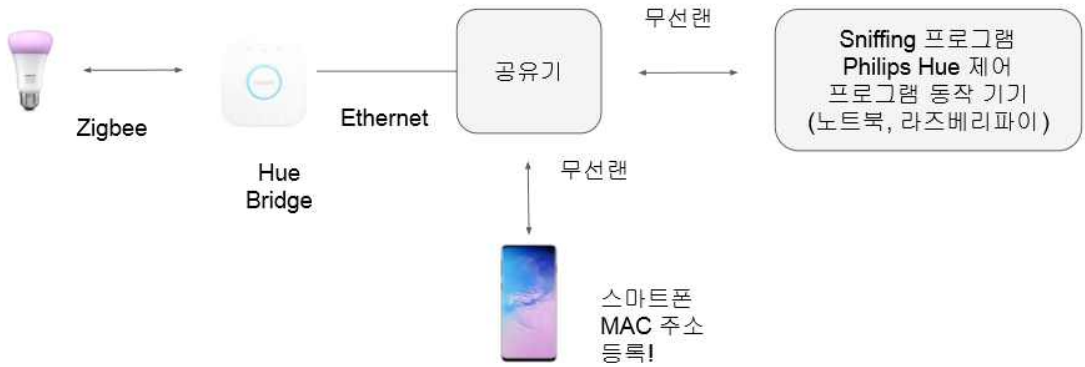
컴퓨터 네트워크

-필립스 휴 자동으로 켜고 끄기-

학번: 201404377
이름: 진승언

과제목표

실험환경



위와 같은 실험환경에서 ARP, DHCP, ping을 사용해서 전구를 제어한다. 스니핑을 하여 패킷을 캡처하고 패킷을 통해 내 스마트폰이 같은 공유기에 접속한 거라면(맥주소로비교) 전구를 켜고 내 핸드폰이 끊기면 전구를 끄면 된다. ARP도 같은 방식으로 ARP 브로드캐스트로 요청하여 해당 IP를 가진 기기가 같은 공유기 내에 있으면 그 기기는 자신의 맥주소를 나에게 보내주게 되고 그 패킷을 통해 내 스마트폰인지 판별하고 전구를 제어하면 됐다.

과제 해결방법

dhcp.py부터 보면 다음과 같다.

```
def main():
    raw_socket = socket.socket(socket.AF_PACKET,
            socket.SOCK_RAW,
            socket.ntohs(0x0003))
    my_macip = "d0b128275cf9"
    #hue bridge connect

    bridge = Bridge('192.168.0.218')
    bridge.connect()
    lights = bridge.lights
```

내 맥주소를 일단 저장해놓고 소켓과 전구를 제어할 휴 브릿지를 연결해 놓는다.

```
#sniffing
while True:
    recv_packet = raw_socket.recvfrom(5000)
    ethernet_protocol = struct.unpack('!6s6sH', (recv_packet[0])[:14])[2]
    if ethernet_protocol == 0x800:
        ip_protocol = struct.unpack('!BBHHHBBH4s4s', recv_packet[0][14:34])[6]
        if ip_protocol == 17:
            udp_src_port = struct.unpack('!H', (recv_packet[0])[34:34+2])[0]
            if udp_src_port == 68:
                packet_data = recv_packet[0][42:]
                #parsing phone mac ip
                come_macip = packet_data.hex()[56:68]
                print("Come MAC Ip => ", come_macip)
                if my_macip == come_macip:
                    print("My MAC IP !!!!!!!!!!")
                    print(recv_packet[0][0:])
                    print("HEX DATA : ", recv_packet[0][0:].hex()[0:])
                    phone_ip = struct.unpack('!BBBB', (recv_packet[0])[296:300])
                    phone_ip = str(phone_ip[0])+ '.'+ str(phone_ip[1]) + '.' + str(phone_ip[2]) +'.' +str(phone_ip[3])
                    print("PHONE => " , phone_ip)
print("PHONE => " , phone_ip)

while True:
    # 1 second sleep
    time.sleep(1)
    # ping to my phone ip that I parsed in packet
    status, result = subprocess.getstatusoutput("ping -c1 -w2 " + phone_ip)
    if status == 0:
        print('respone OK')
        lights[0].on = True
        lights[0].brightness = int(250)
        lights[0].xy = [float(0.9), float(0.9)]
        lights[1].on = True
        lights[1].brightness = int(250)
        lights[1].xy = [float(0.9), float(0.9)]
        lights[2].on = True
        lights[2].brightness = int(250)
        lights[2].xy = [float(0.9), float(0.9)]


    else :
        print("NOT respond")
        lights[0].on = False
        lights[0].brightness = int(0)
        lights[0].xy = [float(0.0), float(0.0)]
        lights[1].on = False
        lights[1].brightness = int(0)
        lights[1].xy = [float(0.0), float(0.0)]
        lights[2].on = False
        lights[2].brightness = int(0)
        lights[2].xy = [float(0.0), float(0.0)]
        break
```

그 후 DHCP 스니핑 하는 코드를 통해 같은 공유기 내에 접속하는 기기의 패킷을 캡처하고 내 핸드폰의 맥주소와 같은지 판별한 후 맞다면 해당 패킷의(내 스마트폰) IP주소를 통해 ping 요청을 보내어 응답이 안 올 때까지 전구를 켠다. 끊기면 끄면 된다.

arp.py

```
raw_socket = socket.socket(socket.PF_PACKET,
        socket.SOCK_RAW,
        socket.htons(0x0800))
raw_socket.bind(("ens33", socket.htons(0x800)))

protocol = 0x0806
source_mac = bytes.fromhex('5076af6f5991')
dest_mac = bytes.fromhex('ffffffffffff')

eth_header = struct.pack("!6s6sH", dest_mac, source_mac, protocol)

hardware_type = 0x0001
protocol_type = 0x0800
hardware_size = 0x06
protocol_size = 0x04

opcode = 1 #request

sender_mac = bytes.fromhex('5076af6f5991')
#d0b128275cf9
target_mac = bytes.fromhex('000000000000')
sender_ip = socket.inet_aton('192.168.0.164')
target_ip = socket.inet_aton('192.168.0.210')

arp_header = struct.pack("!HHBBH6s4s6s4s", hardware_type, protocol_type, hardware_size, protocol_size , opcode, sender_mac, sender_ip, target_mac, target_ip)
packet = eth_header + arp_header
raw_socket.sendall(packet)

#Receive
packet = RecvSocket.recvfrom(65535)
phone_ip = struct.unpack('!BBBB', packet[0][38:42] )
phone_ip = str(phone_ip[0])+ '.'+ str(phone_ip[1]) + '.' + str(phone_ip[2]) +'.' +str(phone_ip[3])
print("My PHONE IP IT : ", phone_ip)
```

먼저 ARP 프로토콜에 맞게 데이터 규격을 짜준다. 보내는 이의 ip와 맥주소는 내 노트북으로 하고 목적지 주소는 브로드캐스트로 보내기 위해 ffffffffffff로 해주고 ip주소는 내 폰의 ip로 한다. 목적지의 맥주소는 모르니 0으로 한다.
그 후 나에게 오는 패킷을 받아 내 폰에서 자신의 맥주소와 함께 보낸 응답(opcode가 2) 맞다면 arp에서 했던 것처럼 전구를 켠다. (이 방식은 dhcp.py와 동일하다.)

유튜브링크

dhcp.py : https://www.youtube.com/watch?v=2MT3q7LuFqc&feature=youtu.be

arp.py : https://www.youtube.com/watch?v=IKI-8rVOUrw&feature=youtu.be