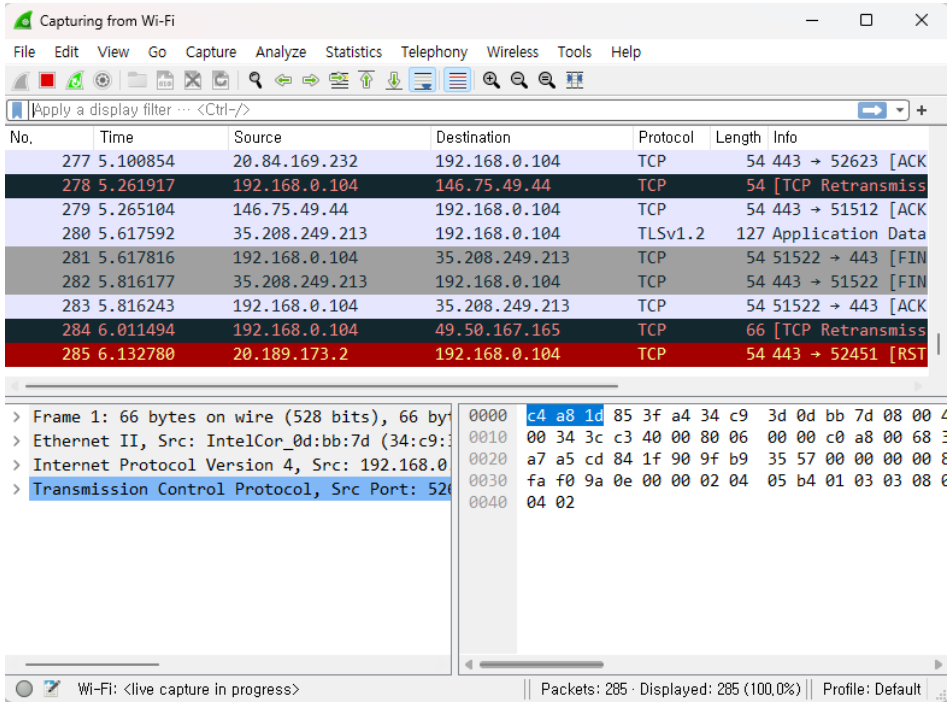


1. 패킷 헤더 확인

홈페이지 접속 패킷 캡처

홈페이지 패킷을 캡처하는 순서는 다음과 같다.

- 시작 화면에서 패킷을 캡처하고 있는 "이더넷"을 더블클릭한다.
- 그러면 패킷 캡처 메인 화면이 나타난다.
- 패킷이 많이 캡처되는 경우에는 인터페이스를 선택하고 캡처 필터를 설정하면 좋다.
- 예를 들어 원하는 홈페이지를 접속할 때에 ARP, DNS, TCP 패킷을 캡처하고 싶은 경우에는 [arp or host DNS 서버의 IP 주소 or host 웹 서버의 IP 주소]와 같이 지정하면 된다.

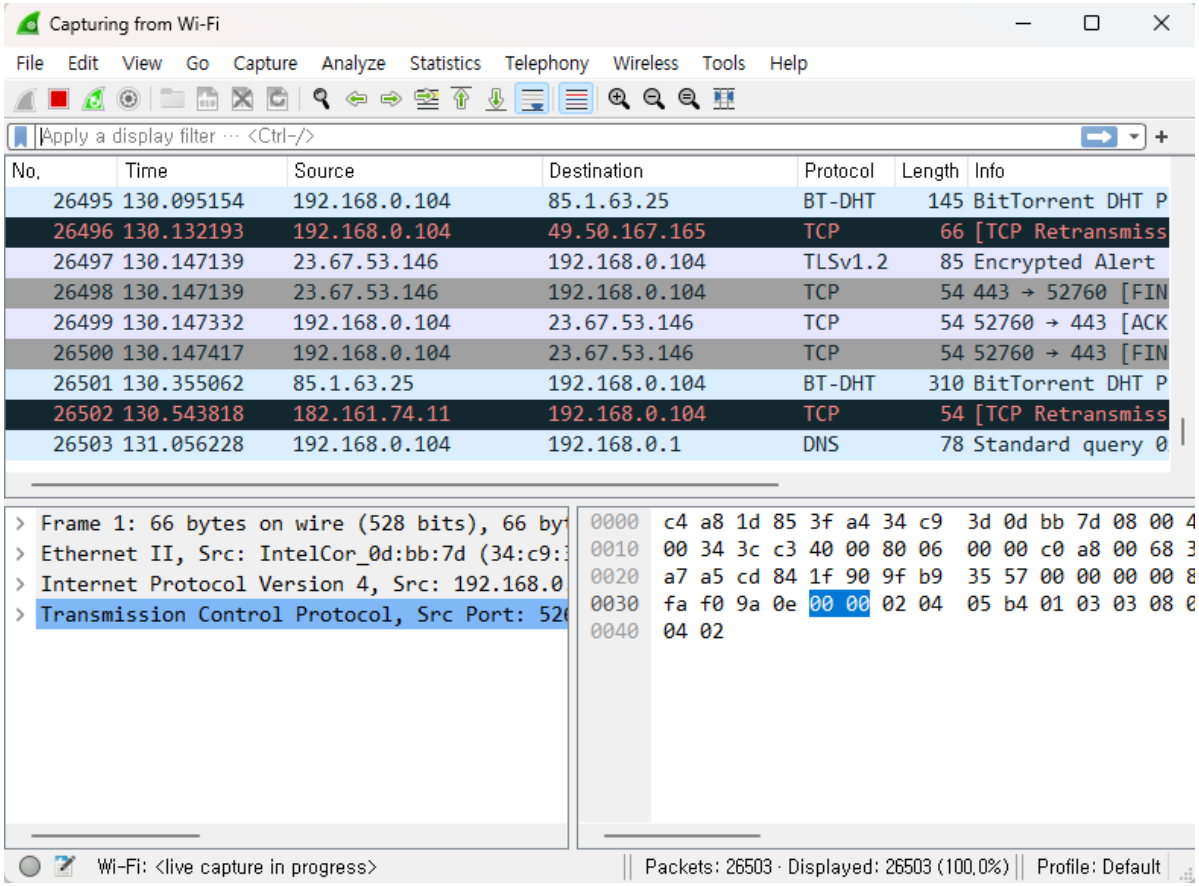


1. 패킷 헤더 확인

홈페이지 접속 패킷 캡처

홈페이지 패킷을 캡처하는 순서는 다음과 같다.

- 캡처된 패킷이 패킷 목록 정보에 차례대로 나타나고 있음을 확인한다.
- 이제 웹 브라우저를 실행하여 원하는 홈페이지를 접속한다. (http://kostat.go.kr)



