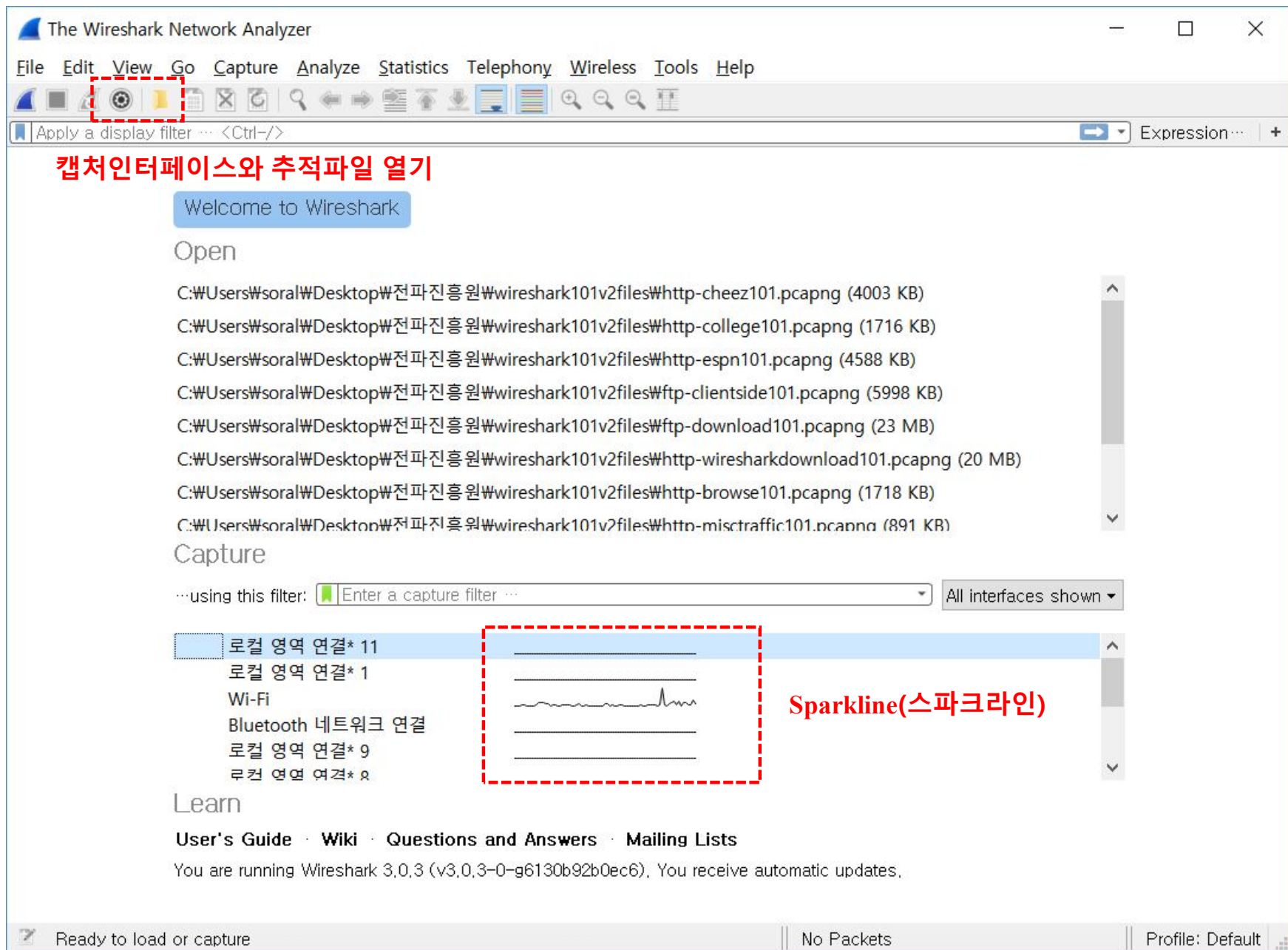


0장. 와이어샹크 핵심요소와 트래픽 흐름



Capturing from Wi-Fi
 ① 타이틀바(제목표시줄)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
 ② 메인 메뉴

③ 메인 툴바

Apply a display filter ... <Ctrl-/>
 ④ 디스플레이필터 영역과 필터 표현식 영역

No.	Time	Source	Destination	Protocol	Length	Info
233	0.000000	172.30.1.29	172.30.1.255	DB-LSP-DISC	211	Dropbox LAN sync Discovery
234	3.044096	40.100.49.18	172.30.1.15	TLSv1.2	97	Application Data
235	0.000001	40.100.48.98	172.30.1.15	TLSv1.2	97	Application Data
236	0.051135	172.30.1.15	40.100.48.98	TCP	54	50756 → https(443) [ACK] Seq
237	0.000001	172.30.1.15	40.100.49.18	TCP	54	50747 → https(443) [ACK] Seq
238	0.693244	172.30.1.12	224.0.0.251	MDNS	222	Standard query response 0x0
239	1.023859	172.30.1.12	224.0.0.251	MDNS	222	Standard query response 0x0
240	0.001810	fe80::1021:ec19:c0...	ff02::fb	MDNS	292	Standard query 0x0000 PTR _

⑤ 패킷 목록(list) 창

> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
 > Ethernet II, Src: IntelCor_51:0b:53 (68:ec:c5:51:0b:53), Dst: Mercury_15:eb:de (b4:a9:4f:15:eb:de)
 > Internet Protocol Version 4, Src: 172.30.1.15, Dst: 69.167.144.15
 ▾ Transmission Control Protocol
 Source Port: 51006 (51006)
 Destination Port: https (443)
 [Stream index: 0]
 [TCP Segment Len: 0]
 ⑥ 패킷 상세(details) 창

0000	b4 a9 4f 15 eb de 68 ec c5 51 0b 53 08 00 45 00	..O...h..Q.S..E.
0010	00 34 2a 94 40 00 80 06 4d 4c ac 1e 01 0f 45 a7	.4* @... ML...E.
0020	90 0f c7 3e 01 bb d9 96 8b f3 00 00 00 00 80 02	...>.....
0030	fe 88 bf 1f 00 00 02 04 05 b4 01 03 03 08 01 01
0040	04 02	..

 ⑦ 패킷 바이트(byte) 창

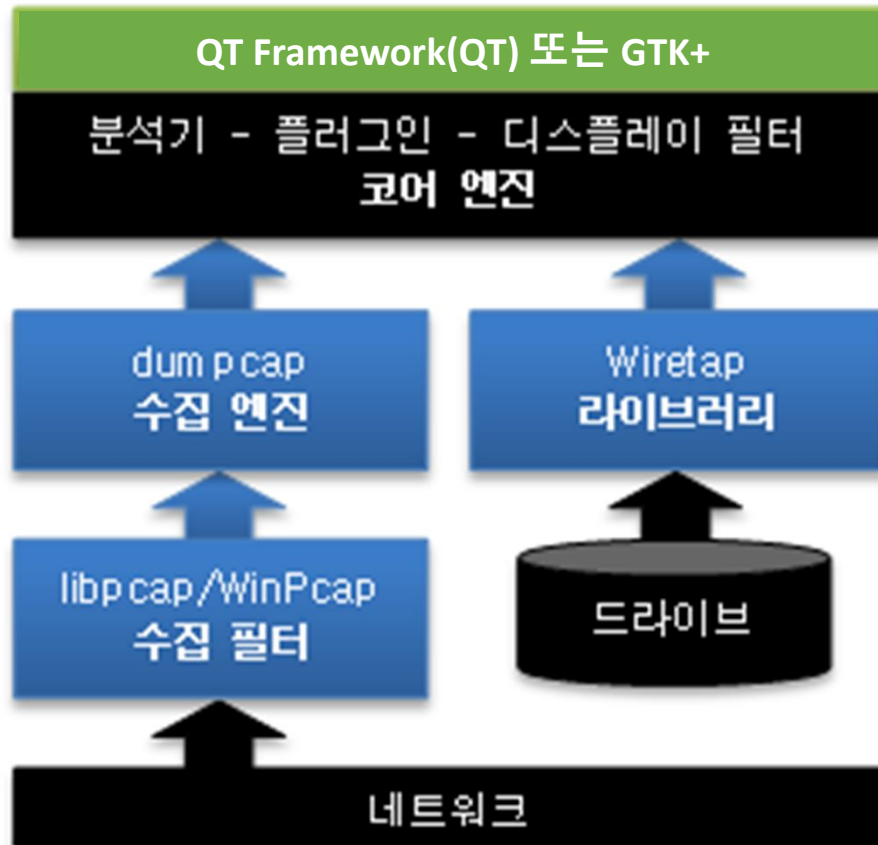
Wi-Fi: <live capture in progress>
 ⑧ 상태바(Status Bar)
 Packets: 240 · Displayed: 240 (100.0%)
 Profile: Default

- 패킷 지시기(Packet indicator)
- 인텔리전트 스크롤바 (Intelligent Scrollbar)

0.1 와이어샹크의 핵심 기능 이해

- 일반 분석 작업
- 문제점 해결 작업
- 보안 분석(네트워크 포렌식) 작업
- 애플리케이션 분석 작업

0.3 와이어샤크가 트래픽을 수집하는 방법



- DumpCap은 특수 링크 계층 드라이버에 의존
- DumpCap 수집엔진은 정지 조건을 지정
- 코어엔진
 - 수천개의 해석기(dissector)를 제공
 - 해석기는 프레임 필드를 쪼개 내용 분석
- QT프레임워크
 - 사용자 인터페이스 제공
- Wiretap 라이브러리
 - 저장된 추적파일을 읽는데 사용

0.4 전형적인 와이어샤크 분석 세션의 이해

- 추적 파일에서 누가 통신하고 있는지 알아낸다.
- 사용 중인 애플리케이션이 무엇인지 알아낸다.
- 관심 있는 대화를 필터링한다.
- 처리율에서 폐기를 보기 위해 IO를 그래프로 나타낸다.
- 문제점을 알아내기 위해 전문자(export)를 연다.
- 경로 전달 지연을 파악하기 위해 왕복 시간을 알아낸다.

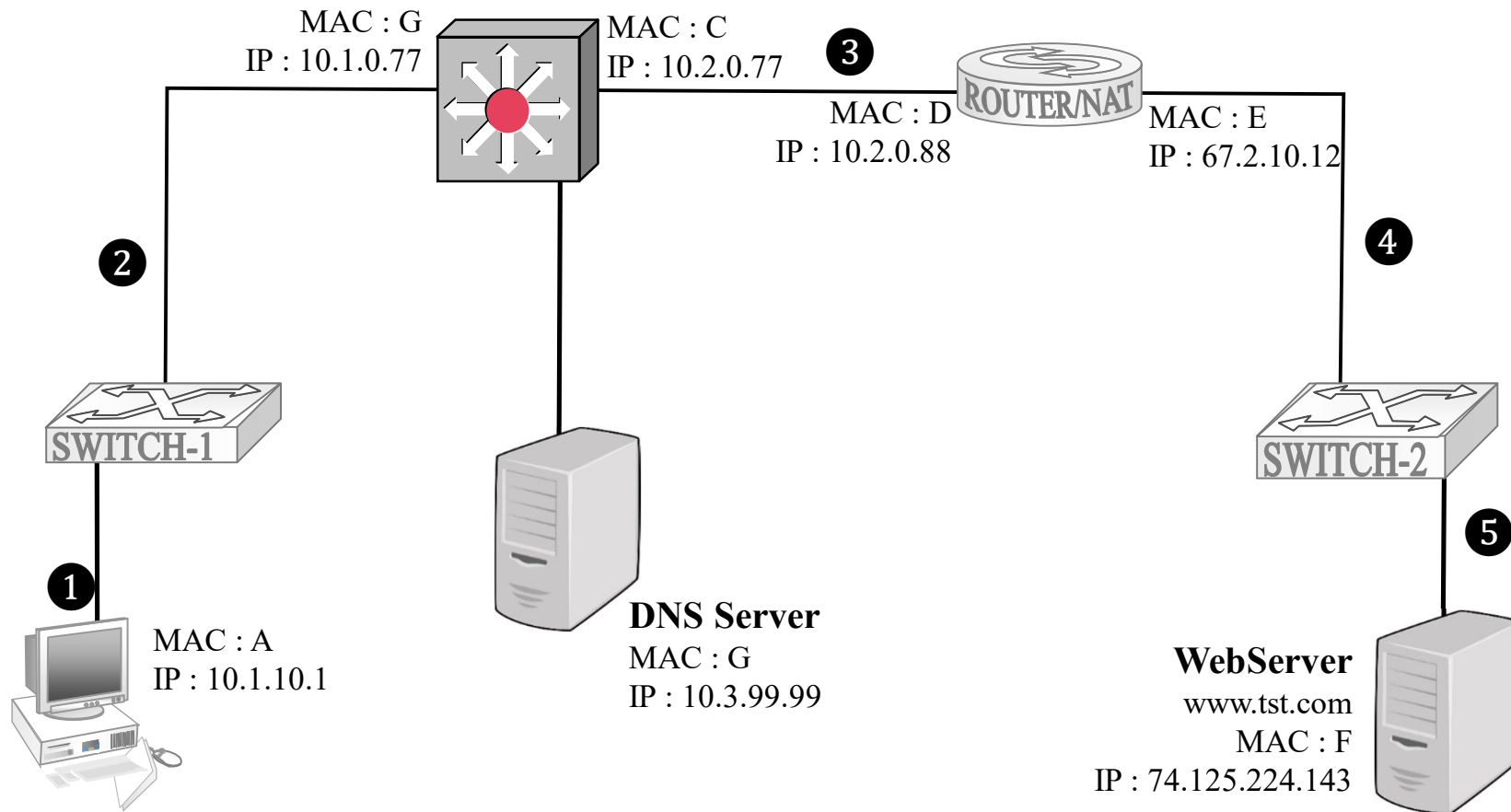
0.5 프레임과 패킷 구분

- ▼ Frame 603: 288 bytes on wire (2304 bits), 288 bytes captured (2304 bits) on interface 0
 - > Interface id: 0 (\Device\NPF_{557124E1-28F6-4C2F-BDCE-1DFD75D011CE})
 - Encapsulation type: Ethernet (1)
 - Arrival Time: Jul 28, 2019 16:53:46.332270000 대한민국 표준시
 - [Time shift for this packet: 0.000000000 seconds]
 - Epoch Time: 1564300426.332270000 seconds
 - [Time delta from previous captured frame: 0.000069000 seconds]
 - [Time delta from previous displayed frame: 0.000000000 seconds]
 - [Time since reference or first frame: 81.608426000 seconds]
 - Frame Number: 603
 - Frame Length: 288 bytes (2304 bits)
 - Capture Length: 288 bytes (2304 bits)
 - [Frame is marked: False]
 - [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_51:0b:53 (68:ec:c5:51:0b:53), Dst: Mercury_15:eb:de (b4:a9:4f:15:eb:de)
- > Internet Protocol Version 4, Src: 172.30.1.15, Dst: 117.18.237.29
- > Transmission Control Protocol
- > Hypertext Transfer Protocol

