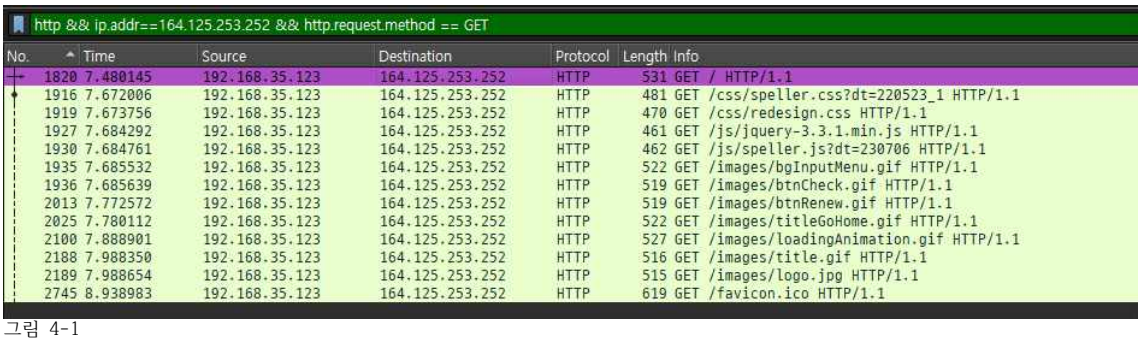
4. HTTP 프로토콜의 요청, 응답 메시지 분석

4.1 요청 메시지



No.	Time	Source	Destination	Protocol	Length	Info
1820	7.480145	192.168.35.123	164.125.253.252	HTTP	531	GET / HTTP/1.1
1916	7.672006	192.168.35.123	164.125.253.252	HTTP	481	GET /css/speller.css?dt=220523_1 HTTP/1.1
1919	7.673756	192.168.35.123	164.125.253.252	HTTP	470	GET /css/redesign.css HTTP/1.1
1927	7.684292	192.168.35.123	164.125.253.252	HTTP	461	GET /js/jquery-3.3.1.min.js HTTP/1.1
1930	7.684761	192.168.35.123	164.125.253.252	HTTP	462	GET /js/speller.js?dt=230706 HTTP/1.1
1935	7.685532	192.168.35.123	164.125.253.252	HTTP	522	GET /images/bgInputMenu.gif HTTP/1.1
1936	7.685639	192.168.35.123	164.125.253.252	HTTP	519	GET /images/btnCheck.gif HTTP/1.1
2013	7.772572	192.168.35.123	164.125.253.252	HTTP	519	GET /images/btnRenew.gif HTTP/1.1
2025	7.780112	192.168.35.123	164.125.253.252	HTTP	522	GET /images/titleGoHome.gif HTTP/1.1
2100	7.888901	192.168.35.123	164.125.253.252	HTTP	527	GET /images/loadingAnimation.gif HTTP/1.1
2188	7.988350	192.168.35.123	164.125.253.252	HTTP	516	GET /images/title.gif HTTP/1.1
2189	7.988654	192.168.35.123	164.125.253.252	HTTP	515	GET /images/logo.jpg HTTP/1.1
2745	8.938983	192.168.35.123	164.125.253.252	HTTP	619	GET /favicon.ico HTTP/1.1

그림 4-1

필터 검색 기능으로 조건문을 작성하여 클라이언트가 웹 서버로 보낸 HTTP 프로토콜의 요청 메시지를 정렬시킬 수 있고 총 13개의 요청 메시지가 전송된 것을 그림 4-1을 통해 확인할 수 있다.