1. 패킷 헤더 확인

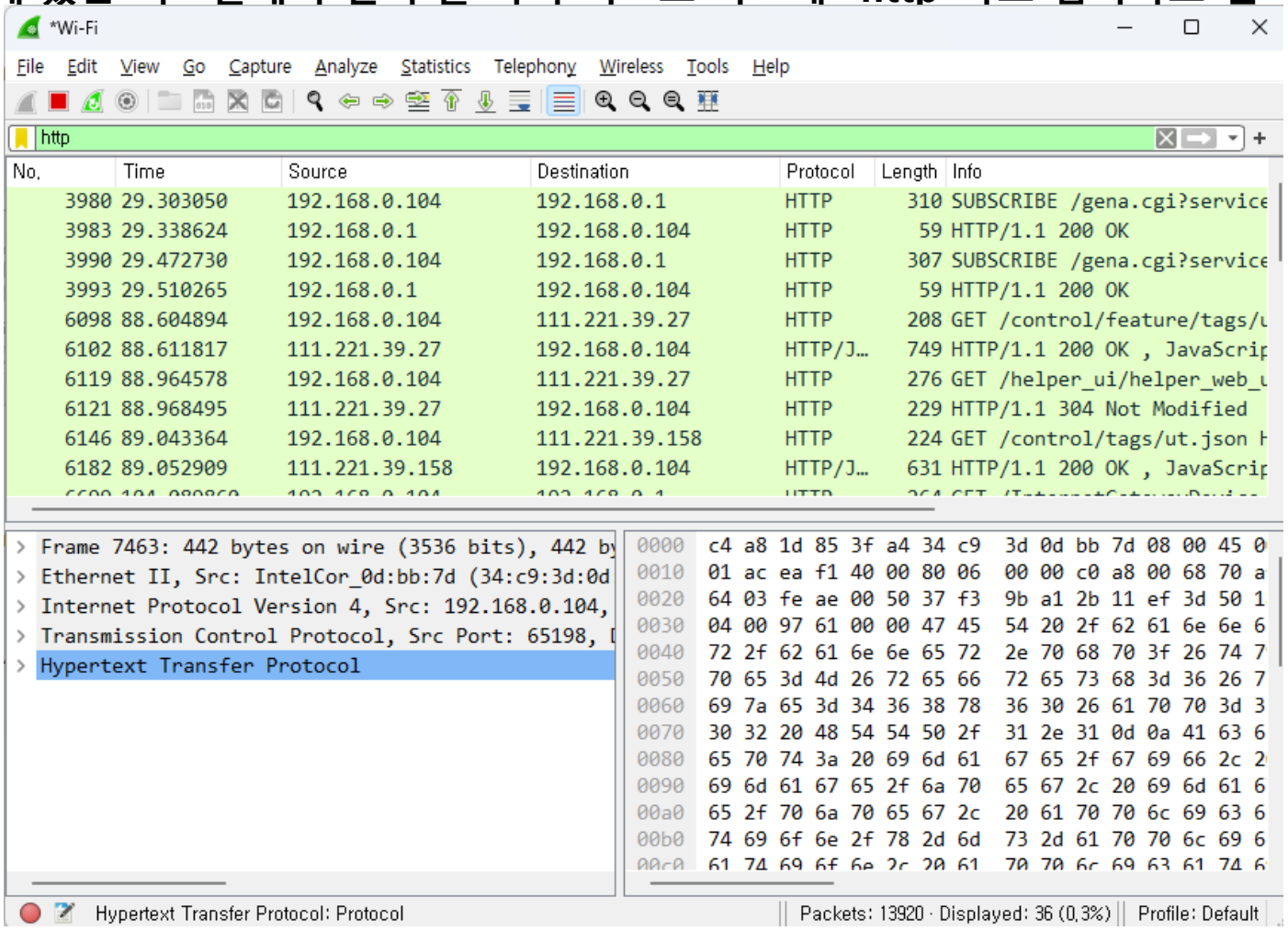
홈페이지 접속 패킷 캡처

- **홈페이지 패킷을 캡처하는 순서는 다음과 같다.**
  - 툴 바의 [STOP] 버튼을 클릭한다.
  - 그러면 패킷 캡처가 종료된다.

- **이제 패킷 캡처 결과를 확인해 보자.**
- **패킷 캡처를 하고 나면 필요치 않은 패킷이 많이 포함되어 있어서 패킷을 바로 해석하기가 쉽지 않다.**
- **따라서 디스플레이 필터를 사용하여 패킷을 제한하면 좋다.**

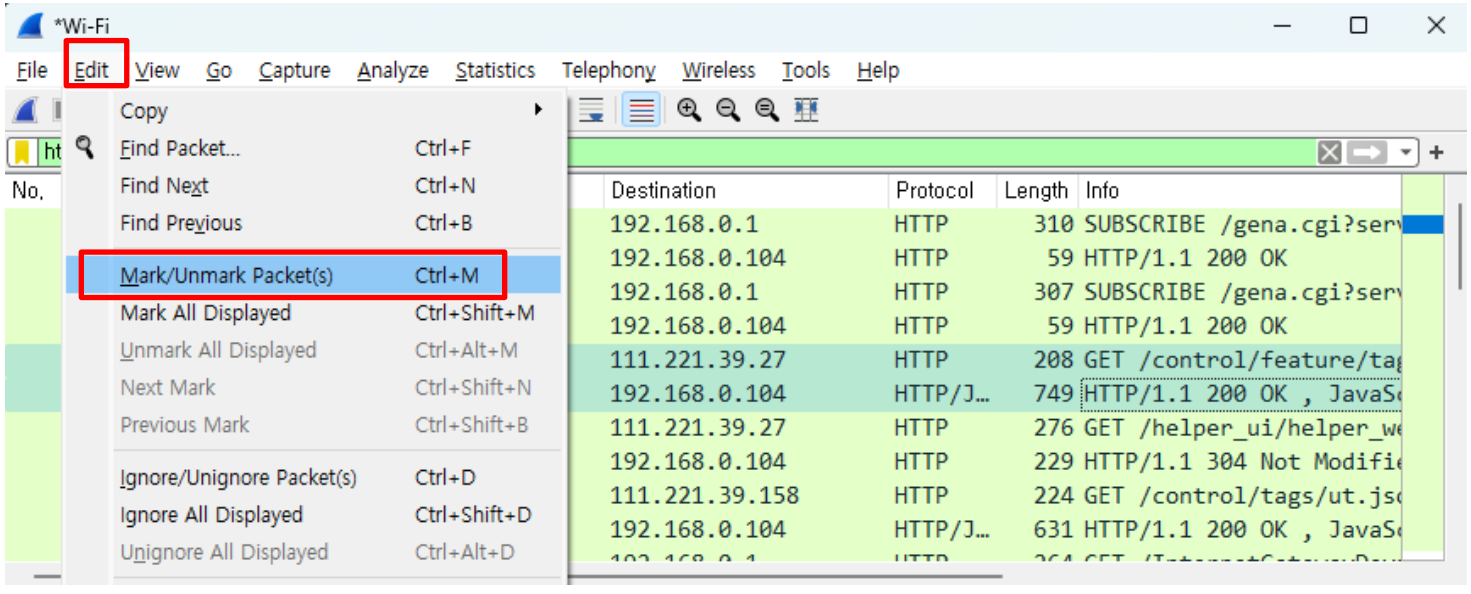
1. 패킷 헤더 확인

- 홈페이지 접속 패킷 캡처
  - 화면 맨 위에 있는 디스플레이 필터 툴 바의 텍스트 박스에 “http"라고 입력하고 엔터키를 누른다.



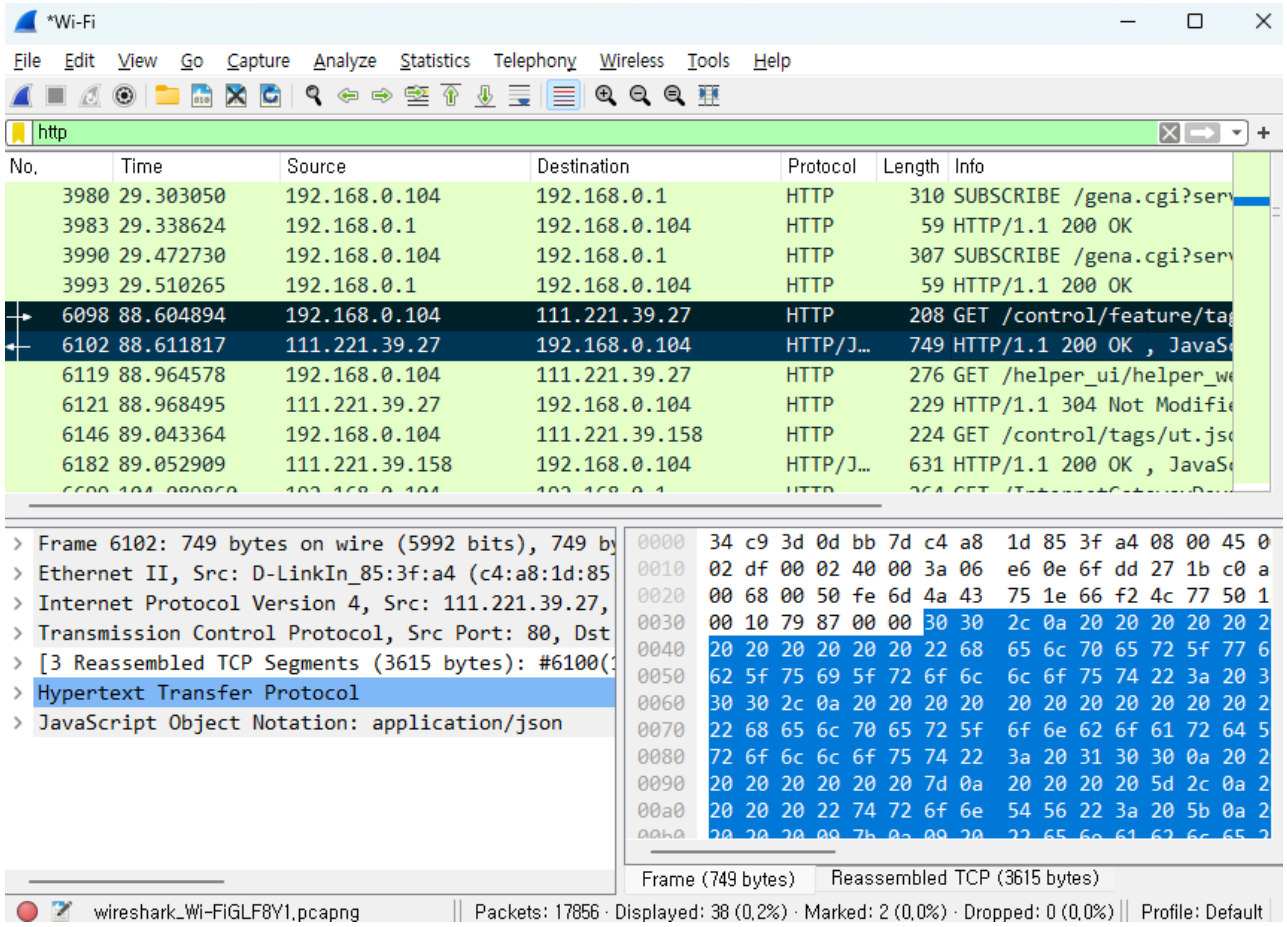
1. 패킷 헤더 확인

- 홈페이지 접속 패킷 캡처
  - 홈페이지의 내용에 따라 다르지만 패킷 목록 정보에는 여러 개의 http 패킷이 나타날 것이다.
  - 여기서 패킷 목록 정보의 Info 열에 [GET /control/feature/tags/ut.json HTTP/1.1], [HTTP/1.1 200 OK, JavaScript Object Notification (application/json)]라고 나타나 있는 패킷을 찾는다.
  - 먼저 [GET /control/feature/tags/ut.json HTTP/1.1] 패킷을 선택하고 메뉴 바에서 [Edit -> Mark/Unmark Packet]을 선택한다.
  - 마찬가지로 [HTTP/1.1 200 OK, JavaScript Object Notification (application/json)] 에 대해서도 동일하게 수행한다.