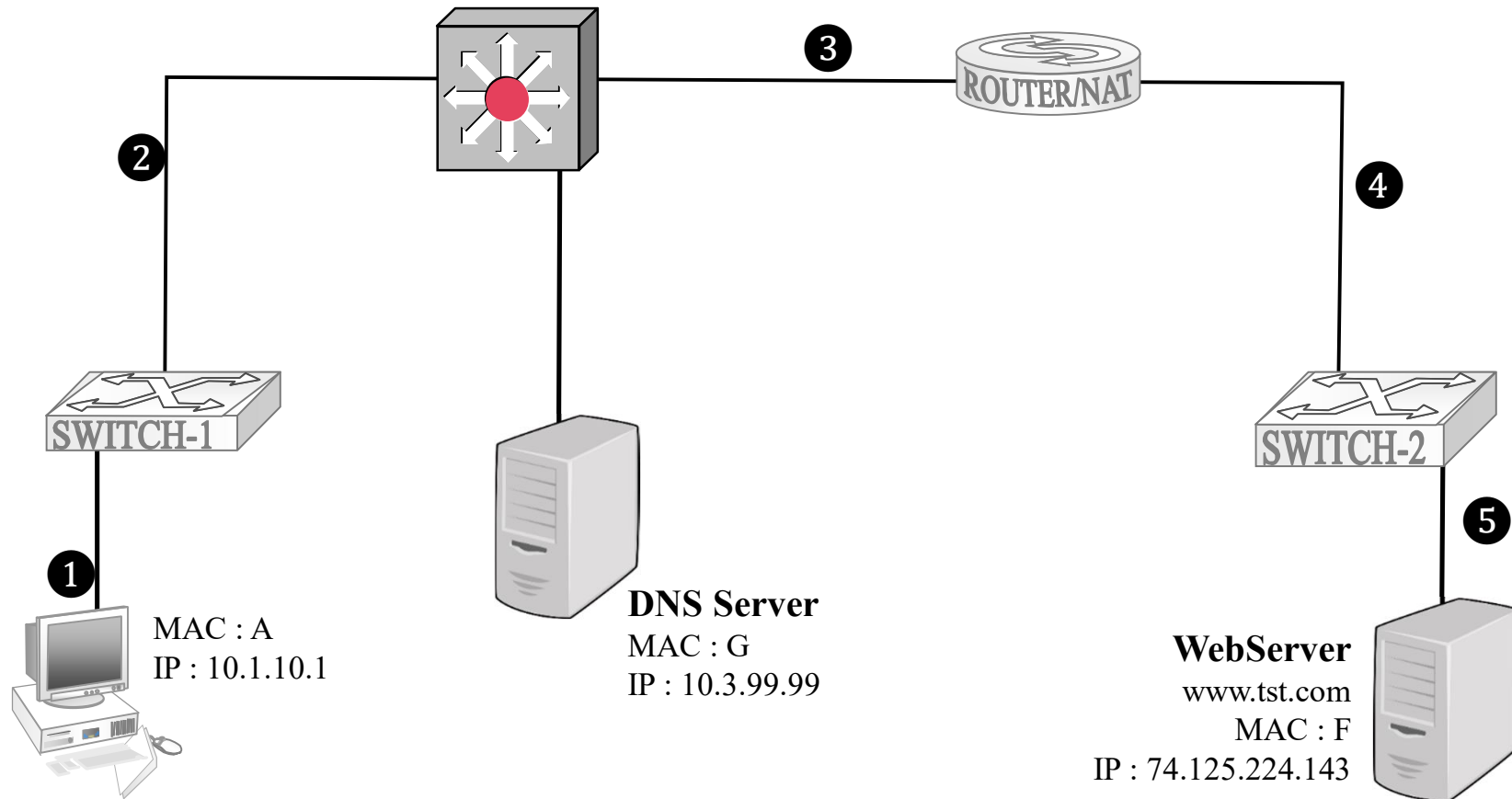
프레임 섹션
: 와이어샤크의 메타 데이터가 들어 있음

* 메타 데이터 : 데이터에 대한 각종 정보(자원의 속성)를 담고있는 데이터

0.6 네트워크를 지나가는 HTTP 패킷 따라가기



0.6 네트워크를 지나가는 HTTP 패킷 따라가기



실습 1. 네트워크 구성도를 완성하려면 패킷을 이용하라.

추적파일 : general101.pcapng

웹 브라우징 트래픽 분석

추적파일 : [http-google101.pcapng](#)

백그라운드 트래픽 분석

.백그라운드 트래픽은 자동화된 프로세스들이 구동 될 때 생성

- 자바 업데이트 시
- 바이러스 탐지 도구 업데이트 시
- 드롭박스가 체크인 시

추적파일 : mybackground.pcapng

실습 2. 사용자 자신의 백그라운드 트래픽 수집과 분류

.자신의 백그라운드 트래픽을 인식한다는 것은 비정상 통신을 조사할 때 고려 사항을

제거하는데 도움이 됨

.문제 해결을 할 때 참조할 수 있게 '정상' 트래픽을 추적파일로 저장

도전과제 (P.94)

추적파일 : challenge101-0.pcapng

1장. 와이어샹크 뷰와 설정 맞춤화

1.1 패킷 목록 창에 칼럼(열) 추가

① 컬럼(열) 추가

- Packet Detail > 특정 프로토콜 필드 선택 > 오른쪽 클릭 > Apply as Column(열로서 적용)
- Edit(편집) > Preference(설정) > 모양 > Columns

② 열 숨김, 삭제, 재배열, 편집

③ 열 내용 정렬

④ 열 데이터 내보내기

파일 > Export Packet Dissections(패킷 분해 결과 내보내기) > AS CSV

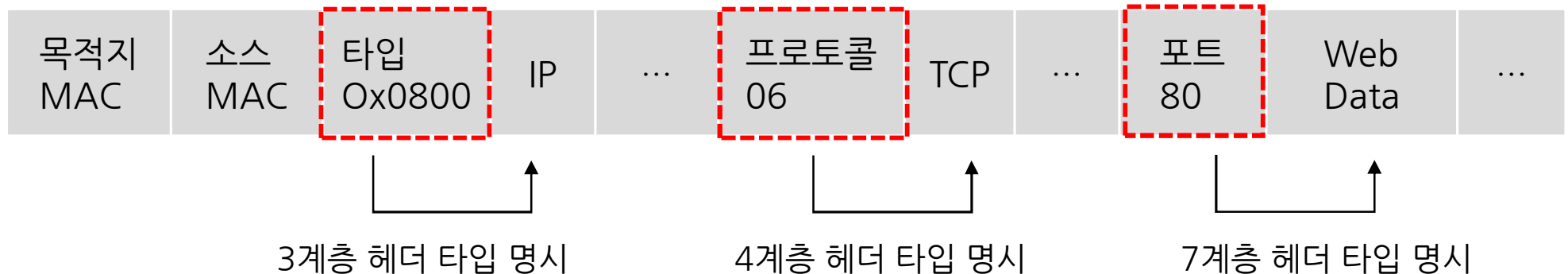
[실습 4] 열에 HTTP 호스트 추가 (P.102)

[실습 30] 추적 파일에서 HTTP Host 필드 값 목록 내보내기(P.291)

1.2 와이어샤크 해석기 해부

<http://chappellu101.pcapng>

- 와이어샤크는 수 많은 패킷 해석기(packet dissect) 를 가지고 있음
- 수집엔진이나 Wiretap 라이브러리에서 코어엔진으로 전해진 패킷들을 기반으로 해석
- 프로토콜 필드 별 해석기를 호출
- SAP 내용을 기반으로 필드를 해석



1.2 와이어샤크 해석기 해부

- 대표적인 해석기
 - 프레임 해석기
 - * 타임스탬프 집합과 같은 추적파일의 기본적인 정보를 조사해서 보여줌
 - 이더넷 해석기
 - IPv4 해석기
 - TCP 해석기
 - HTTP 해석기

1.3 비표준 포트 번호를 사용하는 트래픽 분석

- 해석기 적용 방법 2가지
 - 정적 방법 : 헤더를 검사해 사용할 논리 해석기를 결정
 - 경험적 방법 : 사용할 해석기를 추측해서 결정
- 비표준 포트번호를 사용 시
 - 잘못된 해석기를 적용해서 분석(정적방법)
 - 적절한 해석기를 적용해서 분석(경험적 방법을 사용)
 - 임의의 해석기를 적용 (정적 또는 경험적 방법 모두 적합한 해석기를 결정하지 못한 경우)

1.3 비표준 포트 번호를 사용하는 트래픽 분석

tcp-decodeas.pcapng

Source	Destination	Protocol	Length	Info
207.137.7.104	207.137.7.103	TCP	66	1284 → 137 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1 [ETHERNET FRAME CHECK SEQUEN
207.137.7.103	207.137.7.104	TCP	66	137 → 1284 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460 SACK_PERM=1 [ETHERNET FRAME C
207.137.7.104	207.137.7.103	TCP	64	1284 → 137 [ACK] Seq=1 Ack=1 Win=17520 Len=0 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
207.137.7.103	207.137.7.104	TCP	126	137 → 1284 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=68 [ETHERNET FRAME CHECK SEQUENCE INCORR
207.137.7.104	207.137.7.103	TCP	69	1284 → 137 [PSH, ACK] Seq=1 Ack=69 Win=17452 Len=11 [ETHERNET FRAME CHECK SEQUENCE INCOR
207.137.7.103	207.137.7.104	TCP	93	137 → 1284 [PSH, ACK] Seq=69 Ack=12 Win=17509 Len=35 [ETHERNET FRAME CHECK SEQUENCE INCO
207.137.7.104	207.137.7.103	TCP	72	1284 → 137 [PSH, ACK] Seq=12 Ack=104 Win=17417 Len=14 [ETHERNET FRAME CHECK SEQUENCE INC
207.137.7.103	207.137.7.104	TCP	91	137 → 1284 [PSH, ACK] Seq=104 Ack=26 Win=17495 Len=33 [ETHERNET FRAME CHECK SEQUENCE INC