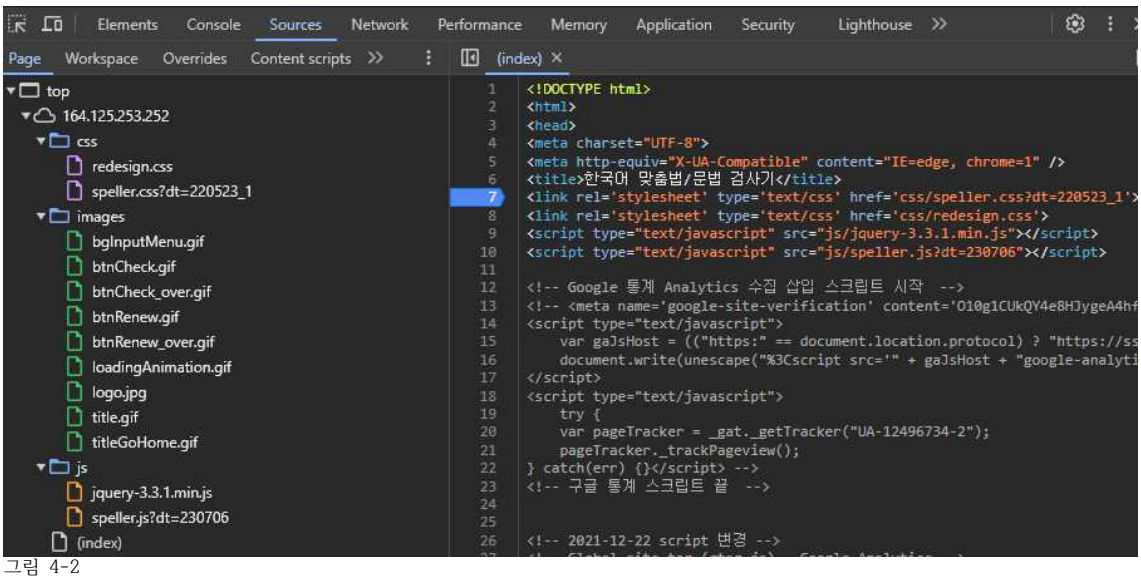
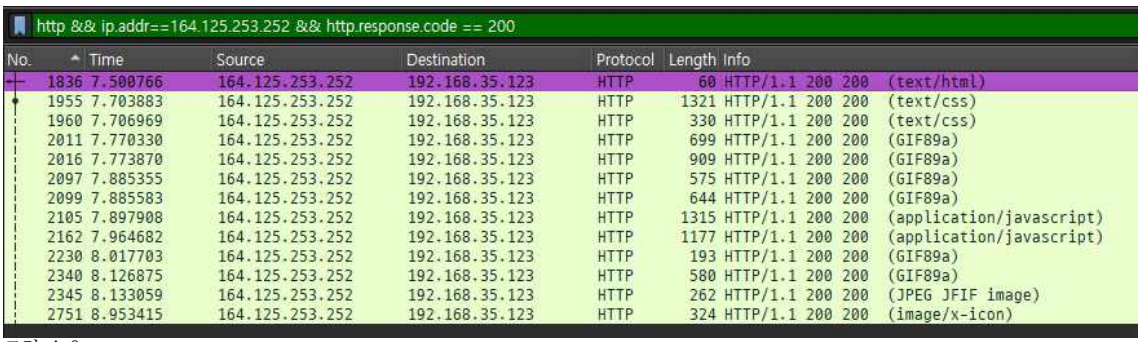


그림 4-2

그림 4-2에서 클라이언트가 웹사이트의 어떤 문서를 요청했는지 요청 메시지의 요청문과 개발자 도구의 Sources 탭에 있는 문서의 이름을 대조하여 직접 확인할 수 있다.

