ARP Table

2020.07.20 김현진

목차

ARP Table

ARP Table
왜 필요해?

Table이 뭐야?

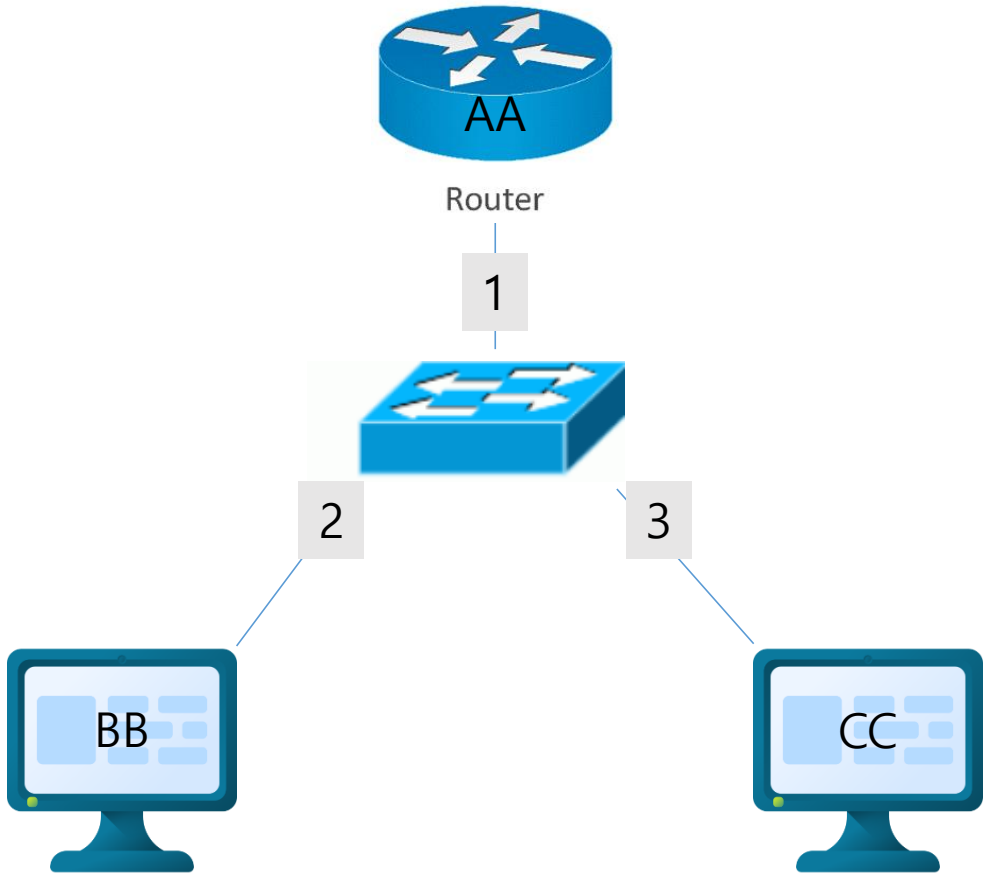
Table

정보를 담는 공간

Ip	Mac
192.168.1.1	AA
192.168.1.10	BB
192.168.1.20	CC

CAM Table

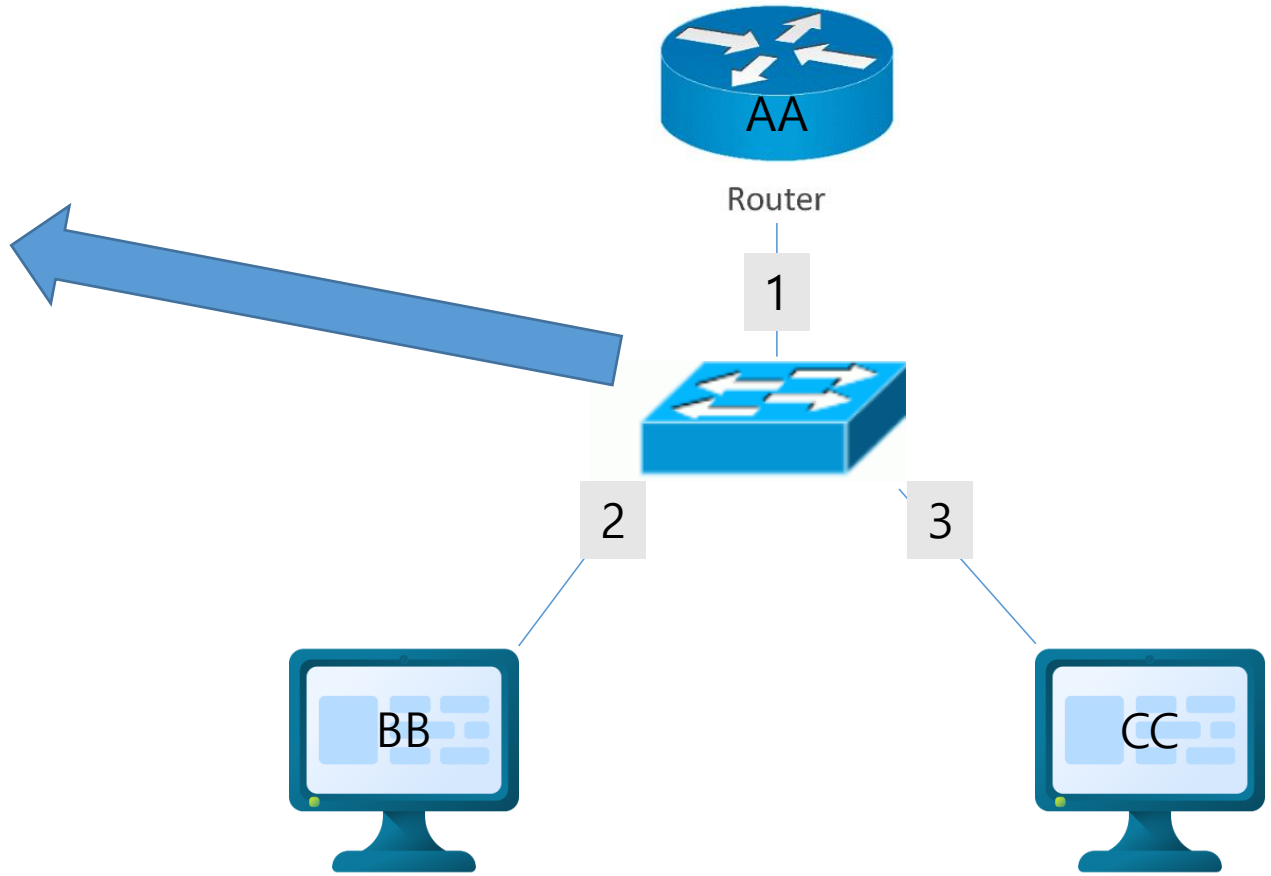
Port–Mac의 매칭 정보를 가지고 있다.



CAM Table

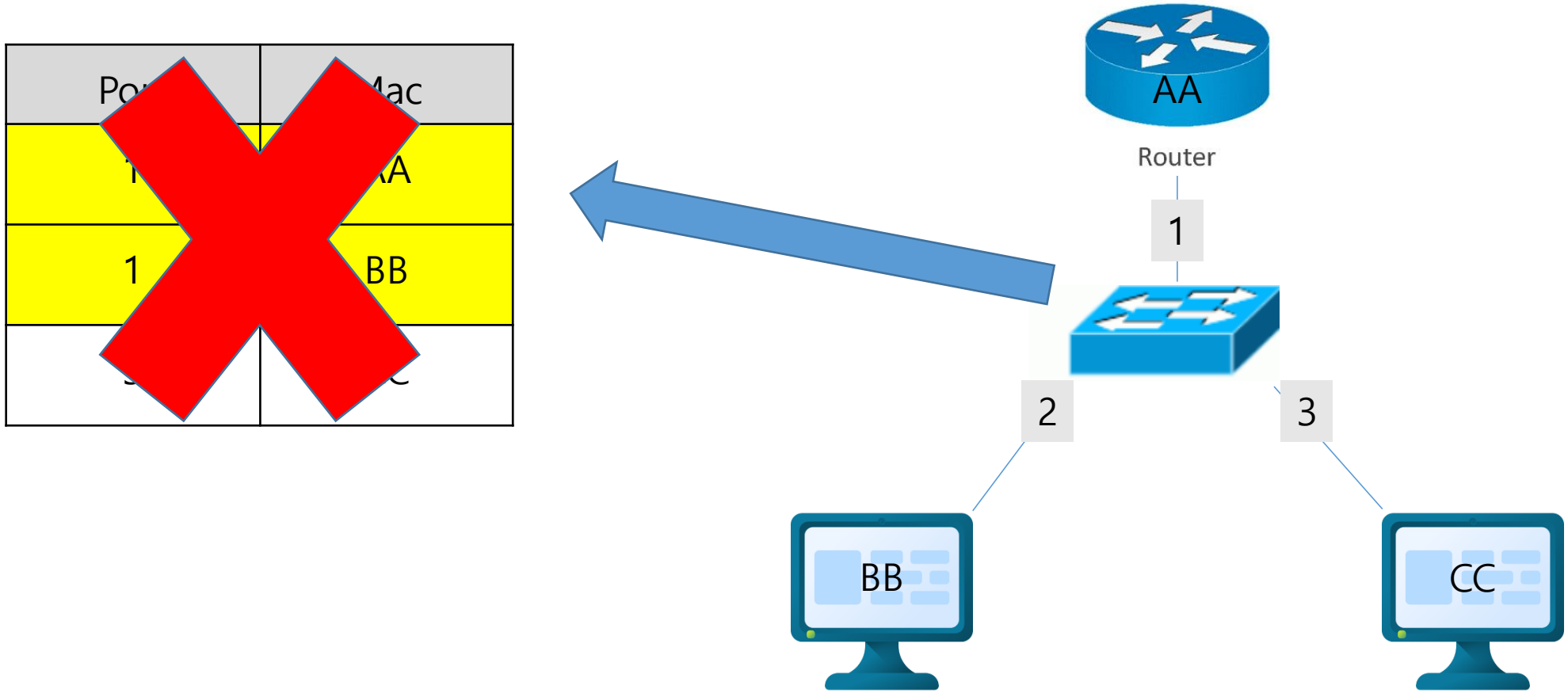
Port–Mac의 매칭 정보를 가지고 있다.