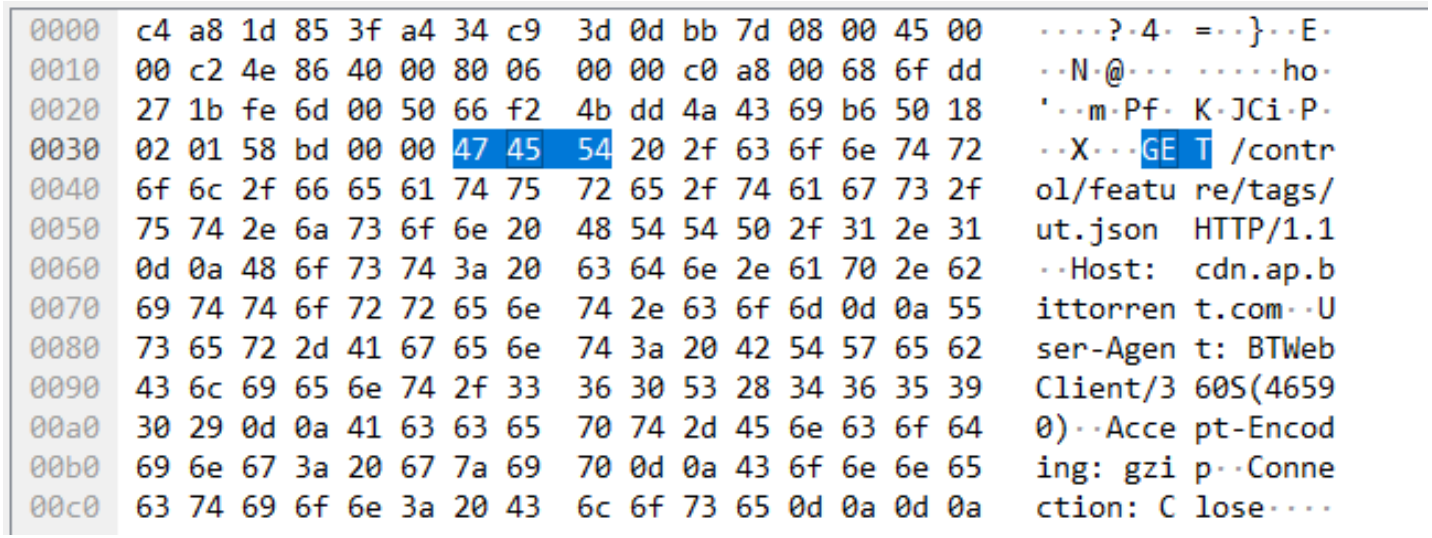
1. 패킷 헤더 확인

- 홈페이지 접속 패킷 캡처
  - 그러면 이 패킷이 검은색으로 반전되어 나타난다.
  - 이것이 앞에서 설명한 [마크(mark)]라는 기능이며 특정 패킷을 강조해서 보여준다.



1. 패킷 헤더 확인

- 홈페이지 접속 패킷 캡처
  - 앞서 강조 표시한 패킷 가운데 Info 열에 [GET /control/feature/tags/ut.json HTTP/1.1]이라고 되어 있는 패킷을 분석해 보자.



```
0000  c4 a8 1d 85 3f a4 34 c9  3d 0d bb 7d 08 00 45 00   ····?·4· =··}··E·
0010  00 c2 4e 86 40 00 80 06  00 00 c0 a8 00 68 6f dd   ··N·@··· ·····ho·
0020  27 1b fe 6d 00 50 66 f2  4b dd 4a 43 69 b6 50 18   '··m·Pf· K·JCi·P·
0030  02 01 58 bd 00 00 47 45  54 20 2f 63 6f 6e 74 72   ··X···GE T /contr
0040  6f 6c 2f 66 65 61 74 75  72 65 2f 74 61 67 73 2f   ol/featu re/tags/
0050  75 74 2e 6a 73 6f 6e 20  48 54 54 50 2f 31 2e 31   ut.json  HTTP/1.1
0060  0d 0a 48 6f 73 74 3a 20  63 64 6e 2e 61 70 2e 62   ··Host:  cdn.ap.b
0070  69 74 74 6f 72 72 65 6e  74 2e 63 6f 6d 0d 0a 55   ittorren t.com··U
0080  73 65 72 2d 41 67 65 6e  74 3a 20 42 54 57 65 62   ser-Agen t: BTWeb
0090  43 6c 69 65 6e 74 2f 33  36 30 53 28 34 36 35 39   Client/3 60S(4659
00a0  30 29 0d 0a 41 63 63 65  70 74 2d 45 6e 63 6f 64   0)··Acce pt-Encod
00b0  69 6e 67 3a 20 67 7a 69  70 0d 0a 43 6f 6e 6e 65   ing: gzi p··Conne
00c0  63 74 69 6f 6e 3a 20 43  6c 6f 73 65 0d 0a 0d 0a   ction: C lose····
```

  - 16진수 2문자는 1바이트(8 비트)가 된다.
  - 패킷 바이트 정보에서는 주소 [0000]을 시작으로 1바이트마다 칸을 띄워 한 줄에 16바이트씩 나타낸다.
  - 이어서 패킷 바이트 정보 우측의 ASCII 정보를 살펴보자.
  - ASCII 문자 코드에 따라 1 바이트 데이터가 1 문자로 나타나 있다.

1. 패킷 헤더 확인

헤더 확인

- 헤더를 확인하기 위해서는 패킷 상세 정보를 확인하면 된다.
- 프로토콜 트리로써 [Frame], [Ethernet II], [Internet Protocol], [Transmission Control Protocol], [Hypertext Transfer Protocol]이 나열되어 있다.
- 이것이 [헤더(Header)]라는 것이다.
- 이 화면을 보면 하나의 패킷에 여러 개의 헤더가 포함되어 있음을 알 수 있다.

```
> Frame 6098: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface \Device\NPF_{8AD3495
> Ethernet II, Src: IntelCor_0d:bb:7d (34:c9:3d:0d:bb:7d), Dst: D-LinkIn_85:3f:a4 (c4:a8:1d:85:3f:a4)
> Internet Protocol Version 4, Src: 192.168.0.104, Dst: 111.221.39.27
> Transmission Control Protocol, Src Port: 65133, Dst Port: 80, Seq: 1, Ack: 1, Len: 154
> Hypertext Transfer Protocol
```

- 와이어샤크는 패킷의 덤프(패킷 바이트 정보)로부터 그 패킷의 [의미]를 해석하고 그것을 패킷 상세 정보에 나타낸다.
- 패킷 형식에 관한 규칙이나 규격은 프로토콜로 정해져 있다.
- 와이어샤크는 이와 같은 인터넷이나 LAN 프로토콜을 기반으로 패킷을 분석하고 그 의미를 패킷 상세 정보에 나타내준다.
- 따라서 패킷 상세 정보에는 패킷 바이트 정보의 내용을 바탕으로 그 패킷의 의미를 해석(디코드)한 내용이 나타난다.