4.2 응답 메시지



No.	Time	Source	Destination	Protocol	Length	Info
1836	7.500766	164.125.253.252	192.168.35.123	HTTP	60	HTTP/1.1 200 200 (text/html)
1955	7.703883	164.125.253.252	192.168.35.123	HTTP	1321	HTTP/1.1 200 200 (text/css)
1960	7.706969	164.125.253.252	192.168.35.123	HTTP	330	HTTP/1.1 200 200 (text/css)
2011	7.770330	164.125.253.252	192.168.35.123	HTTP	699	HTTP/1.1 200 200 (GIF89a)
2016	7.773870	164.125.253.252	192.168.35.123	HTTP	909	HTTP/1.1 200 200 (GIF89a)
2097	7.885355	164.125.253.252	192.168.35.123	HTTP	575	HTTP/1.1 200 200 (GIF89a)
2099	7.885583	164.125.253.252	192.168.35.123	HTTP	644	HTTP/1.1 200 200 (GIF89a)
2105	7.897908	164.125.253.252	192.168.35.123	HTTP	1315	HTTP/1.1 200 200 (application/javascript)
2162	7.964682	164.125.253.252	192.168.35.123	HTTP	1177	HTTP/1.1 200 200 (application/javascript)
2230	8.017703	164.125.253.252	192.168.35.123	HTTP	193	HTTP/1.1 200 200 (GIF89a)
2340	8.126875	164.125.253.252	192.168.35.123	HTTP	580	HTTP/1.1 200 200 (GIF89a)
2345	8.133059	164.125.253.252	192.168.35.123	HTTP	262	HTTP/1.1 200 200 (JPEG JFIF image)
2751	8.953415	164.125.253.252	192.168.35.123	HTTP	324	HTTP/1.1 200 200 (image/x-icon)

그림 4-3

요청 메시지와 마찬가지로 필터 검색 기능으로 웹 서버에서 클라이언트로 보낸 HTTP 프로토콜의 응답 메시지를 정렬시킬 수 있고 총 13개의 성공 응답을 수신한 것으로 모든 요청에 대해 응답받은 것을 그림 4-3을 통해 확인할 수 있다.

4.3 HTTP 요청과 응답

HTTP 요청 메시지에서는 요청하는 문서, HTTP 버전, 클라이언트가 이해할 수 있는 데이터 형식과 사용하는 언어, 웹 브라우저 정보, 지속 연결 여부, 지속 연결 기간, 쿠키 등이 담기고 HTTP 응답 메시지에는 HTTP 버전, 상태코드, 연결 종료 여부, 응답 시간, 마지막 수정 시점, 데이터 형식과 길이, 데이터 등이 담긴다.

4.3.1 index 문서에 대한 요청과 응답

- index 문서의 요청 패킷

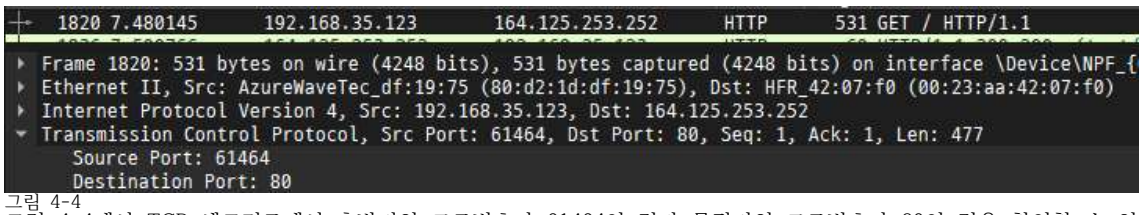


그림 4-4

그림 4-4에서 TCP 세그먼트에서 출발지의 포트번호가 61464인 것과 목적지의 포트번호가 80인 것을 확인할 수 있다.

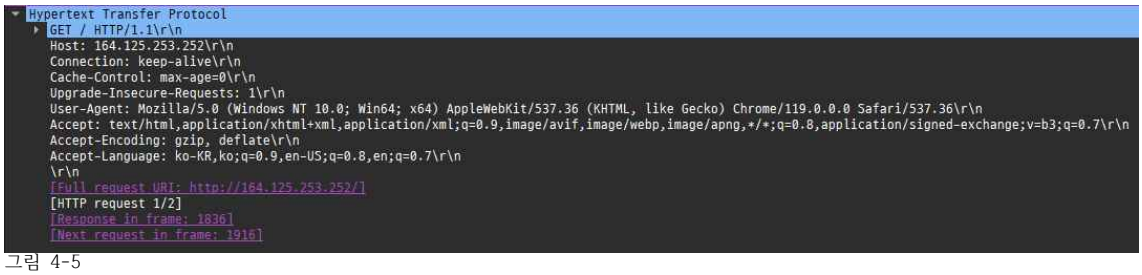


그림 4-5

요청 메시지의 헤더 정보

요청문 : 문서에 대한 정보를 담지 않았으므로 기본 문서인 index를 GET 메소드로 요청한다.