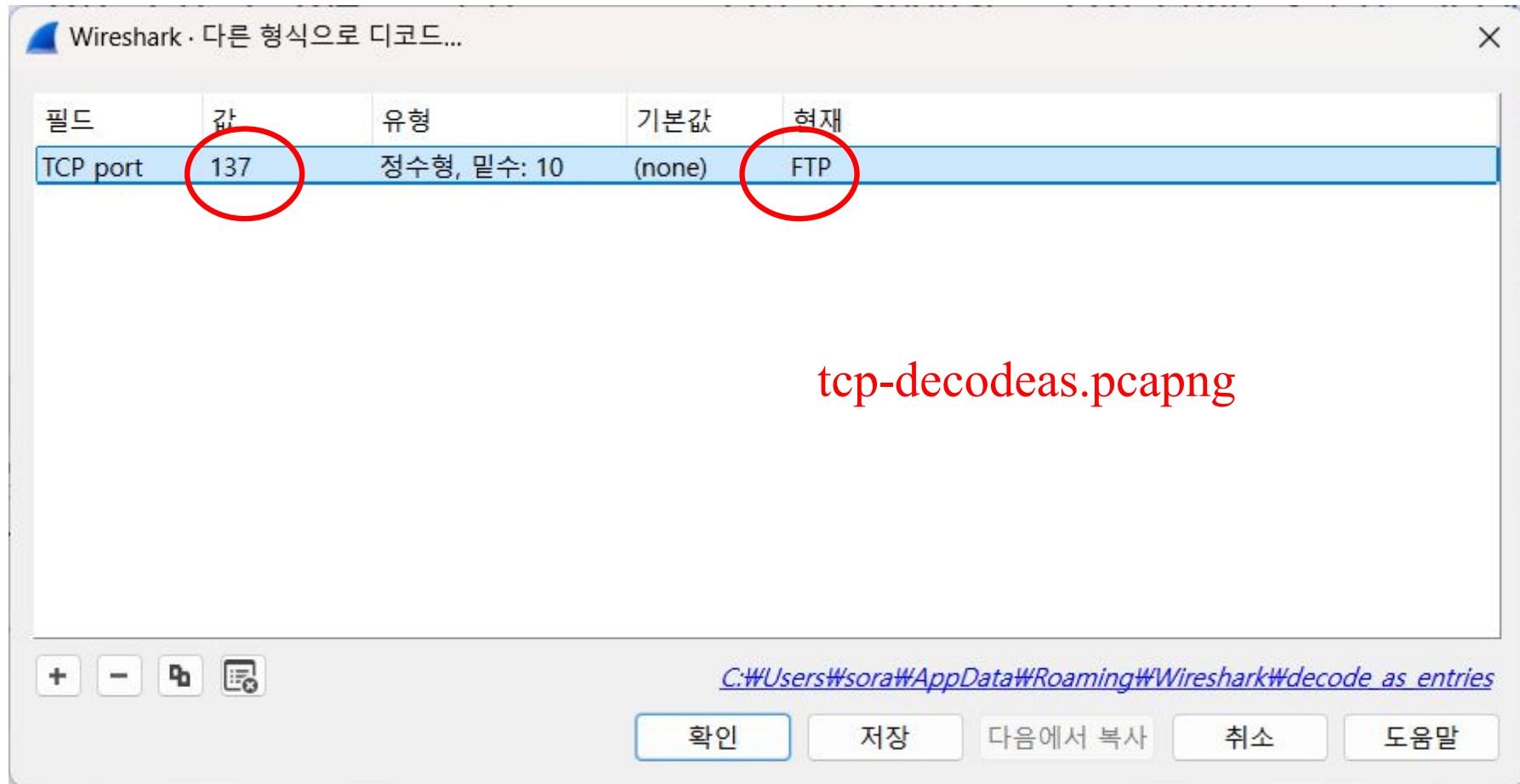
포트 번호 137을 통해 FTP 통신 진행

- 트래픽이 NetBIOS 서비스 트래픽 동작과 일치하지 않음
- 와이어샤크는 TCP 해석 후 트래픽을 더 이상 해석하지 않음

* 애플리케이션에 적용할 적절한 해석지를 확인 할 수 없는 경우 TCP에서 해석을 중단

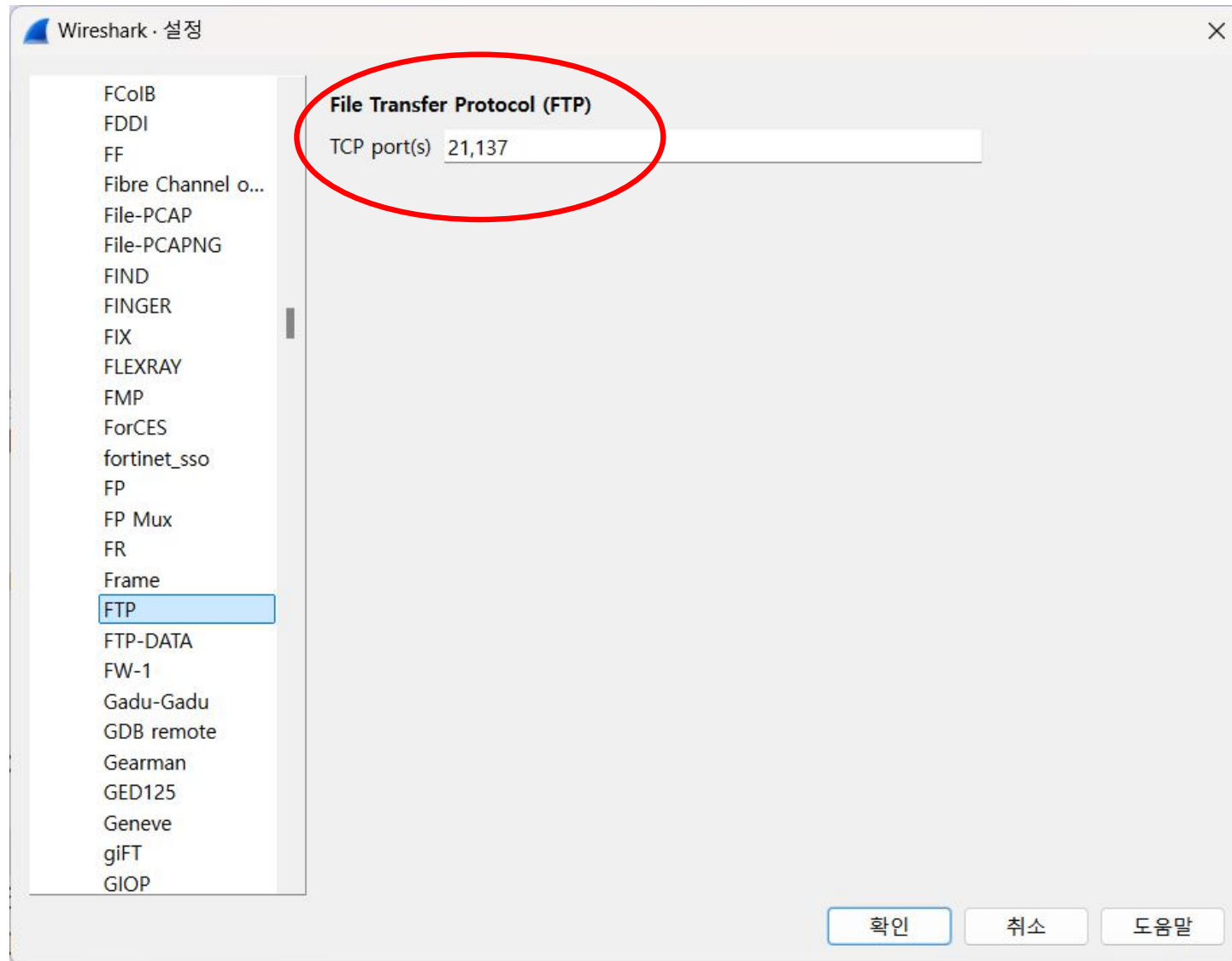
1) 수동으로 해석기 강제 적용

Analyze(분석) > Decode As (다른 형식으로 디코드)



2) 애플리케이션 선호도 설정을 이용한 해석 조절

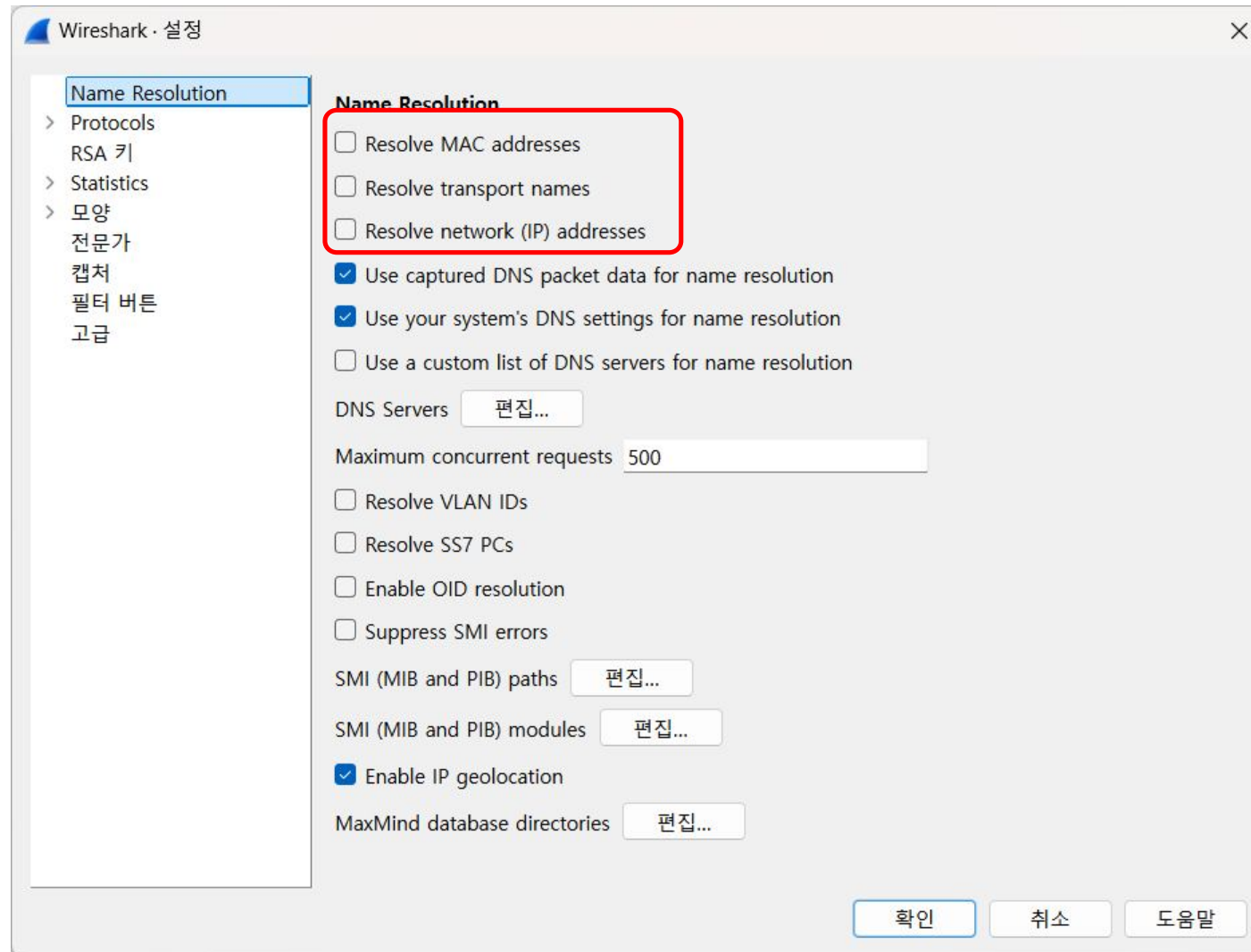
Edit(편집) > Preferences(설정) > Protocol > FTP



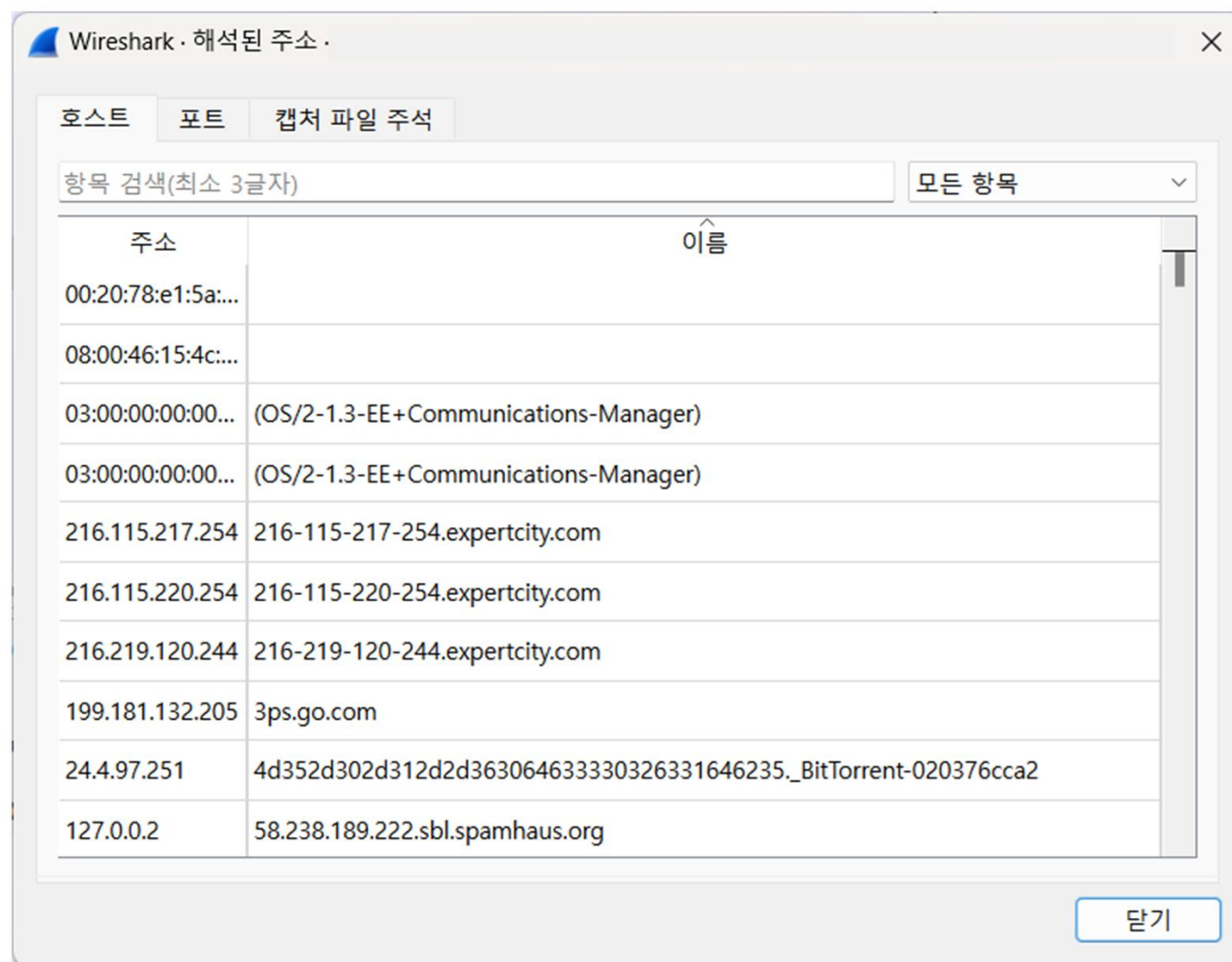
1.4 특정 트래픽 유형을 디스플레이하는 방법 변경

- Filter Expression 버튼 지정

Edit(편집) > Preference(설정) > Name Resolution



[참조] 통계 > 해석된 주소



Wireshark - 해석된 주소

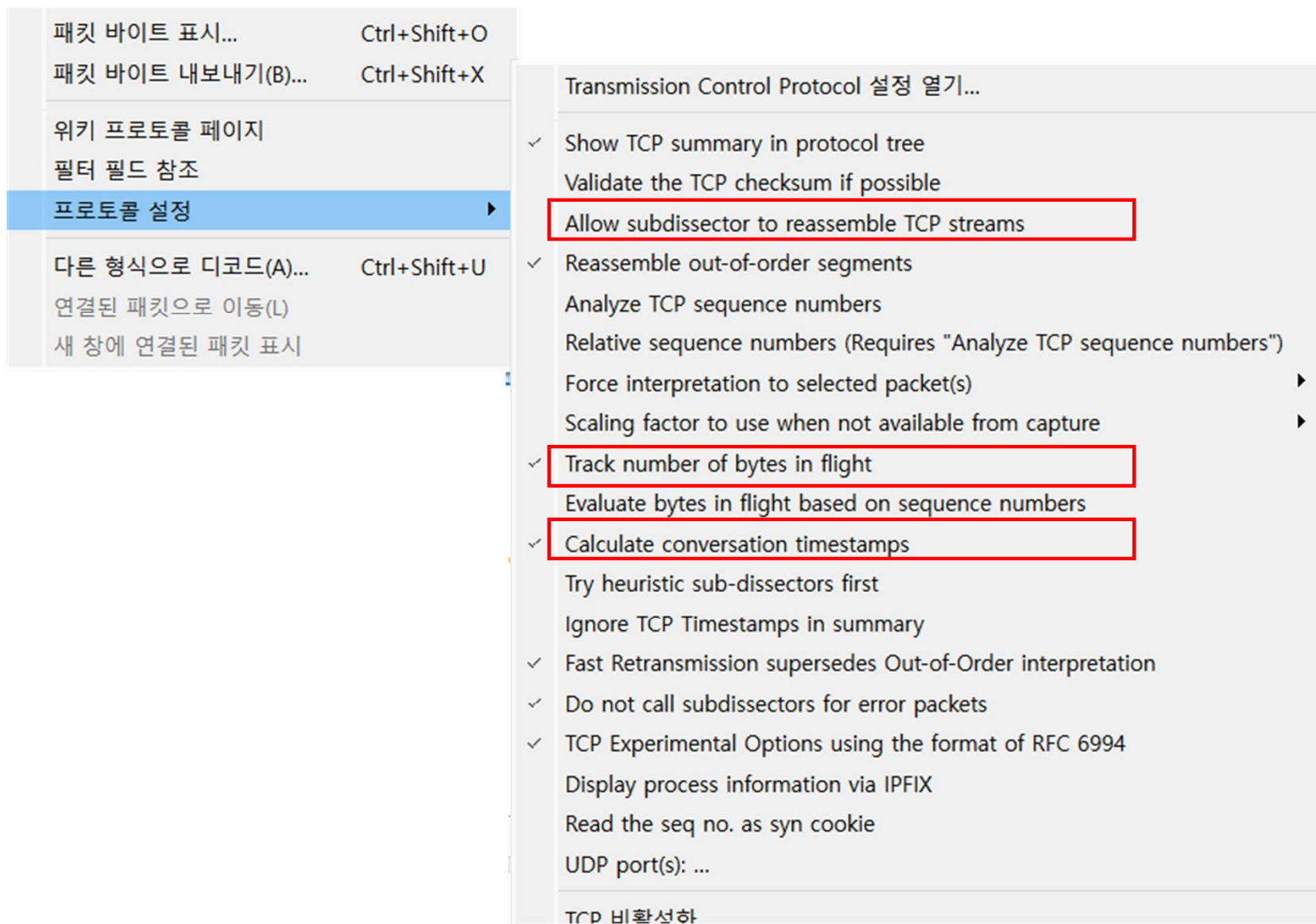
호스트 포트 캡처 파일 주소

항목 검색(최소 3글자) 모든 항목

주소	이름
00:20:78:e1:5a:...	
08:00:46:15:4c:...	
03:00:00:00:00:...	(OS/2-1.3-EE+Communications-Manager)
03:00:00:00:00:...	(OS/2-1.3-EE+Communications-Manager)
216.115.217.254	216-115-217-254.expertcity.com
216.115.220.254	216-115-220-254.expertcity.com
216.219.120.244	216-219-120-244.expertcity.com
199.181.132.205	3ps.go.com
24.4.97.251	4d352d302d312d2d363064633330326331646235_BitTorrent-020376cca2
127.0.0.2	58.238.189.222.sbl.spamhaus.org