1. 패킷 헤더 확인

- **Frame, Ethernet II, 패킷, 세그먼트 헤더**
  - 패킷 상세 정보에 나타나 있는 내용을 더 자세하게 전개해보면 그림과 같이 나타난다.
  - 여기에서 프레임(Frame). Ethernet II, 패킷(Packet). 세그먼트(Segment)를 명확하게 이해할 필요가 있다.
  - 패킷 상세 정보를 확인하면서 간단한 덤프 해석을 해보도록 하자.

```
Frame 6098: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface \Device\NPF_{8AD3495B-1F2A-48E8-A6B8-359B384A11B1}, id 0
    Section number: 1
  > Interface id: 0 (\Device\NPF_{8AD3495B-1F2A-48E8-A6B8-359B384A11B1})
    Encapsulation type: Ethernet (1)
    Arrival Time: Jan 21, 2023 11:56:33.097537000 대한민국 표준시
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1674269793.097537000 seconds
    [Time delta from previous captured frame: 0.000364000 seconds]
    [Time delta from previous displayed frame: 59.094629000 seconds]
    [Time since reference or first frame: 88.604894000 seconds]
    Frame Number: 6098
    Frame Length: 208 bytes (1664 bits)
    Capture Length: 208 bytes (1664 bits)
    [Frame is marked: True]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
> Ethernet II, Src: IntelCor_0d:bb:7d (34:c9:3d:0d:bb:7d), Dst: D-LinkIn_85:3f:a4 (c4:a8:1d:85:3f:a4)
> Internet Protocol Version 4, Src: 192.168.0.104, Dst: 111.221.39.27
> Transmission Control Protocol, Src Port: 65133, Dst Port: 80, Seq: 1, Ack: 1, Len: 154
Hypertext Transfer Protocol
  > GET /control/feature/tags/ut.json HTTP/1.1\r\n
    Host: cdn.ap.bittorrent.com\r\n
    User-Agent: BTWebClient/360S(46590)\r\n
    Accept-Encoding: gzip\r\n
    Connection: Close\r\n
    \r\n
    [Full request URI: http://cdn.ap.bittorrent.com/control/feature/tags/ut.json]
    [HTTP request 1/1]
    [Response in frame: 6102]
```

1. 패킷 헤더 확인

- **Frame, Ethernet Ⅱ, 패킷, 세그먼트 헤더**
  - 먼저 “Frame" 부분은 와이어샤크에 의해 생성된 메타데이터 부분이다.
  - 실제 프레임은 Ethernet II 부분부터 시작된다.
  - “Frame(프레임)"은 매체 접근 제어(MAC; Media Access Control) 헤더부터 MAC 트레일러까지이다.
  - 장치 간의 모든 통신은 프레임을 사용한다.
  - 트래픽을 분석할 때 이더넷 트레일러는 항상 볼 수 있는 것은 아니다.
  - 어떤 운영체제는 이더넷 네트워크 상에서 트레일러를 캡처하는 것을 지원하지 않는다.
  - 와이어샤크는 실제 프레임에 대한 추가 정보를 제공하기 위하여 “Frame" 섹션을 추가하였다.
  - 패킷 상세 정보에서 맨 위에 있는 부분이 Frame 섹션이다.
  - 이 섹션을 확장하면 시간, 색상, 그리고 와이어샤크가 프레임에 추가한 정보를 볼 수 있다

1. 패킷 헤더 확인

- **Frame, Ethernet Ⅱ, 패킷, 세그먼트 헤더**
  - **“Ethernet Ⅱ"라는 이름이 있는 부분이 실제 프레임이다.**
  - **그림을 보면 실제 프레임의 시작과 종료를 정확하게 알 수 있다.**
  - **프레임은 데이터 링크층에서 전송되는 데이터 단위이다.**

  - **패킷 (Packet)은 프레임에 캡슐화되어 있는 내용이다.**
  - **TCP/IP 통신에서 패킷은 IP 헤더에서 시작해서 프레임의 트레일러 바로 앞까지이다.**
  - **보통 사람들은 네트워크 분석을 “패킷 분석”으로 생각한다.**
  - **이는 분석 작업이 대부분 P 헤더에서 시작하기 때문이다.**
  - **마찬가지로 그림을 보면 패킷의 시작과 끝을 정확하게 알 수 있는데 “Internet Protocol version 4 "부터가 패킷의 시작이다.**
  - **패킷은 네트워크 층에서 전송되는 데이터 단위이다.**

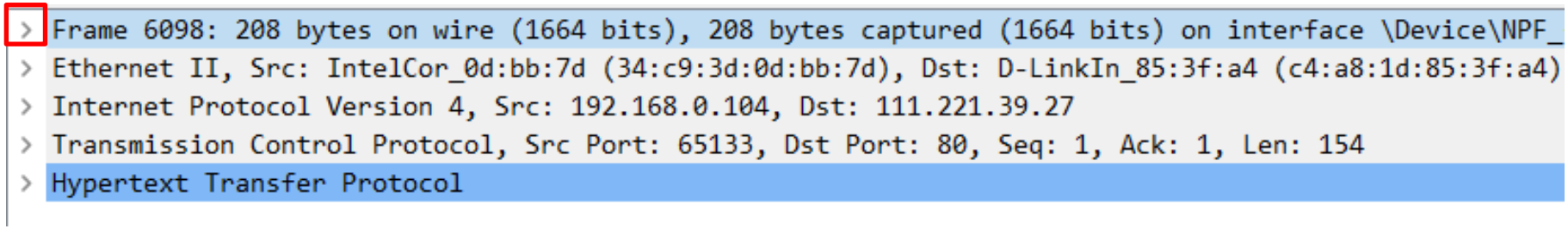
1. 패킷 헤더 확인

- **Frame, Ethernet II, 패킷, 세그먼트 헤더**
  - 세그먼트(segment)는 “Transmission Control Protocol" 헤더에서 시작되는 내용이다.
  - 세그먼트는 HTTP와 같은 응용층 헤더와 데이터를 포함한다.
  - TCP 연결 설정 과정에서 각 TCP 대등(peer)간에 최대 세그먼트 크기 (MSS) 값을 공유한다.
  - 그림을 보면 TCP 세그먼트의 처음과 끝을 정확하게 알 수 있다.
  - 세그먼트는 전송층에서 전송되는 데이터 단위이다.
  - 그 다음은 응용층 프로토콜 중 하나인 HTTP(HypeText Transfer Protocol)의 헤더이다.

1. 패킷 헤더 확인

Frame 헤더

- 패킷 상세 정보에서 헤더 분석을 통하여 간단한 덤프 분석을 해보도록 하자.
- 이제부터 헤더를 하나씩 확인하기로 한다.
- 먼저 [Frame] 부분에 있는 [>]을 클릭하여 전개한다.



1. 패킷 헤더 확인

Frame 헤더

- 그러면 와이어샤크에서 만들어진 메타 정보인 Frame 전체에 대한 설명이 나타난다.

```
Frame 6098: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface \Device\NPF_{8AD3495B-1F2A-48E8-A6B8
  Section number: 1
> Interface id: 0 (\Device\NPF_{8AD3495B-1F2A-48E8-A6B8-359B384A11B1})
  Encapsulation type: Ethernet (1)
  Arrival Time: Jan 21, 2023 11:56:33.097537000 대한민국 표준시
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1674269793.097537000 seconds
  [Time delta from previous captured frame: 0.000364000 seconds]
  [Time delta from previous displayed frame: 59.094629000 seconds]
  [Time since reference or first frame: 88.604894000 seconds]
  Frame Number: 6098
  Frame Length: 208 bytes (1664 bits)
  Capture Length: 208 bytes (1664 bits)
  [Frame is marked: True]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp:http]
  [Coloring Rule Name: HTTP]
  [Coloring Rule String: http || tcp.port == 80 || http2]
```

1. 패킷 헤더 확인

Frame 헤더

그림에 나타난 내용에 대한 설명은 다음과 같다.

- Interface id
  - 패킷을 캡처한 인터페이스 번호.
- Encapsulation type
  - 패킷의 캡술화 유형을 나타낸다.
  - 화면에서는 Ethernet II으로 되어서 Ethernet II의 프레임으로 캡슐화되어 있음을 알 수 있다.
- Arrival Time
  - 프레임을 캡처한 시간.
- Time shift for this Packet
  - 와이어샤크가 생성한 필드
  - 프레임의 디스플레이 시간을 다르게 하는 시간 이동 기능을 이용하는 경우에는 그 시간이 나타난다.
  - 보통 여기는 0초가 된다.
- Epoch Time
  - UNIX 시간 형식의 시리얼 값(1970년 1월 1일 0시 0분 기준으로 한 초 수).

1. 패킷 헤더 확인

Frame 헤더

그림에 나타난 내용에 대한 설명은 다음과 같다.

- Time delta from previous captured frame
  - 와이어샤크가 생성한 필드.
  - 직전에 캡처된 프레임으로부터 간격을 「초」로 나타낸다.
- Time delta from previous displayed frame
  - 와이어샤크가 생성한 필드.
  - 직전에 디스플레이된 프레임으로부터의 간격을 「초」로 나타낸다.
- Time since reference of first frame
  - 와이어샤크가 생성한 필드.
  - 최초의 프레임을 캡처했을 때부터 현재 선택하고 있는 패킷을 캡처했을 때까지의 경과 시간을 「초」로 나타낸다.
- Frame Number
  - 최초에 캡처한 프레임의 번호를 "1"로 하여 그 이후 캡처한 프레임의 순서 번호이다.
- Frame Length
  - 프레임의 길이.

1. 패킷 헤더 확인

Frame 헤더

그림에 나타난 내용에 대한 설명은 다음과 같다.