Port	Mac
1	AA
2	BB
3	CC



CAM Table

Port–Mac의 매칭 정보를 가지고 있다.



ARP Table

ARP Table
왜 필요해?

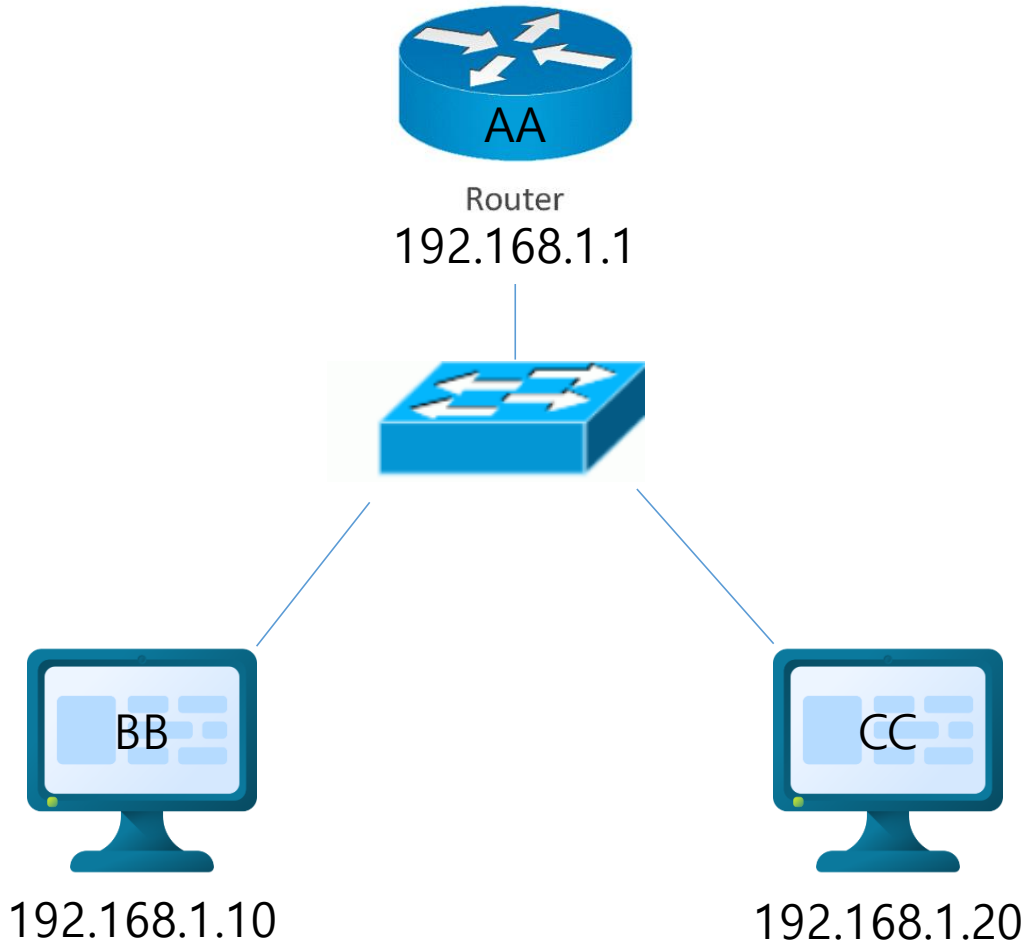
Table이 뭐야?

ARP Table

8731	12.508834	IntelCor_d1:61:7c	Broadcast	ARP	42	Who has 192.1
8732	12.508858	IntelCor_d1:61:7c	Broadcast	ARP	42	Who has 192.1
8737	12.510533	HFR_de:99:91	IntelCor_d1:61:7c	ARP	42	192.168.35.1
11122	14.993690	IntelCor_d1:61:7c	Broadcast	ARP	42	Who has 192.1
11123	14.993719	IntelCor_d1:61:7c	Broadcast	ARP	42	Who has 192.1
11124	14.995625	Chongqin_ad:4a:c3	IntelCor_d1:61:7c	ARP	60	192.168.35.19
12883	17.410414	Microsof_7c:cf:67	IntelCor_d1:61:7c	ARP	42	Who has 192.1
12885	17.410555	IntelCor_d1:61:7c	Microsof_7c:cf:67	ARP	42	192.168.35.20
12886	17.410561	IntelCor_d1:61:7c	Microsof_7c:cf:67	ARP	42	192.168.35.20
20907	28.074468	HFR_de:99:91	Broadcast	ARP	42	Who has 192.1
20916	28.081888	HFR_de:99:91	Broadcast	ARP	42	Who has 192.1
20917	28.081888	HFR_de:99:91	Broadcast	ARP	42	Who has 192.1
20918	28.081888	HFR_de:99:91	Broadcast	ARP	42	Who has 192.1
20919	28.081888	HFR_de:99:91	Broadcast	ARP	42	Who has 192.1
20950	28.140251	HFR_de:99:91	Broadcast	ARP	42	Who has 192.1
21009	28.205311	HFR_de:99:91	Broadcast	ARP	42	Who has 192.1
21010	28.205312	HFR_de:99:91	Broadcast	ARP	42	Who has 192.1
35805	43.204202	IntelCor_d1:61:7c	Broadcast	ARP	60	Who has 192.1
35806	43.204206	IntelCor_d1:61:7c	Broadcast	ARP	60	Who has 192.1
35807	43.205427	HFR_de:99:91	IntelCor_d1:61:7c	ARP	42	192.168.35.1
40180	48.211198	HFR_de:99:91	IntelCor_d1:61:7c	ARP	42	Who has 192.1
40181	48.211588	IntelCor_d1:61:7c	HFR_de:99:91	ARP	60	192.168.35.3
40182	48.211592	IntelCor_d1:61:7c	HFR_de:99:91	ARP	60	192.168.35.3
48885	59.623698	HFR_de:99:91	Broadcast	ARP	42	Who has 192.1

