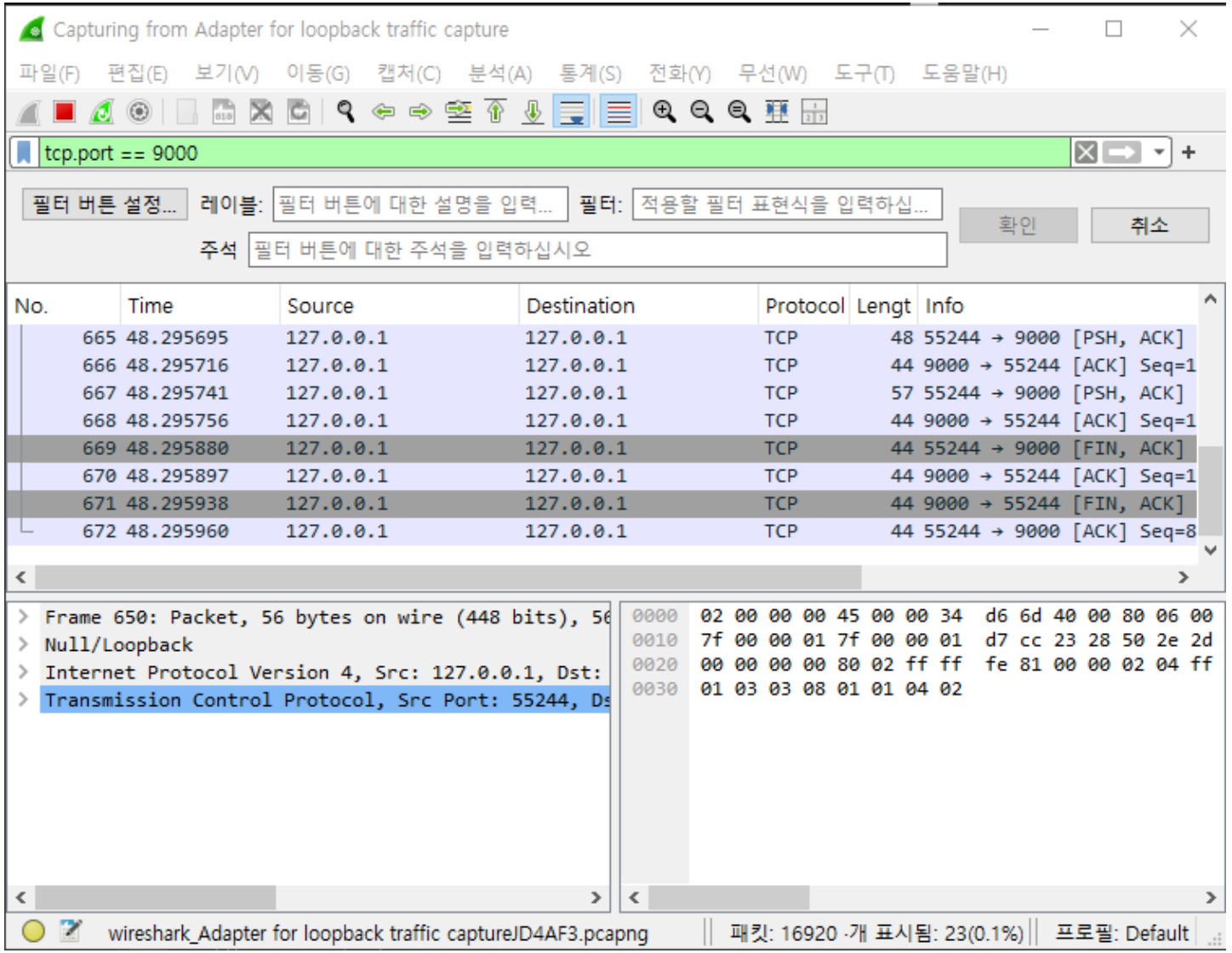
네트워크 프로그래밍 과제_2

2023100795 정세영

테스트용 코드 (고정+가변길이 전송 코드)