- Capture Length
  - 캡처했을 때 프레임 길이로서 보통은 [Frame Number]와 [Frame Length]은 같은 값으로 단위는 [바이트]이다.
- Frame is marked
  - 와이어샤크가 생성한 필드.
  - 와이어샤크에 의해 그 프레임이 마크되었는지의 여부를 True(참) 또는 False(거짓)로 나타낸다.
- Frame is ignored
  - 와이어샤크가 생성한 필드.
  - 와이어샤크에 따라 그 프레임이 무시되었는지의 여부를 True(참) 또는 False(거짓)로 나타낸다.
- protocols in frame
  - 와이어샤크가 생성한 필드.
  - 프레임에 포함된 프로토콜이다.

1. 패킷 헤더 확인

Frame 헤더

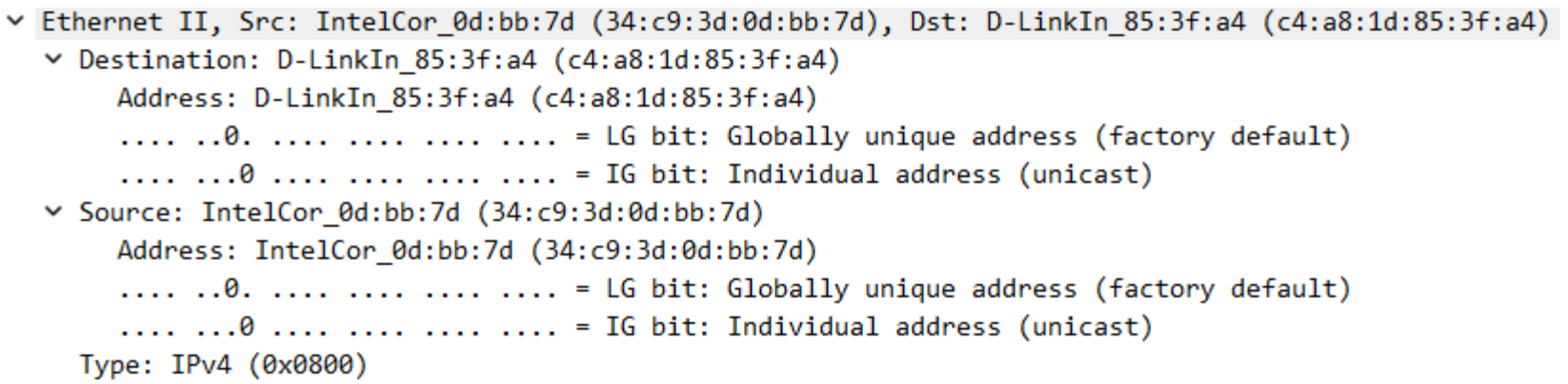
그림에 나타난 내용에 대한 설명은 다음과 같다.

- Coloring Rule Name
  - 와이어샤크가 생성한 필드.
  - 와이어샤크가 색 구별로 사용한 규칙의 이름이다.
- Coloring Rule String
  - 와이어샤크가 생성한 필드.
  - 와이어샤크가 색 구별로 사용한 규칙의 디스플레이 필터이다.

1. 패킷 헤더 확인

Frame 헤더

- 이어서 Ethernet II 부분을 전개해보자.
- Ethernet II 앞에 있는 “>”를 클릭하면 Ethernet II에 관한 설명이 그림과 같이 나타난다.



- 그림에 나타난 내용을 설명하면 다음과 같다.
  - Destination
    - 목적지 LAN 카드(NIC)가 나타난다.
    - LAN 카드 지정에는 LAN 카드에 미리 부여된 MAC 주소를 사용한다.
    - MAC 주소의 길이는 6바이트로서, 앞부분 3바이트는 LAN 카드를 생산한 제조사를 나타내는 번호가 뒷부분 3바이트는 기기를 식별하기 위해 제조사가 부여한 일련번호이다.

1. 패킷 헤더 확인

Frame 헤더

그림에 나타난 내용을 설명하면 다음과 같다.

- Source
  - 발신지 LAN 카드에 있는 MAC 주소가 나타난다.
  - 여기서 Destination Source의 MAC 주소 가운데 7번째 (7 비트째) 위치에 [LG bit]라고 되어 있다.
  - 이것은 원래 부여된 주소인지 아니면 다른 로컬 관리 주소인지 여부를 확인할 수 있다.
  - 여기에는 "0"이 나타나고 [Globally unique address(factory default)]라고 되어 있다.
  - 이것은 LAN 카드의 MAC 주소가 공장 출하 시의 주소 즉 원래 부여된 주소를 사용하고 있음을 나타내고 있다.
  - 또한 2진수로 8 번째 비트 위치에 [IG bit]라고 되어 있다.
  - 이것은 멀티캐스트 통신(특정 그룹에 보내는 1-대-다 통신)인지, 유니캐스트 통신(특정 LAN 카드로 보내는 1-대-1 통신)인지를 나타낸다.
  - 여기에는 "0"이어서 [Individual address(unicast)]라고 되어 있다.
  - 이것은 유니캐스트 통신 임을 알 수 있다.
- Type
  - 이더 타입 (EtherType)이라 불리며, Ethernet II 의 다음에 이어지는 헤더형식을 지정한다.
  - 여기에는 [IPv4(0x0800)]라고 되어있는데 이것은 IP 헤더가 다음에 이어짐을 의미한다.

1. 패킷 헤더 확인

Frame 헤더

- 실제로 Ethernet II에는 와이어샤크가 패킷을 캡처하는 선두 부분에 「프리엠블 필드」가 있다.
- 프리엠블 필드란 패킷의 동기를 맞추기 위한 부분으로 길이는 8바이트로서 그 내용은 [101010...]로 이어진 다음 마지막 1바이트 부분은 [10101011]로 되어있다.
- 이전 페이지 화면 예를 봐도 알 수 있듯이 프리엠블 필드는 패킷을 캡처해도 나타나지 않지만 비동기 방식을 취하는 LAN에서는 타이밍을 맞추는 이 필드가 필수이다.
- 이것은 LAN 표준인 Ethernet II뿐만 아니라 LAN 표준화 기구에서 정해진 IEEE 802.3 형식도 똑같다.
- IEEE 802.3 형식은 맨 앞의 7바이트를 프리엠블, 마지막 1 바이트 부분을 SFD(Start of Frame Delimeter)라고 부르며 구별하고 있다.
- Ethernet II헤더 중에 FCS(Frame Check Sequence)도 패킷 캡처에는 나타나지 않는다.
- FCS는 4바이트로서, 패킷 내용의 오류를 확인한다.
- FCS의 내용을 바탕으로 패킷 내용의 변경 여부를 확인한다.

1. 패킷 헤더 확인

IPv4 헤더

- 이번에는 [Internet Protocol Version 4] 헤더를 전개하기 위해 [>]를 클릭한다.
- [>]를 모두 전개하면 IP 헤더에 관한 설명이 그림과 같이 나타난다.

```
Internet Protocol Version 4, Src: 192.168.0.104, Dst: 111.221.39.158
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
   Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
   Total Length: 210
   Identification: 0x7683 (30339)
   010. .... = Flags: 0x2, Don't fragment
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
   ...0 0000 0000 0000 = Fragment Offset: 0
   Time to Live: 128
   Protocol: TCP (6)
   Header Checksum: 0x0000 [validation disabled]
   [Header checksum status: Unverified]
   Source Address: 192.168.0.104
   Destination Address: 111.221.39.158
```

1. 패킷 헤더 확인

IPv4 헤더

여기서 화면에 나타난 내용을 설명하면 다음과 같다.

- Version
  - IP의 버전을 나타낸다.
  - 화면에는 "4"라고 되어 있다.
  - 이는 IP 버전이 4 임을 알 수 있다.

- Header length
  - IP 헤더의 길이를 나타낸다.
  - 화면에는 [20바이트]로 되어있다.
  - IP 헤더는 가변 길이지만, 인터넷 통신에서 사용되는 IP 헤더는 옵션이 없으면 20바이트가 된다.
  - IP 헤더가 20바이트라고 알아두면 패킷을 분석할 때 편리하다.

1. 패킷 헤더 확인

IPv4 헤더

여기서 화면에 나타난 내용을 설명하면 다음과 같다.

- Differentiated Services Field
  - 패킷의 중요도를 나타낸다.
  - 이 필드를 전개하면 [Differentiated Services Codepoint], [Explicit Congestion Notification]이 나타나는데 여기서는 통신에서 제공되는 서비스 종류와 우선순위를 나타낸다.
  - 화면에는 모두 "0"으로 되어있고 이것은 중요하지 않은 패킷은 폐기하는 특별한 대역 제어나 QoS(서비스 품질 제어)를 하지 않는 이른바 표준 우선순위를 나타내고 있다.
  - 이 [Differentiated Services Field]는 1969년에 인터넷 프로토콜(IP)을 설계할 때는 [TOS(Type Of Service)]라고 불리며 패킷의 우선 제어에 이용되었다.
  - 현재는 [Differentiated Services Field]와 [TOS(Type Of Service)]가 모두 사용되고 있다.

