

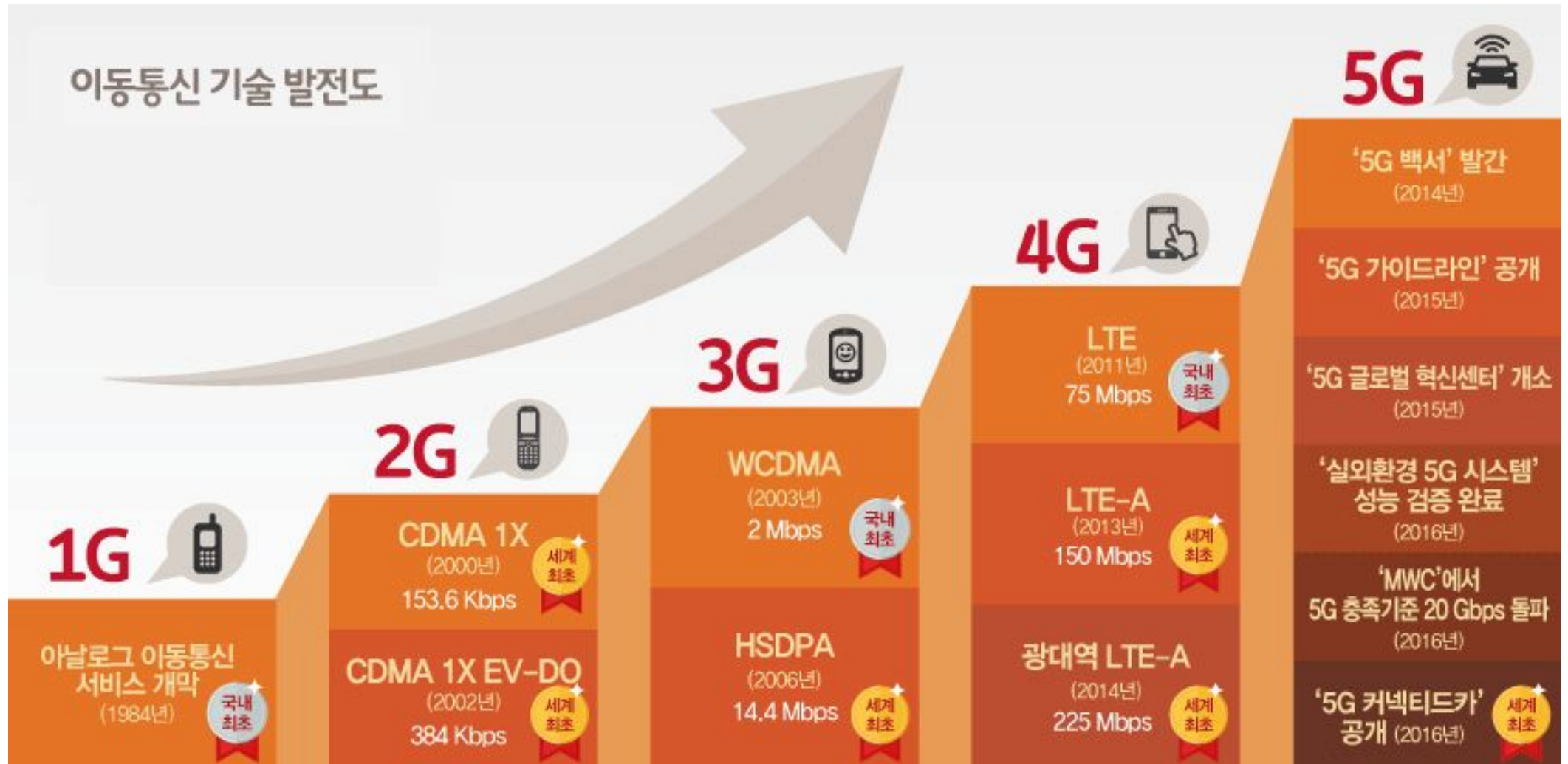
Network

CLI

저녁이 있는 프로젝트
오상훈
6 Hours, 1 Month

이동통신 기술 발전도

- ❖ **bps(bits per second)**, Bps (bytes per second)
 - Kbps (thousands of bps) or Mbps (millions of bps)



wireshark

- ❖ official : <https://www.wireshark.org/>
- ~\$ sudo apt install wireshark
- ❖ Top 10 Wireshark Filters // Filtering with Wireshark
 - <https://youtu.be/68t07-KOH9Y>
- ❖ Password sniffing
 - https://youtu.be/4_7A8lkp5Cc
- ❖ TCP based Robot Operating System protocol (TCPROS)
 - <https://www.wireshark.org/docs/dfref/t/tcpros.html>

wireshark

❖ 알아가기

➤ Filter Network 종류 선택 : eth0 or wlan0 # check ifconfig

➤ Filter Protocol 입력 : **http**

@ <http://www.danawa.com/> 접속 후 wireshark log 확인

The image shows a Wireshark network packet capture window. The top toolbar includes icons for file operations, network analysis, and search. The filter bar at the top shows 'http' selected. The packet list pane displays two packets: packet 2594 (HTTP GET) and packet 2598 (HTTP 302 Moved Temporarily). Packet 2598 is selected, and its details pane shows the following structure:

- Frame 2598: 432 bytes on wire (3456 bits), 432 bytes captured (3456 bits) on interface en0, id 0
- Ethernet II, Src: EFMNetwo_84:b9:10 (70:5d:cc:84:b9:10), Dst: Apple_1a:ac:ac (28:37:37:1a:ac:ac)
- Internet Protocol Version 4, Src: 125.209.222.142, Dst: 192.168.0.6
- Transmission Control Protocol, Src Port: 80, Dst Port: 51361, Seq: 1, Ack: 439, Len: 378
- Hypertext Transfer Protocol
- Line-based text data: text/html (7 lines)

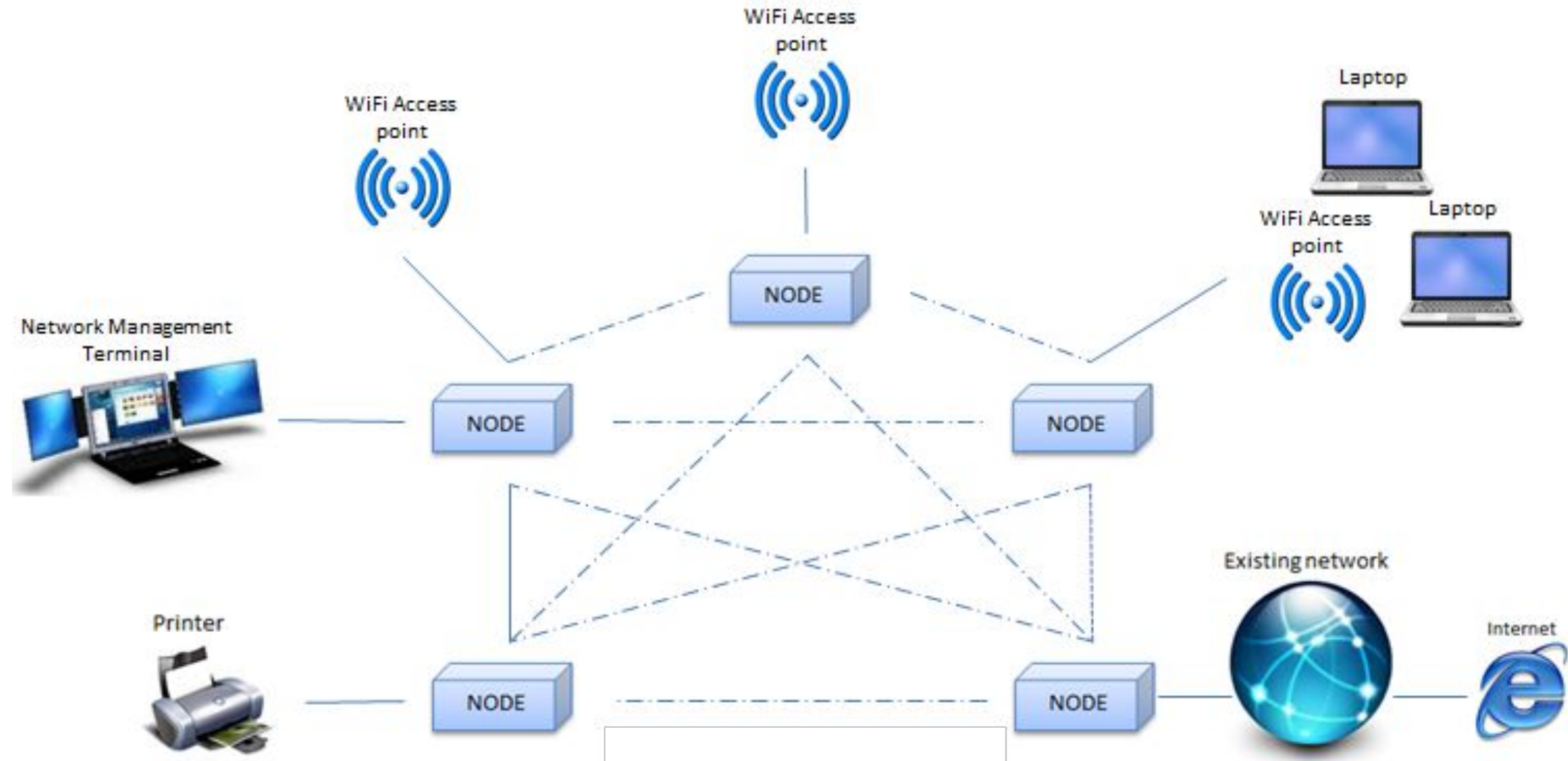
The packet bytes pane shows the raw data of the selected packet, with the Hypertext Transfer Protocol section highlighted in blue. The data is as follows:

```
0080 33 37 20 47 4d 54 0d 0a 43 f6 6e 74 65 6e 74 2d 37 GMT· Content-
0090 54 79 70 65 3a 20 74 65 78 74 2f 68 74 6d 6c 0d Type: te xt/html·
00a0 0a 54 72 61 6e 73 66 65 72 2d 45 6e 63 6f 64 69 ·Transfe r-Encodi
00b0 6e 67 3a 20 63 68 75 6e 6b 65 64 0d 0a 43 f6 6e ng: chun ked·Con
00c0 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c nection: keep-al
00d0 69 76 65 0d 0a 4c f6 63 61 74 69 6f 6e 3a 20 68 ive·Loc ation: h
00e0 74 74 70 73 3a 2f 2f 77 77 77 2e 6e 61 76 65 72 ttps://w ww.naver
00f0 2e 63 6f 6d 2f 0d 0a 56 61 72 79 3a 20 41 63 63 .com/·V ary: Acc
0100 65 70 74 2d 45 6e 63 6f 64 69 6e 67 2c 55 73 65 ept-Enco ding,Use
0110 72 2d 41 67 65 6e 74 0d 0a 0d 0a 38 61 0d 0a 3c r-Agent· ··8a··<
0120 68 74 6d 6c 3e 0d 0a 3c 68 65 61 64 3e 3c 74 69 html>··< head><ti
0130 74 6c 65 3e 33 30 32 20 46 6f 75 6e 64 3c 2f 74 tle>302 Found</t
0140 69 74 6c 65 3e 3c 2f 68 65 61 64 3e 0d 0a 3c 62 itle></h ead>··<b
0150 6f 64 79 3e 0d 0a 3c 63 65 6e 74 65 72 3e 3c 68 ody>··<c enter><h
0160 31 3e 33 30 32 20 46 6f 75 6e 64 3c 2f 68 31 3e 1>302 Fo und</h1>
0170 3c 2f 63 65 6e 74 65 72 3e 0d 0a 3c 68 72 3e 3c </center >··<hr><
0180 63 65 6e 74 65 72 3e 20 4e 57 53 20 3c 2f 63 65 center> NWS </ce
0190 6e 74 65 72 3e 0d 0a 3c 2f 62 6f 64 79 3e 0d 0a nter>··< /body>··
01a0 3c 2f 68 74 6d 6c 3e 0d 0a 0d 0a 30 0d 0a 0d 0a </html>· ··0··
```