연결 형태 : 지속 연결

캐시 유지 시간 : 프록시 서버에 저장되는 캐시를 즉시 만료

보안 연결 : https 페이지로의 전환을 유도

브라우저 식별자 : Mozilla, AppleWebkit, Chrome, Safari

받을 수 있는 문서 형태 : html, xhtml, xml

받을 수 있는 이미지 형태 : avif, webp, apng

압축 형태 : gzip, deflate

사용 언어 : 한국어, 영어

언어의 우선순위 : 한국어 > 미국영어 > 영어

공백 이후의 내용이 없으므로 body부분에 아무런 정보가 담기지 않은 것을 확인할 수 있다.

-index 문서의 응답 패킷

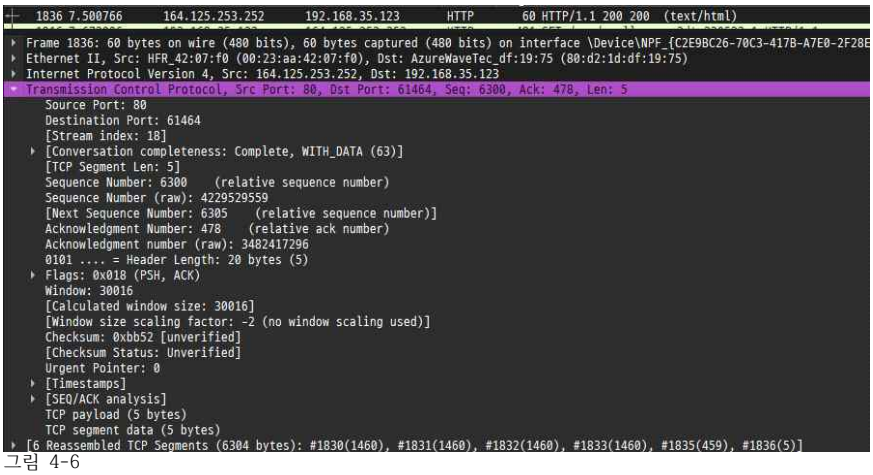


그림 4-6

그림 4-6에서 TCP 세그먼트에서 출발지의 포트번호가 80인 것과 목적지의 포트번호가 61464인 것과 6개의 분할된 세그먼트를 재조립하여 하나의 세그먼트로 재조립한 것을 알 수 있다.

