<컴퓨터 네트워크 3차 프로젝트>

팀명: YonseiAlone

구성인원: 1명

구성원 학번: S20181623

구성원 이름: 김효민

1) 구현 환경

장치 사양

장치 이름	GK-gram
프로세서	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
설치된 RAM	8.00GB(7.87GB 사용 가능)
장치 ID	BC44F7F5-D5BE-49E5-AD88-DFAF35E0B902
제품 ID	00328-20160-00000-AA934
시스템 종류	64비트 운영 체제, x64 기반 프로세서
펜 및 터치	이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.

복사

이 PC의 이름 바꾸기

Windows 사양

에디션	Windows 10 Education
버전	20H2
설치 날짜	2021-05-03
OS 빌드	19042.2006
경험	Windows Feature Experience Pack 120.2212.4180.0

컴퓨터는 위와 같고 파이썬 버전은 3.8.6을 사용했다.

2) 구현

```
def get_ARP_table(self, interface:str, ips:str) -> int:
    # interface: 네트워크 인터페이스의 이름 ex) en0, wl0, 이더넷 등
    # ips: 탐색할 ip의 범위 ex) 192.168.0.1/24는 192.168.0.0 ~ 192.168.0.255까지 256개
    # interface와 ips는 ARP scanning 창으로부터 사용자의 입력값을 받아서 설정됨

    self.ARP_table = list()
    self.interface = interface

    ttt = ips.split("/")
    ip_temp = ttt[0].split(".")

    # todo: scapy의 all verbose를 show하도록 설정하고,
    # todo: scapy의 srp를 사용해 ARP response를 get

    for i in range(256):
        conf.verb = True;
        temp = ip_temp[0] + "." + ip_temp[1] + "." + ip_temp[2] + "." + str(i)
        ans, waste = srp(Ether(dst="ff:ff:ff:ff:ff:ff")/ARP(pdst=temp), timeout=1)
        if(str(ans) != "<Results: TCP:0 UDP:0 ICMP:0 Other:0>"):
            for snd, rcv in ans:
                # todo: arp response (ans)로부터 ip address와 mac address를 get
                mac_addr = rcv[Ether].src
                #self.ARP_table.append((ip_addr, mac_addr))
                self.ARP_table.append((temp, mac_addr))
        # todo: arp response (ans)로부터 ip address와 mac address를 get
```

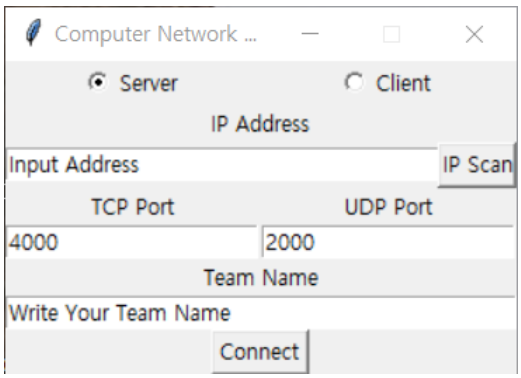
get_ARP_table() 함수의 내용을 보충했다.

우선 인자로 넘겨받은 ips가 1.1.1.1/24 이런식으로 구성되어 있으므로 split() 함수를 이용하여 "/", "." 이 순서대로 문자열을 잘라 준다. 그 후에 subnet의 default 범위가 24까지이므로 0-255까지 총 256번 반복문을 돌면서 앞에서 자른 문자열의 조각들을 마지막 빼고 차례대로 "."과 함께 결합한 뒤, 해당 반복문의 i를 마지막에 붙여줘서 ip address를 완성한다.