At the bottom, the status bar indicates: Frame (432 bytes) De-chunked entity body (138 bytes).

Network

- ❖ 통신 위한 Terminal(단말), Link, Node 집합.
- ❖ 랜(LAN)이나 모뎀 등 통신 설비 갖춘 컴퓨터 이용 서로 연결시켜 주는 조직이나 " " " "



Network Command

- ❖ **ifconfig ?** : interface configurator, view and assign IP Address and Hardware / MAC address
 - ~\$ **ifconfig -a**
 - ~\$ ifdown eth0 # Disable
 - ~\$ ifup eth0 # Enable
 - ~\$ ifconfig eth1 192.168.50.5 netmask 255.255.255.0
- ❖ **host** : find name to IP or IP to name in IPv4 or IPv6 and also query DNS records
 - ~\$ less /etc/hosts # exit Press Key 'q'
 - ...
 - 127.0.0.1 localhost
 - ...
 - ~\$ host www.google.com
 - www.google.com has **address 172.217.24.132**
 - www.google.com has **IPv6 address 2404:6800:4004:806::2004**
 - ~\$ host -t CNAME www.redhat.com
 - www.redhat.com is an alias for ds-www.redhat.com.edgekey.net.
- ❖ 해 보기
 - 다른 URI도 확인
 - ~\$ host www.daum.net

Try - Network Command

~\$ host www.naver.com

or curl -vv naver.com

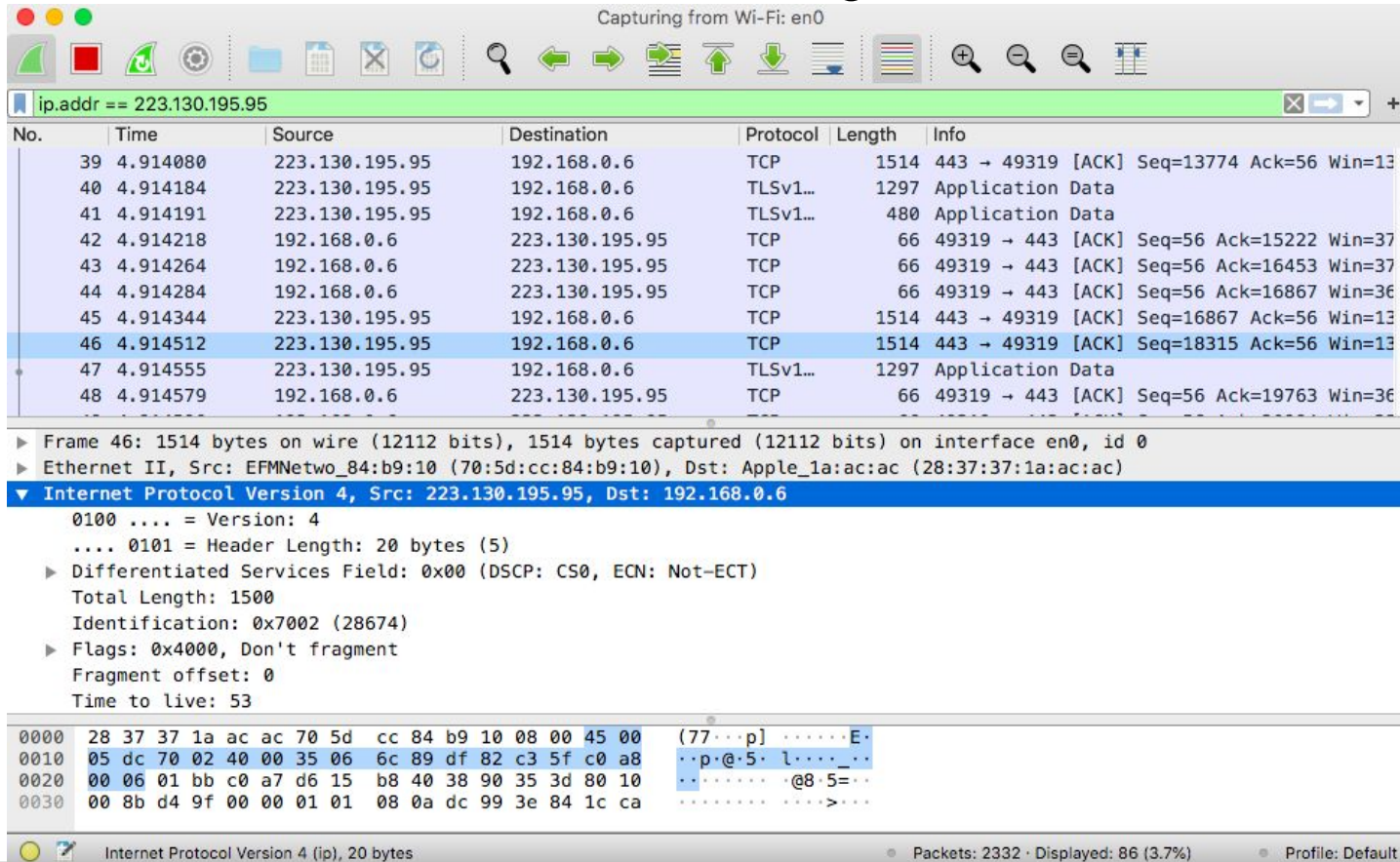
www.naver.com is an alias for www.naver.com.nheos.com.

www.naver.com.nheos.com has address 223.130.195.95

www.naver.com.nheos.com has address 223.130.195.200

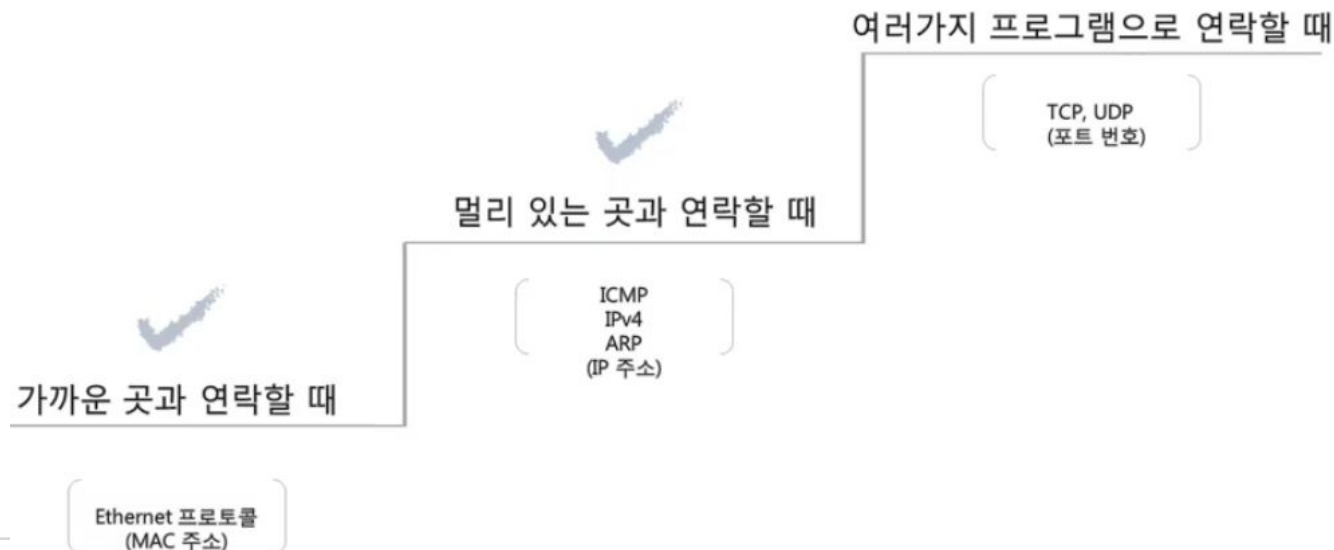
↩ wireshark filter : ip.addr == 223.130.195.95