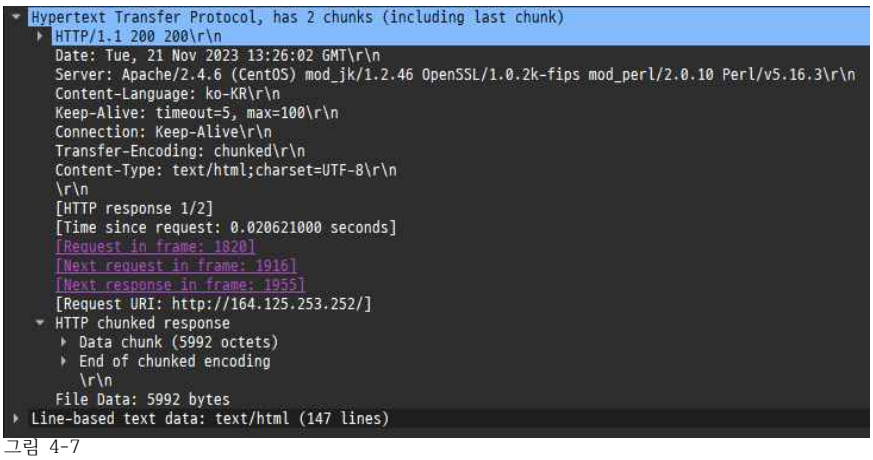


그림 4-7

응답 메시지의 헤더 정보

상태문 : 응답 성공

응답 시간 : 그리니치 천문대의 기준으로 2023/11/21, 13:26:02에 전송된 패킷

서버에서 사용하는 SW : Apache, mod_jk, OpenSSL, mod_perl, Perl

데이터의 언어 : 한국어

지속 연결 시간 : 5초

연결당 최대 요청 횟수 : 100회

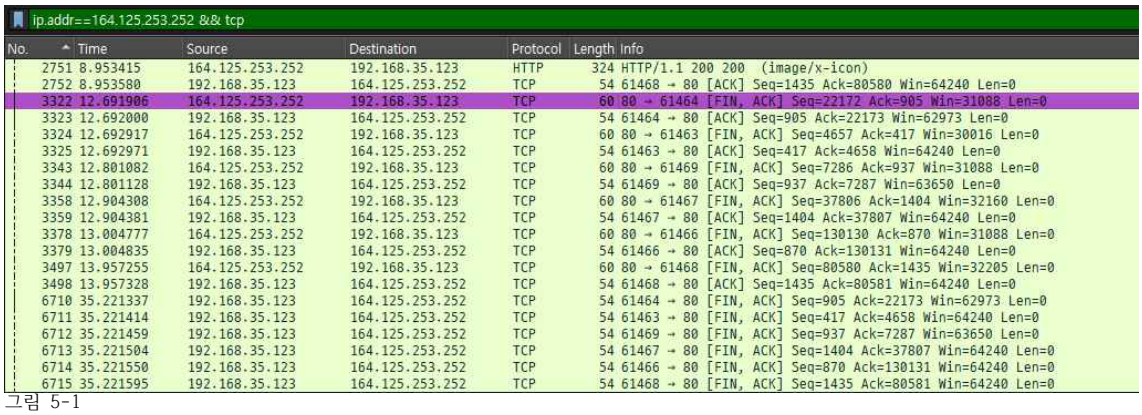
연결 유지 여부 : 연결 유지

메시지의 전송 인코딩 기법 : chunk (여러 조각으로 나누어 보냄)