닫기

프로토콜과 애플리케이션 설정 지정



① Allow subdissector to reassemble TCP streams(재조립 비활성화 시)

8	0.269148	24.6.173.220	198.66.239.146	HTTP	345 GET / HTTP/1.1
9	0.308429	198.66.239.146	24.6.173.220	HTTP	1514 HTTP/1.1 200 OK (text/html)
10	0.309975	198.66.239.146	24.6.173.220	HTTP	1514 Continuation
11	0.309982	198.66.239.146	24.6.173.220	HTTP	1514 Continuation
12	0.309986	198.66.239.146	24.6.173.220	HTTP	1514 Continuation
13	0.310666	24.6.173.220	198.66.239.146	TCP	54 50418 → 80 [ACK] Seq=292 Ack=5841 Win=
14	0.326514	24.6.173.220	198.66.239.146	TCP	66 50419 → 80 [SYN] Seq=0 Win=8192 Len=0
15	0.329102	24.6.173.220	198.66.239.146	TCP	66 50420 → 80 [SYN] Seq=0 Win=8192 Len=0
16	0.329744	24.6.173.220	198.66.239.146	TCP	66 50421 → 80 [SYN] Seq=0 Win=8192 Len=0
17	0.329989	24.6.173.220	198.66.239.146	TCP	66 50422 → 80 [SYN] Seq=0 Win=8192 Len=0
18	0.330160	24.6.173.220	198.66.239.146	TCP	66 50423 → 80 [SYN] Seq=0 Win=8192 Len=0

② Allow subdissector to reassemble TCP streams(재조립 활성화 시)

8	0.269148	24.6.173.220	198.66.239.146	HTTP	345 GET / HTTP/1.1
9	0.308429	198.66.239.146	24.6.173.220	TCP	1514 80 → 50418 [ACK] Seq=1 Ack=292 Win=65700 Len=1460 [TCP segment of a reassembled PDU]
10	0.309975	198.66.239.146	24.6.173.220	TCP	1514 80 → 50418 [ACK] Seq=1461 Ack=292 Win=65700 Len=1460 [TCP segment of a reassembled PDU]
11	0.309982	198.66.239.146	24.6.173.220	TCP	1514 80 → 50418 [ACK] Seq=2921 Ack=292 Win=65700 Len=1460 [TCP segment of a reassembled PDU]
12	0.309986	198.66.239.146	24.6.173.220	TCP	1514 80 → 50418 [ACK] Seq=4381 Ack=292 Win=65700 Len=1460 [TCP segment of a reassembled PDU]
13	0.310666	24.6.173.220	198.66.239.146	TCP	54 50418 → 80 [ACK] Seq=292 Ack=5841 Win=65700 Len=0
14	0.326514	24.6.173.220	198.66.239.146	TCP	66 50419 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
15	0.329102	24.6.173.220	198.66.239.146	TCP	66 50420 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
16	0.329744	24.6.173.220	198.66.239.146	TCP	66 50421 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
17	0.329989	24.6.173.220	198.66.239.146	TCP	66 50422 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
18	0.330160	24.6.173.220	198.66.239.146	TCP	66 50423 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1

① Track number of byte in flight 비활성화 시

▼ [SEQ/ACK analysis]

[\[This is an ACK to the segment in frame: 8\]](#)

[The RTT to ACK the segment was: 0.039281000 seconds]

[iRTT: 0.028051000 seconds]

② Track number of byte in flight 활성화 시

▼ [SEQ/ACK analysis]

[\[This is an ACK to the segment in frame: 8\]](#)

[The RTT to ACK the segment was: 0.039281000 seconds]

[iRTT: 0.028051000 seconds]

[Bytes in flight: 1460]

[Bytes sent since last PSH flag: 1460]

* TCP 통신에서 현재 확인 응답되지 않은 데이터가 얼마나 되는지 보여 주는 설정

① Calculate conversation timestamps 비활성화 시

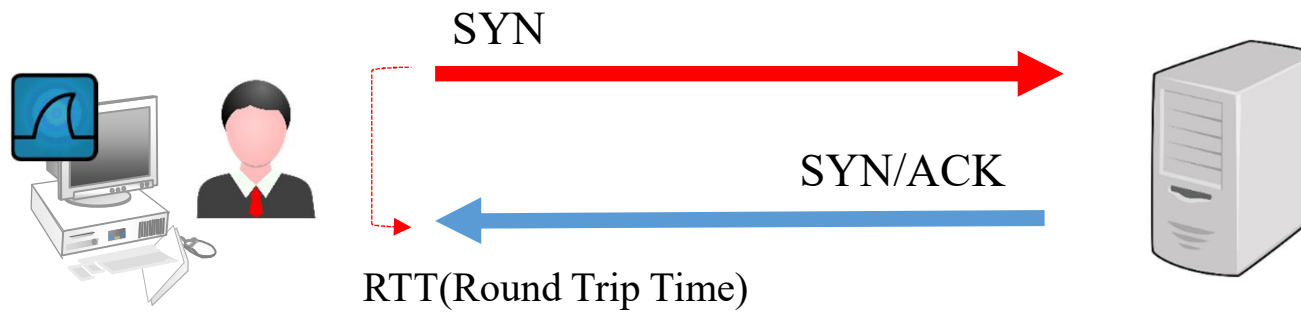
```
> [SEQ/ACK analysis]
  TCP payload (1460 bytes)
  \[Reassembled PDU in frame: 20\]
  TCP segment data (1460 bytes)
```

② Calculate conversation timestamps 활성화 시

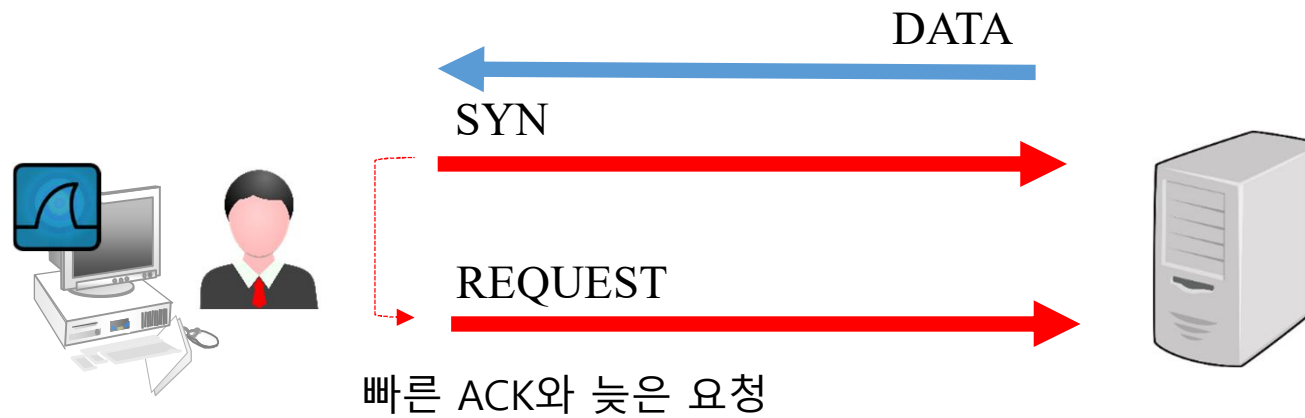
```
> [SEQ/ACK analysis]
✓ [Timestamps]
  [Time since first frame in this TCP stream: 0.068128000 seconds]
  [Time since previous frame in this TCP stream: 0.039281000 seconds]
  TCP payload (1460 bytes)
  \[Reassembled PDU in frame: 20\]
  TCP segment data (1460 bytes)
```


[실습 5] 핵심 와이어샤크 선호도 설정하기(P.116)

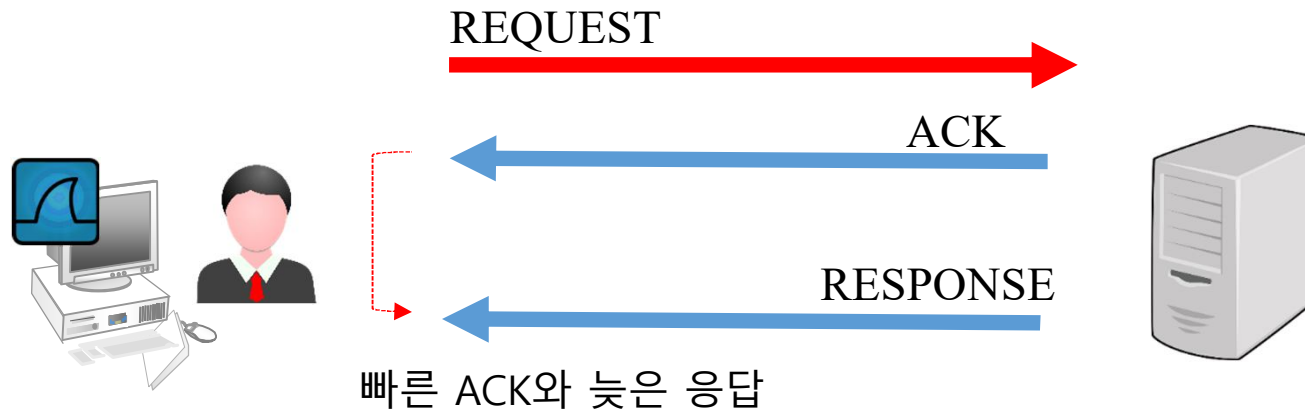
1.7 Time 열을 구성해 지연 문제점 찾아 내기



클라이언트 전달 지연 표시와 원인



서버 전달 지연 표시와 원인



Delta Time

① Time 열 = (현재 패킷이 도착한 시간) - (이전 패킷이 도착한 시간)

② TCP Delta 열

= (현재 패킷이 속한 TCP Stream 내에서 현재 패킷이 도착 시간) - (이전 패킷이 도착한 시간)

* 현재 TCP Stream의 첫 패킷은 TCP Delta 값이 무조건 0

No.	Time
1	0.000000
2	0.878530
3	0.895134
4	0.895469
5	2.050697
6	2.066563

[Second Since Beginning of Capture]

No.	Time	Source
1	0.000000	172.16.16.128
2	0.878530	74.125.95.104
3	0.016604	172.16.16.128
4	0.000335	172.16.16.128
5	1.155228	74.125.95.104
6	0.015866	74.125.95.104

ip.src == 172.16.16.128		
No.	Time	Source
1	0.000000	172.16.16.128
3	0.016604	172.16.16.128
4	0.000335	172.16.16.128

[Second Since Previous Captured Packet]

No.	Time	Source
1	0.000000	172.16.16.128
2	0.878530	74.125.95.104
3	0.016604	172.16.16.128
4	0.000335	172.16.16.128
5	1.155228	74.125.95.104
6	0.015866	74.125.95.104

ip.src == 172.16.16.128		
No.	Time	Source
1	0.000000	172.16.16.128
3	0.895134	172.16.16.128
4	0.000335	172.16.16.128

[Second Since Previous Displayed Packet]

[실습 8] 경로와 서버 전달 지연 문제에 집중하라(P.141)

[도전과제] Challenge101-1.pcapng

필터 사용

- **캡처 필터(Capture Filter)**

- 패킷이 캡처될 때 지정
- 지정된 표현식에 포함/제외된 패킷만 캡처

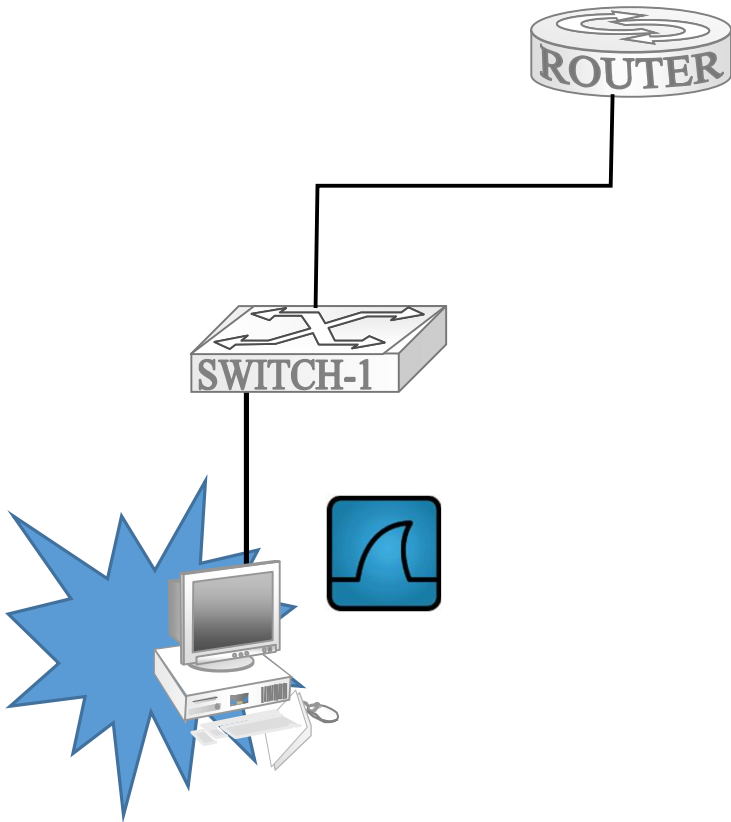
- **디스플레이 필터(Display Filter)**

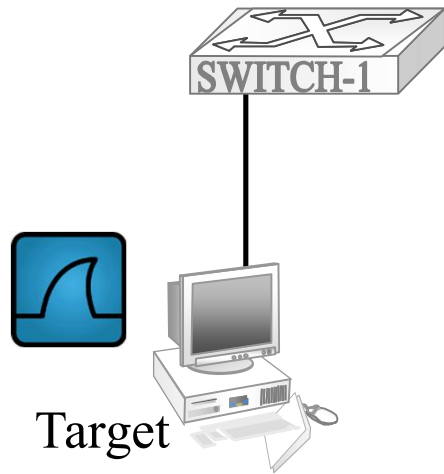
- 원하지 않는 패킷을 숨김
- 지정된 표현식을 기반으로 원하는 패킷을 보기