@ <https://www.naver.com/> 접속 후 wireshark log 확인



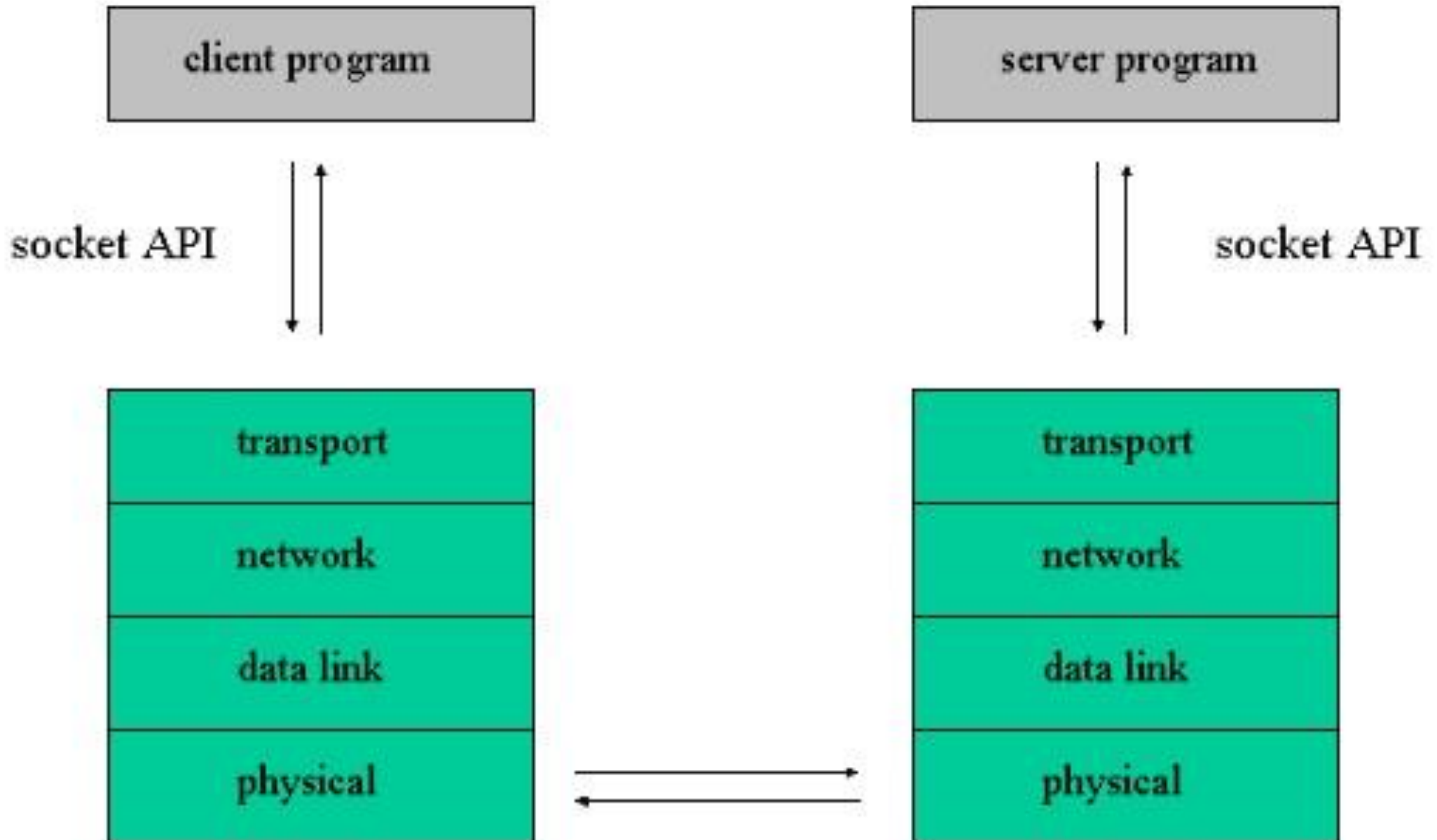
What's Network Protocol !

- ❖ Communication protocol is a system of rules that allow two or more.
 - 이 곳에서 저 곳으로 어떤 것을 보내는데 필요한 양식.(택배, 무역, 전화 등)
 - 네트워크 내에서 특정 사용자 찾는데 필요.
- ❖ 연결 방식 분류
 - Star : 중앙 장비에 모두 연결
 - Mesh : 그물 같이 서로 연결
 - Tree : 나무 구조로 연결
 - Ring, Bus 등
- ❖ 데이터 교환 방식 분류
 - Uni-Cast : 1:1 통신
 - Multi-Cast : 1:N 통신
 - Broad-Cast : 연결된 모두와 통신



Network Service - Client & Server

- ❖ 365/24 응답자와 규칙 필요



OSI 7 계층 모델

❖ 계층별 프로토콜

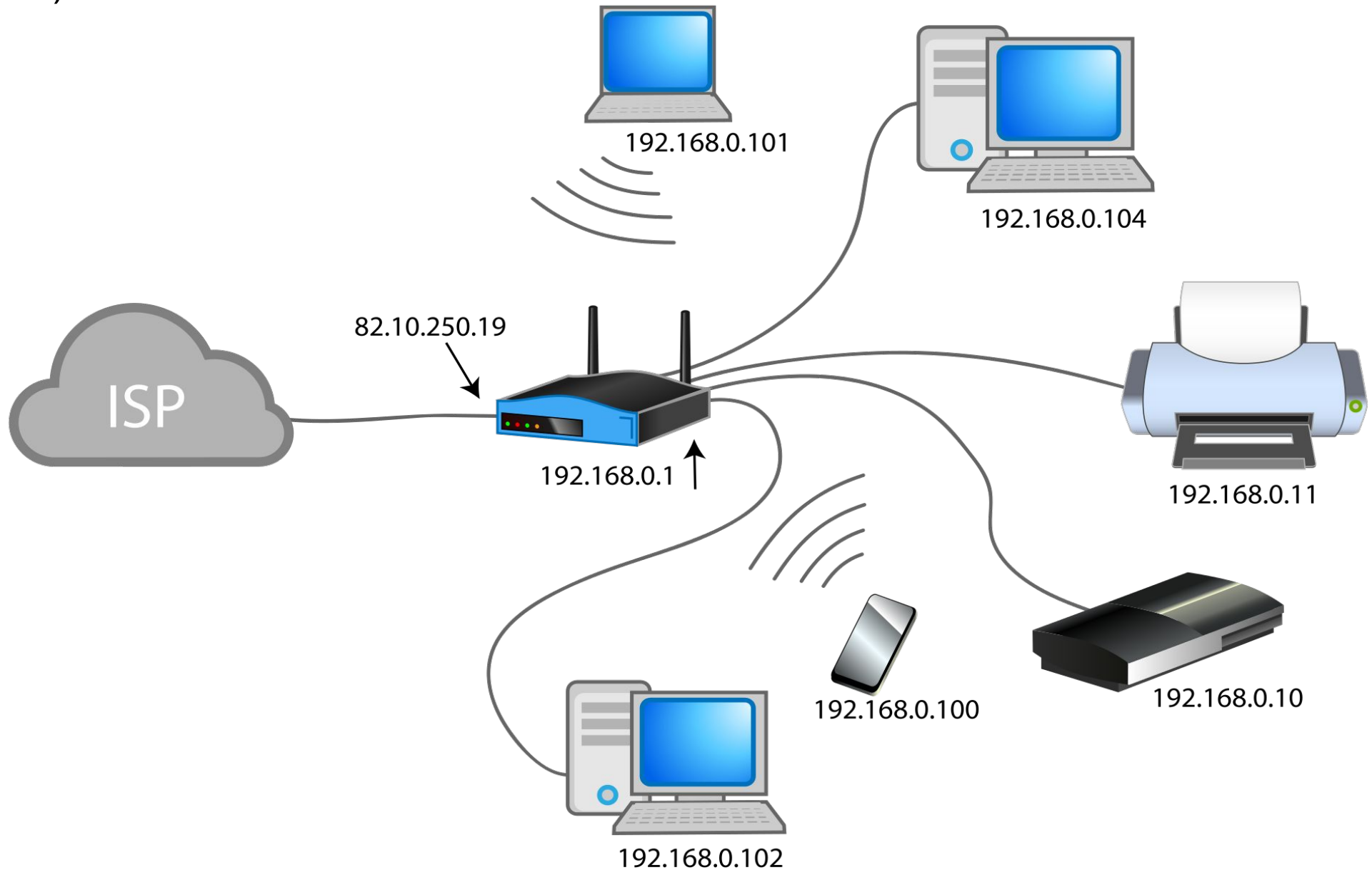
| | | |
|-----|--------|---|
| 7계층 | 응용 | HTTP, SMTP, IMAP, POP, SNMP, FTP, TELNET, SSH |
| 6계층 | 표현 | SMB, AFP, XDR |
| 5계층 | 세션 | NetBIOS |
| 4계층 | 전송 | TCP, UDP, SPX |
| 3계층 | 네트워크 | IP, ICMP, IGMP, X.25, CLNP, ARP, RARP, BGP, OSPF, RIP, IPX, DDP |
| 2계층 | 데이터 링크 | 이더넷, 토큰링, PPP, HDLC, 프레임 릴레이, ISDN, ATM, 무선랜, FDDI |
| 1계층 | 물리 | 전선, 전파, 광섬유, 동축케이블, 도파관, PSTN, 리피터, DSU, CSU, 모뎀 |

Network - Linux Command

- ❖ Refer : Linux Network Configuration and Troubleshooting Commands
 - <https://www.tecmint.com/linux-network-configuration-and-troubleshooting-commands/>
- ❖ ping(Packet INternet Groper) : test connectivity between two nodes
 - ~\$ ping 4.2.2.2
 - PING 4.2.2.2 (4.2.2.2): 56 data bytes
 - 64 bytes from 4.2.2.2: icmp_seq=0 ttl=53 time=93.218 ms
 - ...
 - ~\$ ping -c 5 www.tecmint.com
 - PING www.tecmint.com (104.26.2.23): 56 data bytes
 - 64 bytes from 104.26.2.23: icmp_seq=0 ttl=52 time=137.487 ms
 - ...
- ↵ wireshark filter arp 입력.
 - ~\$ ping 192.168.0.19 # 같은 IP 내 요청
 - ...
- ↵ wireshark log 확인

Route - Small(LAN)

- ❖ Local Area Network(LAN)
ex) StarCraft 게임 시 LAN UDP 사용



NIC(network interface controller)

- ❖ MAC address(media access control address) : unique identifier assigned to network interfaces