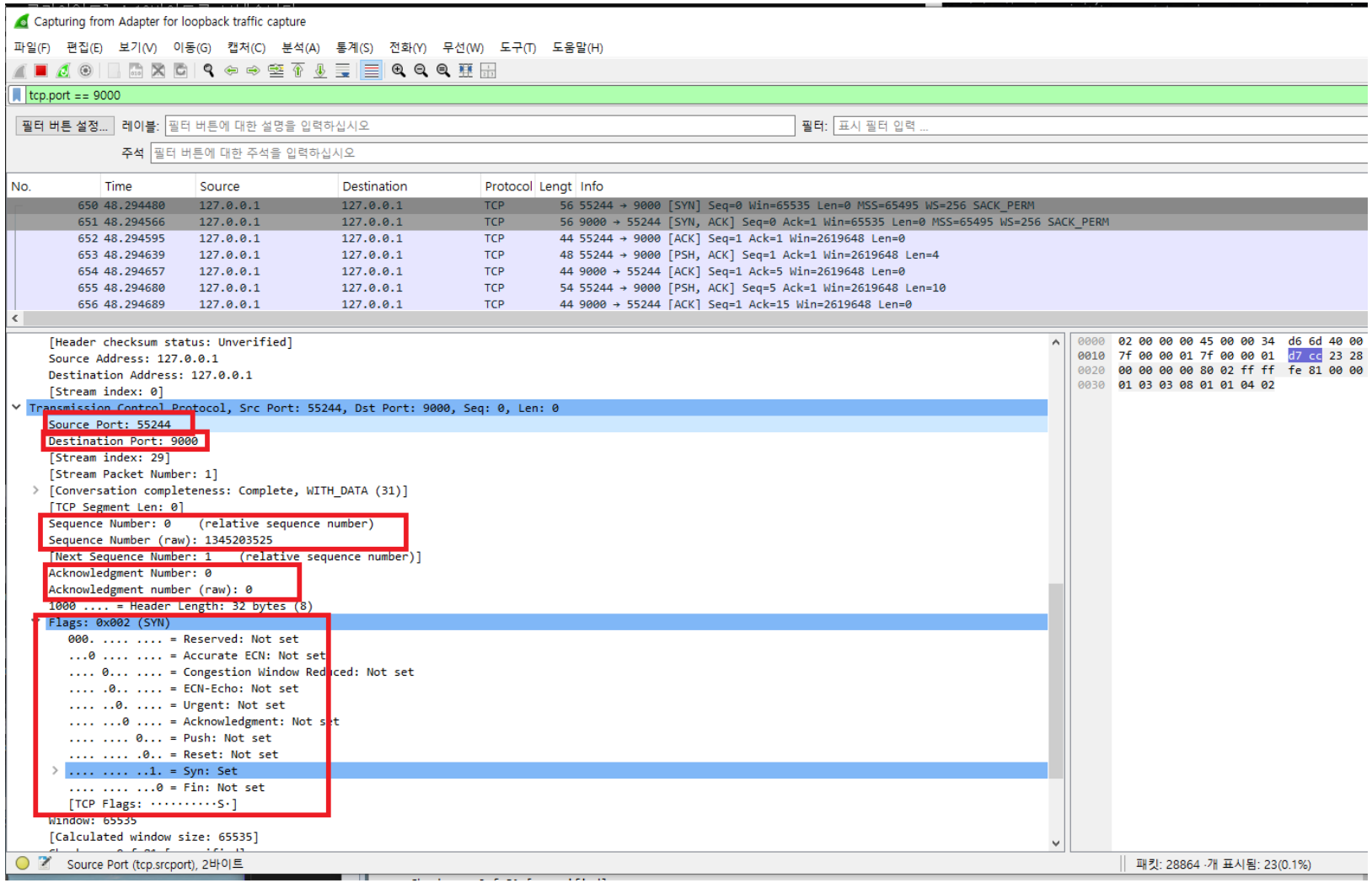
- https://github.com/Crispylux/networkprogrammingwork2