ARP Table

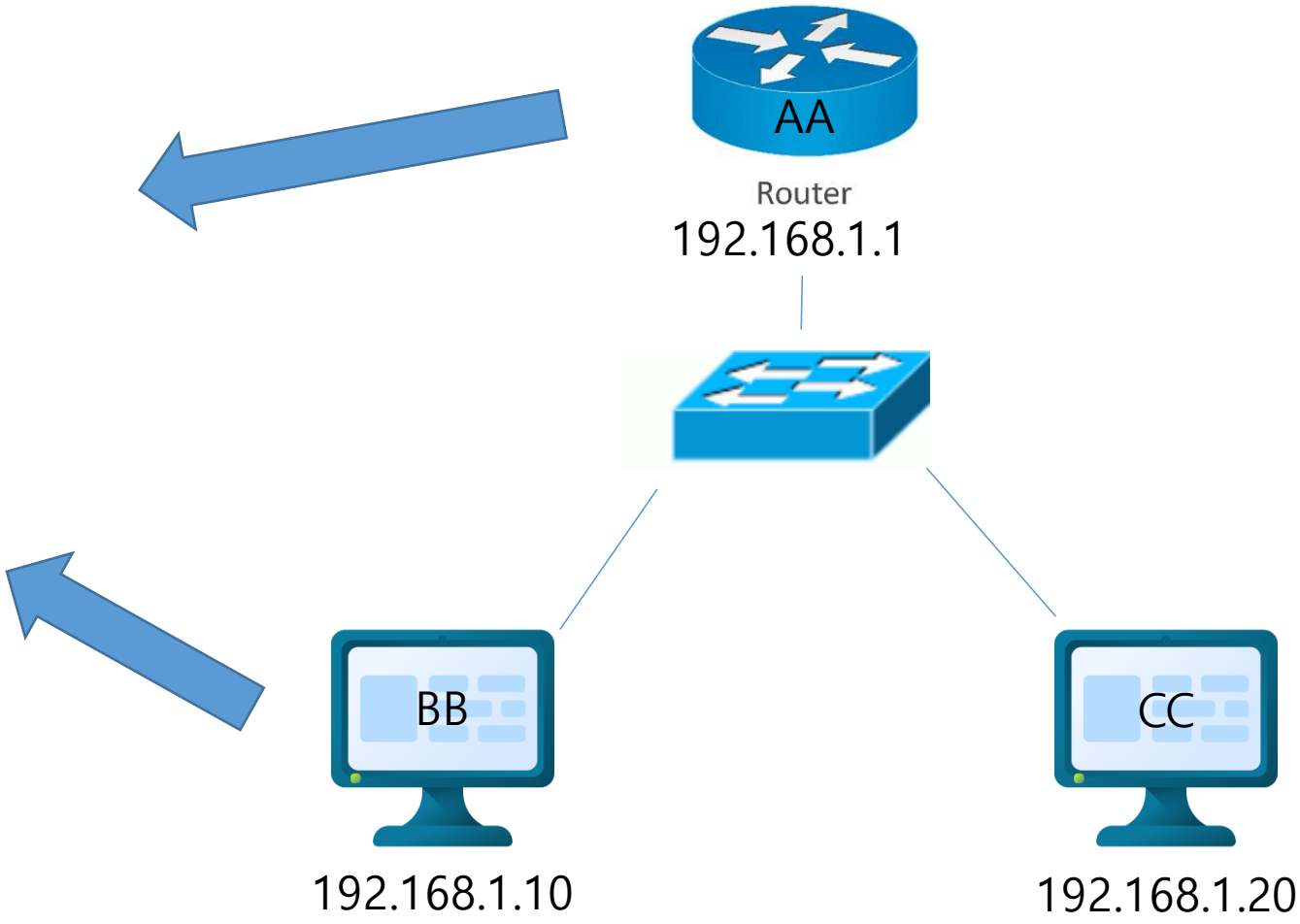
IP–Mac의 1:1 매칭 정보를 가지고 있다.



ARP Table

IP–Mac의 1:1 매칭 정보를 가지고 있다.

Ip	Mac
192.168.1.1	AA
192.168.1.10	BB
192.168.1.20	CC

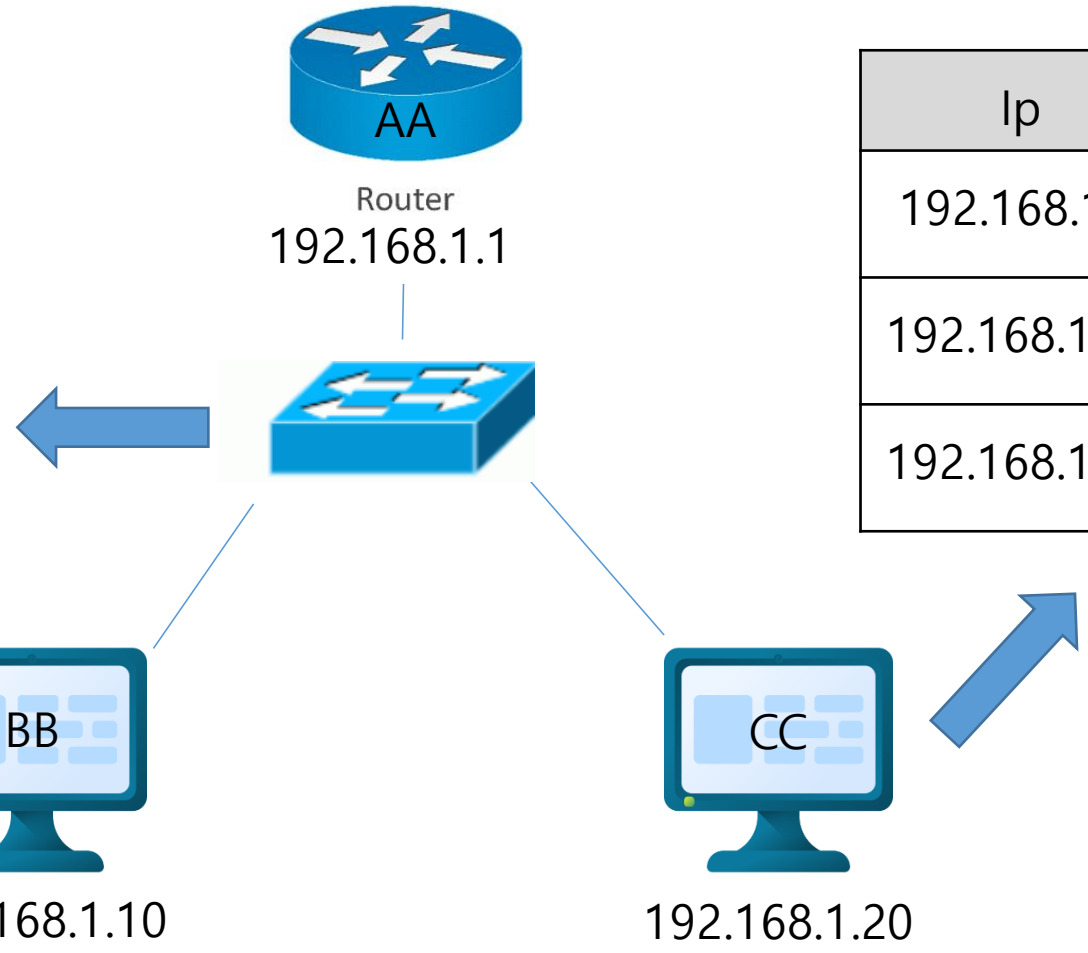


ARP Spoofing

ARP Spoofing

위조된 Reply 패킷을 희생자에게 보내어 패킷을 훔쳐본다.

Port	Mac
1	AA
2	BB
3	CC



Ip	Mac
192.168.1.1	AA
192.168.1.10	BB
192.168.1.20	CC

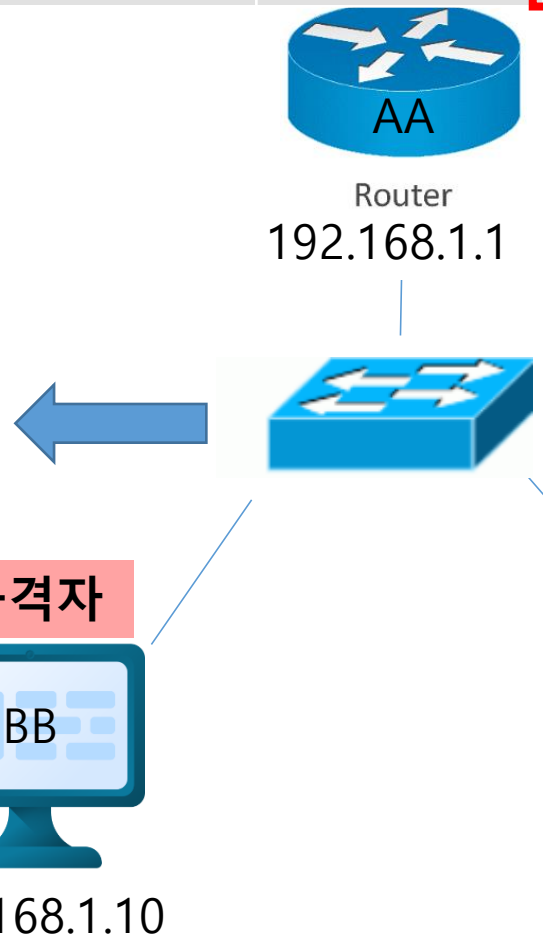
ARP Spoofing

위조된 패킷

Eth_Src	Eth_Dst	Operation	Sender_Ha	Sender_Ip	Target_Ha	Target_Ip
BB	CC	ARP_Reply	BB	1.1	CC	1.20