~\$ ifconfig

en0: flags=8863<UP,BROADCAST,SMART,RUNNING,MULTICAST> mtu 1500

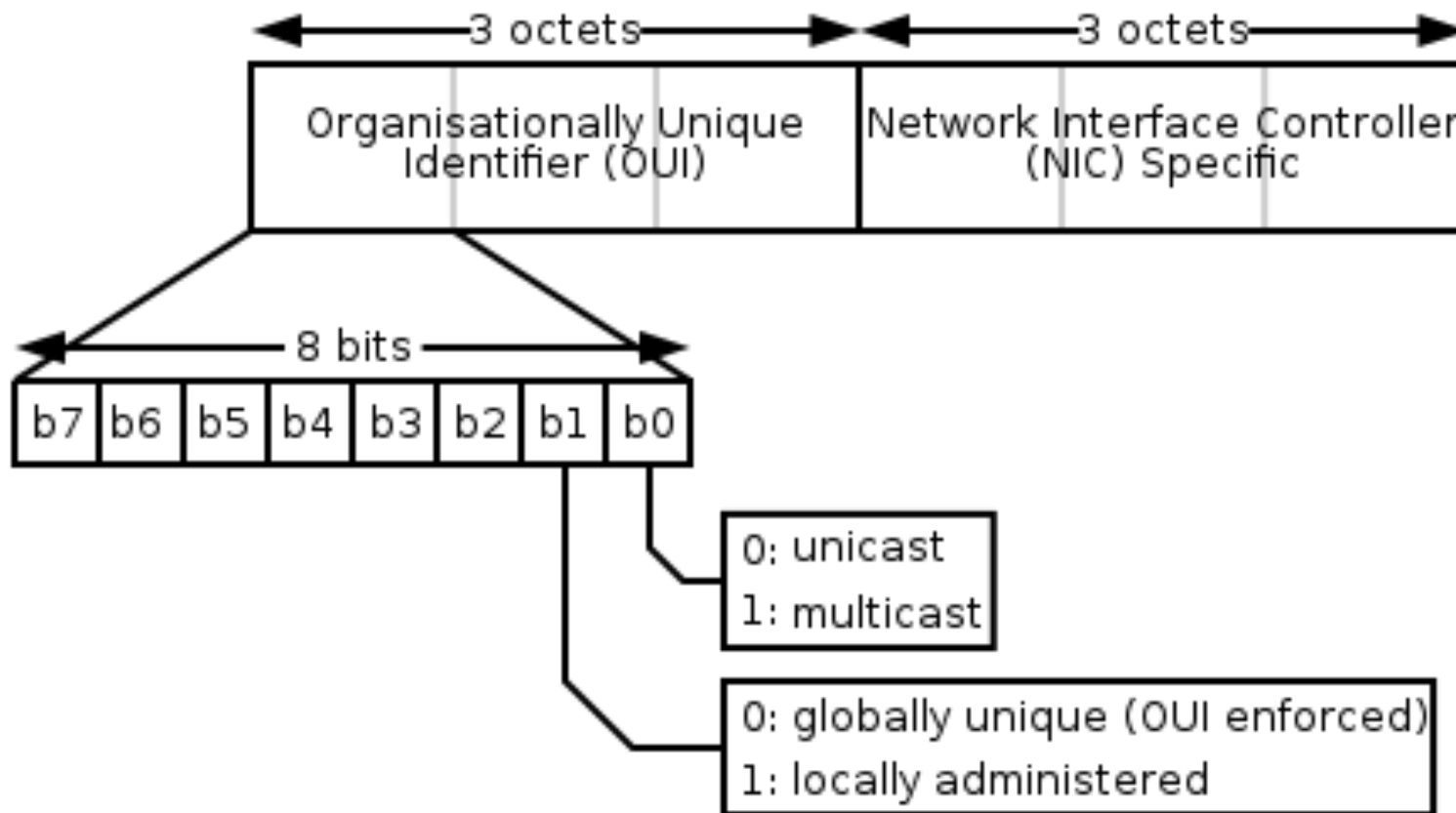
ether 28:37:37:1a:ac:ac

제조업체 식별번호

..

@ <https://ouilookup.com/> > search '**28:37:37:1a:ac:ac**'

결과 확인



ARP

- ❖ Address Resolution Protocol : MAC address 확인 위해 3계층 장비(내부망) 통신, 같은 네트워크서만 사용.

~\$ arp -a

ARP record table

? (192.168.0.1) at 70:5d:cc:84:b9:10 on en0 ifscope [ethernet]

? (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]

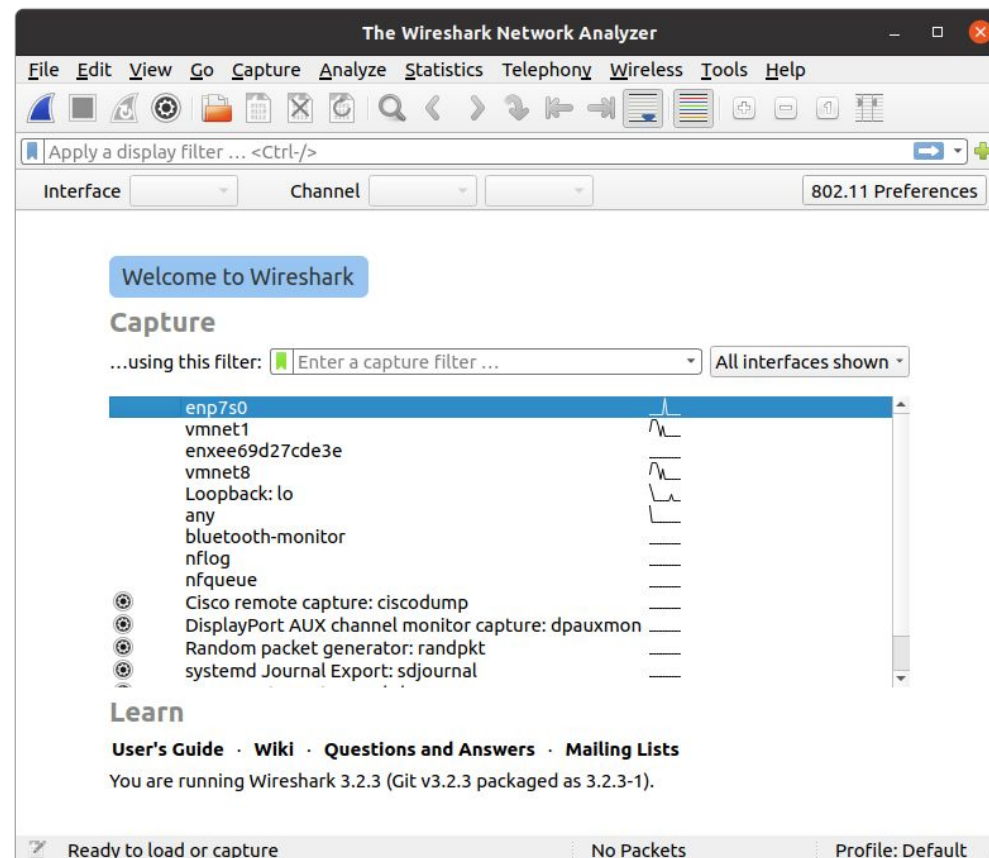
? (239.255.255.250) at 1:0:5e:7f:ff:fa on en0 ifscope permanent [ethernet]

↵ wireshark filter **arp** 입력.

~\$ ping 192.168.0.19 # 같은 IP 내 요청

...

↵ wireshark log 확인



ICMP

- ❖ ICMP(Internet Control Message Protocol) : 특정 대상과 통신 잘되는지 확인
 - 데이터 캡슐화 : 데이터 단위화(패킷), 제어와 사용자 데이터(페이로드)로 구성
 - Ethernet Card MAC address

↓ wireshark filter **icmp** 입력.

~\$ ping 192.168.0.19 -l 7000 -f

같은 IP 대역 내 요청

...

↓ wireshark log 확인

❖ 알아보기

- 가상망 관리자 확인

~\$ traceroute 8.8.8.8

traceroute to 8.8.8.8 (8.8.8.8), ...

1 **192.168.0.1** (192.168.0.1) ...

...