2장. 최선의 수집 방법 결정과 수집 필터 적용

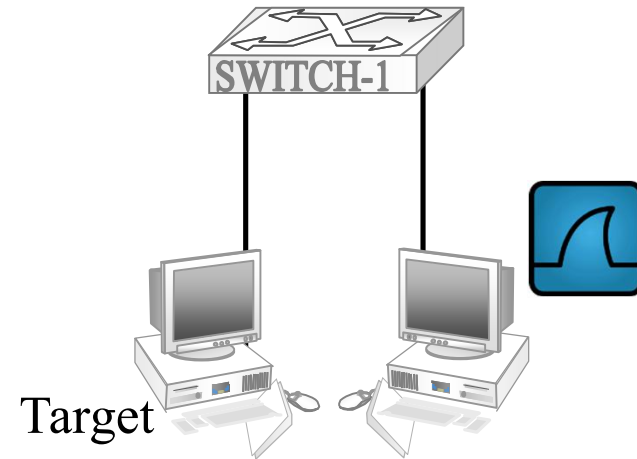
2.1 수집 위치 확인

- 잘못된 지점에 와이어샹크를 위치 시
 - 여러 시간 동안 관련 없는 트래픽 처리
 - False positive(오탐지)를 처리하는데 많은 시간 소비
- 이상적인 시작 포인트
 - 불만이 있거나 의심스러운 호스트에 최대한 가까운 곳에서 수집을 시작
 - 수집 옵션 3가지
 - ① 지연이 발생한 호스트에서 직접 수집
 - ② 호스트의 스위치 포트를 확장(포트미러링)
 - ③ TAP 설정

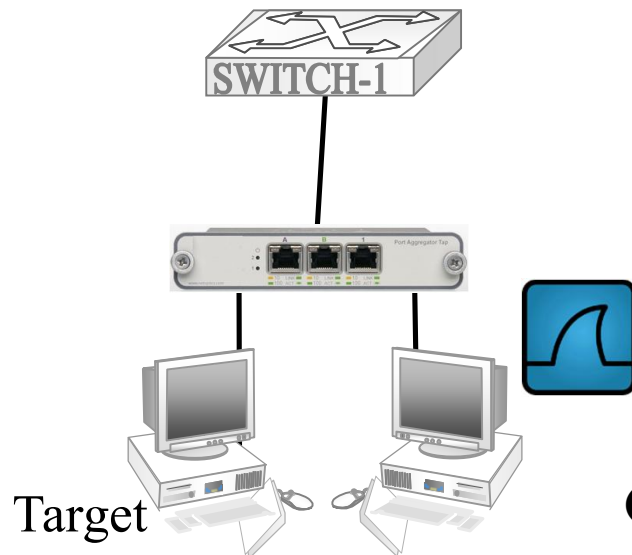




❶ 지연이 발생한 호스트에서 직접 수집



❷ 호스트의 스위치 포트를 확장(포트미러링)



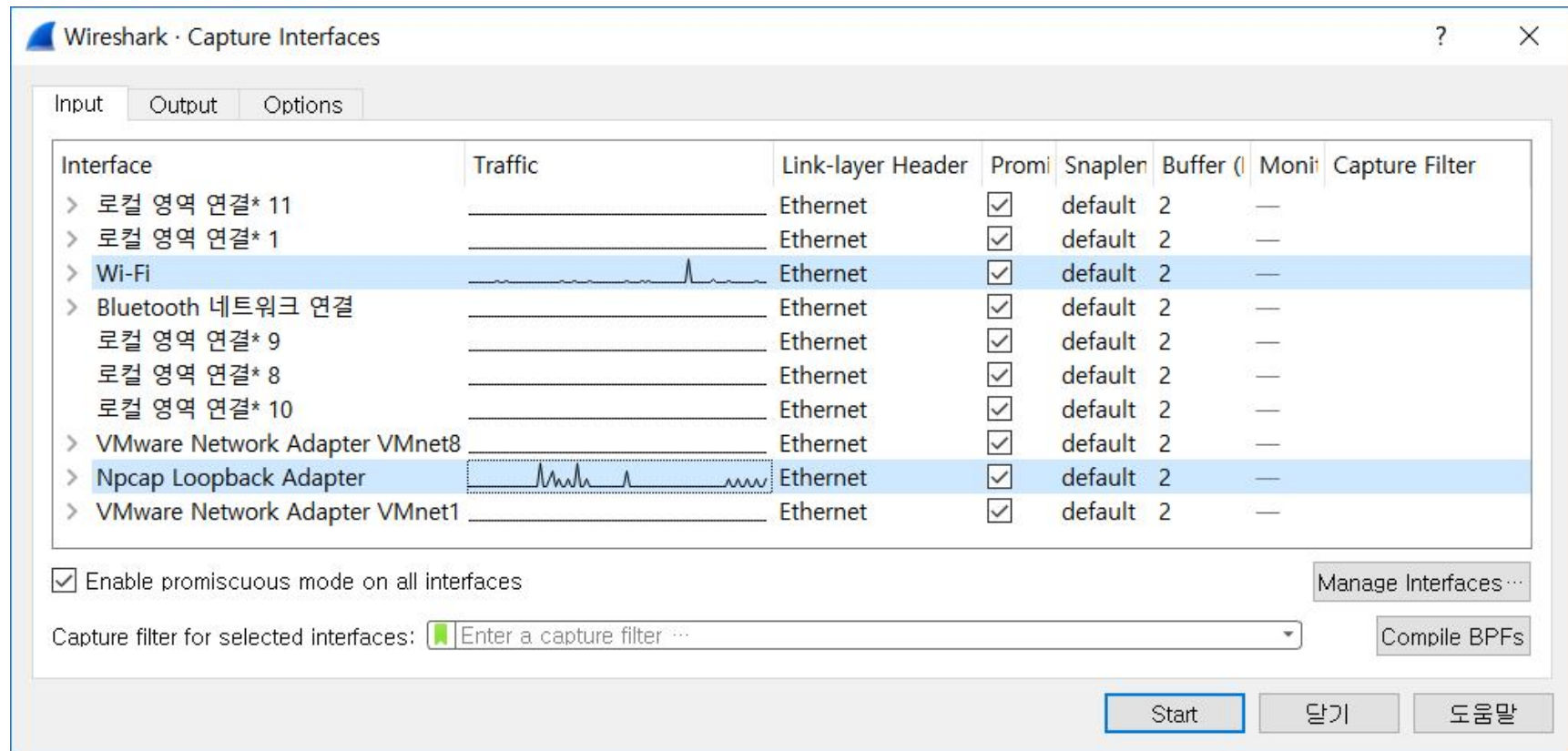
❸ TAP 설정

2.3 무선 네트워크 트래픽 수집

- 스파크 라인을 검사해서 무선 어댑터 확인
- AirPcap 어댑터 사용
- WLAN/Loopback 가시성을 위한 Npcap 드라이버 사용

2.4 동작 중인 인터페이스 파악

- Capture Option 버튼 클릭 또는 스파크 라인 파악
- 복수 어댑터 트래픽 수집 : Ctrl + Click

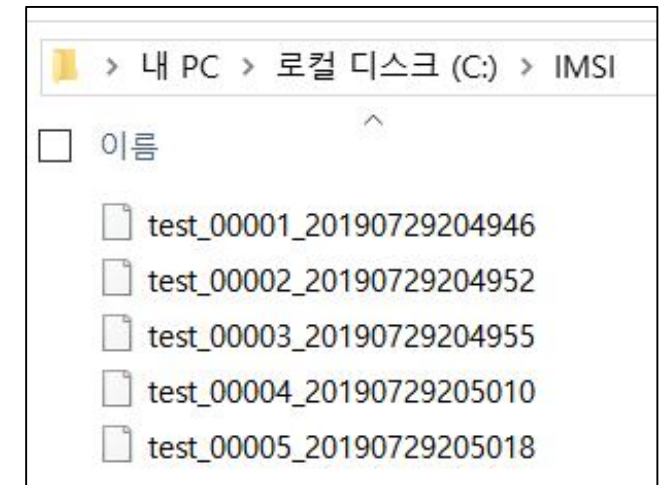
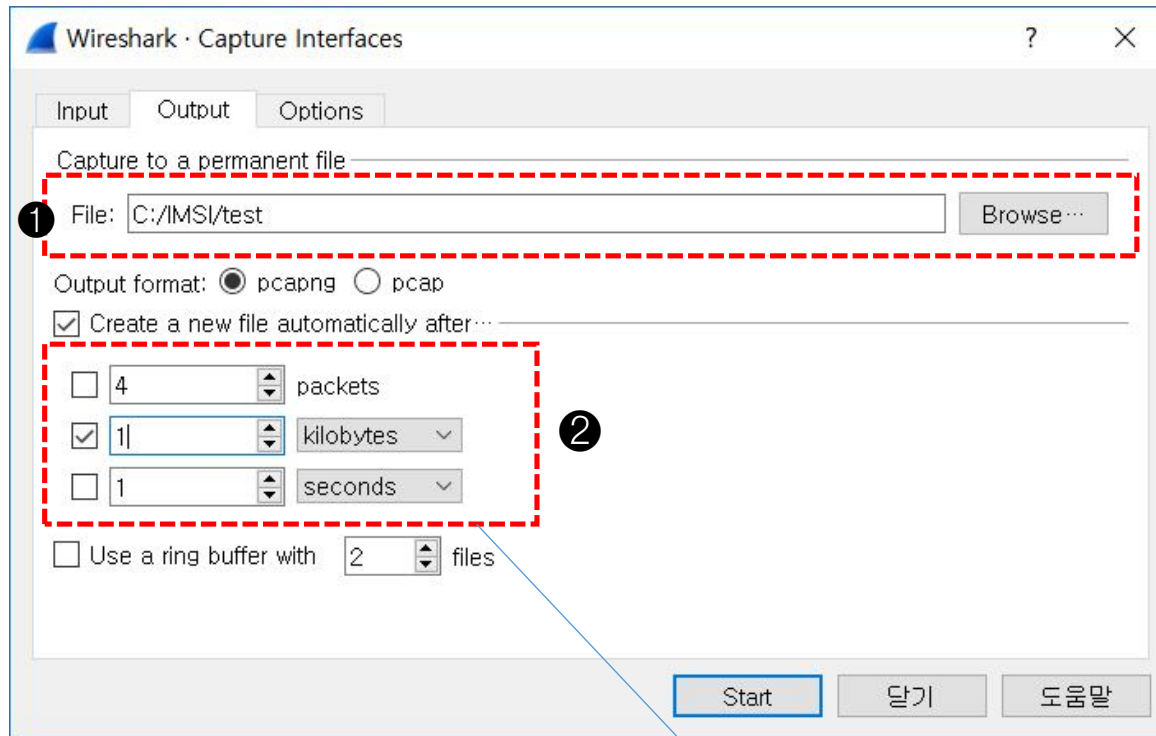


2.5 대량의 트래픽 처리

- 파일 집합으로 수집
- 링 버퍼 사용

파일 집합으로 수집 (p.158)

- Capture(캡처) > Options(옵션) > Output Tab(출력탭)



- File > Open
 - File > File set > List Files

파일당 4개의 패킷
1MB 파일 크기
1초마다

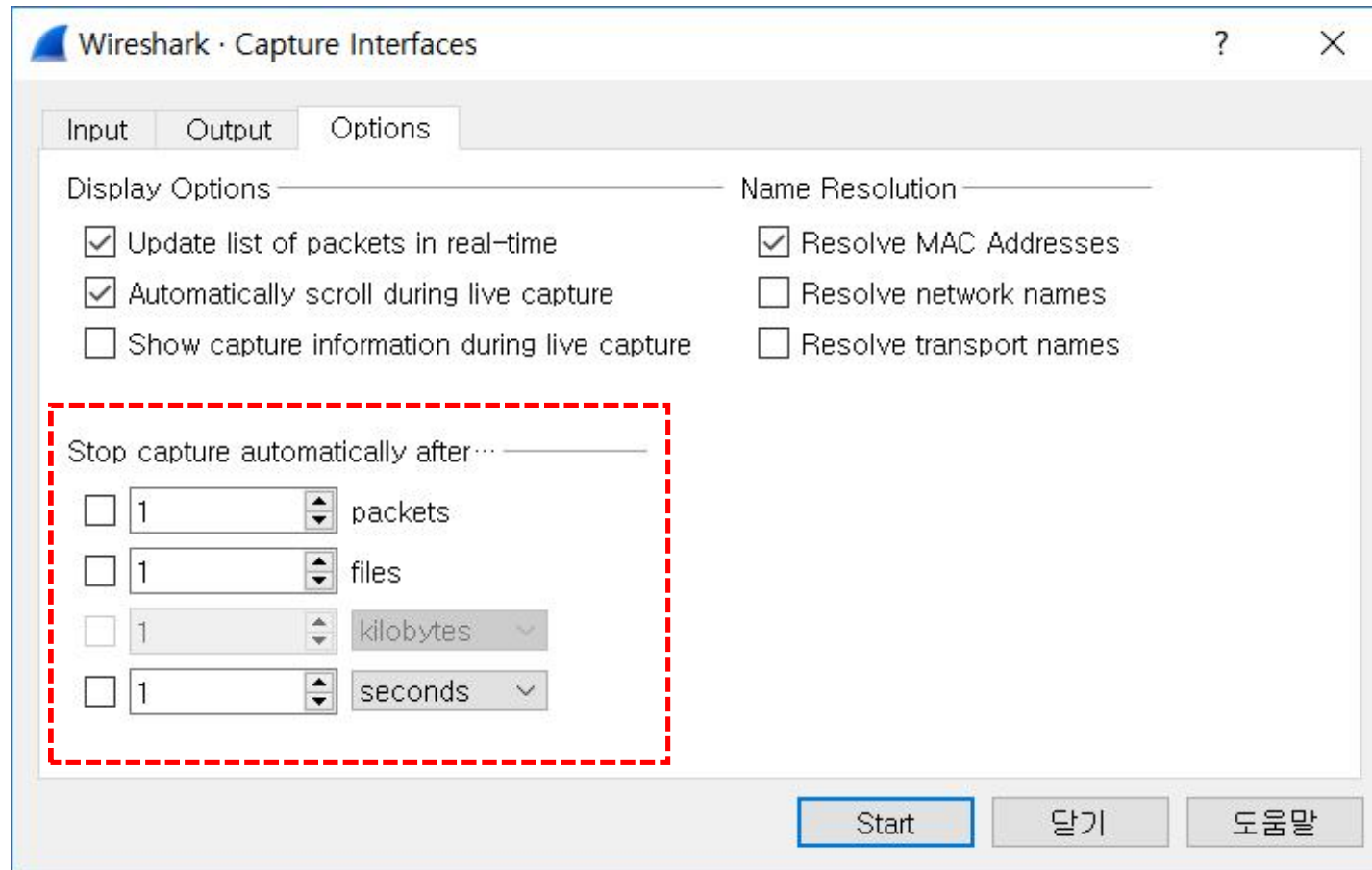
위의 어느 조건이든
먼저 만나면 파일 생성

[실습 9] 파일 집합으로 수집 (P.161)

[실습10] 링 버퍼를 사용해 드라이브 공간 절약 (p.167)

파일 집합으로 수집 (p.158)

- Capture Options > Option Tab > Stop capture automatically after....