3-1. 와이어샤크 화면 상단에 필터 적용(tcp.port == 9000)



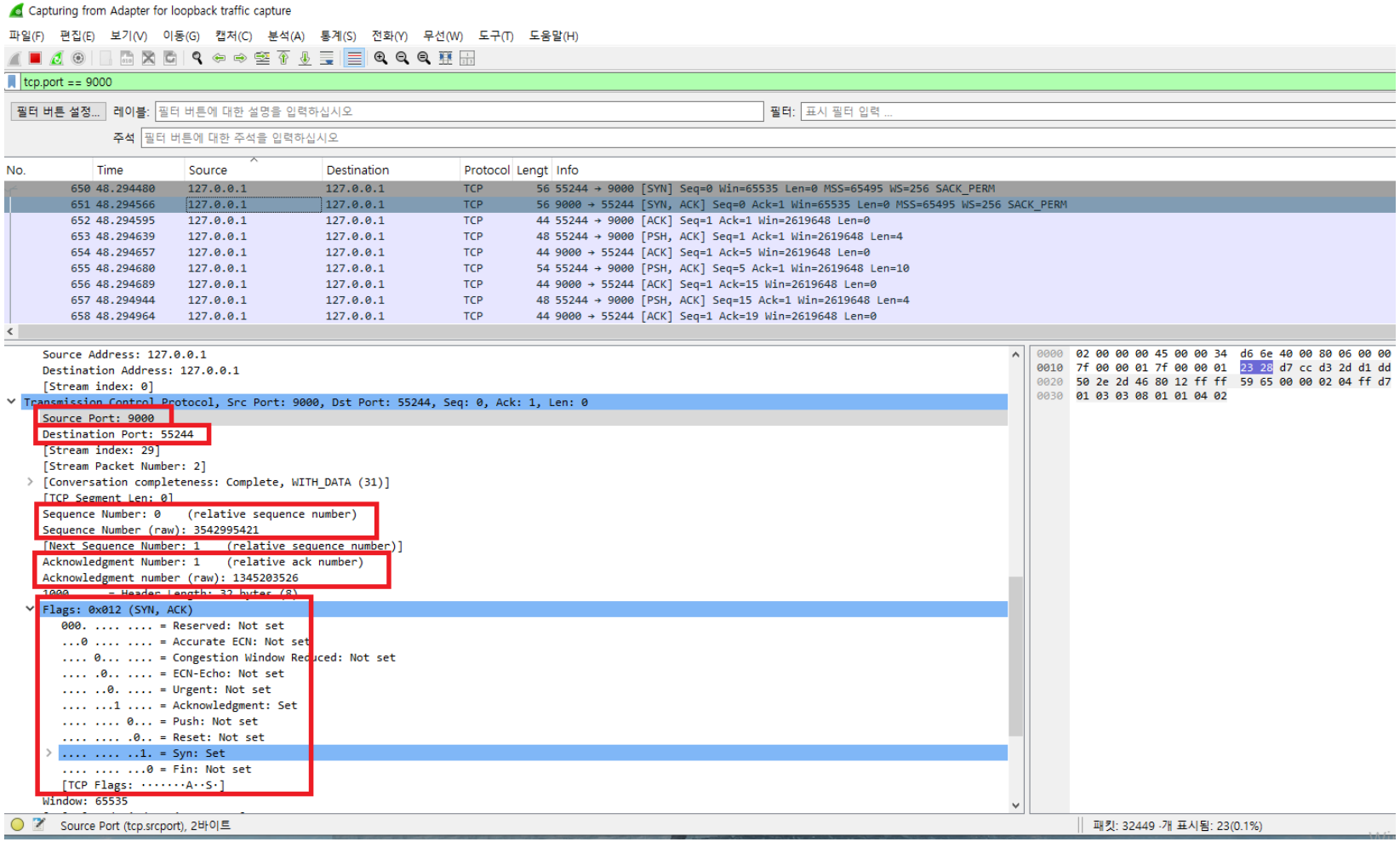
3-2. Client-Server 간의 3-Way Handshake 패킷들을 확인하여 아래 항목들을 나타내시오.