Port	Mac
1	AA
2	BB
3	CC

스위치의 *CAM Table*



Ip	Mac
192.168.1.1	AA
192.168.1.10	BB
192.168.1.20	CC

희생자의 *ARP Table*

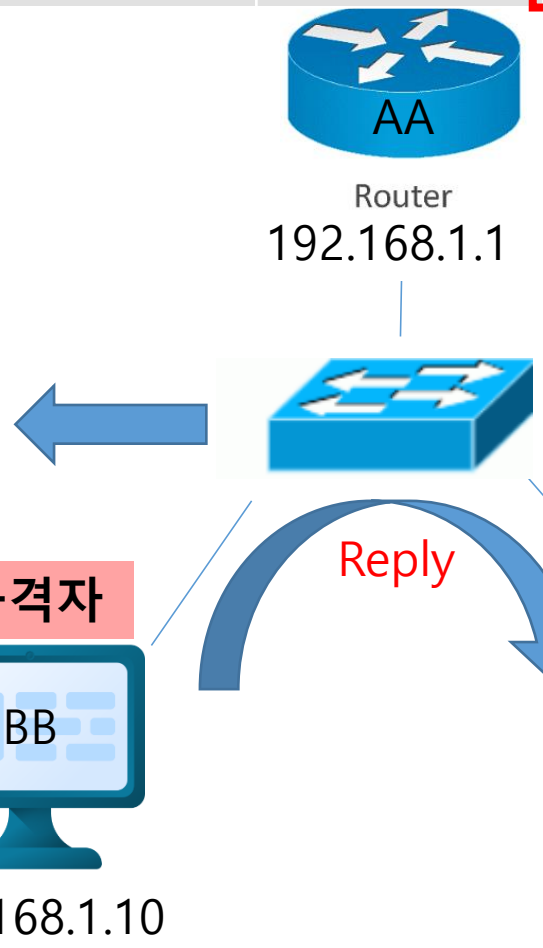
ARP Spoofing

위조된 패킷

Eth_Src	Eth_Dst	Operation	Sender_Ha	Sender_Ip	Target_Ha	Target_Ip
BB	CC	ARP_Reply	BB	1.1	CC	1.20

Port	Mac
1	AA
2	BB
3	CC

스위치의 *CAM Table*



Ip	Mac
192.168.1.1	AA
192.168.1.10	BB
192.168.1.20	CC

희생자의 *ARP Table*

CAM Table

스위치의 *CAM Table*

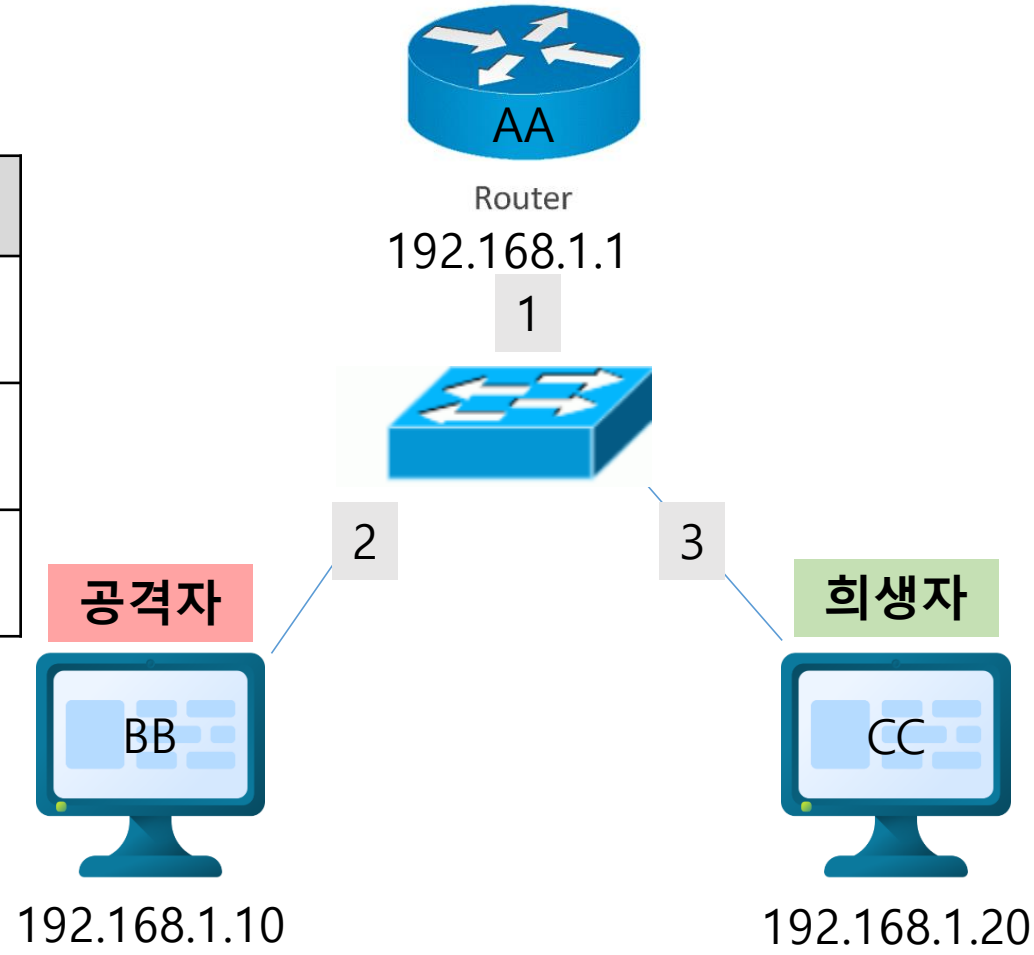
Port	Mac
1	AA
2	BB
3	CC

스위치의 *CAM Table*

Port	Mac
1	AA
2	BB
3	CC

위조된 패킷

Eth_Src	Eth_Dst
BB	CC



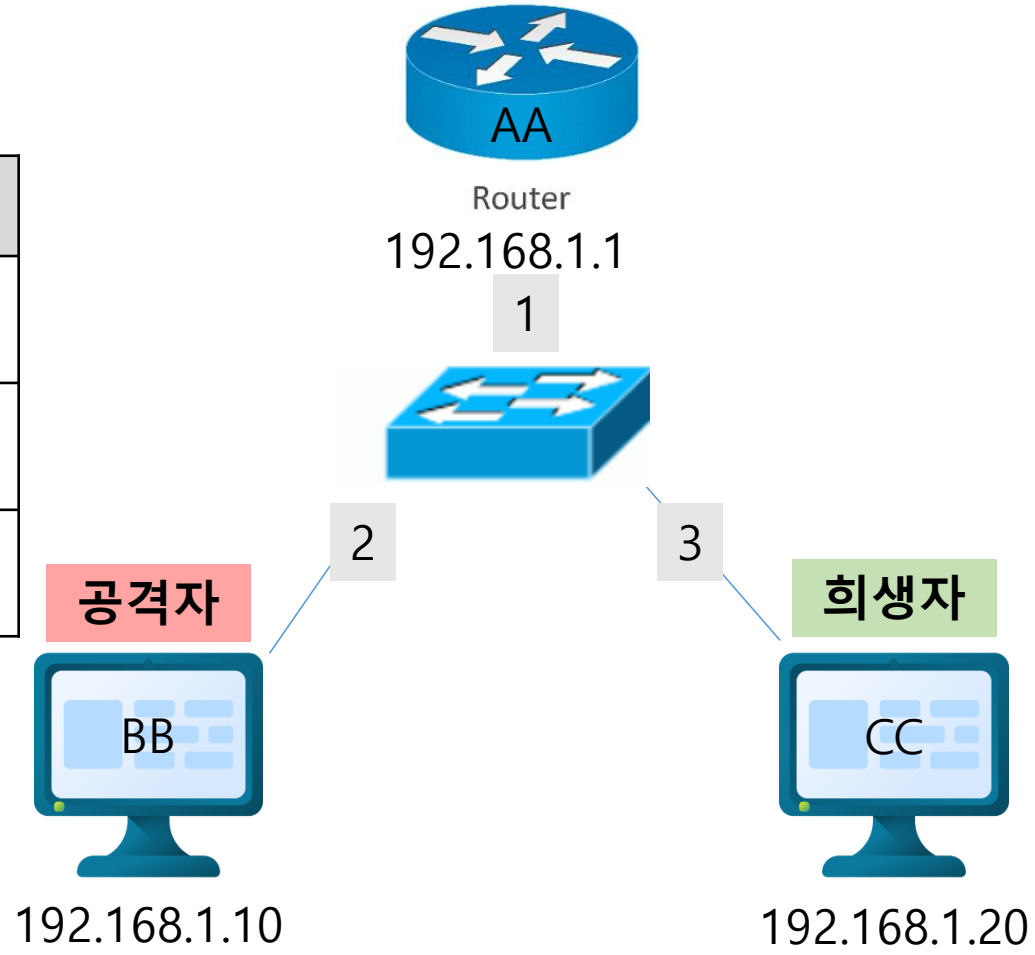
CAM Table

스위치의 *CAM Table*

Port	Mac
1	AA
2	BB
3	CC

스위치의 *CAM Table*

Port	Mac
1	AA
2	BB
3	CC



" 변조 되지 않는 것을 확인 할 수 있다. "