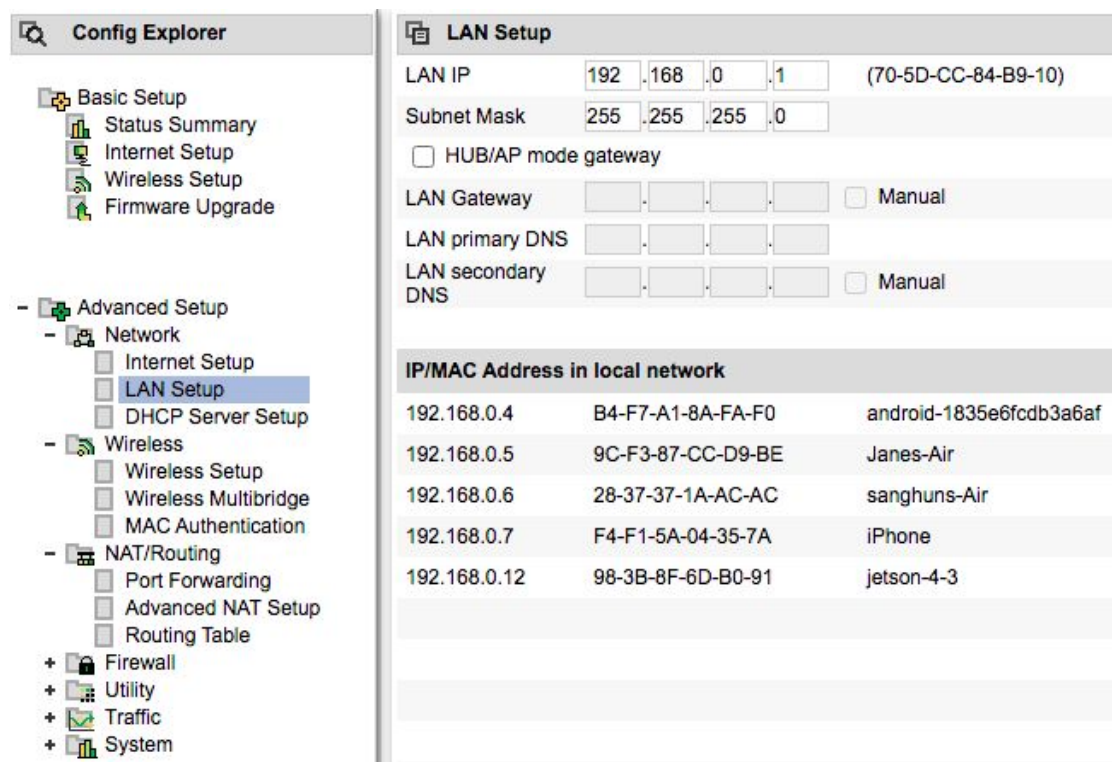
~\$

@ <http://192.168.0.1/>

id : admin

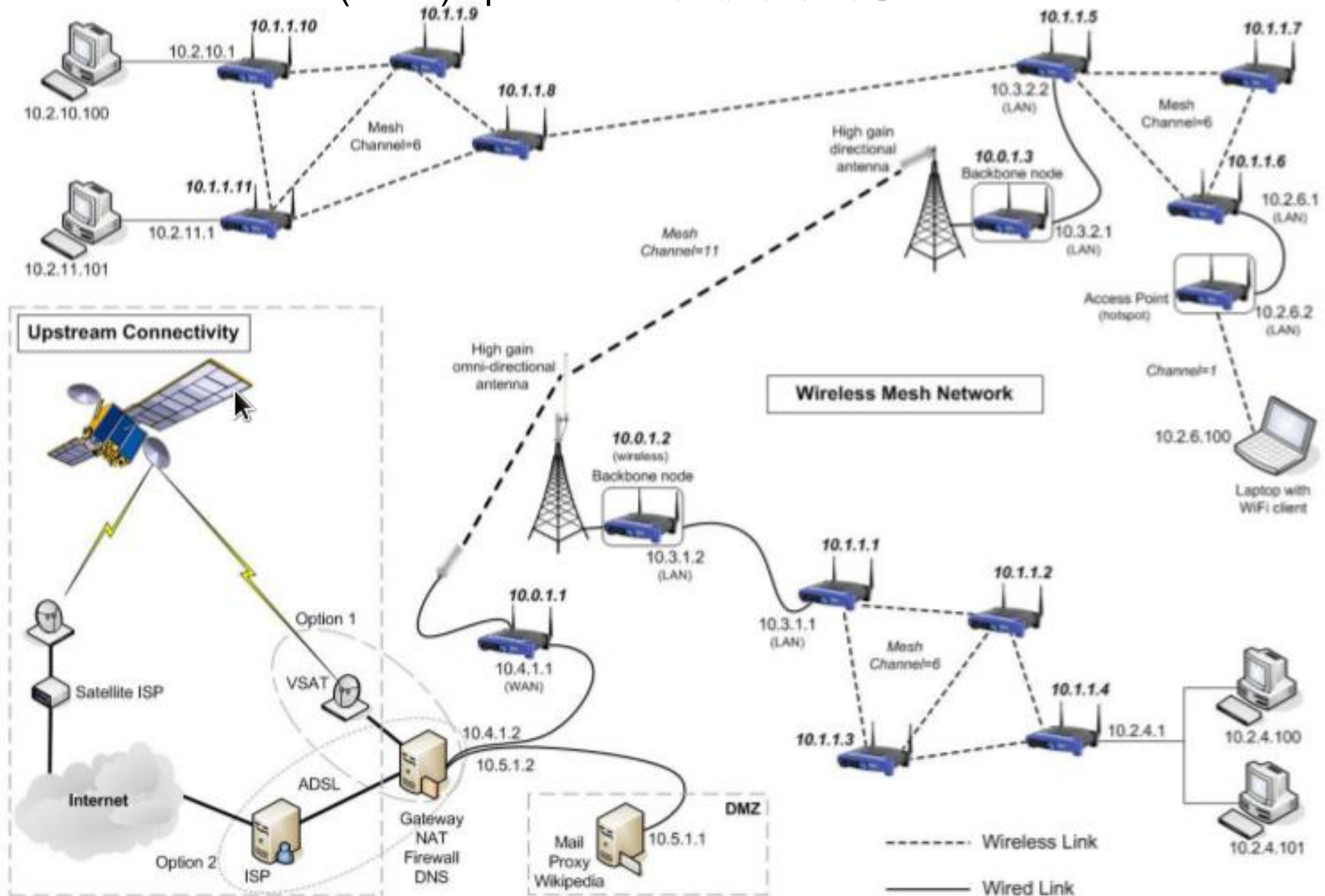
대개

pw : *****



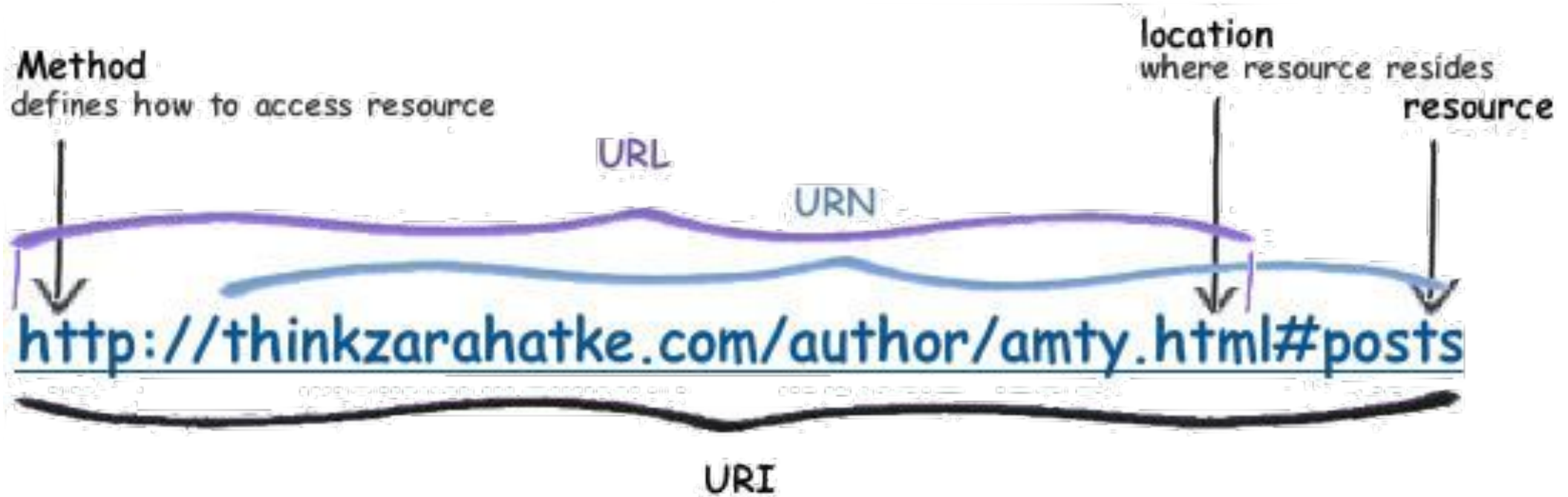
Big(WAN)

- ❖ Wire Area Network(WAN) : packet 단위 데이터 이동.



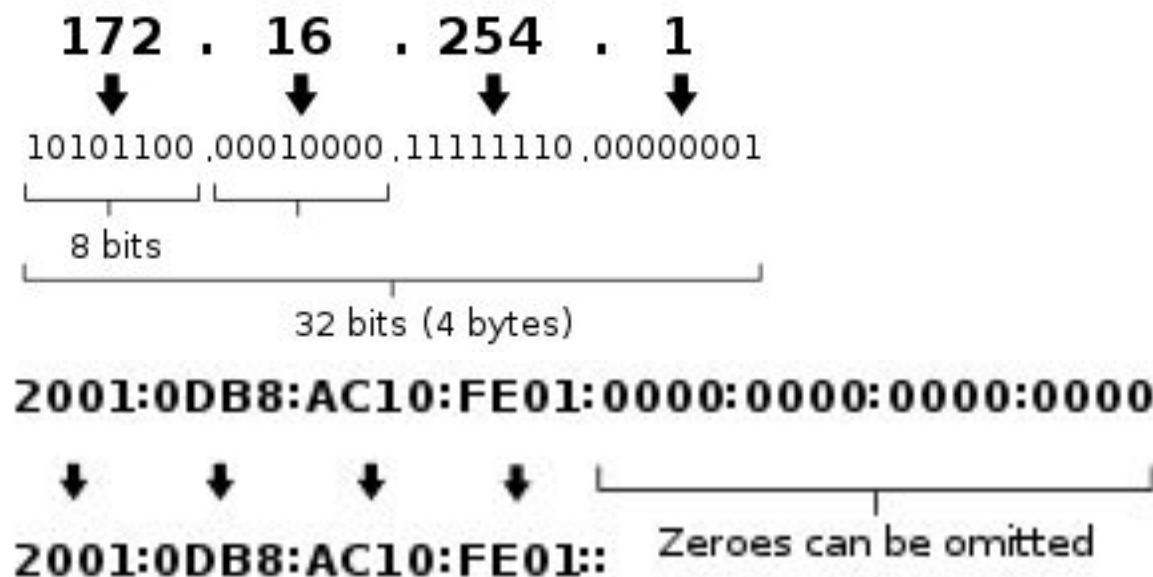
URI vs URL vs URN

❖ URI : 인간 위한 표기



❖ IP address : 컴퓨터 위한 표기

- IPv4 addresses : 2^{32}
ex) 192.168.0.255
- IPv6 addresses : 2^{128}
ex) fe80::41:295e:2140:4b30