1) Syn 패킷의 Source Port, Destination Port, Flags, Sequence Number, Acknowledgment Number



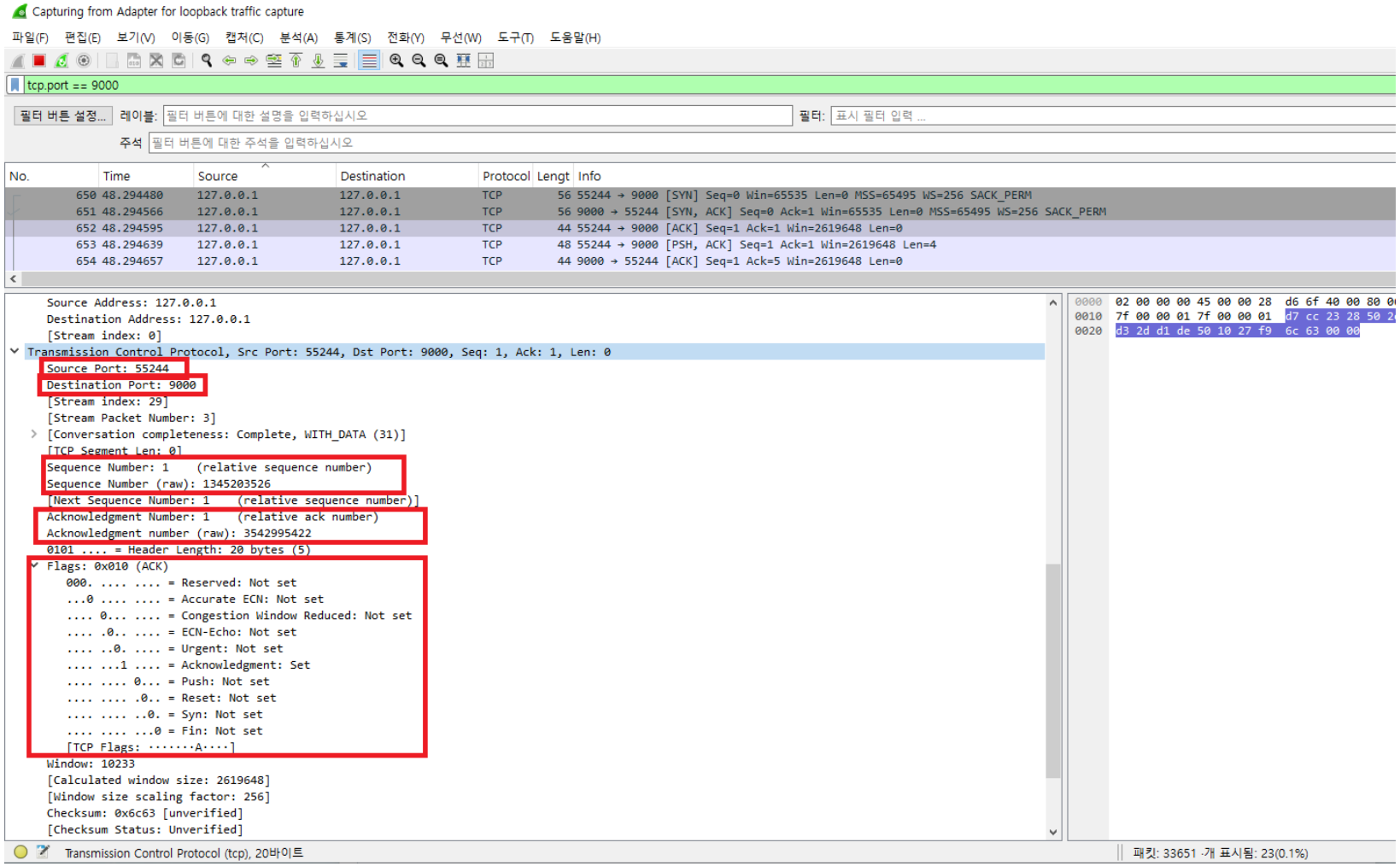
3-2. Client-Server 간의 3-Way Handshake 패킷들을 확인하여 아래 항목들을 나타내시오.