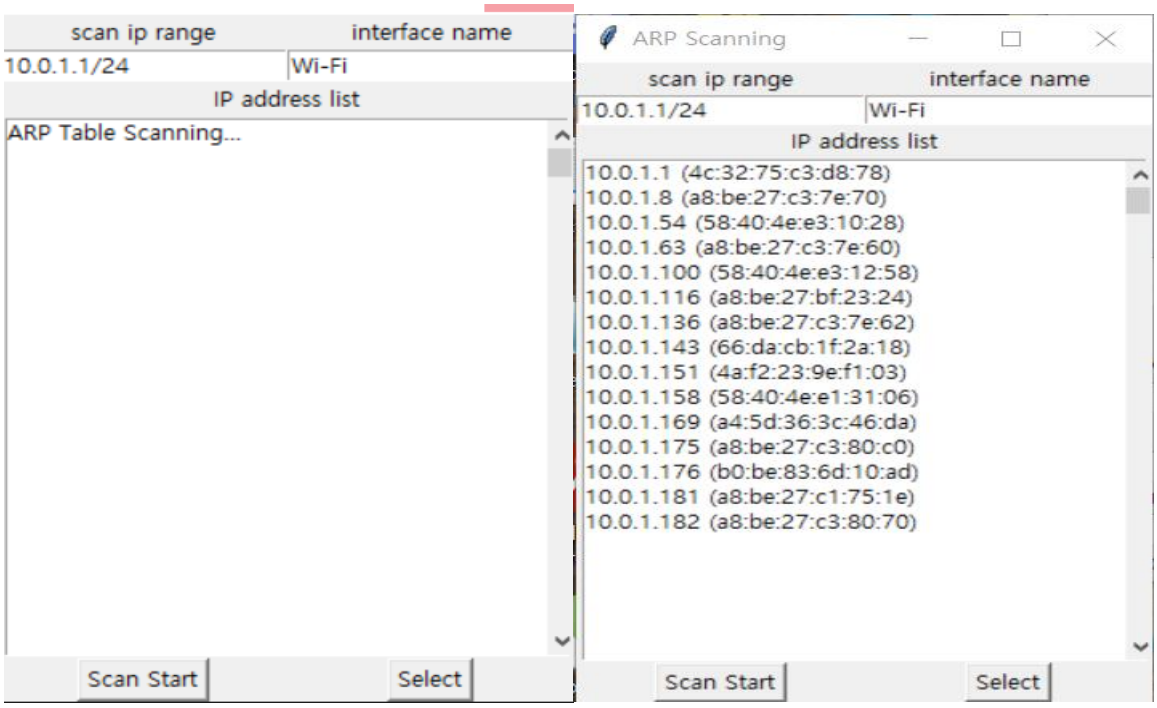
그 후에 srp(), ARP(), Ether()를 이용해서 ans를 구하게 되고 ans가 특정 조건을 만족할 때만 for 문을 돌면서 ans에서 MAC address를 추출한다. 그리고 ARP_table에 위에서 최종적으로 만든 ip address와 MAC address를 넣어준다.

그러고 다시 i의 값을 늘려가면서 위의 과정을 반복한다.

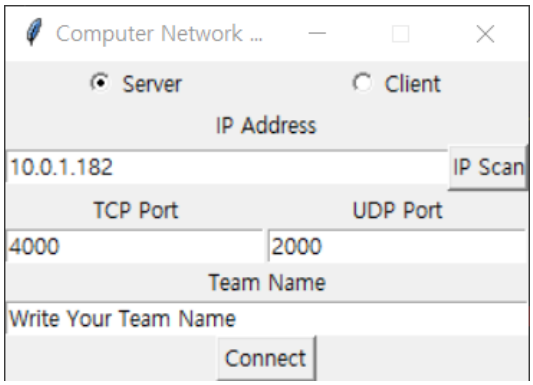
3) 정상 동작 스크린샷



초기 상황이다. 여기서 IP Scan 버튼을 누르면 아래와 같은 화면이 나타난다.