- Total Length
  - 패킷의 길이를 바이트로 나타낸다.

- Identification
  - 패킷의 식별 정보를 나타낸다.

1. 패킷 헤더 확인

IPv4 헤더

여기서 화면에 나타난 내용을 설명하면 다음과 같다.

- Flags
  - 패킷을 단편화하는 데 이용된다.
  - Reserved bit는 앞으로 IP 프로토콜 확장을 위해 예약되어 있는 부분이다.
  - 이 부분은 무시해도 좋다.
  - 이어서 Don't fragment는 패킷의 분할을 금지하기 위한 비트로 [DF 비트]라고 한다.
  - 여기가 "1"인 경우에는 패킷의 분할이 금지된다.
  - "0"인 경우에는 패킷의 분할이 허용된다.
  - 그 아래의 [More fragments]는 분할한 패킷이 이후에 이어짐을 나타내기 위한 비트로 [MF 비트]라고 한다.
  - 여기가 "0"인 경우는 이어지는 패킷이 없음을 나타낸다.
  - "1"인 경우는 분할되어 이어지는 패킷이 있음을 의미한다.

- Fragment offset
  - 분할된 패킷 가운데 현재 패킷의 위치가 [바이트]로 나타난다.
  - 화면에서는 "0"으로 되어있는데 이것은 첫 번째 패킷임을 나타내고 있다.

1. 패킷 헤더 확인

IPv4 헤더

여기서 화면에 나타난 내용을 설명하면 다음과 같다.

- Time to live
  - [생존 시간(TTL)]이라고 하며 패킷의 수명을 나타낸다.
  - 이 생존 시간은 패킷이 라우터를 중계하면서 네트워크에 송신될 때에 값이 하나씩 줄어드는 구조로 되어있다.
  - 그리고 값이 "0"이 되면 패킷은 자동적으로 중계하는 라우터나 계층 3스위치에 의해 폐기된다.
  - TTL은 패킷이 목적지에 순조롭게 전달되지 못하고 네트워크를 계속 순환하는 미아 패킷을 생존 시간에 따라 관리하여 네트워크가 폭주하지 않도록 하고 있다.
  - 최초의 TTL 값은 OS에 의해 설정된다.
  - 참고로 윈도우즈의 표준 초기 TTL 값은 128 이다.

- protocol
  - [프로토콜 필드]라고 하며 IP 이후에 이어지는 헤더 형식을 지정한다.
  - 화면에는 16진수로 "6"이 들어가 있다.
  - 이로써 IP 이후에 TCP 헤더가 이어짐을 알 수 있다.
  - 프로토콜 필드에 들어가는 값은 인터넷(IANA) 에서 정해진 정수이며, 6이라면 TCP이다.

1. 패킷 헤더 확인

IPv4 헤더

여기서 화면에 나타난 내용을 설명하면 다음과 같다.

- Header checksum
  - [헤더 검사합]으로 IP 헤더의 내용을 기본으로 계산된 값과 헤더의 검사합 값을 비교하여 패킷의 IP 헤더가 변경되지 않았는지 확인한다.
  - 두 개의 값이 같으면 IP 헤더는 변경되지 않았고, 반대로 값이 다르면 IP 헤더가 변경되었다는 것을 의미한다.
  - 헤더 검사합을 계산하는 방법으로 CRC-32 라는 방식이 사용되고 있다.
  - Wireshark 4.x에서는 디폴트로 검사하지 않는다.
  - [Header checksum : 검사합 값] 다음에 (validation disabled)라 나타나 있으며 검사합 계산이 무효로 되어있음을 알 수 있다.
  - 또한 이어지는 (Header checksum Status :) 다음에는 [Unverified]라 되어있으므로 검사합 검사를 하고 있지 않음을 알 수 있다(와이어샤크가 생성한 필드).

- Source
  - 발신지 컴퓨터의 IP 주소를 나타낸다.

- Destination
  - 목적지 컴퓨터의 IP 주소를 나타낸다.
  - 경우에 따라 호스트명이 나타난다.

1. 패킷 헤더 확인

TCP 헤더

- [Transmission Control Protocol] 헤더를 전개하기 위해 [>]를 클릭한다.
- [>]를 클릭하면 전개되어 그림과 같이 TCP 헤더에 관한 내용이 나타난다.

```
▼ Transmission Control Protocol, Src Port: 65138, Dst Port: 80, Seq: 1, Ack: 1, Len: 170
    Source Port: 65138
    Destination Port: 80
    [Stream index: 202]
    [Conversation completeness: Complete, WITH_DATA (63)]
    [TCP Segment Len: 170]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 509571949
    [Next Sequence Number: 171    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 448478949
    0101 .... = Header Length: 20 bytes (5)
  ▶ Flags: 0x018 (PSH, ACK)
    Window: 513
    [Calculated window size: 131328]
    [Window size scaling factor: 256]
    Checksum: 0x5950 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  ▶ [Timestamps]
  ▼ [SEQ/ACK analysis]
      [iRTT: 0.004542000 seconds]
      [Bytes in flight: 170]
      [Bytes sent since last PSH flag: 170]
    TCP payload (170 bytes)
```

1. 패킷 헤더 확인

TCP 헤더

그림에 나타난 내용에 대한 설명은 다음과 같다.

- Source port
  - 발신지 프로세스의 포트 번호이다.
  - 프로세스란 프로그램을 구성하는 단위이다.

- Destination port
  - 목적지 프로세스의 포트 번호이다.
  - TCP는 프로세스 지정에 포트 번호를 사용한다.
  - 포트 번호는 클라이언트 측의 프로그램이나 서버 측의 서비스를 지정하기 위해 이용된다.
  - 즉, TCP는 포트 변호에 의해 애플리케이션을 지정한다고 할 수 있다.

- Stream index
  - 와이어샤크에서는 TCP 연결 추적이나 분석을 쉽게 하기 위해 TCP 연결 설정부터 종료까지를 하나의 TCP 스트림으로 보고 패킷에 나타난 순서로 스트림의 번호를 붙여 분석한다.
  - 여기서는 TCP 스트림 인덱스 번호가 "20"으로 되어 있다.

1. 패킷 헤더 확인

TCP 헤더

그림에 나타난 내용에 대한 설명은 다음과 같다.

- TCP Segments Len
  - 와이어샤크가 TCP의 헤더와 페이로드 등에서 계산한 TCP 세그먼트 길이를 바이트 단위로 나타낸다.

- Sequence number
  - [순서 번호]라고 하며 현재 송신하고 있는 TCP 세그먼트의 위치를 숫자로 나타낸다.
  - 이번에는 첫 번째 데이터이므로 화면에는 "1"이 나타나 있다.
  - 순서 번호는 연결 설정 시에 [초기 순서 번호 + 1]로 설정되어 송신 세그먼트의 바이트 수가 더해져 간다.

- Acknowledgement number
  - [확인응답 번호]라고 하며 수신한 TCP 세그먼트의 위치가 번호로 나타난다.
  - 실제로는 "수신한 바이트 수 +1 "이 확인응답 번호가 된다.

- Header length
  - TCP 헤더의 길이가 4바이트(32 비트) 배수로 나타난다.
  - TCP 헤더는 가변 길이인데 IP와 마찬가지로 옵션이 없는 헤더 길이는 20바이트가 된다.

1. 패킷 헤더 확인

TCP 헤더

그림에 나타난 내용에 대한 설명은 다음과 같다.

- Flags
  - 통신을 제어하는 데 이용된다.
  - [>]를 클릭해서 전개하면 3 비트의 [Reserved(예약)], Nonce(비표 값)], [Congestion Window Reduced(CWR)], [ECN-Echo], [Urgent], [Acknowledgement], [Push], [Reset], [Syn], [Fin] 등이 있다.

- Windows
  - [RWIN]이라고도 하며 연속해서 TCP 패킷을 수신하기 위한 수신 버퍼를 나타낸다.

- checksum
  - TCP 헤더와 세그먼트의 내용을 확인한다. Wireshark 4.x는 디폴트로 검사합 확인을 하지 않아서 [Unverified]라고 나타난다.
  - 또한, [checksum Status]에 대해서도 [Unverified]라고 나타난다.

1. 패킷 헤더 확인

TCP 헤더

그림에 나타난 내용에 대한 설명은 다음과 같다.

- Urgent pointer
  - TCP의 URG 플래그과 함께 이용되는 필드로 긴급 데이터의 위치를 나타낸다.
  - 보통은 "0"이 들어간다.

- SEQ/ ACK analysis
  - 와이어샤크가 TCP의 순서 번호, 확인 응답 번호와 플래그의 상태에 따라 계산하여 부가하는 필드이다.
  - 엑스퍼트 기능 등에 의해 와이어샤크가 부가한 헤더는 [ ]으로 나타난다.

- TCP payload
  - TCP 페이로드 부분의 크기를 나타낸다.