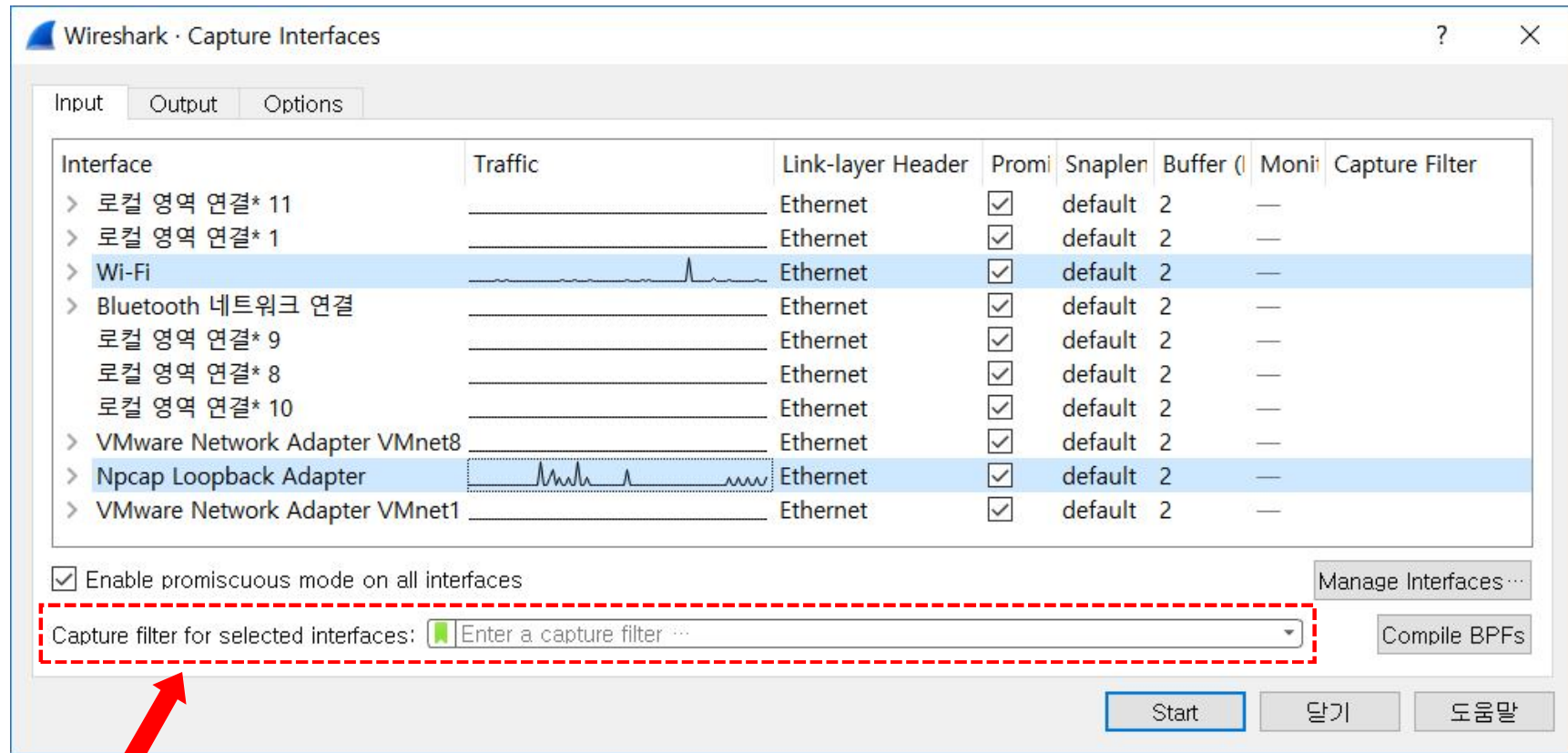
2.7 수집 필터를 이용한 트래픽 용량 줄이기

- 수집 필터는 대량의 데이터를 처리하기 위한 것
- 수집해야 할 패킷 수를 줄이며 와이어샹크의 부하를 줄임
 - 트래픽이 dumpcap로부터 가져오는 것보다 빠르면 문제 발생
- 수집 필터 문법
 - BPF(Berkeley Packet Filtering) 문법을 사용
 - dumpcap에서 지원하는 형식

BPF 구문 표현식(Expression)

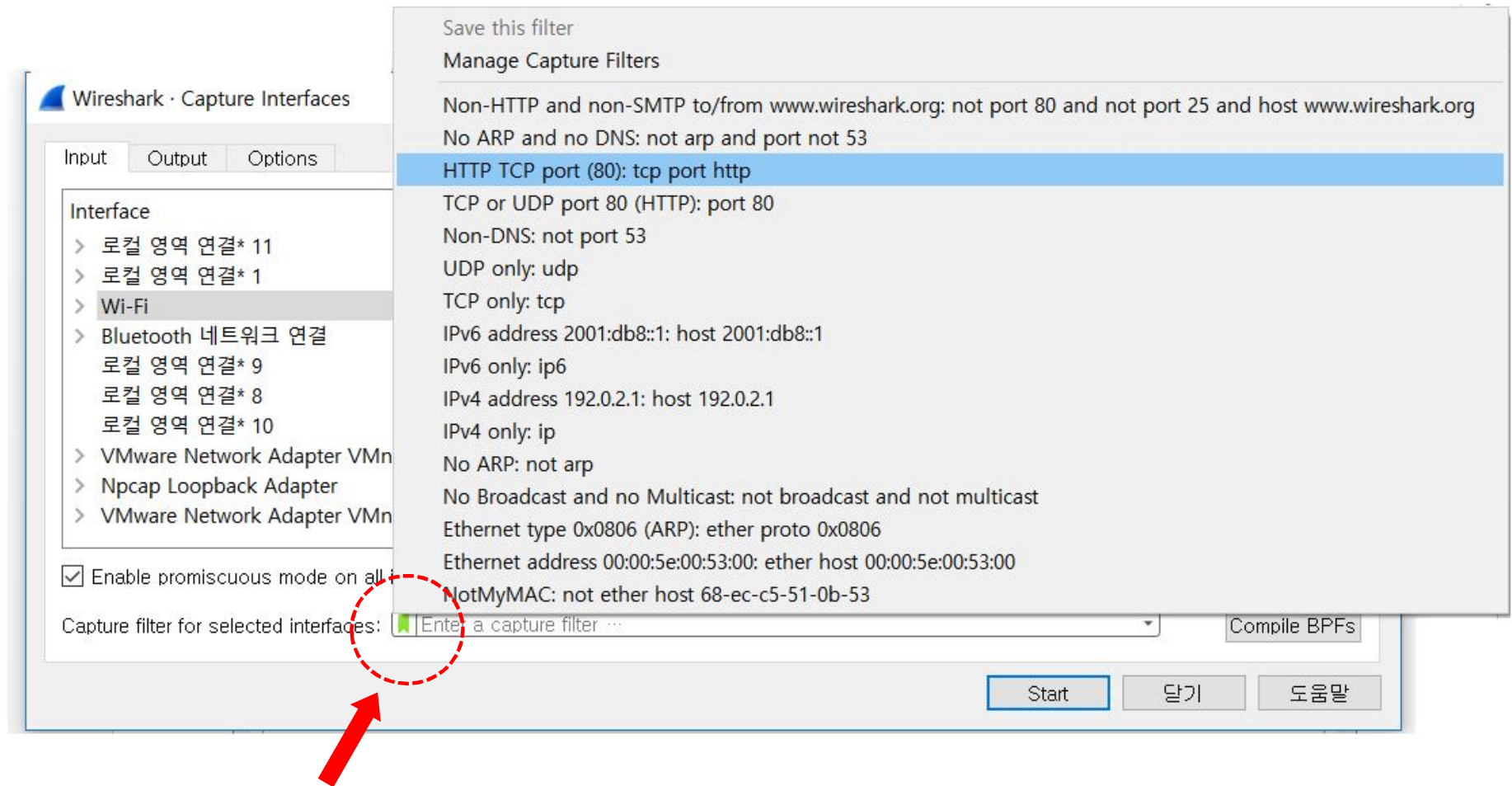
Primitive			연산자	Primitive		
<u>dst</u>	<u>host</u>	<u>192.168.0 10</u>	&&	<u>tcp</u>	<u>port</u>	<u>80</u>
한정자	한정자	ID		한정자	한정자	ID

Capture Option 창에서 수집 필터 적용



Capture Option 창에서 수집 필터 적용

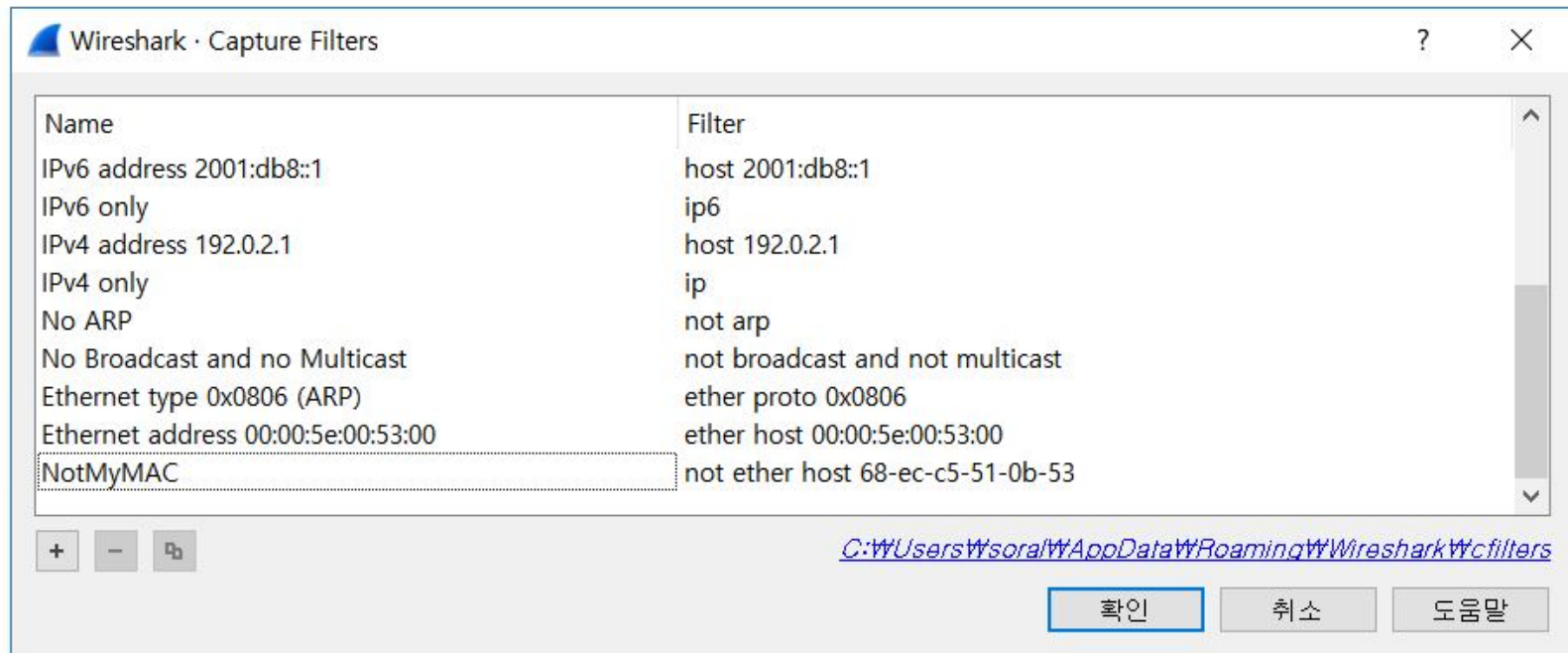
① Capture Option(캡처옵션) > 수집 필터 체크할피 화살표



수집 필터 체크할피 화살표

Capture Option 창에서 수집 필터 적용

② 주메뉴 > Capture(캡처) > Capture Filters(캡처필터)



2.8 주소 기반의 트래픽 수집

1) 특정 IP 주소에서/로 오는 트래픽 수집

- host 10.3.1.1
- host 2406:da00:ff00::6b16:f02d
- not host 10.3.1.1
- src host 10.3.1.1
- dst host 10.3.1.1
- host 10.3.1.1 or host 10.3.1.2
- host www.espn.com

2) IP 주소 범위에서/로 오는 트래픽 수집

- net 10.3.0.0/16
- net 10.3.0.0 mask 255.255.0.0
- ipv6 net 2406:da00:ff00::/64
- not dst net 10.3.0.0/16
- dst net 10.3.0.0/16
- src net 10.3.0.0/16

3) 브로드캐스트 또는 멀티캐스트 트래픽 수집

- ip broadcast
- ip multicast
- dst host ff02::1
- dst host ff02::2

4) MAC 주소 기반의 트래픽 수집

- ether host 00:08:15:00:08:15
- ether src 00:08:15:00:08:15
- ether dst 00:08:15:00:08:15
- not ether host 00:08:15:00:08:15