2) Syn, Ack 패킷의 Source Port, Destination Port, Flags, Sequence Number, Acknowledgment Number



3-2. Client-Server 간의 3-Way Handshake 패킷들을 확인하여 아래 항목들을 나타내시오.

3) Ack 패킷의 Source Port, Destination Port, Flags, Sequence Number, Acknowledgment Number



3-3. 3-Way Handshake 패킷에 표시되는 각 Flag들의 의미는 무엇인가?

Reserved : 미래 확장을 위해 예약된 필드(항상 0)

Accurate ECN : 혼잡 상황을 더 정확히 알리기 위한 ECN 확장 비트

Congestion Window Reduced : 송신자가 ecn을 받아 혼잡 윈도우를 줄였음을 알림

Ecn-Echo : ECN기능이 활성화되어 있으며 네트워크 혼잡이 감지되었음을 송신자에게 알림

Urgent : 긴급 포인터 필드가 유효함을 의미

Acknowledgment : 확인번호 필드가 유효함을 의미

Push : 수신 측이 데이터를 버퍼링하지 않고 즉시 상위응용계층에 전달하도록 요청

Reset : 비정상적인 연결을 즉시 종료.

Syn : 연결 설정 요청, 세션을 시작하면 초기 시퀀스번호(ISN)을 교환

Fin : 연결 종료 요청

3-4. Sequence Number, Acknowledgment Number는 어떠한 규칙으로 부여되는가? (패킷들의 예를 들면서 설명하시오.)

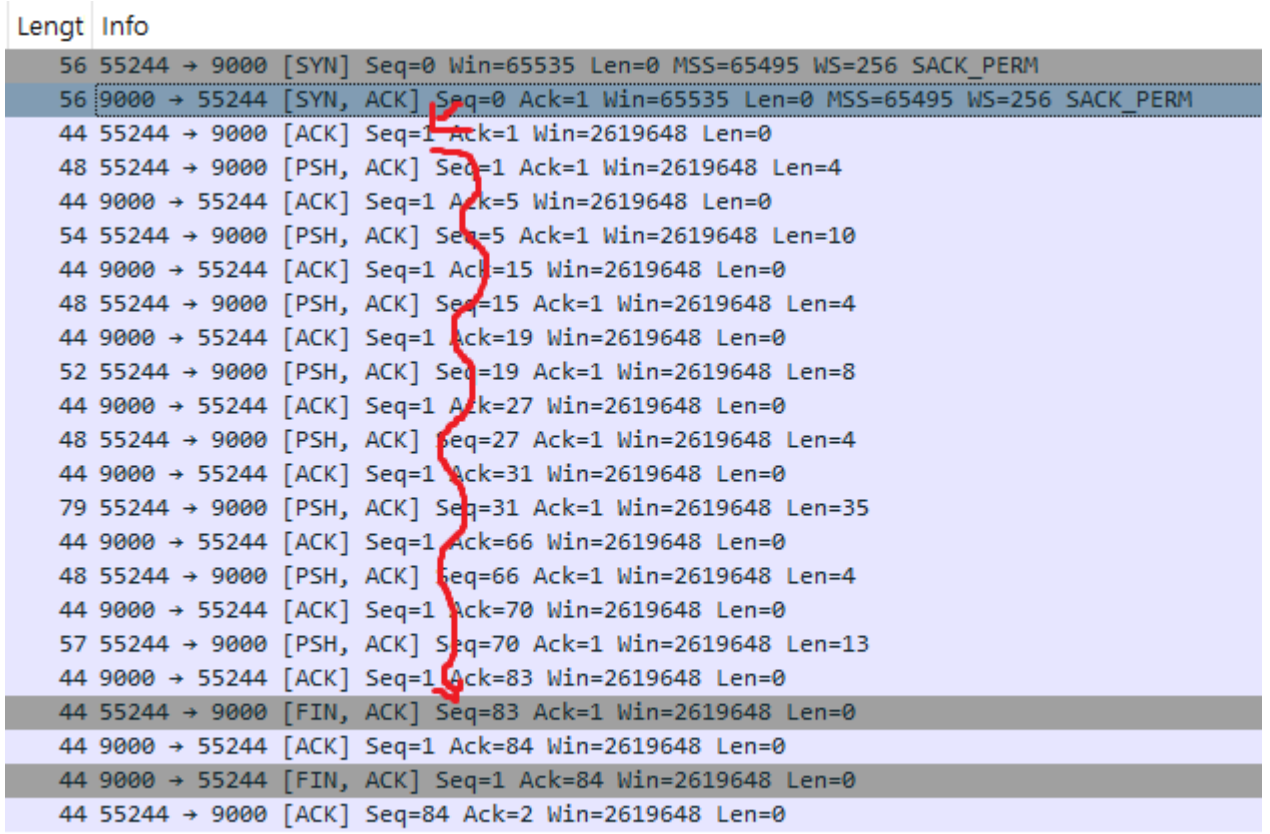
Sequence Number(Seq)는 데이터 첫 바이트 번호를 의미한다
Acknowledgment Number(Ack)는 다음에 수신하고자 하는 바이트의 번호를 의미한다(지금까지 수신한 데이터 +1)