데이터 형태 : html, UTF-8 인코딩

body부분에 html 문서가 담김

5. 웹서버와 TCP 연결해제과정



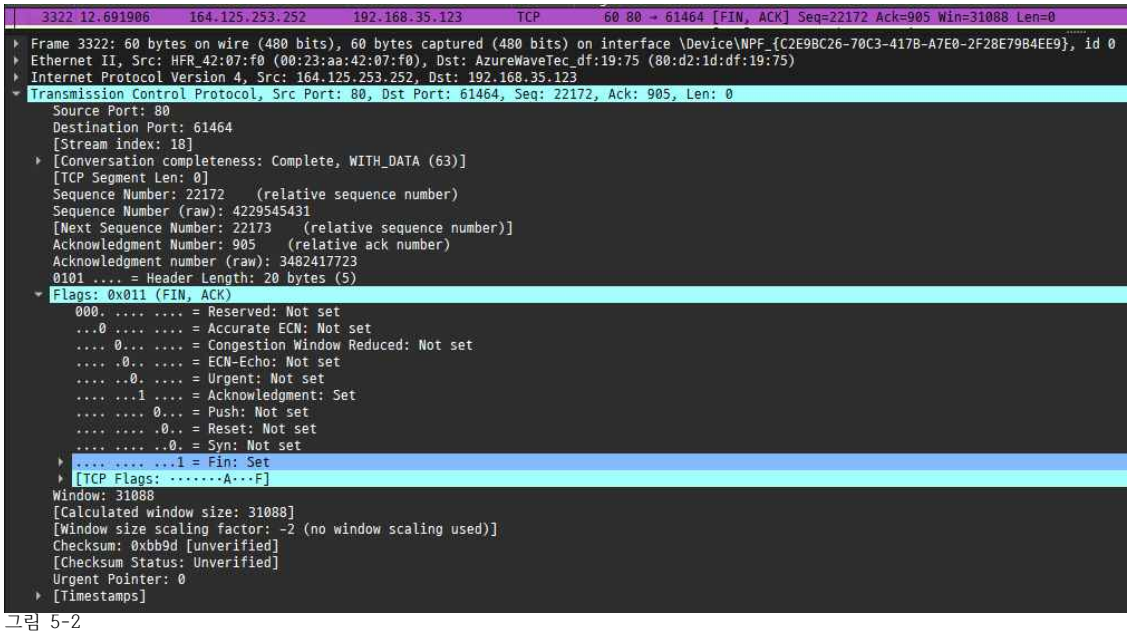
ip.addr==164.125.253.252 && tcp

No.	Time	Source	Destination	Protocol	Length	Info
2751	8.953415	164.125.253.252	192.168.35.123	HTTP	324	HTTP/1.1 200 200 (image/x-icon)
2752	8.953580	192.168.35.123	164.125.253.252	TCP	54	61468 → 80 [ACK] Seq=1435 Ack=80580 Win=64240 Len=0
3322	12.691906	164.125.253.252	192.168.35.123	TCP	60	80 → 61464 [FIN, ACK] Seq=22172 Ack=905 Win=31088 Len=0
3323	12.692000	192.168.35.123	164.125.253.252	TCP	54	61464 → 80 [ACK] Seq=905 Ack=22173 Win=62973 Len=0
3324	12.692917	164.125.253.252	192.168.35.123	TCP	60	80 → 61463 [FIN, ACK] Seq=4657 Ack=417 Win=30016 Len=0
3325	12.692971	192.168.35.123	164.125.253.252	TCP	54	61463 → 80 [ACK] Seq=417 Ack=4658 Win=64240 Len=0
3343	12.801082	164.125.253.252	192.168.35.123	TCP	60	80 → 61469 [FIN, ACK] Seq=7286 Ack=937 Win=31088 Len=0
3344	12.801128	192.168.35.123	164.125.253.252	TCP	54	61469 → 80 [ACK] Seq=937 Ack=7287 Win=63650 Len=0
3358	12.904308	164.125.253.252	192.168.35.123	TCP	60	80 → 61467 [FIN, ACK] Seq=37806 Ack=1404 Win=32160 Len=0
3359	12.904381	192.168.35.123	164.125.253.252	TCP	54	61467 → 80 [ACK] Seq=1404 Ack=37807 Win=64240 Len=0
3378	13.004777	164.125.253.252	192.168.35.123	TCP	60	80 → 61466 [FIN, ACK] Seq=130130 Ack=870 Win=31088 Len=0
3379	13.004835	192.168.35.123	164.125.253.252	TCP	54	61466 → 80 [ACK] Seq=870 Ack=130131 Win=64240 Len=0
3497	13.957255	164.125.253.252	192.168.35.123	TCP	60	80 → 61468 [FIN, ACK] Seq=80580 Ack=1435 Win=32205 Len=0
3498	13.957328	192.168.35.123	164.125.253.252	TCP	54	61468 → 80 [ACK] Seq=1435 Ack=80581 Win=64240 Len=0
6710	35.221337	192.168.35.123	164.125.253.252	TCP	54	61464 → 80 [FIN, ACK] Seq=905 Ack=22173 Win=62973 Len=0
6711	35.221414	192.168.35.123	164.125.253.252	TCP	54	61463 → 80 [FIN, ACK] Seq=417 Ack=4658 Win=64240 Len=0
6712	35.221459	192.168.35.123	164.125.253.252	TCP	54	61469 → 80 [FIN, ACK] Seq=937 Ack=7287 Win=63650 Len=0
6713	35.221504	192.168.35.123	164.125.253.252	TCP	54	61467 → 80 [FIN, ACK] Seq=1404 Ack=37807 Win=64240 Len=0
6714	35.221550	192.168.35.123	164.125.253.252	TCP	54	61466 → 80 [FIN, ACK] Seq=870 Ack=130131 Win=64240 Len=0
6715	35.221595	192.168.35.123	164.125.253.252	TCP	54	61468 → 80 [FIN, ACK] Seq=1435 Ack=80581 Win=64240 Len=0

그림 5-1

그림 5-1에서 FIN과 ACK가 설정된 패킷을 찾아 TCP 연결 해제 요청 패킷임을 확인할 수 있다. 뿐만 아니라 클라이언트 측의 61463, 61464, 61466, 61467, 61468, 61469번 포트에서 총 6개의 TCP 다중화가 발생한 것으로 나타난다.