IP address

❖ 특수한 IP

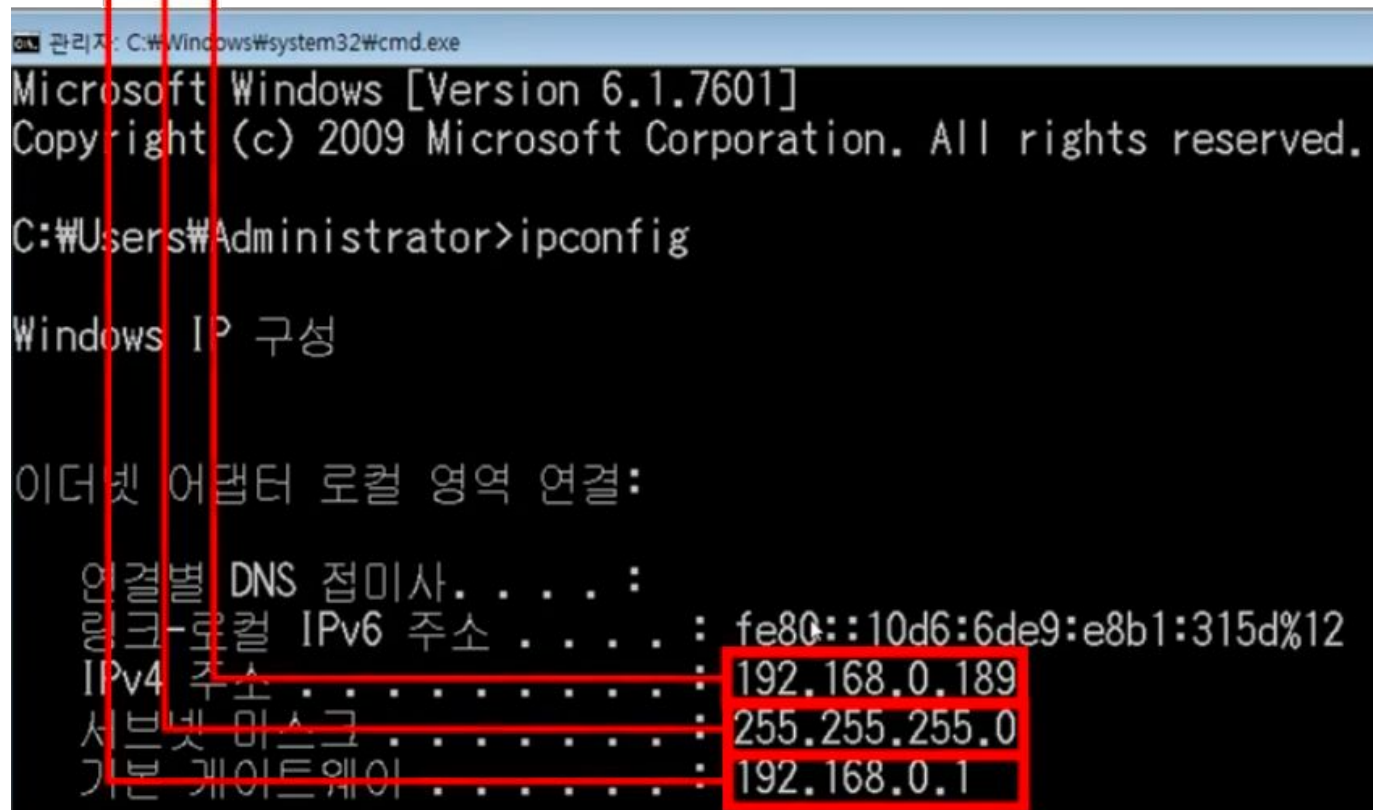
- Wildcard : 0.0.0.0
- 내 자신 : 127.0.0.1

~\$ ifconfig # or ip addr

→ IPv4 주소 : 현재 PC에 할당된 IP주소

→ 서브넷 마스크 : IP 주소에 대한 네트워크의 대역을 규정하는 것

→ 게이트웨이 주소 : 외부와 통신할 때 사용하는 네트워크의 출입구



The screenshot shows a Windows command prompt window titled "관리자: C:\Windows\system32\cmd.exe". The user has entered the command "ipconfig". The output shows the configuration for the "Ethernet adapter 로컬 영역 연결:" (Ethernet adapter Local Area Connection:). The output is as follows:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ipconfig

Windows IP 구성

이더넷 어댑터 로컬 영역 연결:

    연결별 DNS 접미사. . . . . : 
    링크-로컬 IPv6 주소 . . . . . : fe80::10d6:6de9:e8b1:315d%12
    IPv4 주소 . . . . . : 192.168.0.189
    서브넷 마스크 . . . . . : 255.255.255.0
    기본 게이트웨이 . . . . . : 192.168.0.1
```

Red arrows and boxes are used to highlight specific information in the output:

- A red arrow points from the text "IPv4 주소 : 현재 PC에 할당된 IP주소" to the "IPv4 주소" line in the output.
- A red arrow points from the text "서브넷 마스크 : IP 주소에 대한 네트워크의 대역을 규정하는 것" to the "서브넷 마스크" line in the output.
- A red arrow points from the text "게이트웨이 주소 : 외부와 통신할 때 사용하는 네트워크의 출입구" to the "기본 게이트웨이" line in the output.
- The values "192.168.0.189", "255.255.255.0", and "192.168.0.1" are enclosed in red boxes.

Route

- ❖ 네트워크 장비 간 연결 하는 다리 역할자.
 - 보내야 하는 네트워크 정보 보유.
 - NAT(Network Address Translation) Table에 기록

❖ route : shows and manipulate ip routing table

~\$ route add -net 10.10.10.0/24 gw 192.168.0.1

del

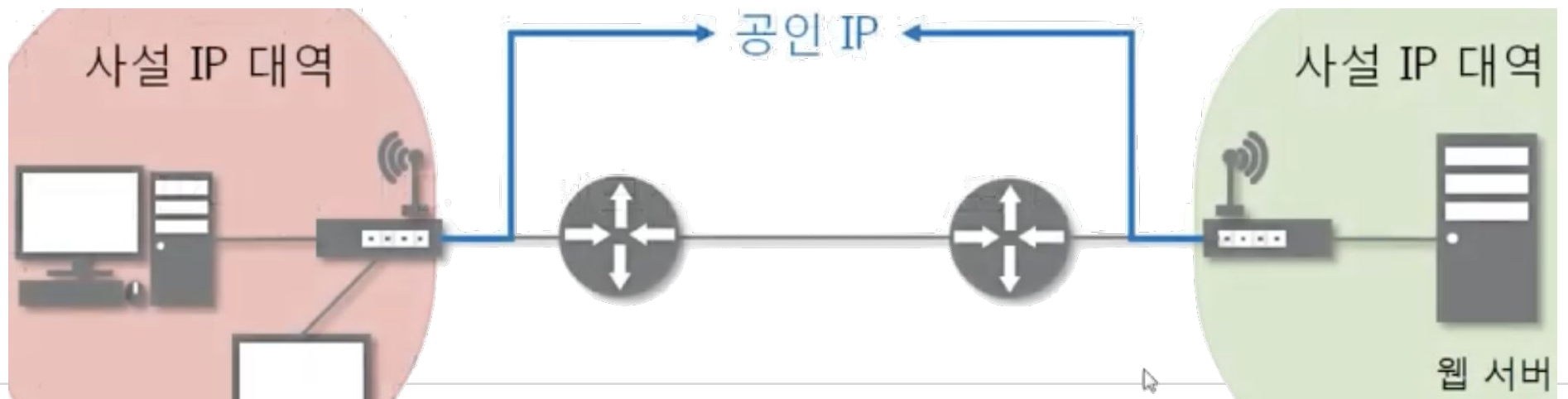
~\$ route add default gw 192.168.0.1

~\$ netstat -r

Kernel IP routing table

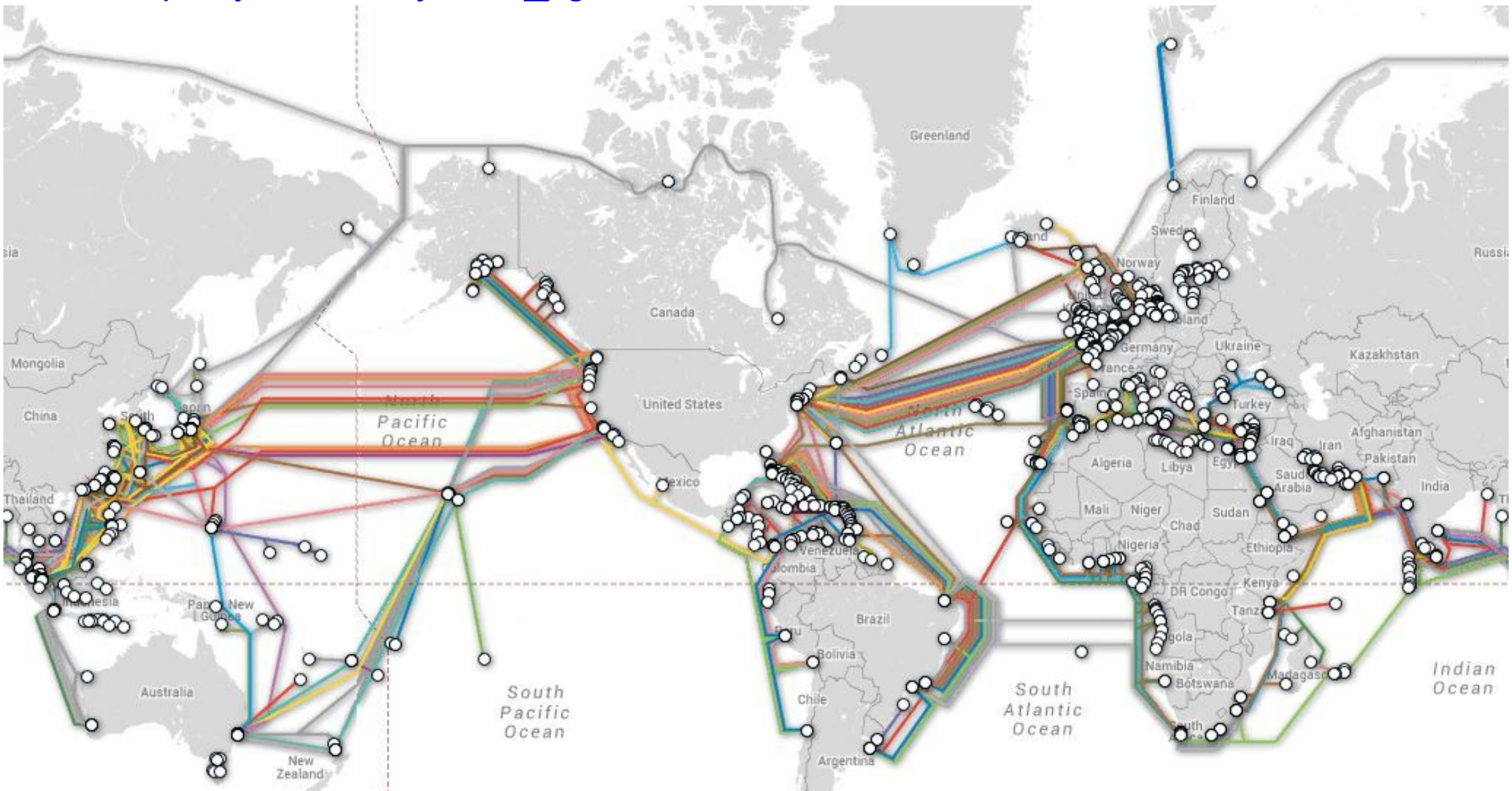
| Destination | Gateway | Genmask | Flags | MSS | Window | irtt | Iface |
|-------------|-------------|---------------|-------|-----|--------|------|--------|
| default | 192.168.0.1 | 0.0.0.0 | UG | 0 | 0 | 0 | enp7s0 |
| link-local | 0.0.0.0 | 255.255.0.0 | U | 0 | 0 | 0 | enp7s0 |
| 192.168.0.0 | 0.0.0.0 | 255.255.255.0 | U | 0 | 0 | 0 | enp7s0 |

...



Submarine Cable Map

- ❖ <https://www.submarinecablemap.com/>
- ❖ https://youtu.be/RyR-5O_jrgw



Network Command

❖ traceroute ? : network troubleshooting utility, using response icmp

~\$ sudo apt install traceroute

~\$ traceroute 8.8.8.8

traceroute to 8.8.8.8 (8.8.8.8), 64 hops max, 52 byte packets

1 **192.168.0.1** (192.168.0.1) 4.464 ms 1.209 ms 1.019 ms

...

5 112.190.109.225 (112.190.109.225) 2.750 ms

112.190.108.65 (112.190.108.65) 2.139 ms

112.190.110.229 (112.190.110.229) 2.093 ms

6 * * *

...

10 dns.google (8.8.8.8) 37.904 ms 36.179 ms 32.229 ms

~\$

@ <http://192.168.0.1/>

access your gateway

❖ 해 보기

~\$ traceroute www.google.com

~\$ traceroute www.daum.net

~\$ traceroute www.rapa.or.kr

Port Forward

❖ NAT 응용 : 특정 IP와 Port를 다른 IP와 Port 변환

~\$ ~/hello_web\$ python3 manage.py runserver 0:**8090** # set your own port number

@ <https://www.myip.com/>

→ if ip 121.113.52.129

@ <http://121.113.52.129:80>

Config Explorer

- Basic Setup
 - Status Summary
 - Internet Setup
 - Wireless Setup
 - Firmware Upgrade
- Advanced Setup
 - Network
 - Wireless
 - NAT/Router
 - Port Forwarding**
 - Advanced NAT Setup
 - Routing Table
 - Firewall
 - Utility
 - Traffic
 - System

Port Forwarding User-defined rules

| Prio. | Name(User) | LAN IP | External port | Internal port | Del |
|-------|------------|-------------|---------------|---------------|--------------------------|
| 1 | test_web | 192.168.0.6 | TCP(8848) | TCP(8848) | <input type="checkbox"/> |

+ Add new rule

Rule Name: Port Forward(User) ☐ Rule disable

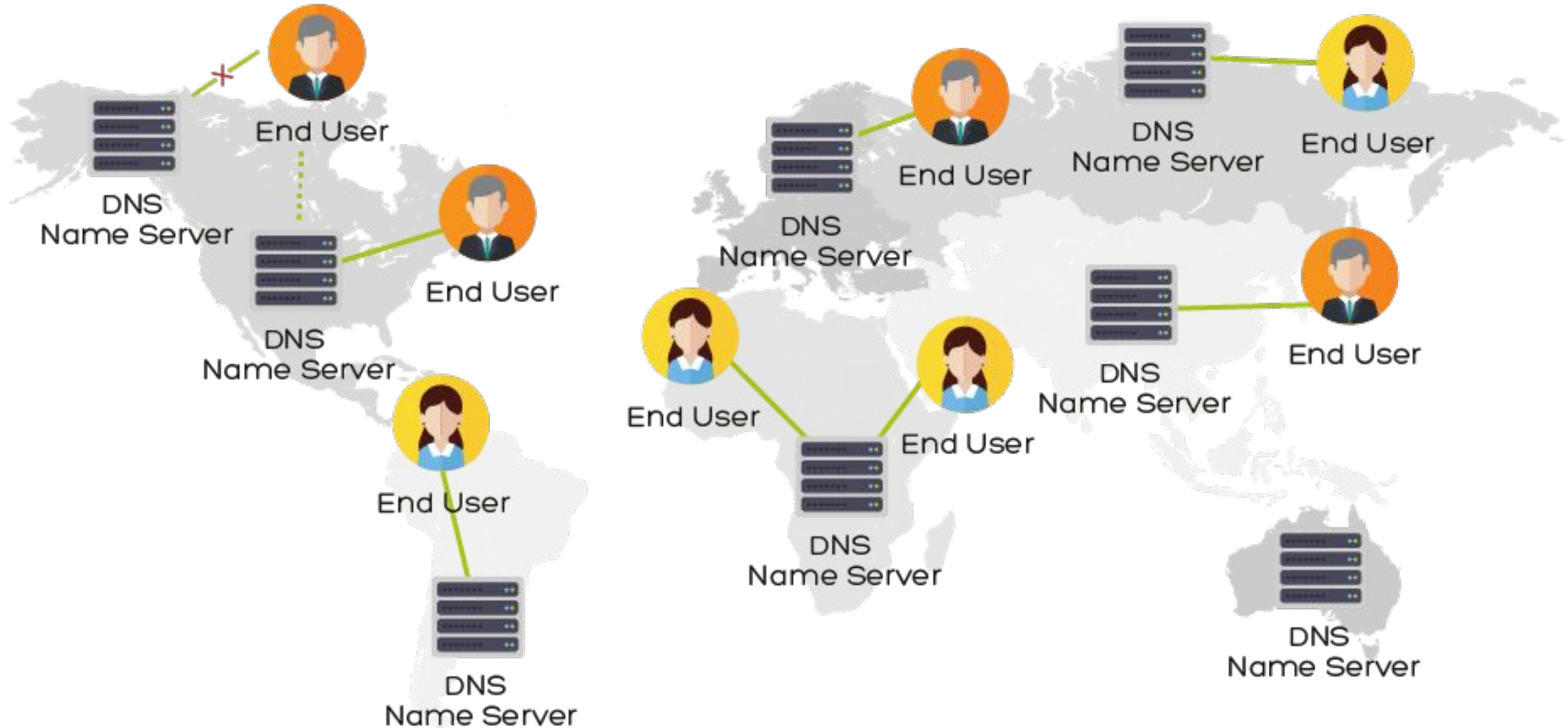
Internal IP: ☐ Currently connected IP address

Protocol: External port: ~ Internal port: ~

PC<-Save PC->Restore Choose File No file chosen New Apply Cancel

DNS(Domain Name System)

- ❖ 도메인 이름을 네트워크 주소 변환 역할 : 컴퓨터 Number URL 사용
- ❖ DNS Server : <https://www.lifewire.com/free-and-public-dns-servers-2626062>



Set Static IP

❖ refer : <https://danielmiessler.com/study/manually-set-ip-linux/>