[SYN] Seq=0 (초기 연결)
[SYN, ACK] Seq=0 Ack=1 (서버 응답)
[ACK] Seq=1 Ack=1 Len=0 (연결 완료)

[PSH, ACK] Seq=1 Ack=1 Len=4 (4바이트 전송)
[ACK] Seq=1 Ack=5 (4바이트 받았다고 확인)

[PSH, ACK] Seq=5 Ack=1 Len=10 (10바이트 전송)
[ACK] Seq=1 Ack=15 (10바이트 받음, 다음은 15번)

식으로 값이 증가한다.



3-5. 고정 길이 데이터(4바이트 고정 길이)가 전달되는 패킷 중에 하나를 확인하고 해당 패킷의 IP Total Length, IP Header Length, TCP Header Length 필드를 확인하고 4바이트 페이로드가 전달됨을 나타내시오.

참고: TCP 데이터 크기 = IP Total Length - IP Header Length - TCP Header Length

No.	Time	Source	Destination	Protocol	Lengt	Info
650	48.294480	127.0.0.1	127.0.0.1	TCP	56	55244 → 9000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
651	48.294566	127.0.0.1	127.0.0.1	TCP	56	9000 → 55244 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
652	48.294595	127.0.0.1	127.0.0.1	TCP	44	55244 → 9000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
653	48.294639	127.0.0.1	127.0.0.1	TCP	48	55244 → 9000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=4
654	48.294657	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=5 Win=2619648 Len=0
655	48.294680	127.0.0.1	127.0.0.1	TCP	54	55244 → 9000 [PSH, ACK] Seq=5 Ack=1 Win=2619648 Len=10
656	48.294689	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=15 Win=2619648 Len=0
657	48.294944	127.0.0.1	127.0.0.1	TCP	48	55244 → 9000 [PSH, ACK] Seq=15 Ack=1 Win=2619648 Len=4
658	48.294964	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=19 Win=2619648 Len=0
659	48.294989	127.0.0.1	127.0.0.1	TCP	52	55244 → 9000 [PSH, ACK] Seq=19 Ack=1 Win=2619648 Len=8
660	48.295003	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=27 Win=2619648 Len=0
661	48.295482	127.0.0.1	127.0.0.1	TCP	48	55244 → 9000 [PSH, ACK] Seq=27 Ack=1 Win=2619648 Len=4
662	48.295497	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=31 Win=2619648 Len=0
663	48.295524	127.0.0.1	127.0.0.1	TCP	79	55244 → 9000 [PSH, ACK] Seq=31 Ack=1 Win=2619648 Len=35
664	48.295538	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=66 Win=2619648 Len=0
665	48.295695	127.0.0.1	127.0.0.1	TCP	48	55244 → 9000 [PSH, ACK] Seq=66 Ack=1 Win=2619648 Len=4
666	48.295716	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=70 Win=2619648 Len=0
667	48.295741	127.0.0.1	127.0.0.1	TCP	57	55244 → 9000 [PSH, ACK] Seq=70 Ack=1 Win=2619648 Len=13
668	48.295756	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=83 Win=2619648 Len=0
669	48.295880	127.0.0.1	127.0.0.1	TCP	44	55244 → 9000 [FIN, ACK] Seq=83 Ack=1 Win=2619648 Len=0
670	48.295897	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=84 Win=2619648 Len=0
671	48.295938	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [FIN, ACK] Seq=1 Ack=84 Win=2619648 Len=0
672	48.295960	127.0.0.1	127.0.0.1	TCP	44	55244 → 9000 [ACK] Seq=84 Ack=2 Win=2619648 Len=0