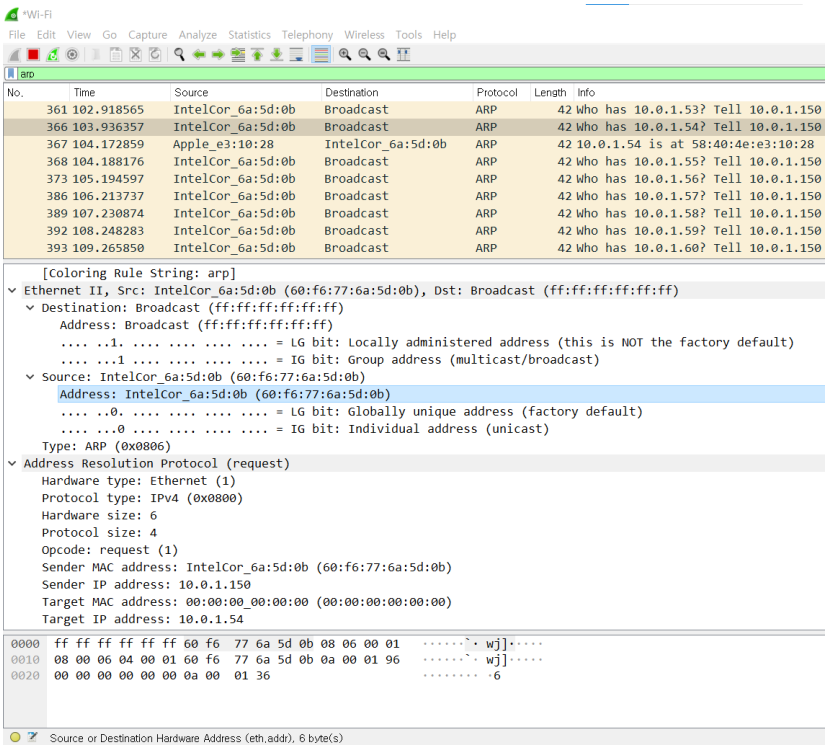
좌측은 Scan Start 버튼을 누른 후고 우측은 해당 Scan이 끝난 후의 table 상태를 의미한다.



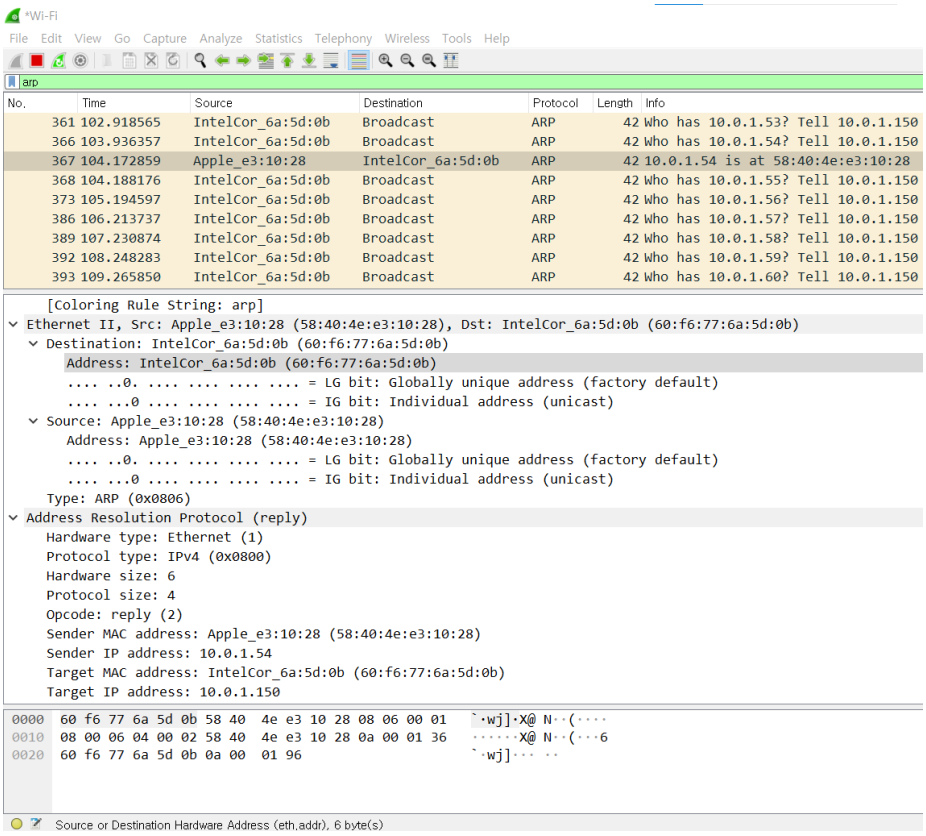
그 후에 Select 버튼을 누르게 되면 이와 같이 IP Addrees가 바뀌게 된다.