- 이상이 TCP 헤더의 주된 내용이다.
- 필요에 따라 이후에 [Options]이 이어진다.
- 단, 일반적인 TCP 패킷에는 [Options]이 붙지 않는다.

1. 패킷 헤더 확인

- HTTP 헤더
  - 이제 HTTP 헤더를 분석해 보자.
  - [HyperText Trasnfer Protocol[을 전개하기 위해 [>]를 클릭한다.
  - HTTP 헤더 해석은 지금까지 살펴본 Ethernet II 나 IP, TCP 분석과는 조금 다르다.
  - 또, HTTP 헤더에 있는 필드의 종류나 수는 웹 브라우저나 os 에 따라 다르다.
  - HTTP 프로토콜은 홈페이지를 캡처하는 순서가 실제로 알파벳을 사용한 명령으로 나타나 있다.
  - 암호화되어 있지 않아서 이것을 [평문(plain text)]이라고도 한다.
  - 패킷 바이트 정보를 살펴보면 [GET /sample.html HTTP...] 등과 같은 알파벳이 나타나 있다.
  - 이와 같은 문자를 사용한 1바이트 단위의 명령이나 응답 부분은 [바이트스트림]이라고 한다.

1. 패킷 헤더 확인

HTTP 헤더

- [>]를 선택하면 [Hyper Transfer Protocol] 헤더에 관한 설명이 그림과 같이 나타난다.

```
Hypertext Transfer Protocol
  GET /control/tags/ut.json HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /control/tags/ut.json HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /control/tags/ut.json
    Request Version: HTTP/1.1
  Host: cdn.ap.bittorrent.com\r\n
  User-Agent: BTWebClient/360S(46590)\r\n
  Accept-Encoding: gzip\r\n
  Accept-Language: ko-KR\r\n
  Connection: Close\r\n
  \r\n
  [Full request URI: http://cdn.ap.bittorrent.com/control/tags/ut.json]
  [HTTP request 1/1]
  [Response in frame: 6182]
```

- 여기에 나타난 내용에 대해 설명한다.
- HTTP 요청 헤더는 [요청 행]이라는 첫 번째 행에 웹 브라우저에서 웹 서버로 보내는 명령이 들어있다.
- 구체적으로는 홈페이지를 살펴보거나 어느 홈페이지를 접속하는 장소 지정이 이루어진다.

1. 패킷 헤더 확인

- HTTP 헤더
  - 요청 행은 [GET /control/tags/ut.json HTTP /1.1₩r₩n]이라고 나타난 부분이다.
  - 이어서 와이어샤크가 HTTP의 수신 내용을 추출하여 [Expert Info (Chat/Sequence) :)라는 생성 헤더가 나타나 있고, 그 내용은 [GET /control/tags/ut.json HTTP /1.1₩r₩n]로 되어있다.
  - 이 부분을 전개하면, 맨 먼저 [Request Method]에서는 홈페이지를 캡처하는 방법을 나타낸다.
  - 여기서는 [GET] Method가 지정되어 있다.
  - 「요청 행」 가운데 [Request URI:]는 취득하고 싶은 홈페이지의 파일 장소를 지정한다.
  - 파일 장소는 단일 자원 식별자(URI; Uniform Resource Identifier)라고 한다.
  - 이번 패킷에는 URI가 [/test/sample.html]로 지정되어 있다.
  - 다음은 「요청 행」의 마지막인 [Request Version :]은 HTTP의 버전을 지정한다.
  - 여기에는 “1.1”로 되어있다.

1. 패킷 헤더 확인

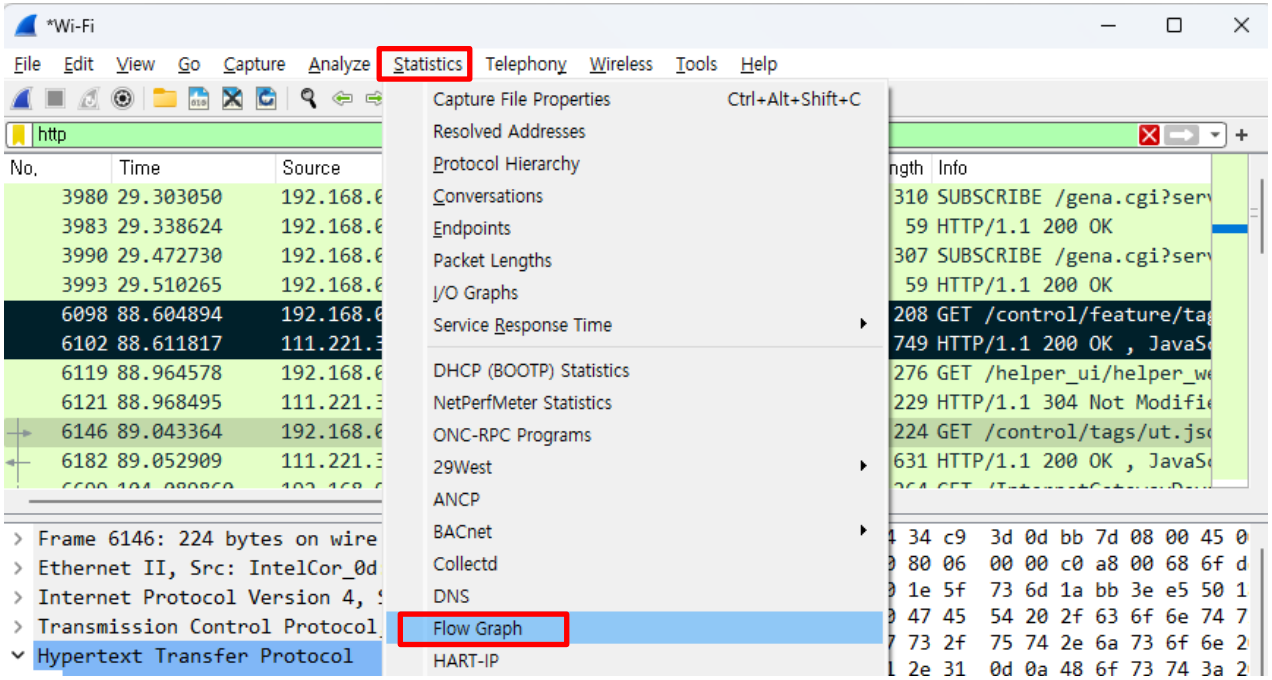
- **HTTP 헤더**
  - 그 다음은 [HTTP 요청 헤더]라는 부분이 이어진다.
  - 각 헤더는 [헤더명: 값₩r₩n]이라는 형식을 취한다.
  - 이 부분은 웹 브라우저가 임의로 부가한 부분이기 때문에 웹 브라우저에 따라 다르게 나타날 수 있다.
  - HTTP 요청 헤더에는 [Host:], [Connection:], [Upgrade-InsecureRequests:], [User-Agent], [Accept:], [Accept-Encoding:], [Accept-Language:], [Cookie:]가 있다.
  - 또한, 와이어샤크가 부가하는 헤더에는 [Full request URI:], [Response in frame], [Next request in frame]이 있다.

2. 덤프 분석

- 덤프 분석 (dump analysis) 이란 네트워크를 통하여 전달되는 패킷을 캡처하여 통신 내용을 분석하는 것을 말한다.
- 덤프 분석의 목적은 통신을 통하여 전송되는 통신 내용을 정확하게 파악하는 데 있다.
- 예를 들어 네트워크에 트러블이 발생했을 때 나타나는 오류 화면이나 로그 분석과 달리 실제로 어떤 통신이 이루어지고 있는지를 확인하려면 반드시 덤프 분석이 필요하다.
- 또 애플리케이션 개발자가 자신이 개발한 통신 프로그램의 동작을 확인하거나 콘텐츠 개발자가 통신 내용을 파악하거나 운영 담당자가 트래픽을 조사할 때는 모두 덤프 분석을 통해서 이루어진다.
- 여기서 먼저 홈페이지에 접속했을 때에 통신 흐름을 따라서 패킷에 포함된 프로토콜 내용에 대해 덤프 분석을 해보자.