```
~$ sudo nmcli device wifi connect AI_Multicopter_5G password rapa1\#
```

```
~$ nmcli device status
```

```
DEVICE TYPE    STATE    CONNECTION
```

```
wlan0  wifi    connected  skiptimeB911 1
```

```
...
```

```
~$ ifconfig wlan0
```

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
```

```
    inet 192.168.0.12 netmask 255.255.255.0 broadcast 192.168.0.255
```

```
...
```

```
~$ ifconfig wlan0 192.168.0.12 netmask 255.255.255.0 up
```

```
~$ ifconfig wlan0
```

```
~$ ping 192.168.0.1 # check gateway
```

```
~$ route add default gw 192.168.0.1
```

```
~$ ping google.com
```

```
~$ systemctl restart network
```

```
~$ ifconfig wlan0
```

```
~$ route
```

Port

❖ Program(Process) 간 통신 위해 사용

➤ Well-Known Port, Registered Port, Dynamic Port

↓ wireshark filter **tcp.port == 443** 입력.

@ www.google.com

↓ wireshark log 확인

~\$ sudo lsof -i -P -n

| COMMAND | PID | USER | FD | TYPE | DEVICE | SIZE/OFF | NODE | NAME |
|-----------|-------|------|----|------|---|----------|------|---------------|
| systemd-r | 708 | ... | | UDP | 127.0.0.53:53 | | | |
| systemd-r | 708 | ... | | TCP | 127.0.0.53:53 | | | |
| ... | | | | | | | | |
| chrome | 2230 | ... | | | 192.168.0.146:45160->142.250.199.65:443 | | | (ESTABLISHED) |
| chrome | 2296 | ... | | | 192.168.0.146:50168->40.81.94.43:443 | | | (ESTABLISHED) |
| ... | | | | | | | | |
| remmina | 24369 | ... | | | 192.168.0.146:37204->192.168.0.178:5901 | | | (CLOSE_WAIT) |
| ssh | 26494 | | | TCP | 192.168.0.146:55138->192.168.0.178:22 | | | (ESTABLISHED) |

~\$

Kill Process

❖ 알아보기

➤ start Django Server

...

Django version 3.1.2, using settings 'web_project.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CONTROL-C.

...

~\$ sudo lsof -i -P -n

| COMMAND | PID | USER | FD | TYPE | DEVICE | SIZE/OFF | NODE | NAME |
|---------|--------------|------|-----|----------------------------------|---------------|----------|------|------|
| python3 | 40843 | ... | TCP | 127.0.0.1:38803 | (LISTEN) | | | |
| python3 | 40843 | ... | | 127.0.0.1:38803->127.0.0.1:40368 | (ESTABLISHED) | | | |

~\$ kill -9 **40843**

❖ 해 보기

➤ kill ssh Server

~\$ lsof -nP -iTCP

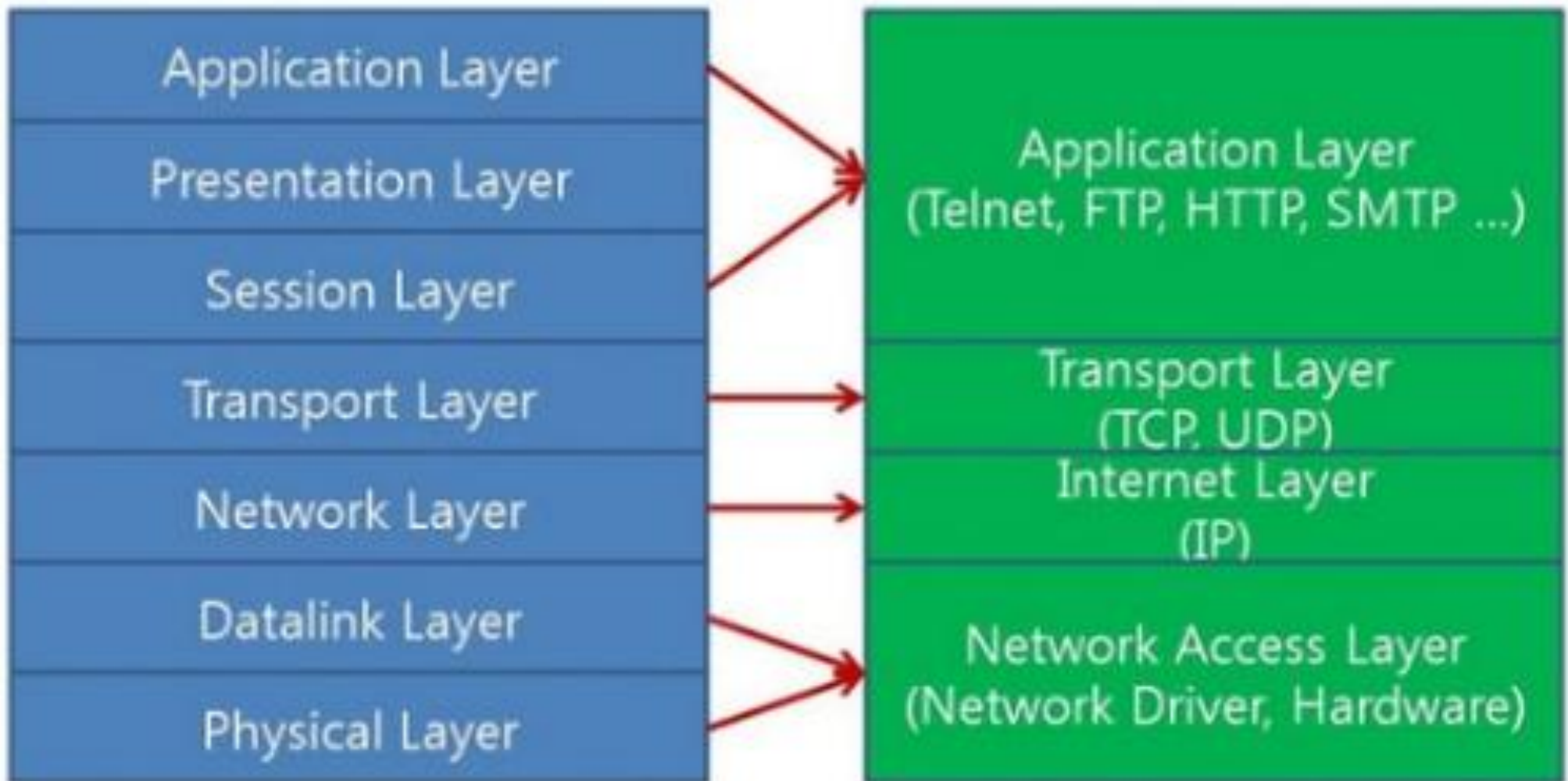
or lsof -t -i:8000

~\$ kill -9 <pid>

or kill \$(sudo lsof -t -i:8000)