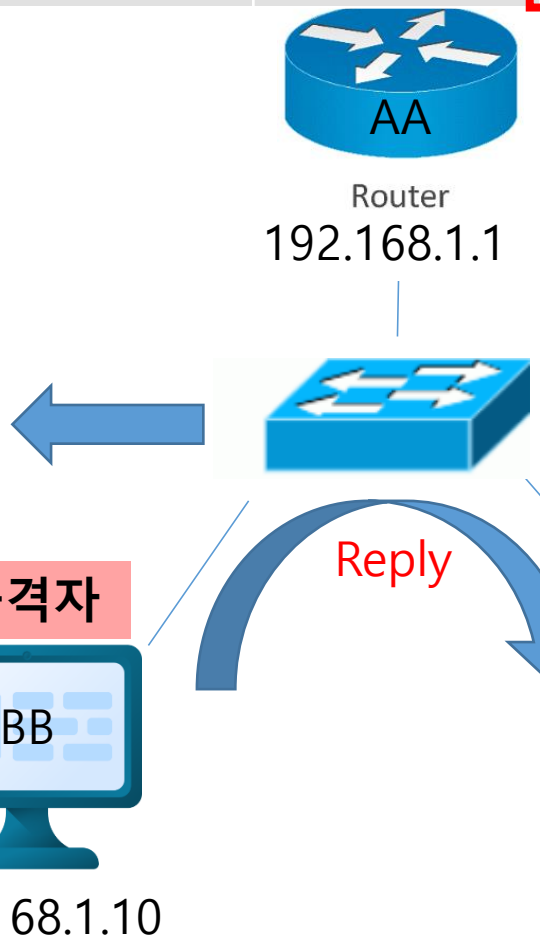
ARP Spoofing

위조된 패킷

Eth_Src	Eth_Dst	Operation	Sender_Ha	Sender_Ip	Target_Ha	Target_Ip
BB	CC	ARP_Reply	BB	1.1	CC	1.20

Port	Mac
1	AA
2	BB
3	CC

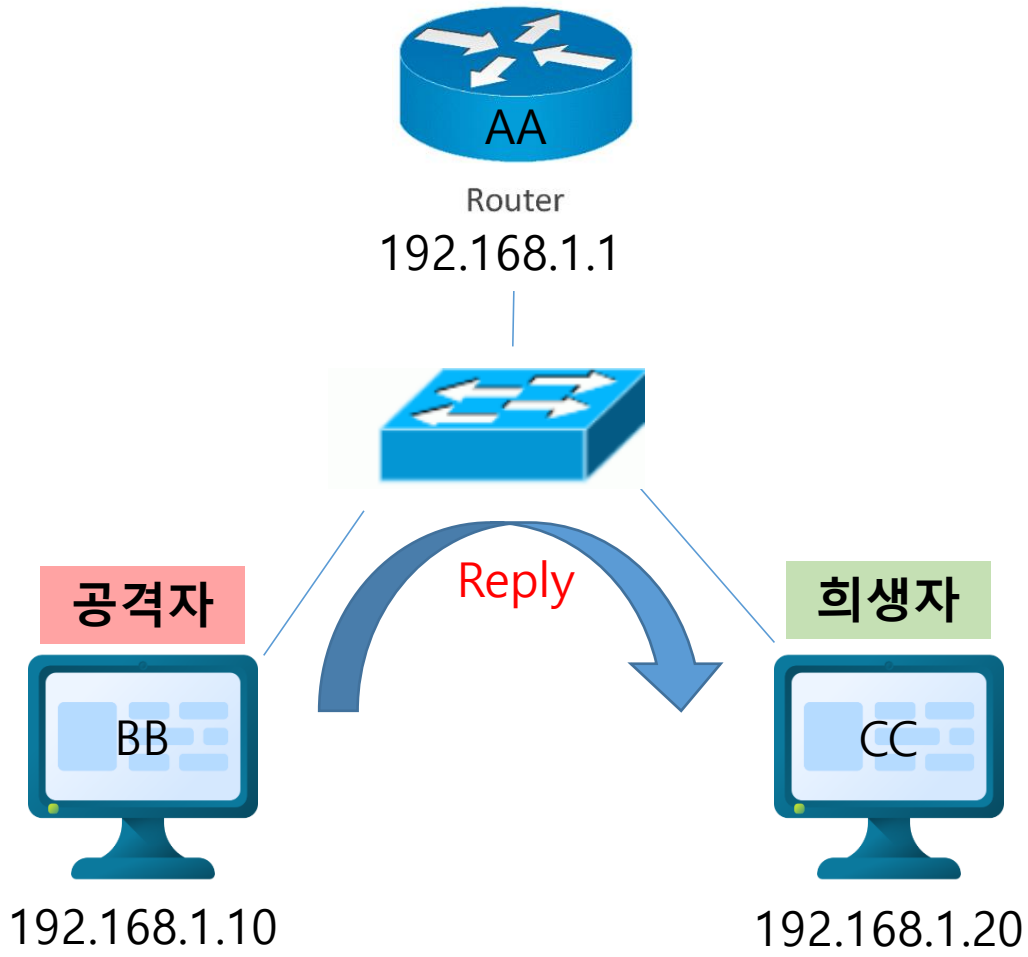
스위치의 CAM Table



Ip	Mac
192.168.1.1	AA
192.168.1.10	BB
192.168.1.20	CC

희생자의 ARP Table

ARP Table



희생자의 ARP Table

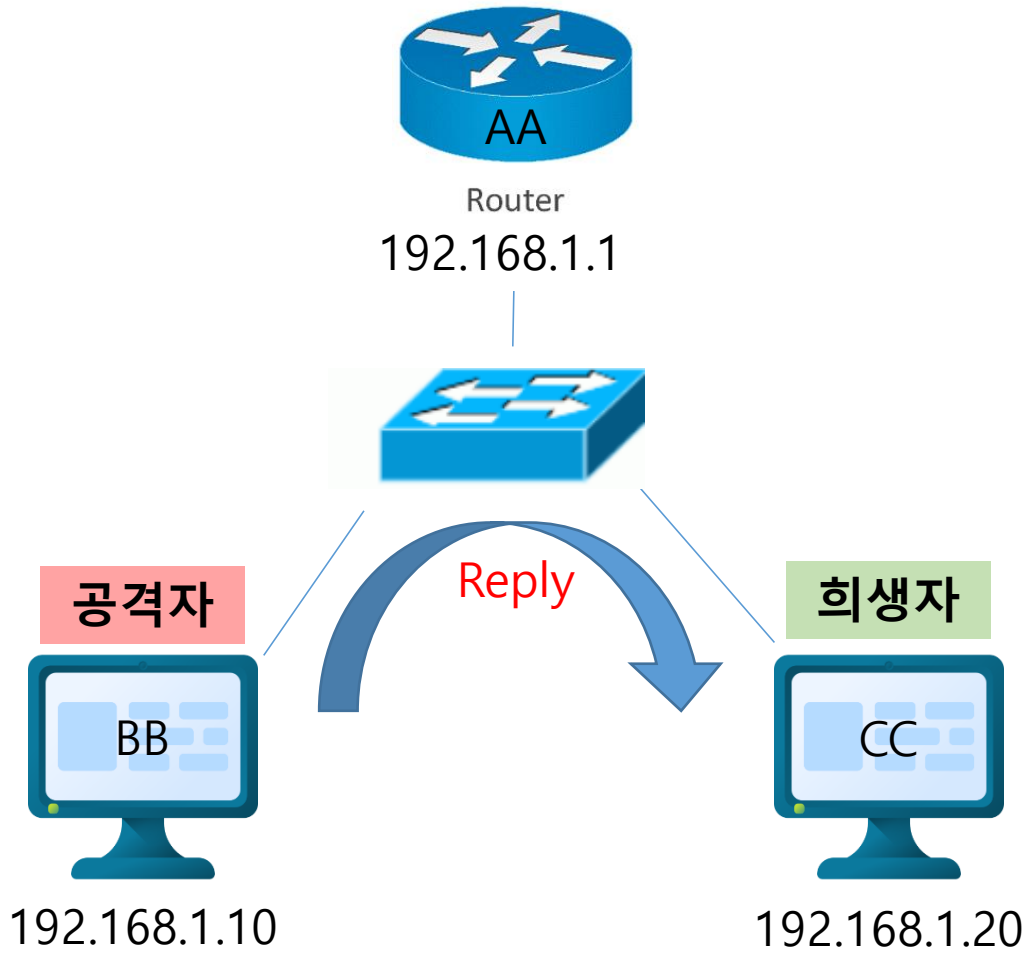
Ip	Mac
192.168.1.1	AA
192.168.1.10	BB
192.168.1.20	CC

희생자의 ARP Table

Ip	Mac
192.168.1.1	AA
192.168.1.1	BB
192.168.1.20	CC

Operation	Sender_Ha	Sender_Ip	Target_Ha	Target_Ip
ARP_Reply	BB	1.1	CC	1.20

ARP Table



희생자의 ARP Table

Ip	Mac
192.168.1.1	AA
192.168.1.10	BB
192.168.1.20	CC

희생자의 ARP Table

Ip	Mac
192.168.1.1	AA
192.168.1.1	BB
192.168.1.20	CC

" 중복된 Mac 주소를 발견할 수 있다. "

K-XARP

K-XARP

Reply 온 패킷의 맥주소가 중복되면 의심 가능!

