

서버프로그래밍 4주차 과제

202127064 이지호

[통신 여부]

주소 \ 프로토콜	118.36.250.254 (이하 118)	168.126.63.1 (이하 168)	8.8.8.8 (이하 8)	127.0.0.1 (이하 127)
ARP	O	X	X	X
ICMP	O	O	O	X

[표와 같은 결과가 나오는 이유]

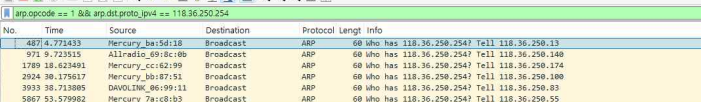
- 118은 내 IP 주소이므로 실제 네트워크를 거치지 않고 ARP, ICMP 요청을 하는 것처럼 보입니다.
- 168과 8은 각각 KT와 Google의 DNS 서버로 ARP 요청을 하지 않고(게이트웨이의 MAC 주소를 모를 때 요청함) ICMP 요청 및 응답은 게이트웨이를 통해 전달됩니다.
- 127은 루프백 주소로 통신이 운영체제 내부에서만 처리되므로 WireShark에서 ARP, ICMP로 확인할 수 없고 Adaptar for loopback에서 확인할 수 있습니다.

1-1. 118.36.250.254(집 공유기) - ARP

```
PS C:\WINDOWS\system32> arp -d
PS C:\WINDOWS\system32> ping 118.36.250.254

Ping 118.36.250.254 32바이트 데이터 사용:
118.36.250.254의 응답: 바이트=32 시간=2ms TTL=64
118.36.250.254의 응답: 바이트=32 시간=2ms TTL=64
118.36.250.254의 응답: 바이트=32 시간=2ms TTL=64
118.36.250.254의 응답: 바이트=32 시간=1ms TTL=64

118.36.250.254에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 1ms, 최대 = 2ms, 평균 = 1ms
PS C:\WINDOWS\system32>
```



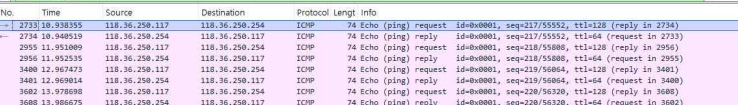
No.	Time	Source	Destination	Protocol	Length	Info
4871	4.721433	Mercury_ba:5d:18	Broadcast	ARP	60	Who has 118.36.250.254? Tell 118.36.250.13
971	9.723535	Allradio_69:8c:0b	Broadcast	ARP	60	Who has 118.36.250.254? Tell 118.36.250.148
1789	18.623491	Mercury_cc:62:99	Broadcast	ARP	60	Who has 118.36.250.254? Tell 118.36.250.174
2024	38.175817	Mercury_b0:87:51	Broadcast	ARP	60	Who has 118.36.250.254? Tell 118.36.250.188
3933	38.713885	DAVOLLINK_06:99:11	Broadcast	ARP	60	Who has 118.36.250.254? Tell 118.36.250.83
5867	53.579962	Mercury_7a:c8:b3	Broadcast	ARP	60	Who has 118.36.250.254? Tell 118.36.250.55

1-2. 118.36.250.254(집 공유기) - ICMP

```
PS C:\WINDOWS\system32> ping 118.36.250.254

Ping 118.36.250.254 32바이트 데이터 사용:
118.36.250.254의 응답: 바이트=32 시간=2ms TTL=64
118.36.250.254의 응답: 바이트=32 시간=1ms TTL=64
118.36.250.254의 응답: 바이트=32 시간=1ms TTL=64
118.36.250.254의 응답: 바이트=32 시간=8ms TTL=64

118.36.250.254에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 1ms, 최대 = 8ms, 평균 = 3ms
PS C:\WINDOWS\system32>
```