5.1 서버 측의 TCP 연결 해제 요청



```
3322 12.691906  164.125.253.252  192.168.35.123  TCP  60 80 → 61464 [FIN, ACK] Seq=22172 Ack=905 Win=31088 Len=0
▸ Frame 3322: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{C2E9BC26-70C3-417B-A7E0-2F28E79B4EE9}, id 0
▸ Ethernet II, Src: HFR_42:07:f0 (00:23:aa:42:07:f0), Dst: AzureWaveTec_df:19:75 (80:d2:1d:df:19:75)
▸ Internet Protocol Version 4, Src: 164.125.253.252, Dst: 192.168.35.123
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 61464, Seq: 22172, Ack: 905, Len: 0
    Source Port: 80
    Destination Port: 61464
    [Stream index: 18]
  ▸ [Conversation completeness: Complete, WITH_DATA (63)]
    [TCP Segment Len: 0]
    Sequence Number: 22172    (relative sequence number)
    Sequence Number (raw): 4229545431
    [Next Sequence Number: 22173    (relative sequence number)]
    Acknowledgment Number: 905    (relative ack number)
    Acknowledgment number (raw): 3482417723
    0101 .... = Header Length: 20 bytes (5)
  ▾ Flags: 0x011 (FIN, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
    ▸ .... .... ...1 = Fin: Set
    ▸ [TCP Flags: ·····A···F]
    Window: 31088
    [Calculated window size: 31088]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0xbb9d [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  ▸ [Timestamps]
```

그림 5-2

그림 5-2에서 세그먼트에 담긴 플래그 중 ACK와 FIN이 1로 설정되어 있는 것을 볼 수 있고, 출발지와 목적지의 ip 또는 포트번호를 보고 클라이언트와 서버 중 어느쪽에서 TCP 연결 해제를 요청했는지 알 수 있다. 이 경우 서버에서 Sequence number가 22172인 연결 해제 요청 패킷을 보낸 것을 알 수 있다.