IP비교 MAC 수집 및 검색

Request에 대응하는 Reply 패킷을 수집해서 Mac주소가 중복되는지 확인하는 로직이 필요!

C언어

```c
int ip_comparison(uint8_t d_dump[100],uint8_t s_dump[100]) {
    int i,result=0;
    for (i=0;i<4;i++) {
        if (d_dump[i] == s_dump[i]){
            result+=1;
        }
    }
    return result;
}

int main(int argc, char * argv[])  {

    if (argc!=2) {
        usage();
        return -1;
    }
```

```c
    if (eth_type == ETHERTYPE_ARP) { // ARP = 0x0806

        if (arp_opcode == ARP_REQUEST) { // request
            memcpy(d_dump,arp->dst_ip,4);
            printf("q_dump = %d %d %d %d \n", d_dump[0], d_dump[1], d_dump[2], d_dump[3]);
        }
        if (arp_opcode == ARP_REPLY) { // reply
            printf("\nOP OK\n");
            memcpy(s_dump,arp->src_ip,4);
            printf("p_dump = %d %d %d %d \n", s_dump[0], s_dump[1], s_dump[2], s_dump[3]);

            if (ip_comparison(d_dump, s_dump) == 4) {    ⚠ incompatible pointer types pass
                memcpy(&smac,ether->ether_shost,6);   // smac Collection

                printf("--------- [reply OK] --------- \n\n");
                printf("Hardware : %02x \n",arp_hardware);
                printf("Protocol : %02x \n",arp_protocol);
                printf("Hlen : %d \n",arp_hlen);
                printf("Plen : %d \n",arp_plen);
                printf("Opcode : %d \n",arp_opcode);
                printf("Src Mac : %02x:%02x:%02x:%02x:%02x:%02x \n", ether->ether_shost[0]
                printf("Src Ip : %d.%d.%d.%d \n",arp->src_ip[0],arp->src_ip[1],arp->src_ip
                printf("Dst Mac : %02x:%02x:%02x:%02x:%02x:%02x \n",ether->ether_dhost[0],
                printf("Dst Ip : %d.%d.%d.%d \n",arp->dst_ip[0],arp->dst_ip[1],arp->dst_ip
```

C언어

```c
int ip_comparison(uint8_t d_dump[100],uint8_t s_dump[100]) {
    int i,result=0;
    for (i=0;i<4;i++) {
        if (d_dump[i] == s_dump[i]){
            result+=1;
        }
    }
    return result;
}

int main(int argc, char * ar

    if (argc!=2) {
        usage();
        return -1;
    }
```

IP에 대응하는 MAC주소가

중복되는지 체크하기 어려움!

```c
{ // ARP = 0x0806

// request

", d_dump[0], d_dump[1], d_dump[2], d_dump[3]);

// reply

%d %d %d \n", s_dump[0], s_dump[1], s_dump[2], s_dump[3]);

mp, s_dump) == 4) {    ⚠ incompatible pointer types pass
cpy(&sma      >ether_shost,6);   // smac Collection

intf("--------- [ eply OK] --------- \n\n");
rintf("Hardware : %02x \n",arp_hardware);
rintf("Protocol : %02x \n",arp_protocol);
printf("Hlen : %d \n",arp_hlen);
printf("Plen : %d \n",arp_plen);
printf("Opcode : %d \n",arp_opcode);
printf("Src Mac : %02x:%02x:%02x:%02x:%02x:%02x \n", ether->ether_shost[0]
printf("Src Ip : %d.%d.%d.%d \n",arp->src_ip[0],arp->src_ip[1],arp->src_ip
printf("Dst Mac : %02x:%02x:%02x:%02x:%02x:%02x \n",ether->ether_dhost[0],
printf("Dst Ip : %d.%d.%d.%d \n",arp->dst_ip[0],arp->dst_ip[1],arp->dst_ip
```

Python

```
************  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.3
Target MAC :  00:23:aa:de:99:91
Target IP :  192.168.35.1
192.168.35.3 00:24:d6:d1:61:7c

#####     right reply       ######

{'192.168.35.1': '00:23:aa:de:99:91', '192.168.35.197':
 '40:5b:d8:ad:4a:c3', '192.168.35.203': '00:24:d6:d1:61
:7c', '192.168.35.3': '00:24:d6:d1:61:7c'}
```

Python

```
***********  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.3
Target MAC :  00:23:aa:de:99:91
Target IP :  192.168.35.1
192.168.35.3 00:24:d6:d1:61:7c

#####     right reply       ######

{'192.168.35.1': '00:23:aa:de:99:91', '192.168.35.197':
 '40:5b:d8:ad:4a:c3', '192.168.35.203': '00:24:d6:d1:61
:7c', '192.168.35.3': '00:24:d6:d1:61:7c'}
```

pypcap

libpcap 용 단순화 된 객체 지향 Python

Graphic | Table

Library	Time taken(seconds)	Packets per second
libpcap	0.141	3546099
libtins	**0.273**	**1831501**
pcapplusplus	0.38	1315789
dpkt	8.132	61485

Python

```
***********  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.3
Target MAC :  00:23:aa:de:99:91
Target IP :  192.168.35.1
192.168.35.3 00:24:d6:d1:61:7c

#####     right reply       ######

{'192.168.35.1': '00:23:aa:de:99:91', '192.168.35.197':
 '40:5b:d8:ad:4a:c3', '192.168.35.203': '00:24:d6:d1:61
:7c', '192.168.35.3': '00:24:d6:d1:61:7c'}
```

dpkt

기본 TCP / IP 프로토콜에 대한 정의와 함께
빠르고 간단한 패킷 생성 / 파싱을위한 파이썬 모듈

dpkt 모듈

- dpkt.ah 모듈
- dpkt.aim 모듈
- dpkt.aoe 모듈
- dpkt.aoeata 모듈
- dpkt.aoecfg 모듈
- dpkt.arp 모듈
- dpkt.asn1 모듈

Python

```python
import pcap
import dpkt

ARP_Table={}

def Right_Reply(sIP,sMAC):
    if sIP in ARP_Table :
        print("exist")
    else :
        ARP_Table[sIP] = sMAC
        print(ARP_Table)

def main():

    dev = pcap.findalldevs()[2]

    for timestamp, buf in pcap.pcap(name=dev):

        eth = dpkt.ethernet.Ethernet(buf)
        arp = eth.data
```

IP와 MAC을 조회하고 없으면 쌍을 추가해주는 함수

Python

```
21          arp = eth.data
22
23          if eth.type == 0x0806:
24              print("\n***********  ARP  ************")
25
26              print('Arp_hln : ', arp.hln)
27              print('Arp_hrd : ', arp.hrd)
28              print('Arp_pln : ', arp.pln)
29              print('Arp_pro : ', arp.pro)
30              print('\nOPcode : ', arp.op)
31              print('Sender MAC : ', ":".join(['%02x' % arp.sha[0],'%02x' % arp.sha[1],'%02x' % arp.sha[2],'%02x' % arp.sha[3],'%02x' % arp.sha[4],
32              print('Sender IP : ', str(int(arp.spa[0]))+"."+str(int(arp.spa[1]))+"."+str(int(arp.spa[2]))+"."+str(int(arp.spa[3])))  # ip
33              print('Target MAC : ', ":".join(['%02x' % arp.tha[0],'%02x' % arp.tha[1],'%02x' % arp.tha[2],'%02x' % arp.tha[3],'%02x' % arp.tha[4],
34              print('Target IP : ', str(int(arp.tpa[0]))+"."+str(int(arp.tpa[1]))+"."+str(int(arp.tpa[2]))+"."+str(int(arp.tpa[3])))  # ip
35
36              if arp.op == 1 :
37                  tIP = str(int(arp.tpa[0]))+"."+str(int(arp.tpa[1]))+"."+str(int(arp.tpa[2]))+"."+str(int(arp.tpa[3]))
38                  print(tIP)
39
40              if arp.op == 2 :
41                  sIP = str(int(arp.spa[0]))+"."+str(int(arp.spa[1]))+"."+str(int(arp.spa[2]))+"."+str(int(arp.spa[3]))
42                  sMAC = ":".join(['%02x' % arp.sha[0],'%02x' % arp.sha[1],'%02x' % arp.sha[2],'%02x' % arp.sha[3],'%02x' % arp.sha[4],'%02x' % arp
43                  print(sIP,sMAC)
```