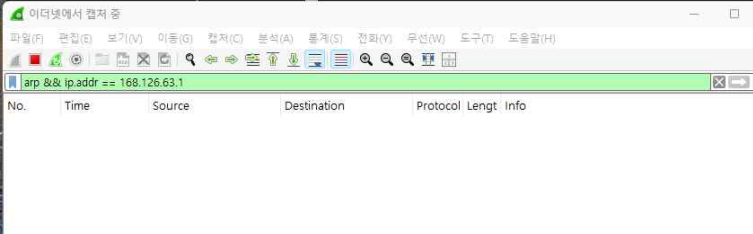
No.	Time	Source	Destination	Protocol	Length	Info
2733	18.938355	118.36.250.117	118.36.250.254	ICMP	74	Echo (ping) request id=0x0001, seq=217/55552, ttl=64 (request in 2733)
2734	18.948519	118.36.250.254	118.36.250.117	ICMP	74	Echo (ping) reply id=0x0001, seq=217/55552, ttl=64 (reply in 2733)
2955	11.953089	118.36.250.117	118.36.250.254	ICMP	74	Echo (ping) request id=0x0001, seq=218/55888, ttl=64 (request in 2955)
2956	11.952535	118.36.250.254	118.36.250.117	ICMP	74	Echo (ping) reply id=0x0001, seq=218/55888, ttl=64 (reply in 2955)
3480	12.967473	118.36.250.117	118.36.250.254	ICMP	74	Echo (ping) request id=0x0001, seq=219/56064, ttl=64 (request in 3480)
3481	12.969814	118.36.250.254	118.36.250.117	ICMP	74	Echo (ping) reply id=0x0001, seq=219/56064, ttl=64 (reply in 3480)
3682	13.978698	118.36.250.117	118.36.250.254	ICMP	74	Echo (ping) request id=0x0001, seq=220/56320, ttl=64 (request in 3682)
3688	13.986675	118.36.250.254	118.36.250.117	ICMP	74	Echo (ping) reply id=0x0001, seq=220/56320, ttl=64 (reply in 3682)

2-1. 168.126.63.1 - ARP

```
PS C:\WINDOWS\system32> arp -d
PS C:\WINDOWS\system32> ping 168.126.63.1

Ping 168.126.63.1 32바이트 데이터 사용:
168.126.63.1의 응답: 바이트=32 시간=1ms TTL=59
168.126.63.1의 응답: 바이트=32 시간=2ms TTL=59
168.126.63.1의 응답: 바이트=32 시간=1ms TTL=59
168.126.63.1의 응답: 바이트=32 시간=1ms TTL=59

168.126.63.1에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 1ms, 최대 = 2ms, 평균 = 1ms
PS C:\WINDOWS\system32>
```

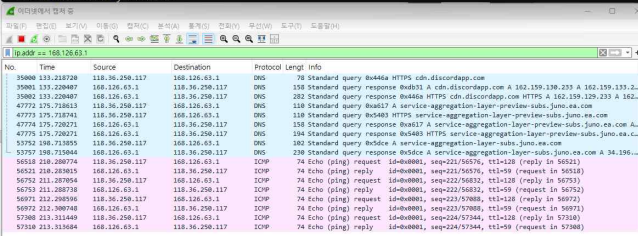


2-2. 168.126.63.1 - ICMP

```
PS C:\WINDOWS\system32> arp -d
PS C:\WINDOWS\system32> ping 168.126.63.1

Ping 168.126.63.1 32바이트 데이터 사용:
168.126.63.1의 응답: 바이트=32 시간=2ms TTL=59
168.126.63.1의 응답: 바이트=32 시간=1ms TTL=59
168.126.63.1의 응답: 바이트=32 시간=2ms TTL=59
168.126.63.1의 응답: 바이트=32 시간=2ms TTL=59

168.126.63.1에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 1ms, 최대 = 2ms, 평균 = 1ms
PS C:\WINDOWS\system32>
```

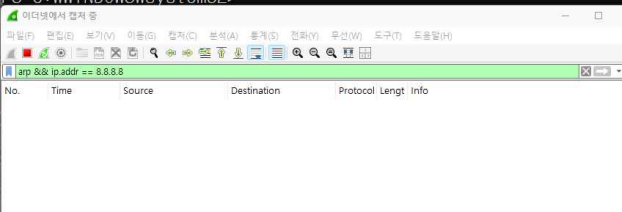


3-1. 8.8.8.8 - ARP

```
PS C:\WINDOWS\system32> ping 8.8.8.8

Ping 8.8.8.8 32바이트 데이터 사용:
8.8.8.8의 응답: 바이트=32 시간=34ms TTL=58
8.8.8.8의 응답: 바이트=32 시간=34ms TTL=58
8.8.8.8의 응답: 바이트=32 시간=34ms TTL=58
8.8.8.8의 응답: 바이트=32 시간=34ms TTL=58

8.8.8.8에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 34ms, 최대 = 34ms, 평균 = 34ms
PS C:\WINDOWS\system32>
```