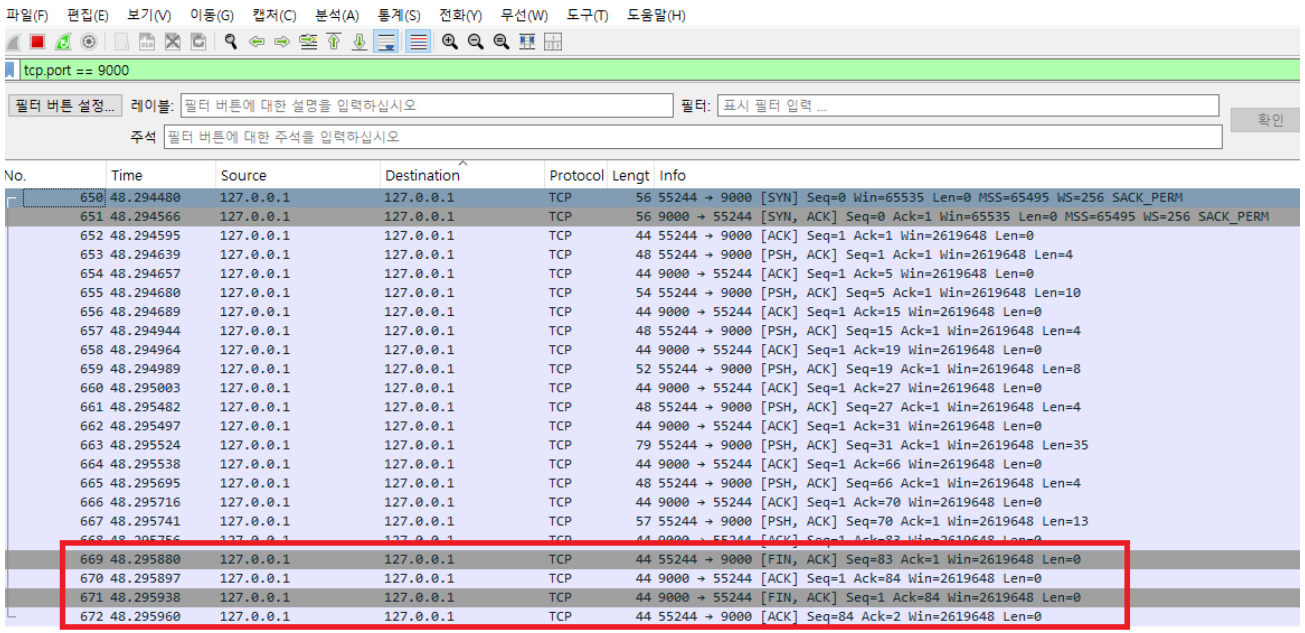
```
    Character encoding: ASCII (0)
    [Coloring Rule Name: TCP]
    [Coloring Rule String: tcp]
▾ Null/Loopback
    Family: IP (2)
▾ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▾ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 44
    Identification: 0xd670 (54896)
  ▾ 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: 127.0.0.1
```

```
▾ Transmission Control Protocol, Src Port: 55244, Dst Port: 9000, Seq: 1, Ack: 1, Len: 4
    Source Port: 55244
    Destination Port: 9000
    [Stream index: 29]
    [Stream Packet Number: 4]
  ▸ [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 4]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 1345203526
    [Next Sequence Number: 5    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3542995422
    0101 .... = Header Length: 20 bytes (5)
  ▾ Flags: 0x018 (PSH, 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
      .... .... 1... = Push: Set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
```