Python

```
21          arp = eth.data
22
23          if eth.type == 0x0806:
24              print("\n***********  ARP  ************")
25
26              print('Arp_hln : ', arp.hln)
27              print('Arp_hrd : ', arp.hrd)
28              print('Arp_pln : ', arp.pln)
```

```
print( "-".join(["a", "b", "c"]) ) # a-b-c
```

```
a-b-c
```

```
":".join(['%02x' % arp.sha[0],
```

```
sha[4],
tha[4],
."+str(int(arp.tpa[1]))+"."+str(int(arp.tpa[2]))+"."+str(int(arp.tpa[3])))  # ip
```

```
36              if arp.op == 1 :
37                  tIP = str(int(arp.tpa[0]))+"."+str(int(arp.tpa[1]))+"."+str(int(arp.tpa[2]))+"."+str(int(arp.tpa[3]))
38                  print(tIP)
39
40              if arp.op == 2 :
41                  sIP = str(int(arp.spa[0]))+"."+str(int(arp.spa[1]))+"."+str(int(arp.spa[2]))+"."+str(int(arp.spa[3]))
42                  sMAC = ":".join(['%02x' % arp.sha[0],'%02x' % arp.sha[1],'%02x' % arp.sha[2],'%02x' % arp.sha[3],'%02x' % arp.sha[4],'%02x' % arp
43                  print(sIP,sMAC)
```

Python

```
21          arp = eth.data
22
23          if eth.type == 0x0806:
24              print("\n***********  ARP  ************")
25
26              print('Arp_hln : ', arp.hln)
27              print('Arp_hrd : ', arp.hrd)
28              print('Arp_pln : ', arp.pln)
29              print('Arp_pro : ', arp.pro)
30              print('\nOPcode : ', arp.op)
31              print('Sender MAC : ', ":".join(['%02x' % arp.sha[0],'%02x' % arp.sha[1],'%02x' % arp.sha[2],'%02x' % arp.sha[3],'%02x' % arp.sha[4],
32              print('Sender IP : ', str(int(arp.spa[0]))+"."+str(int(arp.spa[1]))+"."+str(int(arp.spa[2]))+"."+str(int(arp.spa[3])))  # ip
33              print('Target MAC : ', ":".join(['%02x' % arp.tha[0],'%02x' % arp.tha[1],'%02x' % arp.tha[2],'%02x' % arp.tha[3],'%02x' % arp.tha[4],
34              print('Target IP : ', str(int(arp.tpa[0]))+"."+str(int(arp.tpa[1]))+"."+str(int(arp.tpa[2]))+"."+str(int(arp.tpa[3])))  # ip
35
36              if arp.op == 1 :
37                  tIP = str(int(arp.tpa[0]))+"."+str(int(arp.tpa[1]))+"."+str(int(arp.tpa[2]))+"."+str(int(arp.tpa[3]))
38                  print(tIP)
39
40              if arp.op == 2 :
41                  sIP = str(int(arp.spa[0]))+"."+str(int(arp.spa[1]))+"."+str(int(arp.spa[2]))+"."+str(int(arp.spa[3]))
42                  sMAC = ":".join(['%02x' % arp.sha[0],'%02x' % arp.sha[1],'%02x' % arp.sha[2],'%02x' % arp.sha[3],'%02x' % arp.sha[4],'%02x' % arp
43                  print(sIP,sMAC)
```

Python

```
21          arp = eth.data
22
23          if eth.type == 0x0806:
24              print("\n***********  ARP  ************")
25
26  if arp.op == 1 :
27      tIP = str(int(arp.tpa[0]))+"."+str(int(arp.tpa[1]))+"."+st
28      print(tIP)
29
30  if arp.op == 2 :
31      sIP = str(int(arp.spa[0]))+"."+str(int(arp.spa[1]))+"."+st
32      sMAC = ":".join(['%02x' % arp.sha[0],'%02x' % arp.sha[1],'
33      print(sIP,sMAC)
34
35          if arp.op == 2 :
36              sIP = str(int(arp.spa[0]))+"."+str(int(arp.spa[1]))+"."+str(int(arp.spa[2]))+"."+str(int(arp.spa[3]))
37              sMAC = ":".join(['%02x' % arp.sha[0],'%02x' % arp.sha[1],'%02x' % arp.sha[2],'%02x' % arp.sha[3],'%02x' % arp.sha[4],'%02x' % arp
38              print(sIP,sMAC)
```

Request 일때 IP와
Reply 일때 IP를 저장

Python

```
45                  if tIP == sIP :
46                      print("\n######     right reply        ######\n")
47                      Right_Reply(sIP,sMAC)
48
49
50  if __name__=='__main__' :
51      main()
```

Request 일때 Target IP와
Reply 일때 Sender IP가 같으면
Right_Reply() 함수 실행

```
************  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  1
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.203
Target MAC :  00:00:00:00:00:00
Target IP :  192.168.35.1
192.168.35.1

************  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  1
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.203
Target MAC :  00:00:00:00:00:00
Target IP :  192.168.35.1
192.168.35.1

************  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:23:aa:de:99:91
Sender IP :  192.168.35.1
Target MAC :  00:24:d6:d1:61:7c
Target IP :  192.168.35.203
192.168.35.1 00:23:aa:de:99:91

#####     right reply       ######

{'192.168.35.1': '00:23:aa:de:99:91'}
```

Python

```
ARP         42 Who has 192.168.35.1? Tell 192.168.35.203
ARP         42 Who has 192.168.35.1? Tell 192.168.35.203
ARP         42 192.168.35.1 is at 00:23:aa:de:99:91
ARP         42 Who has 192.168.35.197? Tell 192.168.35.203
ARP         42 Who has 192.168.35.197? Tell 192.168.35.203
ARP         60 192.168.35.197 is at 40:5b:d8:ad:4a:c3
ARP         42 Who has 192.168.35.203? Tell 192.168.35.105
ARP         42 192.168.35.203 is at 00:24:d6:d1:61:7c
ARP         42 192.168.35.203 is at 00:24:d6:d1:61:7c
ARP         42 Who has 192.168.35.2? Tell 192.168.35.1
ARP         42 Who has 192.168.35.2? Tell 192.168.35.1
ARP         42 Who has 192.168.35.2? Tell 192.168.35.1
ARP         42 Who has 192.168.35.2? Tell 192.168.35.1
ARP         42 Who has 192.168.35.2? Tell 192.168.35.1
ARP         42 Who has 192.168.35.2? Tell 192.168.35.1
ARP         42 Who has 192.168.35.2? Tell 192.168.35.1
ARP         42 Who has 192.168.35.2? Tell 192.168.35.1
ARP         42 Who has 192.168.35.2? Tell 192.168.35.1
ARP         60 Who has 192.168.35.1? Tell 192.168.35.3
ARP         60 Who has 192.168.35.1? Tell 192.168.35.3
ARP         42 192.168.35.1 is at 00:23:aa:de:99:91
ARP         42 Who has 192.168.35.3? Tell 192.168.35.1
ARP         60 192.168.35.3 is at 00:24:d6:d1:61:7c
ARP         60 192.168.35.3 is at 00:24:d6:d1:61:7c
ARP         42 Who has 192.168.35.2? Tell 192.168.35.1
```

Python

```
************  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  1
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.203
Target MAC :  00:00:00:00:00:00
Target IP :  192.168.35.1
```

```
************  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:23:aa:de:99:91
Sender IP :  192.168.35.1
Target MAC :  00:24:d6:d1:61:7c
Target IP :  192.168.35.203
192.168.35.1 00:23:aa:de:99:91

#####     right reply       ######

{'192.168.35.1': '00:23:aa:de:99:91'}
```

```
OPcode :  2
Sender MAC :  00:23:aa:de:99:91
Sender IP :  192.168.35.1
Target MAC :  00:24:d6:d1:61:7c
Target IP :  192.168.35.203
192.168.35.1 00:23:aa:de:99:91

#####     right reply       ######

{'192.168.35.1': '00:23:aa:de:99:91'}
```

```
ARP        42 Who has 192.168.35.1? Tell 192.168.35.203
ARP        42 Who has 192.168.35.1? Tell 192.168.35.203
ARP        42 192.168.35.1 is at 00:23:aa:de:99:91
ARP        42 Who has 192.168.35.197? Tell 192.168.35.203
ARP        42 Who has 192.168.35.197? Tell 192.168.35.203
ARP        60 192.168.35.197 is at 40:5b:d8:ad:4a:c3
ARP        42 Who has 192.168.35.203? Tell 192.168.35.105
```

```
ARP        42 Who has 192.168.35.1? Tell 192.168.35.203
ARP        42 Who has 192.168.35.1? Tell 192.168.35.203
ARP        42 192.168.35.1 is at 00:23:aa:de:99:91
```

```
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        60 Who has 192.168.35.1? Tell 192.168.35.3
ARP        60 Who has 192.168.35.1? Tell 192.168.35.3
ARP        42 192.168.35.1 is at 00:23:aa:de:99:91
ARP        42 Who has 192.168.35.3? Tell 192.168.35.1
ARP        60 192.168.35.3 is at 00:24:d6:d1:61:7c
ARP        60 192.168.35.3 is at 00:24:d6:d1:61:7c
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
```