4) Wireshark 실습

* Sender

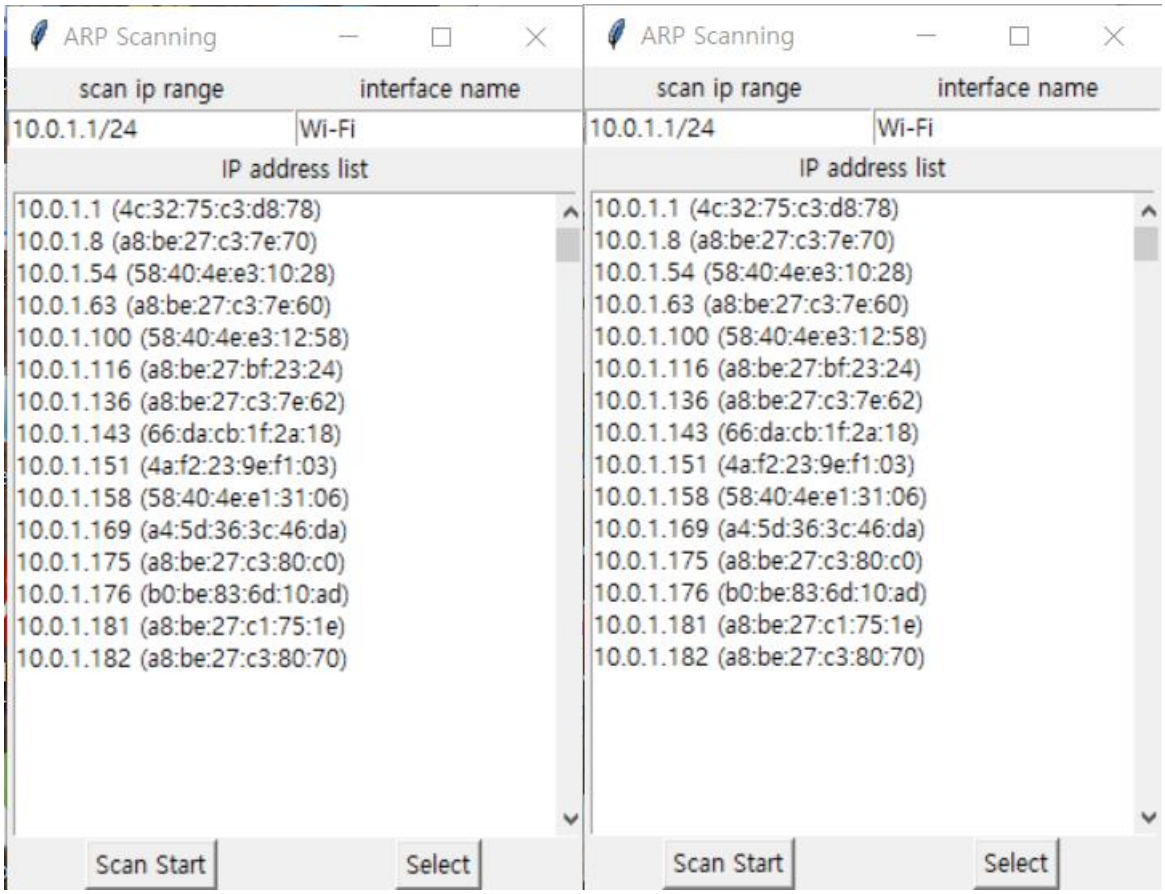


* Reciever



5) Mobility에 따른 IP Address 및 ARP table 확인

5-1) 같은 장소에서 WIFI 연결을 해제했다가 다시 연결했을 때

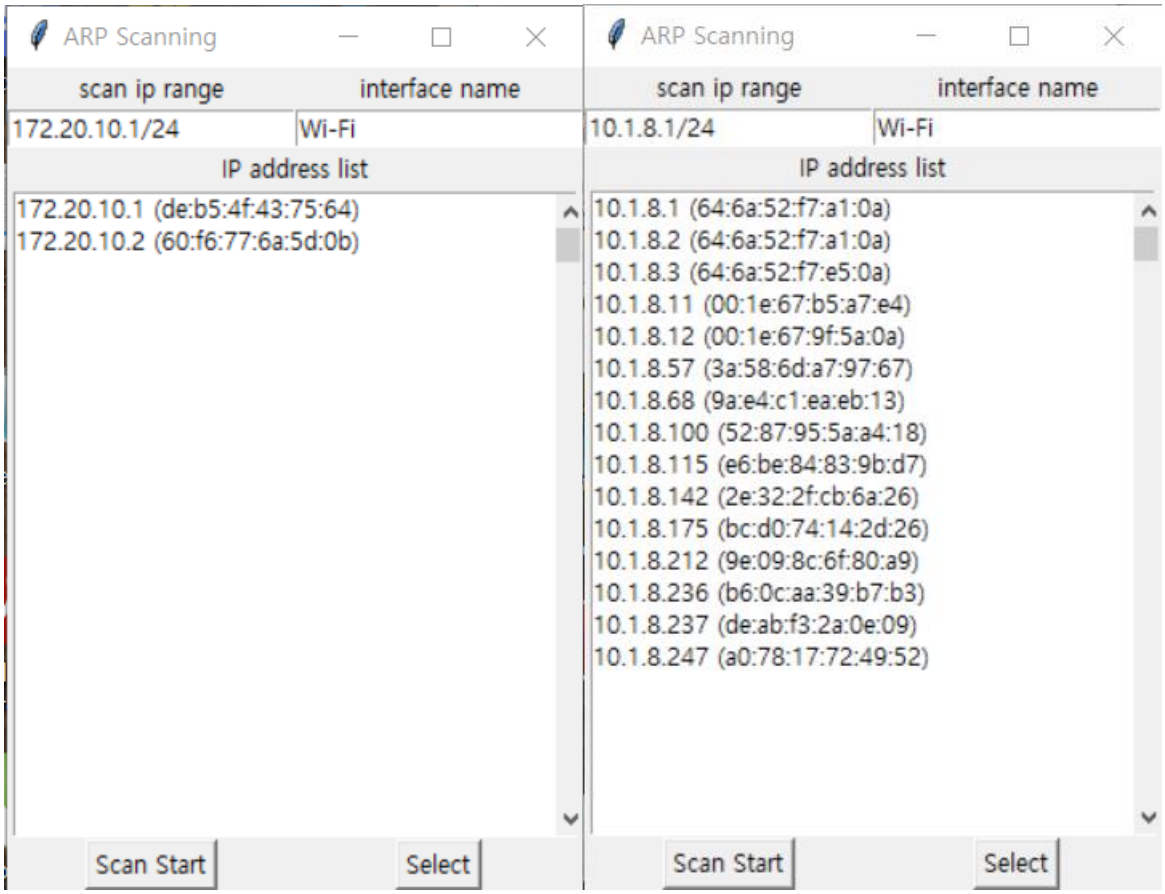


좌측 사진은 노트북의 WIFI를 해제하기 전의 ARP table을 나타내고 있고 우측 사진은 다시 WIFI를 연결 했을 때의 ARP table을 나타낸다. 보시다시피 그 둘 사이에는 차이가 없음을 알 수 있다.

여기서 네트워크는 별도의 공유기를 사용하는 WIFI로 설정했다.

5-2) 장소를 이동한 경우

본 실험에서는 장소를 이동하지 않고 네트워크를 모바일 테더링, 교내 와이파이로 변경하여 다시 ARP table을 스캔해 보았고 그 결과는 아래와 같다.



좌측이 모바일 테더링, 우측이 교내의 WIFI에서 ARP table을 스캔한 경우를 나타낸다.

위에서 보이는 바와 같이 3가지 경우 전부 subnet이 달라짐을 알 수 있다.

5-3) 5-1, 5-2의 차이점 및 이유

1의 경우에는 ARP table이 바뀌지 않고, 2에서는 전부 다 다른 ARP table을 가지게 된다.

이러한 차이는 기존과 같은 subnet에 연결하는지, 다른 subnet에 연결하는지에 따라 발생한다고 할 수 있다. 왜냐하면, subnet이 바뀌게 된다면 그것이 ARP table entry에까지 영향을 줄 수 있기 때문이다.