[참고] TCP Flag 관련 트래픽 수집

32	16	8	4	2	1
u	a	p	r	s	f
r	c	s	s	y	i
g	k	h	t	n	n

- $\text{tcp}[13] \& 32 == 32$
- $\text{tcp}[13] \& 16 == 16$
- $\text{tcp}[13] \& 8 == 8$
- $\text{tcp}[13] \& 4 == 4$
- $\text{tcp}[13] \& 2 == 2$
- $\text{tcp}[13] \& 1 == 1$
- $\text{tcp}[13] == 18$

//TCP SYN-ACK

[실습 11] 자신의 IP 주소에서/로 오는 트래픽만 수집 (P.177)

[실습12] 자신을 제외한 모든 MAC 주소에서/로 오는 트래픽만 수집
(P.179)

실습 11. 자신의 IP 주소에서/로 오는 트래픽만 수집

실습 12. 자신을 제외한 모든 MAC 주소에서/로 오는 트래픽만 수집

[실습 9] 파일 집합으로 수집 (P.161)

[실습10] 링 버퍼를 사용해 드라이브 공간 절약 (p.167)

2.9 특정 애플리케이션에 대한 트래픽 수집

- port 53
- not port 53
- port 80
- udp port 67
- tcp port 21
- portrange 1-80
- tcp portrange 1-80
- port 20 or port 21
- host 10.3.1.1 and port 80
- host 10.3.1.1 and not port 80
- udp src port 68 and udp dst port 67

2.10 특정 ICMP 트래픽 수집

- icmp
- icmp[0]=8
- imcp[0]=17
- icmp[0]=8 or imcp[0]=0
- icmp[0]=4 and not icmp[1]=4

[실습 13] DNS 수집 필터 생성과 저장 및 적용 (P.184)

3장. 특정 트래픽을 위한 디스플레이 필터 적용

3.1 적절한 디스플레이 필터 문법 사용

- 디스플레이 필터와 수집 필터 문법은 다름
 - 필터 검용으로 동작 하는 경우도 있음
 - 수집필터는 BPF 형식
 - 디스플레이 필터는 특허 받은 형식
- 디스플레이 필터 오류 탐지 메커니즘
- 필드이름을 기반으로 필터 적용
- 자동 완성 기능을 사용해 디스플레이 필터 구축
- 디스플레이 필터와 7개이 연산자 비교
- Expression을 사용한 디스플레이 필터 구축

디스플레이 필터 오류 탐지 메커니즘(P.239)

- 대소문자 구분
- 적색 배경
 - 문법 검사 실패
 - 동작하지 않음
- 녹색 배경
 - 문법 이상 없음
 - ‘논리 검사’는 하지 않음 (예) `http && udp`
- 황색 배경
 - 필터가 원하는 대로 동작하지 않는 것을 경고 (예) `ip.addr != 10.1.1.1`

캡처 필터 vs 디스플레이 필터

캡처 필터 구문 예제	디스플레이 필터 예제
host 172.16.1.1	ip.host == 172.16.1.1
src host 172.16.1.1	ip.src == 172.16.1.1
dst host 172.16.1.1	ip.dst == 172.16.1.1
port 8080	tcp.port == 8080
!port 8080	!tcp.port == 8080
tcp[13] & 1 == 1	tcp.flag.fin == 1

디스플레이필터와 연산자 비교 (p.197)

연산자	영어표기	예제
==	eq	ip.src == 10.2.2.2
!=	ne	tcp.srcport != 80
>	gt	frame.time_relative > 1
<	lt	tcp.window_size < 1460
>=	ge	dns.count.answers >=10
<=	lt	ip.ttl < 10
	contains	http contain "GET"

표현식(expression)을 사용한 디스플레이 필터 구축

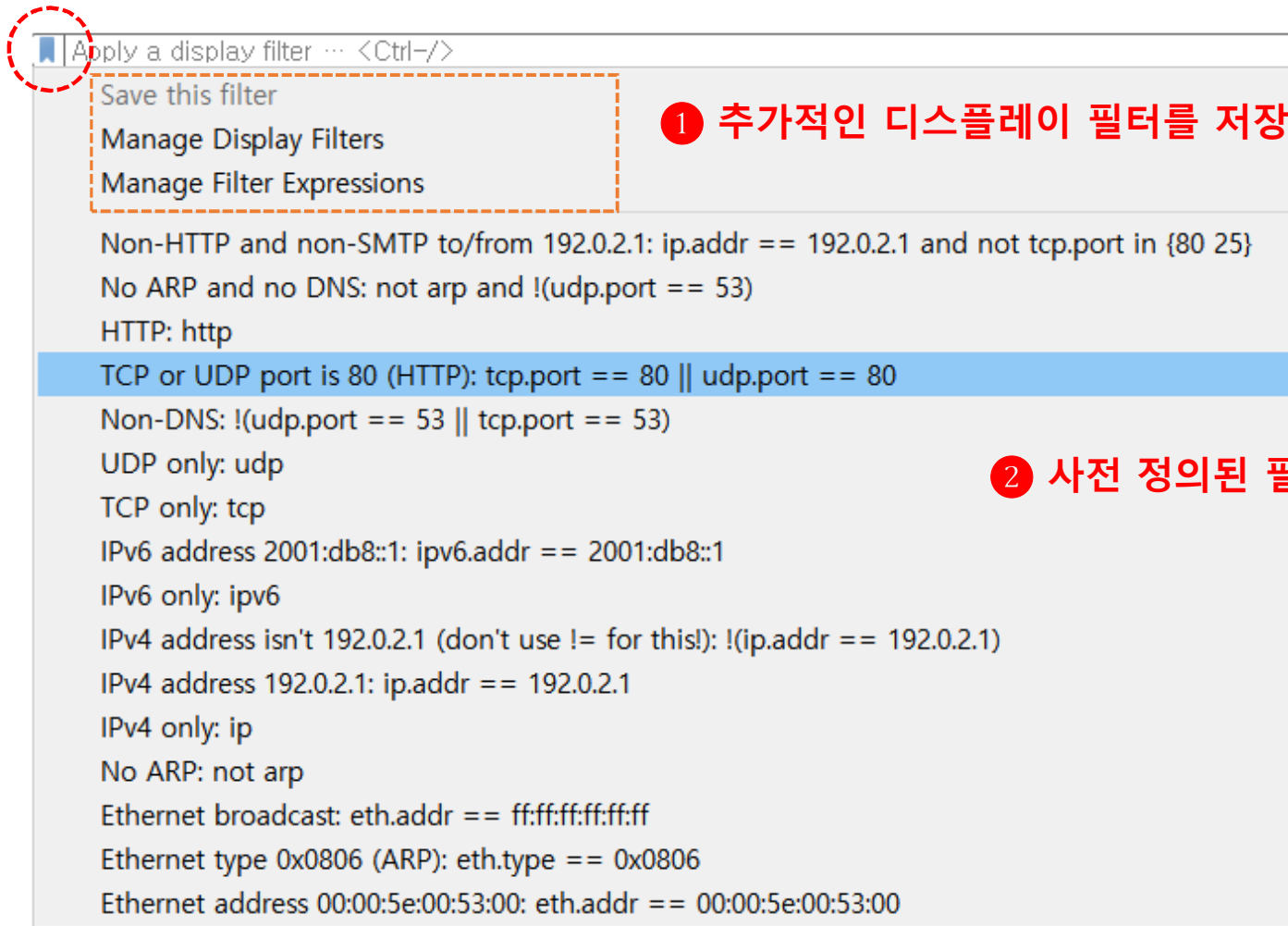
분석 > 표시 필터 표현식

The image shows the 'Wireshark · Display Filter Expression' dialog box. It is divided into several sections:

- Field Name:** A list of fields. 'tcp.flags.syn · Syn' is selected and highlighted with a red dashed box and a circled '1'.
- Relation:** A list of relations. '==' is selected and highlighted with a red dashed box and a circled '2'.
- Value (Boolean):** A text input field containing '1', highlighted with a red dashed box.
- Predefined Values:** A list of predefined values. 'Set' is selected and highlighted with a red dashed box and a circled '3'.
- Search:** A text input field containing 'tcp.flags.syn == 1', highlighted with a red dashed box and a circled '4'.

At the bottom right, there are three buttons: '확인' (OK), '취소' (Cancel), and '도움말' (Help).

3.2 디폴트 디스플레이 필터 편집과 사용



The image shows the 'Apply a display filter' dialog box in Wireshark. The title bar includes a blue icon and the text 'Apply a display filter ... <Ctrl-/>'. The dialog is divided into two main sections. The top section, outlined with a dashed orange border, contains three options: 'Save this filter', 'Manage Display Filters', and 'Manage Filter Expressions'. To the right of this section is a red circular icon with the number '1' and the text '추가적인 디스플레이 필터를 저장' (Save additional display filters). The bottom section contains a list of default display filters. The filter 'TCP or UDP port is 80 (HTTP): tcp.port == 80 || udp.port == 80' is highlighted in blue. To the right of this list is a red circular icon with the number '2' and the text '사전 정의된 필터 확인' (Check predefined filters).

Apply a display filter ... <Ctrl-/>

Save this filter
Manage Display Filters
Manage Filter Expressions

Non-HTTP and non-SMTP to/from 192.0.2.1: ip.addr == 192.0.2.1 and not tcp.port in {80 25}
No ARP and no DNS: not arp and !(udp.port == 53)
HTTP: http
TCP or UDP port is 80 (HTTP): tcp.port == 80 || udp.port == 80
Non-DNS: !(udp.port == 53 || tcp.port == 53)
UDP only: udp
TCP only: tcp
IPv6 address 2001:db8::1: ipv6.addr == 2001:db8::1
IPv6 only: ipv6
IPv4 address isn't 192.0.2.1 (don't use != for this!): !(ip.addr == 192.0.2.1)
IPv4 address 192.0.2.1: ip.addr == 192.0.2.1
IPv4 only: ip
No ARP: not arp
Ethernet broadcast: eth.addr == ff:ff:ff:ff:ff:ff
Ethernet type 0x0806 (ARP): eth.type == 0x0806
Ethernet address 00:00:5e:00:53:00: eth.addr == 00:00:5e:00:53:00

① 추가적인 디스플레이 필터를 저장

② 사전 정의된 필터 확인

3.3 HTTP 트래픽의 적절한 필터링

- http
- tcp.port == XX

3.4 DHCP 디스플레이 필터가 동작하지 않는 이유

- bootp

3.5 IP 주소, 주소범위, 서브넷 기반으로 디스플레이 필터 적용

1) 단순 IP 주소 호스트에게/부터의 트래픽 필터링

- `ip.addr == 10.3.1.1`
- `!ip.addr == 10.3.1.1`
- `ipv6.addr == 2406:da00:ff00::6b16:f02d`
- `ip.src == 10.3.1.1`
- `ip.dst == 10.3.1.1`
- `ip.host == www.wireshark.org`

3.5 IP 주소, 주소범위, 서브넷 기반으로 디스플레이 필터 적용

2) 주소 범위에게/부터의 트래픽 필터링

- `ip.addr > 10.3.0.1 && ip.addr < 10.3.0.5`
- `(ip.addr >= 10.3.0.1 && ip.addr <= 10.3.0.6) && !ip.addr == 10.3.0.3`
- `ipv6.addr == fe80:: && ipv6.addr < fec0::`

3.5 IP 주소, 주소범위, 서브넷 기반으로 디스플레이 필터 적용

3) IP 서브넷에서/으로부터 트래픽 필터링

- `ip.addr == 10.3.0.0/16`
- `ip.addr == 10.3.0.0/16 && !ip.addr == 10.3.0.3`
- `!ip.addr == 10.3.0.0/16 && !ip.addr == 10.2.0.0/16`

3.6 패킷 안에 있는 필드를 이용한 빠른 필터링

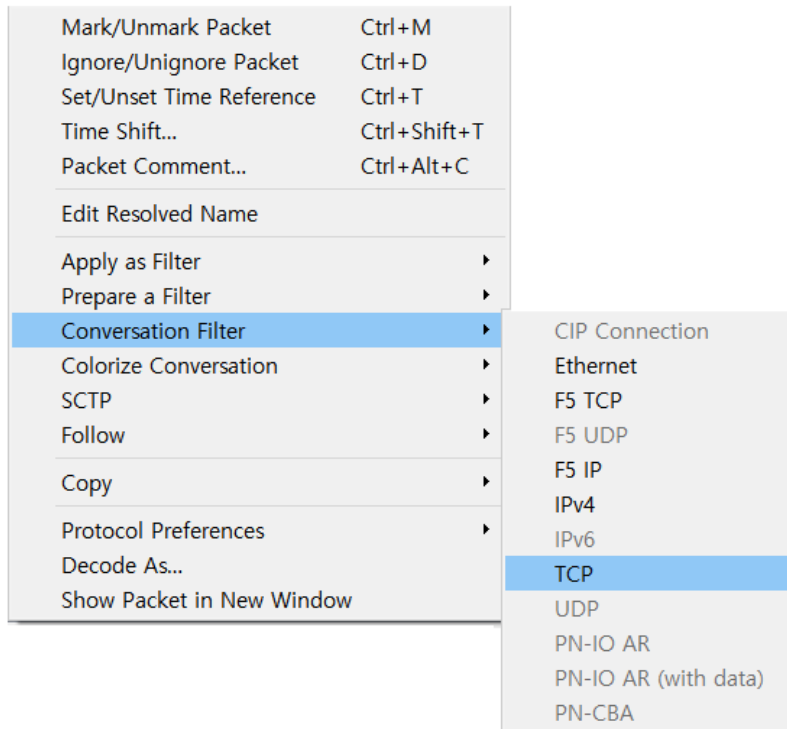
- 1) Apply as Filter
- 2) Prepare a Filter

TCP/UDP Conversation Filter 방법 (p.227)

- 관심 있는 데이터를 빠르게 분석 가능
- 필터 방법 2가지
 - Conversation
 - Stream Follow

3.7 단일 TCP나 UDP 대화 필터링(Conversation Filtering)

❶ 패킷 리스트 > 패킷선택 > 오른쪽 마우스 클릭 > Conversation Filter > TCP



추적파일 : [http-espn101.pcapng](http://espn101.pcapng)