Python

```
***********  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  1
Sender MAC :  f0:6e:0b:7c:cf:67
Sender IP :  192.168.35.105
Target MAC :  00:24:d6:d1:61:7c
Target IP :  192.168.35.203
192.168.35.203

***********  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.203
Target MAC :  f0:6e:0b:7c:cf:67
Target IP :  192.168.35.105
192.168.35.203 00:24:d6:d1:61:7c

#####     right reply       ######

{'192.168.35.1': '00:23:aa:de:99:91', '192.168.35.197': '40:5b:d8:ad:4a:c3'
, '192.168.35.203': '00:24:d6:d1:61:7c'}

***********  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.203
Target MAC :  f0:6e:0b:7c:cf:67
Target IP :  192.168.35.105
192.168.35.203 00:24:d6:d1:61:7c

#####     right reply       ######

exist
```

```
ARP      42 Who has 192.168.35.1? Tell 192.168.35.203
ARP      42 Who has 192.168.35.1? Tell 192.168.35.203
ARP      42 192.168.35.1 is at 00:23:aa:de:99:91
ARP      42 Who has 192.168.35.197? Tell 192.168.35.203
ARP      42 Who has 192.168.35.197? Tell 192.168.35.203
ARP      60 192.168.35.197 is at 40:5b:d8:ad:4a:c3
ARP      42 Who has 192.168.35.203? Tell 192.168.35.105
ARP      42 192.168.35.203 is at 00:24:d6:d1:61:7c
ARP      42 192.168.35.203 is at 00:24:d6:d1:61:7c
ARP      42 Who has 192.168.35.2? Tell 192.168.35.1
ARP      42 Who has 192.168.35.2? Tell 192.168.35.1
ARP      42 Who has 192.168.35.2? Tell 192.168.35.1
ARP      42 Who has 192.168.35.2? Tell 192.168.35.1
ARP      42 Who has 192.168.35.2? Tell 192.168.35.1
ARP      42 Who has 192.168.35.2? Tell 192.168.35.1
ARP      42 Who has 192.168.35.2? Tell 192.168.35.1
ARP      42 Who has 192.168.35.2? Tell 192.168.35.1
ARP      42 Who has 192.168.35.2? Tell 192.168.35.1
ARP      60 Who has 192.168.35.1? Tell 192.168.35.3
ARP      60 Who has 192.168.35.1? Tell 192.168.35.3
ARP      42 192.168.35.1 is at 00:23:aa:de:99:91
ARP      42 Who has 192.168.35.3? Tell 192.168.35.1
ARP      60 192.168.35.3 is at 00:24:d6:d1:61:7c
ARP      60 192.168.35.3 is at 00:24:d6:d1:61:7c
ARP      42 Who has 192.168.35.2? Tell 192.168.35.1
```

Python

```
************  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  1
Sender MAC :  f0:6e:0b:7c:cf:67
Sender IP :  192.168.35.105
Target MAC :  00:24:d6:d1:61:7c
Target IP :  192.168.35.203
192.168.35.203

************  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.203
Target MAC :  f0:6e:0b:7c:cf:67
Target IP :  192.168.35.105
192.168.35.203 00:24:d6:d1:61:7c

######     right reply       ######

{'192.168.35.1': '00:23:aa:de:99:91', '192.168.35.197': '40:5b:d8:ad:4a:c3'
, '192.168.35.203': '00:24:d6:d1:61:7c'}

************  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.203
Target MAC :  f0:6e:0b:7c:cf:67
Target IP :  192.168.35.105
192.168.35.203 00:24:d6:d1:61:7c

######     right reply       ######

exist
```

```
ARP        42 Who has 192.168.35.1? Tell 192.168.35.203
ARP        42 Who has 192.168.35.1? Tell 192.168.35.203
ARP        42 192.168.35.1 is at 00:23:aa:de:99:91
ARP        42 Who has 192.168.35.197? Tell 192.168.35.203
ARP        42 Who has 192.168.35.197? Tell 192.168.35.203
ARP        42 Who has 192.168.35.203? Tell 192.168.35.105
ARP        42 192.168.35.203 is at 00:24:d6:d1:61:7c
ARP        42 192.168.35.203 is at 00:24:d6:d1:61:7c
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
ARP        60 Who has 192.168.35.1? Tell 192.168.35.3
ARP        60 Who has 192.168.35.1? Tell 192.168.35.3
ARP        42 192.168.35.1 is at 00:23:aa:de:99:91
ARP        42 Who has 192.168.35.3? Tell 192.168.35.1
ARP        60 192.168.35.3 is at 00:24:d6:d1:61:7c
ARP        60 192.168.35.3 is at 00:24:d6:d1:61:7c
ARP        42 Who has 192.168.35.2? Tell 192.168.35.1
```

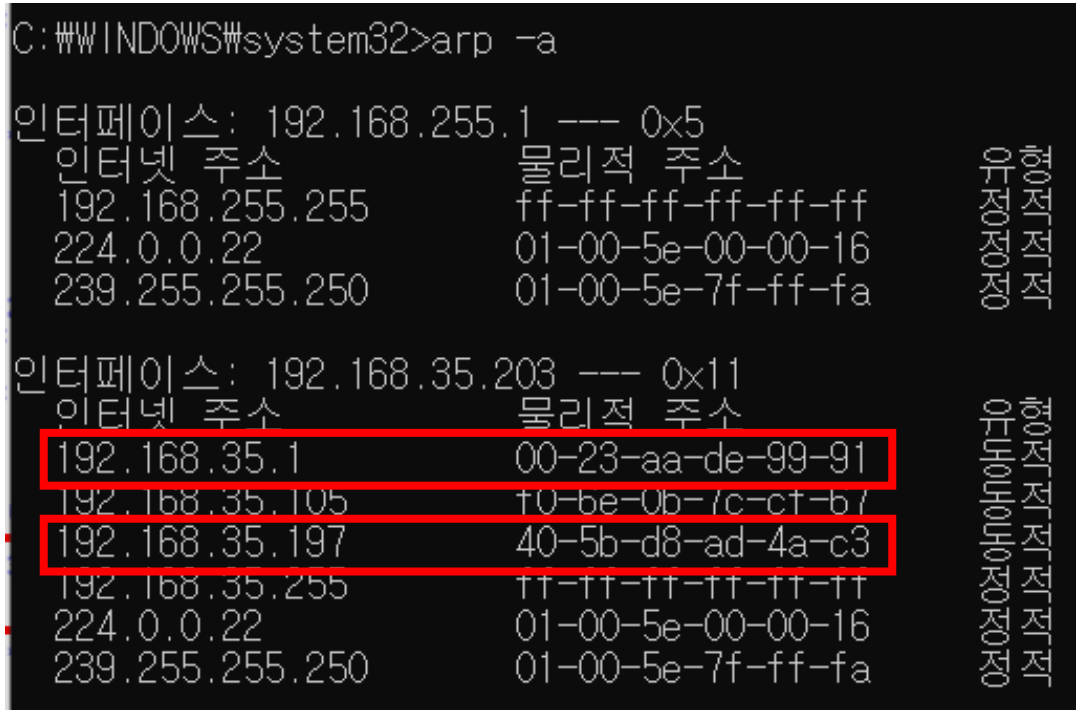
Python

```
************  ARP  ************
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.3
Target MAC :  00:23:aa:de:99:91
Target IP :  192.168.35.1
192.168.35.3 00:24:d6:d1:61:7c

#####     right reply       ######

{'192.168.35.1': '00:23:aa:de:99:91', '192.168.35.197':
 '40:5b:d8:ad:4a:c3', '192.168.35.203': '00:24:d6:d1:61
:7c', '192.168.35.3': '00:24:d6:d1:61:7c'}
```



Python

```
***********  ARP  ***********
Arp_hln :  6
Arp_hrd :  1
Arp_pln :  4
Arp_pro :  2048

OPcode :  2
Sender MAC :  00:24:d6:d1:61:7c
Sender IP :  192.168.35.3
Target MAC :  00:23:aa:de:99:91
Target IP :  192.168.35.1
192.168.35.3 00:24:d6:d1:61:7c

#####     right reply       ######

{'192.168.35.1': '00:23:aa:de:99:91', '192.168.35.197':
 '40:5b:d8:ad:4a:c3', '192.168.35.203': '00:24:d6:d1:61
:7c', '192.168.35.3': '00:24:d6:d1:61:7c'}
```

Q & A

Thank you ☺