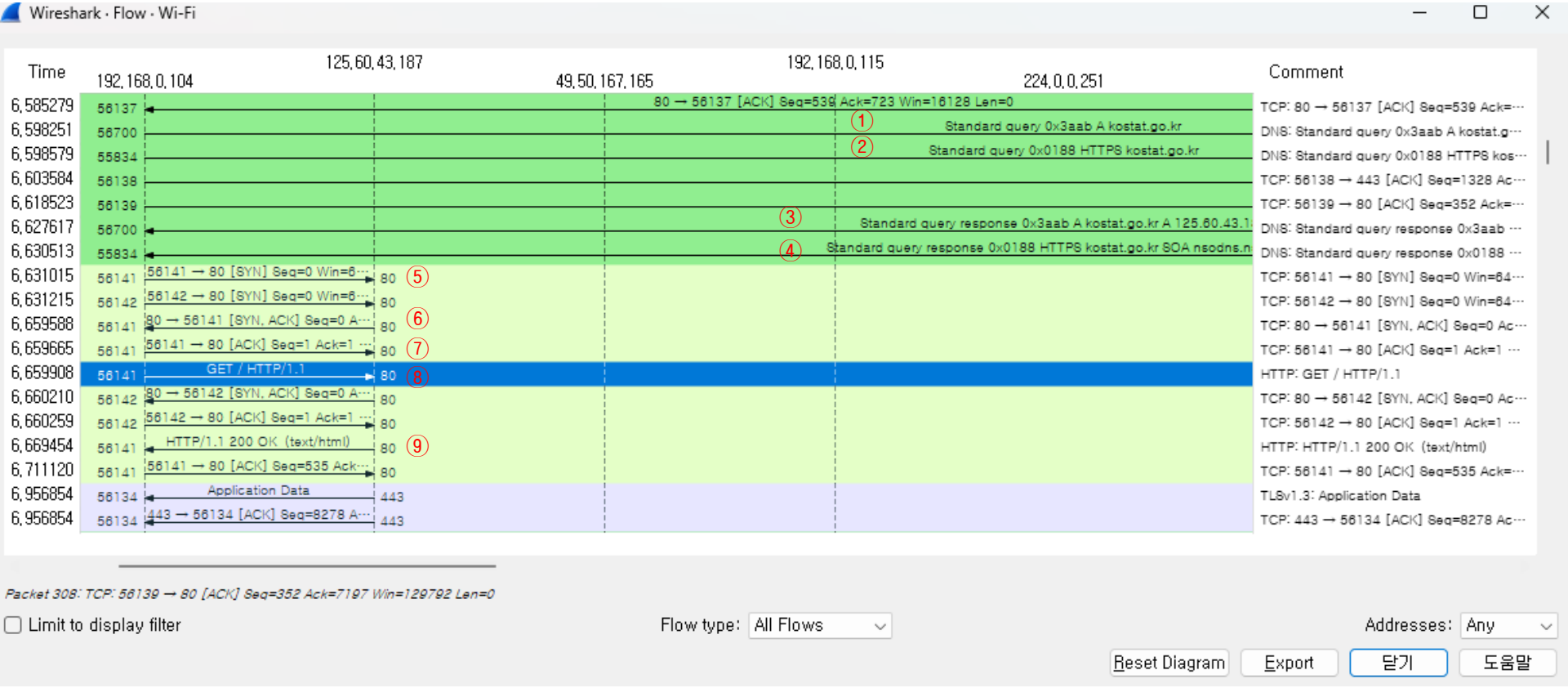
2. 덤프 분석

- 그림에 나타난 메인 화면의 패킷 목록 정보는 홈페이지 (http://kostat.go.kr)에 접속했을 때에 캡처된 패킷 목록들이 차례로 나타난 것이다.
- 여기서 캡처된 패킷을 Time 열을 기준으로 [Flow Graph]로 살펴보자.
- 교환되는 통신 흐름을 [통신 순서]라고 하는데 통신 순서를 추적함으로써 보다 자세하게 덤프 분석을 할 수 있다.
- 이를 위해서는 먼저 메뉴 바에 있는 Statistics에서 Flow Graph 를 선택한다.



2. 덤프 분석

- 그림과 같이 그래프 분석 (Graph Analysis) 화면이 나타난다.



2. 덤프 분석

- 그러면 웹페이지 (http://kostat.go.kr) 에 접속했을 때의 캡처 예의 그래프 분석 화면을 보고 시간에 따라 통신 순서를 정리하면 다음과 같다.
  - ① 클라이언트 PC에서 DNS 서버로 보내는 DNS 질의 요청 메시지와 DNS 서버로부터 클라이언트 PC로 보내는 DNS 질의 응답 메시지 순서 (①~④)
  - ② 클라이언트 PC에서 웹 서버로 연결 설정을 위한 3-방향 확인응답 방식 (⑤~⑦)
  - ③ 클라이언트 PC에서 웹 서버로 HTTP 요청 메시지와 ACK 및 [200 OK] HTTP 응답 메시지(⑧, ⑨)

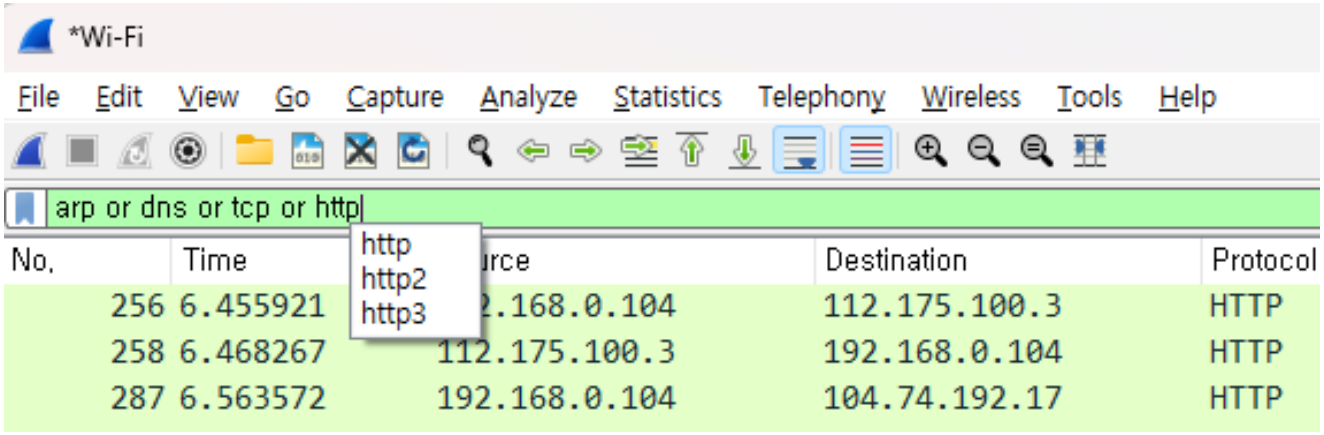
2. 덤프 분석

- 뿐만 아니라 그래프분석 화면인 그림을 살펴보고, 통신 흐름에 대한 구체적인 내용을 요약하면 다음 표와 같다.

프레임	송신자	수신자	프로토콜	개요
①	클라이언트 PC	DNS 서버	DNS	DNS 질의 요청
②	클라이언트 PC	DNS 서버	DNS	HTTPS로 변환
③	DNS 서버	클라이언트 PC	DNS	DNS 질의 응답
④	DNS 서버	클라이언트 PC	DNS	HTTPS로 변환
⑤	클라이언트 PC	웹서버	TCP	3-단계 핸드쉐이크(SYN)
⑥	웹서버	클라이언트 PC	TCP	3-단계 핸드쉐이크(SYN-ACK)
⑦	클라이언트 PC	웹서버	TCP	3-단계 핸드쉐이크(ACK)
⑧	클라이언트 PC	웹서버	HTTP	HTTP 요청 메시지
⑨	웹서버	클라이언트 PC	HTTP	HTTP 응답 메시지

2. 덤프 분석

- 윈도우즈에서 덤프 분석할 때에 특히 주의해야 할 점이 있다.
- 윈도우에서 웹브라우저를 실행시켜놓고 홈페이지를 방문하지 않아도 정기적으로 제어용 패킷 등 애플리케이션이나 시스템에서 필요한 통신을 하고 있다.
- 따라서 원하는 패킷을 캡처해도 다른 패킷이 섞여 있는 경우가 많다.
- 이러한 경우에는 그림에 나타난 것처럼 필터 툴바의 텍스트 상자에 [arp or dns or tcp or http]라고 입력한 다음 엔터키를 누르거나 (Apply) 버튼을 클릭하면 패킷 목록 정보에 나타나는 패킷은 ARP, DNS, TCP, HTTP 프로토콜을 포함하는 것만으로 제한할 수 있다.



2. 덤프 분석

- 또한 ARP, DNS 및 HTTP 패킷은 항상 캡처할 수 있는 것은 아니다.
- 윈도우즈에는 이미 실행한 ARP 결과를 일시적으로 저장해 두는 [캐시]라는 기능이 있으며, 저장 시간은 표준에는 2분으로 되어 있다.
- 이 시간 동안에는 ARP 캐시에 있는 내용을 이용하기 때문에 ARP 패킷을 캡처할 수 없다.
- 마찬가지로 DNS 질의 결과나 HTML 홈페이지도 캐시된다.
- 따라서 패킷을 제대로 캡처할 수 없는 경우에는 잠시 시간을 두고 다시 패킷을 캡처하거나 웹 브라우저의 임시 파일이나 이력 등을 클리어하거나 홈페이지를 열 때에 슈퍼 리로드 기능을 이용하면 된다.
- 또한 구내 LAN에서 프록시 서버를 경유하여 접속한 경우에는 프록시 서버와의 통신 패킷이 캡처된다.

Thank You