(ip.addr eq 24.6.173.220 and ip.addr eq 199.181.132.250) and (tcp.port eq 19941 and tcp.port eq 80)						
No.	Time	Source	Destination	Protocol	Length	Info
5	0.000000	24.6.173.220	199.181.132.250	TCP	66	19941 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=
6	0.031335	199.181.132.250	24.6.173.220	TCP	66	80 → 19941 [SYN, ACK] Seq=0 Ack=1 Win=4380
7	0.000126	24.6.173.220	199.181.132.250	TCP	54	19941 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=
8	0.000665	24.6.173.220	199.181.132.250	HTTP	603	GET / HTTP/1.1
9	0.041099	199.181.132.250	24.6.173.220	HTTP	484	HTTP/1.1 301 Moved Permanently (text/html
31	0.199860	24.6.173.220	199.181.132.250	TCP	54	19941 → 80 [ACK] Seq=550 Ack=431 Win=65268
4891	68.873340	24.6.173.220	199.181.132.250	TCP	54	19941 → 80 [RST, ACK] Seq=550 Ack=431 Win=

3.7 단일 TCP나 UDP 대화 필터링(Conversation Filtering)

② Statistics > Conversation Filter

추적파일 : [http-espn101.pcapng](#)

Wireshark · Conversations · http-espn101.pcapng

Ethernet · 1IPv4 · 37IPv6TCP · 63UDP · 82

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
24.6.173.220	75.75.75.75	180	22 k	90	6973	90	15 k	0.000000	21.8143	2557	5526
24.6.173.220	199.181.132.250	7	1381	5	831	2	550	0.030245	69.1464	96	63
24.6.173.220	68.71.216.176	127	134 k	38	7147	89	127 k	0.168701	24.5121	2332	41 k
24.6.173.220	184.84.222.48	720	649 k	265	43 k	455	605 k	0.322923	70.0159	5024	69 k
24.6.173.220	143.127.102.125	10	1229	5	514	5	715	0.377829	0.1381	29 k	41 k
24.6.173.220	70.42.13.100	12	2578	7	1903	5	675	2.433476	14.8802	1023	362
24.6.173.220	68.71.212.151	7	1293	5	828	2	465	2.437970	66.7377	99	55
24.6.173.220	74.125.224.59	142	115 k	51	9643	91	105 k	2.843065	66.3320	1162	12 k
24.6.173.220	184.84.222.152	303	286 k	110	25 k	193	261 k	3.261301	70.9168	2865	29 k

Wireshark · Conversations · http-espn101.pcapng

Ethernet · 1IPv4 · 37IPv6TCP · 63UDP · 82

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
24.6.173.220	143.127.102.125	10	1229	5	514	5	715	0.377829	0.1381	29 k	41 k
24.6.173.220	68.71.212.151	7	1293								
24.6.173.220	199.181.132.250	7	1381								
24.6.173.220	107.20.148.253	10	1400								
24.6.173.220	184.84.222.64	8	1422								
24.6.173.220	184.51.159.181	10	1437	6	888						
24.6.173.220	184.84.183.147	8	1573	5	699						
24.6.173.220	107.22.175.32	10	1602	6	1068						
24.6.173.220	184.84.222.112	8	2120	5	701	3	141				
24.6.173.220	64.95.73.7	9	2311	6	1304	3	100				
24.6.173.220	68.71.220.175	7	2355	5	1171	2	118				

3.7 단일 TCP나 UDP 대화 필터링(Conversation Filtering)

② Statistics > Conversation Filter

- Packets 필드를 기준으로 내림 차순으로 정렬
- 첫 번째 패킷 선택
- 오른쪽 마우스 클릭 > Apply as Filter > Selected > A → B

Wireshark · Conversations · http-espn101.pcapng

Ethernet · 1											
IPv4 · 37											
IPv6											
TCP · 63											
UDP · 82											
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	
24.6.173.220	184.84.222.88	1,855	2020 k	533							3765
24.6.173.220	184.84.222.48	720	649 k	265							5024
24.6.173.220	184.84.222.120	613	628 k	195							1882
24.6.173.220	184.84.222.10	371	382 k	119							980
24.6.173.220	184.84.222.152	303	286 k	110							2865
24.6.173.220	75.75.75.75	180	22 k	90	6973	90					2557
24.6.173.220	74.125.224.59	142	115 k	51	9643	91					1162
24.6.173.220	68.71.216.157	132	20 k	66	3672	66	16 k	21.802866	4	Any ↔ B	658
24.6.173.220	68.71.216.176	127	134 k	38	7147	89	127 k	0.168701	2	Any → B	2332
24.6.173.220	184.84.222.16	41	36 k	15	1768	26	35 k	7.951909	6	B → Any	231
24.6.173.220	50.17.254.18	37	7626	22	5637	15	1989	9.581595	2.8209		15 k
24.6.173.220	184.84.222.75	36	33 k	12	1602	24	32 k	5.377013	63.7964		200

3.7 단일 TCP나 UDP 대화 필터링(Conversation Filtering)

② Statistics > Conversation Filter

http-esp101.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src==24.6.173.220 && ip.dst==184.84.222.88

No.	Time	Source	Destination	Protocol	Length	Info
2708	0.000000	24.6.173.220	184.84.222.88	TCP	66	19996 → 80 [SYN] Seq=0 Win=0
2710	0.030156	24.6.173.220	184.84.222.88	TCP	54	19996 → 80 [ACK] Seq=1 Ack=1
2711	0.000703	24.6.173.220	184.84.222.88	HTTP	1514	GET /ads/SEA_ad_111222_Tos
2712	0.000008	24.6.173.220	184.84.222.88	HTTP	134	Continuation
2717	0.020801	24.6.173.220	184.84.222.88	TCP	54	19996 → 80 [ACK] Seq=1541
2720	0.000979	24.6.173.220	184.84.222.88	TCP	54	19996 → 80 [ACK] Seq=1541
2728	0.020029	24.6.173.220	184.84.222.88	TCP	54	19996 → 80 [ACK] Seq=1541
2731	0.000930	24.6.173.220	184.84.222.88	TCP	54	19996 → 80 [ACK] Seq=1541
2734	0.034379	24.6.173.220	184.84.222.88	TCP	54	19996 → 80 [ACK] Seq=1541
2738	0.001807	24.6.173.220	184.84.222.88	TCP	54	19996 → 80 [ACK] Seq=1541
2741	0.000957	24.6.173.220	184.84.222.88	TCP	54	19996 → 80 [ACK] Seq=1541

> Frame 2708: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

> Ethernet II, Src: HewlettP_a7:bf:a3 (d4:85:64:a7:bf:a3), Dst: Cadant_31:bb:c1 (00:01:5c:31:bb:c1)

> Internet Protocol Version 4, Src: 24.6.173.220, Dst: 184.84.222.88

Offset	Raw Data
0000	00 01 5c 31 bb c1 d4 85 64 a7 bf a3 08 00 45 00 ..\1.... d.....E.
0010	00 34 66 04 40 00 80 06 00 00 18 06 ad dc b8 54 .4f.@... ..T
0020	de 58 4e 1c 00 50 f2 ad 09 5f 00 00 00 00 80 02 .XN..P.. _.....
0030	20 00 5c b6 00 00 02 04 05 b4 01 03 03 02 01 01 .\.....

http-esp101.pcapng | Packets: 4900 · Displayed: 533 (10.9%) | Profile: Default

[참고] Endpoints

Statistics > Endpoints

Wireshark · Endpoints · http-espn101.pcapng

Ethernet · 2 IPv4 · 38 IPv6 TCP · 100 UDP · 83

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	AS Number	AS Organization
24.6.173.220	4,900	4490 k	1,700	201 k	3,200	4288 k	—	—	—	—
184.84.222.88	1,855	2020 k	1,322	1989 k	533	30 k	—	—	—	—
184.84.222.48	720	649 k	455	605 k	265	43 k	—	—	—	—
184.84.222.120	613	628 k	418	614 k	195	14 k	—	—	—	—
184.84.222.10	371	382 k	252	374 k	119	7502	—	—	—	—
184.84.222.152	303	286 k	193	261 k	110	25 k	—	—	—	—
75.75.75.75	180	22 k	90	15 k	90	6973	—	—	—	—
74.125.224.59	142	115 k	91	105 k	51	9643	—	—	—	—
68.71.216.157	132	20 k	66	16 k	66	3672	—	—	—	—
68.71.216.176	127	134 k	89	127 k	38	7147	—	—	—	—
184.84.222.16	41	36 k	26	35 k	15	1768	—	—	—	—
50.17.254.18	37	7626	15	1989	22	5637	—	—	—	—
184.84.222.75	36	33 k	24	32 k	12	1602	—	—	—	—
138.108.7.20	31	24 k	20	23 k	11	1675	—	—	—	—
184.84.222.137	30	19 k	16	17 k	14	1638	—	—	—	—
68.71.216.171	29	24 k	17	23 k	12	1007	—	—	—	—
96.17.110.92	25	8423	12	4121	13	4302	—	—	—	—
96.17.148.114	24	13 k	12	10 k	12	3575	—	—	—	—
96.17.110.102	17	2650	6	672	11	1978	—	—	—	—
66.235.138.59	15	7482	7	2816	8	4666	—	—	—	—

☐ Name resolution ☐ Limit to display filter

Endpoint Types

Copy Map 닫기 도움말

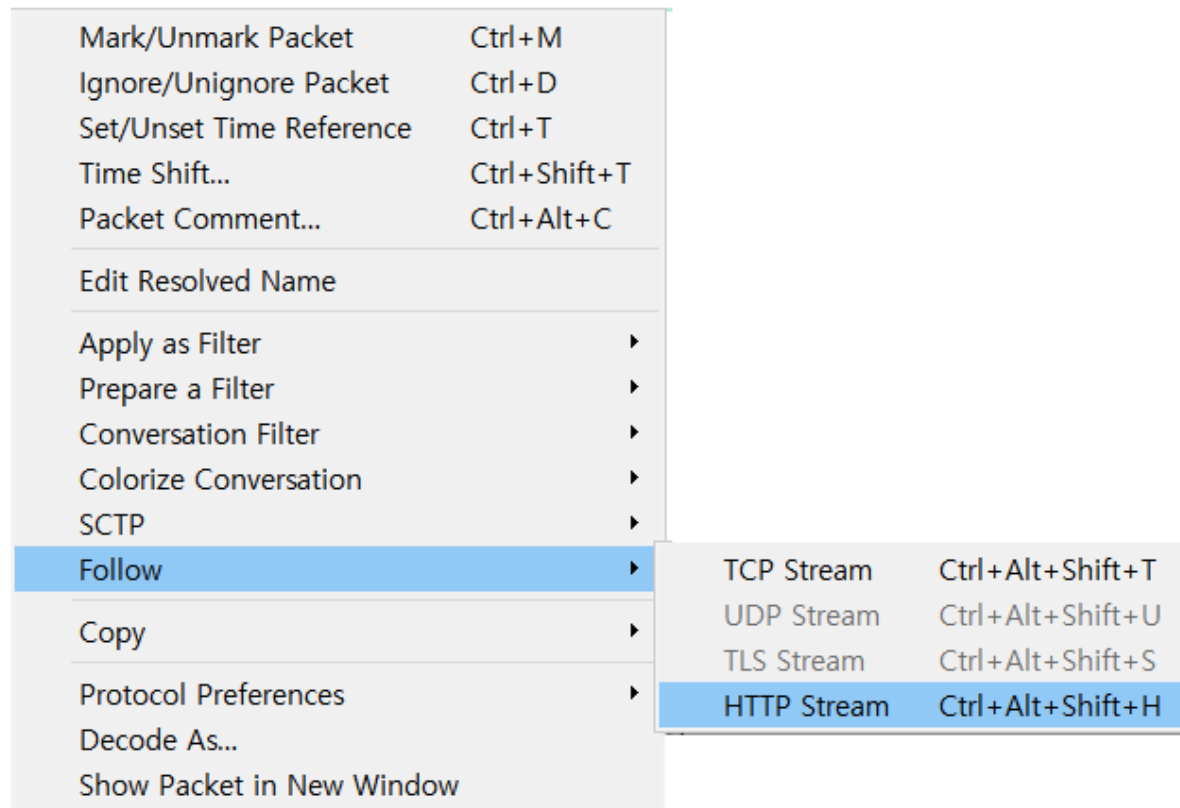
3.7 단일 TCP나 UDP 대화 필터링

* 스트림 따라가기(Stream Follow)

- 여러 패킷의 데이터를 통합해 쉽게 읽을 수 있는 형식으로 재구성 (재조립)
- 4가지 유형의 스트림
 - TCP stream
 - UDP Stream
 - SSL Stream
 - HTTP Stream

3.7 단일 TCP나 UDP 대화 필터링

TCP 또는 HTTP 패킷 선택 > 오른쪽 마우스 클릭 > Follow > HTTP Stream



추적파일 : http-espn101.pcapng

Wireshark · Follow HTTP Stream (tcp.stream eq 0) · http-espn101.pcapng

```
GET / HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg,
application/x-ms-xbap, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/
msword, */*
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; GTB7.2; SLCC2;
.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; HPDPDF;
.NET4.0C; InfoPath.3; MS-RTC LM 8; BRI/2)
Accept-Encoding: gzip, deflate
Host: www.espn.com
Connection: Keep-Alive

HTTP/1.1 301 Moved Permanently
Date: Sat, 07 Jan 2012 21:59:44 GMT
Server: Apache
Location: http://espn.go.com/
Content-Length: 227
X-Cnection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="http://espn.go.com/">here</a>.</p>
</body></html>
```

client pkt(s), server pkt(s), turn(s)

Entire conversation (979 bytes) Show and save data as ASCII

Find: Find Next

Filter Out This Stream Print Save as... Back 달기 도움말

http-espn101.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Info
5	0.000000	24.6.173.220	199.181.132.250	TCP	66	19941 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=
6	0.031335	199.181.132.250	24.6.173.220	TCP	66	80 → 19941 [SYN, ACK] Seq=0 Ack=1 Win=4380 Len=0 M
7	0.000126	24.6.173.220	199.181.132.250	TCP	54	19941 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
8	0.000665	24.6.173.220	199.181.132.250	HTTP	603	GET / HTTP/1.1
9	0.041099	199.181.132.250	24.6.173.220	HTTP	484	HTTP/1.1 301 Moved Permanently (text/html)
31	0.199860	24.6.173.220	199.181.132.250	TCP	54	19941 → 80 [ACK] Seq=550 Ack=431 Win=65268 Len=0
4891	68.873340	24.6.173.220	199.181.132.250	TCP	54	19941 → 80 [RST, ACK] Seq=550 Ack=431 Win=0 Len=0

3.8 다중 포함/배제 조건으로 디스플레이 필터 확장 (P.233)

ip.addr != 10.2.2.2 //송수신주소가 10.2.2.2가 아닌 패킷 수집

 //다른 주소의 패킷들을 수집

!ip.addr==10.2.2.2

!tcp.flags.syn ==1 //UDP와 ARP 패킷이 수집 가능

Tcp.flags.sync != 1 //syn이 0으로 설정된 패킷 수집