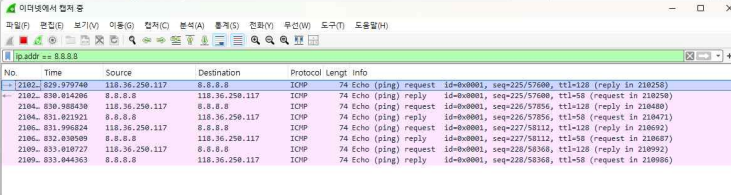


3-2. 8.8.8.8 - ICMP

```
PS C:\WINDOWS\system32> arp -d
PS C:\WINDOWS\system32> ping 8.8.8.8

Ping 8.8.8.8 32바이트 데이터 사용:
8.8.8.8의 응답: 바이트=32 시간=34ms TTL=58
8.8.8.8의 응답: 바이트=32 시간=33ms TTL=58
8.8.8.8의 응답: 바이트=32 시간=33ms TTL=58
8.8.8.8의 응답: 바이트=32 시간=33ms TTL=58

8.8.8.8에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 33ms, 최대 = 34ms, 평균 = 33ms
PS C:\WINDOWS\system32>
```



The screenshot shows a Windows command prompt window with the following output:

```
PS C:\WINDOWS\system32> arp -d
PS C:\WINDOWS\system32> ping 8.8.8.8

Ping 8.8.8.8 32바이트 데이터 사용:
8.8.8.8의 응답: 바이트=32 시간=34ms TTL=58
8.8.8.8의 응답: 바이트=32 시간=33ms TTL=58
8.8.8.8의 응답: 바이트=32 시간=33ms TTL=58
8.8.8.8의 응답: 바이트=32 시간=33ms TTL=58

8.8.8.8에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 33ms, 최대 = 34ms, 평균 = 33ms
PS C:\WINDOWS\system32>
```

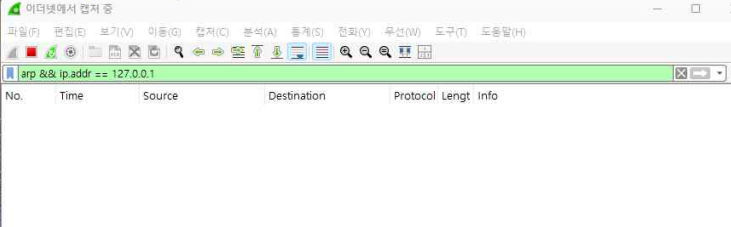
Below the command prompt is a Wireshark network traffic capture window. The filter is set to 'ip.addr == 8.8.8.8'. The packet list shows several ICMP Echo (ping) requests and replies. The packet details pane shows the selected packet (No. 2182) as an ICMP Echo (ping) request from 118.36.258.117 to 8.8.8.8.

4-1. 127.0.0.1 - ARP

```
PS C:\WINDOWS\system32> arp -d
PS C:\WINDOWS\system32> ping 127.0.0.1

Ping 127.0.0.1 32바이트 데이터 사용:
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128

127.0.0.1에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 0ms, 최대 = 0ms, 평균 = 0ms
PS C:\WINDOWS\system32>
```



The screenshot shows a Windows command prompt window with the following output:

```
PS C:\WINDOWS\system32> arp -d
PS C:\WINDOWS\system32> ping 127.0.0.1

Ping 127.0.0.1 32바이트 데이터 사용:
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128

127.0.0.1에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 0ms, 최대 = 0ms, 평균 = 0ms
PS C:\WINDOWS\system32>
```

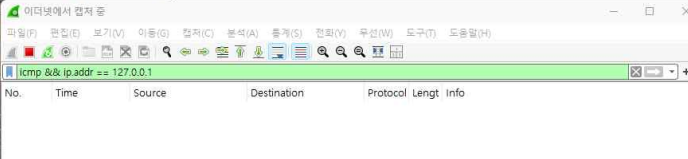
Below the command prompt is a Wireshark network traffic capture window. The filter is set to 'arp && ip.addr == 127.0.0.1'. The packet list is empty, indicating no traffic was captured.

4-2. 127.0.0.1 - ICMP

```
PS C:\WINDOWS\system32> arp -d
PS C:\WINDOWS\system32> ping 127.0.0.1

Ping 127.0.0.1 32바이트 데이터 사용:
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128

127.0.0.1에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 0ms, 최대 = 0ms, 평균 = 0ms
PS C:\WINDOWS\system32>
```



The screenshot shows a Windows command prompt window with the following output:

```
PS C:\WINDOWS\system32> arp -d
PS C:\WINDOWS\system32> ping 127.0.0.1

Ping 127.0.0.1 32바이트 데이터 사용:
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128

127.0.0.1에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 0ms, 최대 = 0ms, 평균 = 0ms
PS C:\WINDOWS\system32>
```

Below the command prompt is a Wireshark network traffic capture window. The filter is set to 'icmp && ip.addr == 127.0.0.1'. The packet list is empty, indicating no traffic was captured.