5.2 클라이언트 측의 TCP 연결 해제 요청에 대한 응답

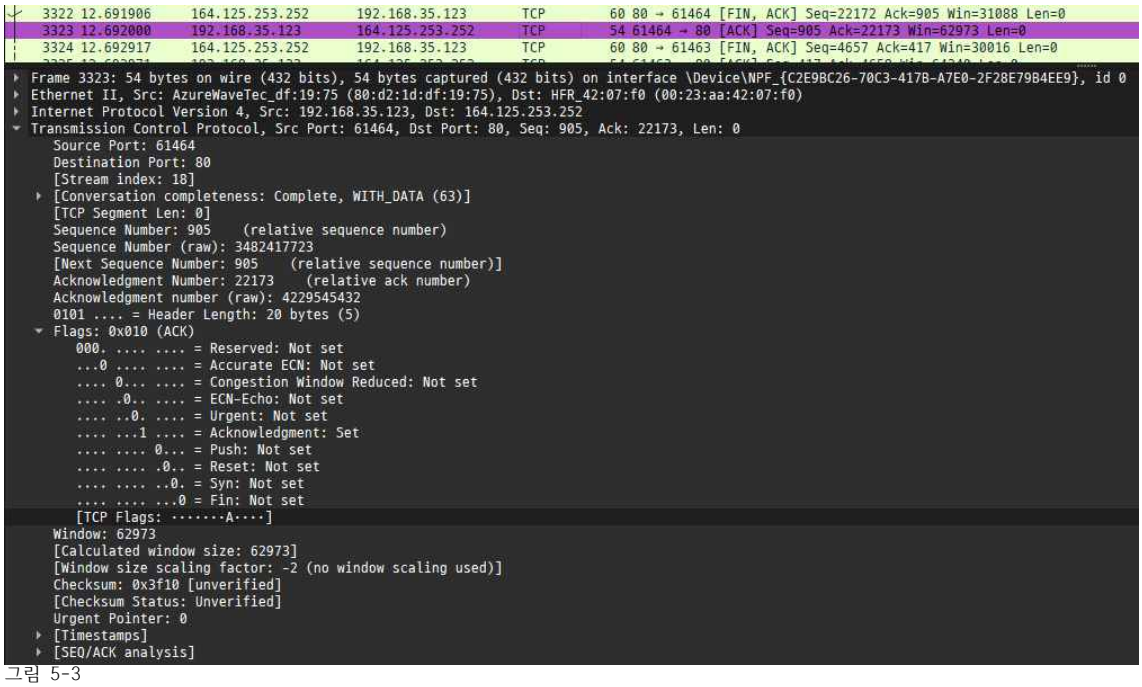


그림 5-3

그림 5-3에서 클라이언트가 서버로 보내는 TCP 세그먼트의 ACK 플래그가 1로 설정되어 있고 Acknowledge Number가 22173인 것으로 보아 이 패킷은 Sequence Number가 22172인 TCP 연결 해제 요청 패킷에 대한 응답 패킷임을 알 수 있다. 따라서 해당 패킷은 TCP 연결 해제 응답 패킷이다.

5.3 클라이언트측 TCP 연결 해제 알림

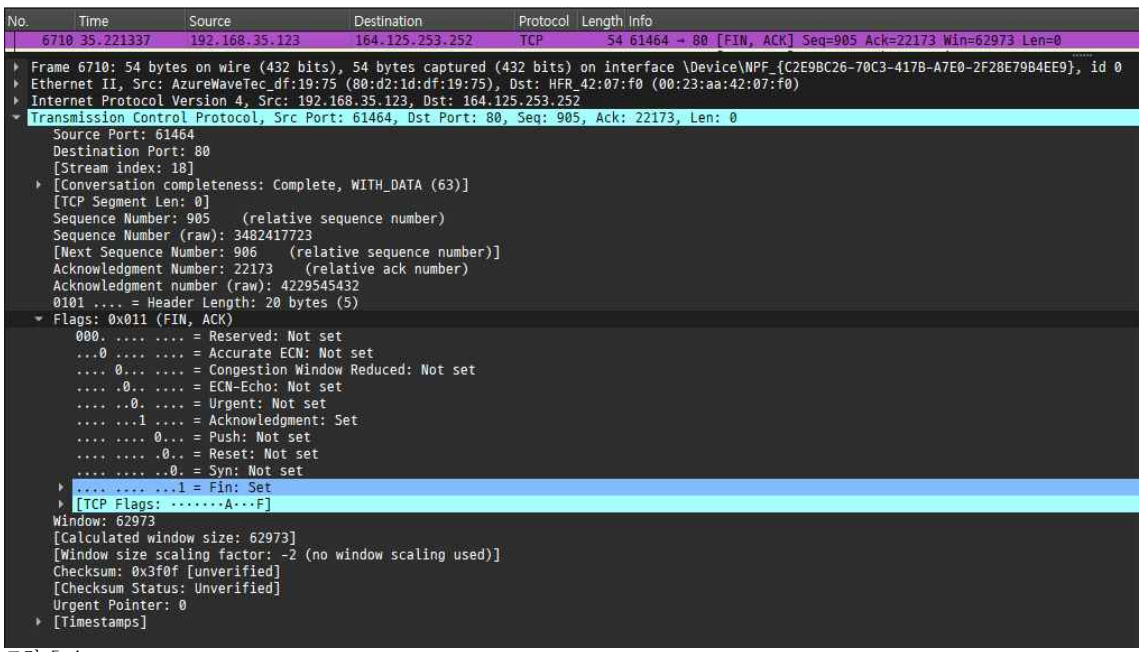


그림 5-4

그림 5-4에서 Acknowledge Number가 22173으로 TCP 연결 해제 요청에 대한 응답 패킷의 것과 같다. 이것으로 클라이언트도 서버에게 더는 요청을 보내지 않을 것이라고 알린다.