TCP 데이터 크기 = 44 – 20 – 20 = 4byte

따라서 4바이트 페이로드가 전달된다!

3-6. Wireshark에서 4-Way Handshake를 통한 연결 종료 과정을 분석하시오.



No.	Time	Source	Destination	Protocol	Lengt	Info
650	48.294480	127.0.0.1	127.0.0.1	TCP	56	55244 → 9000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
651	48.294566	127.0.0.1	127.0.0.1	TCP	56	9000 → 55244 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
652	48.294595	127.0.0.1	127.0.0.1	TCP	44	55244 → 9000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
653	48.294639	127.0.0.1	127.0.0.1	TCP	48	55244 → 9000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=4
654	48.294657	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=5 Win=2619648 Len=0
655	48.294680	127.0.0.1	127.0.0.1	TCP	54	55244 → 9000 [PSH, ACK] Seq=5 Ack=1 Win=2619648 Len=10
656	48.294689	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=15 Win=2619648 Len=0
657	48.294944	127.0.0.1	127.0.0.1	TCP	48	55244 → 9000 [PSH, ACK] Seq=15 Ack=1 Win=2619648 Len=4
658	48.294964	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=19 Win=2619648 Len=0
659	48.294989	127.0.0.1	127.0.0.1	TCP	52	55244 → 9000 [PSH, ACK] Seq=19 Ack=1 Win=2619648 Len=8
660	48.295003	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=27 Win=2619648 Len=0
661	48.295482	127.0.0.1	127.0.0.1	TCP	48	55244 → 9000 [PSH, ACK] Seq=27 Ack=1 Win=2619648 Len=4
662	48.295497	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=31 Win=2619648 Len=0
663	48.295524	127.0.0.1	127.0.0.1	TCP	79	55244 → 9000 [PSH, ACK] Seq=31 Ack=1 Win=2619648 Len=35
664	48.295538	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=66 Win=2619648 Len=0
665	48.295695	127.0.0.1	127.0.0.1	TCP	48	55244 → 9000 [PSH, ACK] Seq=66 Ack=1 Win=2619648 Len=4
666	48.295716	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=70 Win=2619648 Len=0
667	48.295741	127.0.0.1	127.0.0.1	TCP	57	55244 → 9000 [PSH, ACK] Seq=70 Ack=1 Win=2619648 Len=13
669	48.295880	127.0.0.1	127.0.0.1	TCP	44	55244 → 9000 [FIN, ACK] Seq=83 Ack=1 Win=2619648 Len=0
670	48.295897	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [ACK] Seq=1 Ack=84 Win=2619648 Len=0
671	48.295938	127.0.0.1	127.0.0.1	TCP	44	9000 → 55244 [FIN, ACK] Seq=1 Ack=84 Win=2619648 Len=0
672	48.295960	127.0.0.1	127.0.0.1	TCP	44	55244 → 9000 [ACK] Seq=84 Ack=2 Win=2619648 Len=0

55244 -> 9000 [FIN, ACK] : 클라이언트가 서버에게 종료하자고 요청함

9000 -> 55244 [ACK] : 서버가 확인했다고 보냄

9000 -> 55244 [FIN, ACK] : 서버가 클라이언트한테 나도 종료한다고 함

55244 -> 9000 [ACK] : 클라이언트가 확인했다고 보냄