OSI 7계층(Open System Interconnection 7 Layer)

- ❖ 1984년 통신 위한 이상적 프로토콜 모델 발표.
- ❖ 기존 SNA, 토큰링, FDDI 다양한 네트워크 사이 연결 호환성 위해 등장.
- ❖ Transport Layer
 - TCP(Transmission Control Protocol), UDP(User Datagram Protocol)

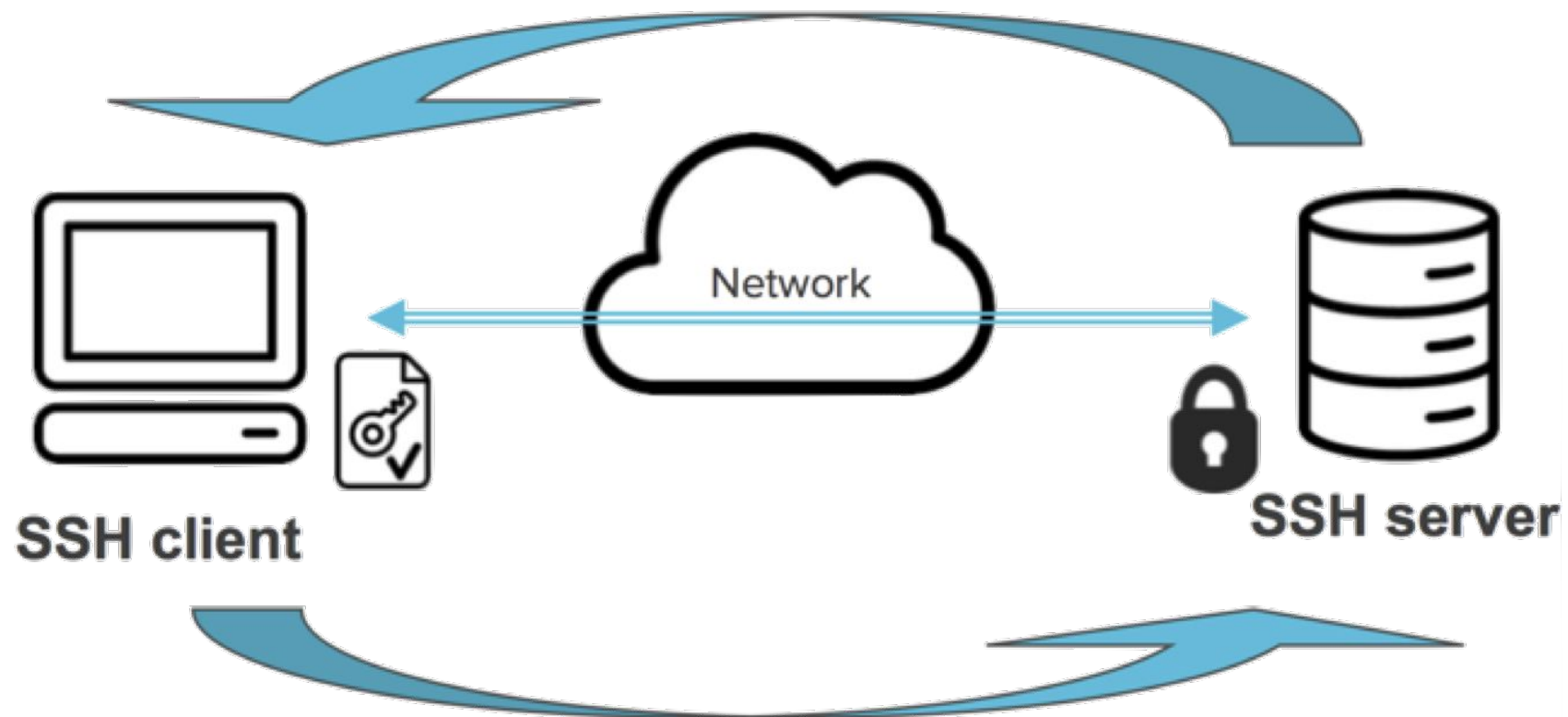


SSH(Secure Shell)

- ❖ A cryptographic network protocol for operating network services
 - `ssh host_ip_address`
- ❖ Key Pair

1) **Server** authentication:

Server proves its identity to the client



2) **User** authentication:

Client proves user's identity to the server

Try - SSH

```
~$ apt install -y openssh-server
```

```
~$ sudo vim /etc/ssh/sshd_config
```

```
#PermitRootLogin prohibit-password      # Change PermitRootLogin yes
```

```
...
```

```
~$ sudo passwd                          # When Access denied
```

```
~$ systemctl status ssh && systemctl restart ssh
```

➤ Client (Linux OS)

```
~$ ssh -p 50532 root@13.209.98.218
```

```
...
```

```
~$ who -u -H                            # or who -a          → 현재 접속자 확인
```

❖ 해보기

➤ 자신 PC와 goom.io IP 알리기(Google Drive)

➤ 상대 PC와 goom.io Server에 file로 흔적 남기기(ex. ohsanghun.me)

❖ 알아가기 SCP (secure copy)

➤ **scp** [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2

```
~$ scp ./dump02 jetbot@192.168.0.178:/home/jetbot
```

```
jetbot@192.168.0.5's password:
```

```
dump02                                100% 1431   86.6KB/s   00:00
```

```
~$ scp ./temp01.txt -P 53052 root@13.125.218.119:/root
```

UDP

- ❖ refer : https://linuxhint.com/send_receive_udp_packets_linux_cli/
 - ❖ User(Universal) Datagram Protocol, 신뢰성 낮고, 데이터 중복 / 누락 발생.
 - Using DNS Server, Video Streaming Service
- ```
jetbot@jetson-4-3:~$ nc -u -l 9999 # Server : Protocol UDP, Port 9999
 # Waiting for Message
~$ nc -u 192.168.0.12 9999 # Client : Protocol UDP, ServerIP ...0.12 Port 9999
 # Waiting for Message
↓ wireshark filter udp.port == 9999 && ip.addr == 192.168.0.12 입력.
~$ nc -u 192.168.0.12 9999 # Client : input text below
Hello There.
안녕하세요.
jetbot@jetson-4-3:~$ nc -u -l 9999 # Server : Check below with Message
Hello There.
안녕하세요.
↓ wireshark check log
```
- 
- ❖ 알아보기 : Check Sending UDP
- ```
~$ echo -n "echo hello" >/dev/udp/192.168.0.12/9999  # unreachable on wireshark
```

UDP Socket Programming - Server/Client

```
jetbot@jetson-4-3:~$ vi UDPServer.py
```

```
import socket
UDPServer = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
UDPServer.bind(("192.168.0.12", 20001))          # Bind to Server address and ip
while(True):                                   # Listen for incoming datagrams
    bytesAddressPair = UDPServer.recvfrom(1024)
    message = bytesAddressPair[0]
    address = bytesAddressPair[1]
    print("Message from Client:{}, IP:{}".format(message, address))
    UDPServer.sendto(str.encode("Hello UDP Client"), address)    # reply to client
```

```
~$ vi UDPClient.py
```

```
import socket
UDPClient = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
UDPClient.sendto(str.encode("Hello UDP Server"), ("192.168.0.12", 20001))
msgFromServer = UDPClient.recvfrom(1024)
print("Message from Server {}".format(msgFromServer[0]))
```

```
jetbot@jetson-4-3:~$ python3 UDPServer.py
```

```
~$ python3 UDPClient.py
```

UDP with Webcam - sender

❖ refer : <https://awakening95.tistory.com/1>

```
~$ sudo sysctl -w net.core.rmem_max=65535
```

```
~$ senderUDP.py
```

```
import socket
```

```
from cv2 import cv2 as cv
```

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
cap = cv.VideoCapture(0)
```

```
while cap.isOpened():
```

```
    ret, frame = cap.read()          # frame (480, 640, 3)
```

```
    dim = frame.flatten()
```

```
    str = dim.tostring()
```

```
    for i in range(20):              # ((480*640*3)/20=46080) < 65535
```

```
        sock.sendto(bytes([i]) + str[i*46080:(i+1)*46080], ('192.168.0.5', 1234))
```

```
cap.release()
```

❖ UDP with CSI-cam - sender

```
cam_id = 0
```

```
camSet='nvarguscamerasrc sensor-id='+str(cam_id)+' ! video/x-raw(memory:NVMM),
```

```
width=3264, height=2464, framerate=21/1,format=NV12 ! nvvidconv flip-method=0 !
```

```
video/x-raw, width=640, height=480, format=BGRx ! videoconvert ! video/x-raw,
```

```
format=BGR ! appsink'
```

```
cap = cv.VideoCapture(camSet)
```

UDP with Webcam - receiver

```
~$ sudo sysctl -w net.core.rmem_max=65535
```

```
~$ receiverUDP.py
```

```
import socket
```

```
import numpy
```

```
from cv2 import cv2 as cv
```

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
sock.bind(('192.168.0.5', 1234))
```

```
str = [b'\xff' * 46080 for x in range(20)]
```

```
while True:
```

```
    picture = b"
```

```
    data, addr = sock.recvfrom(46081)
```

```
    str[data[0]] = data[1:46081]
```

```
    if data[0] == 19:
```

```
        for i in range(20):
```

```
            picture += str[i]
```

```
        frame = numpy.fromstring(picture, dtype=numpy.uint8)
```

```
        frame = frame.reshape(480, 640, 3)
```

```
        cv.imshow("frame", frame)
```

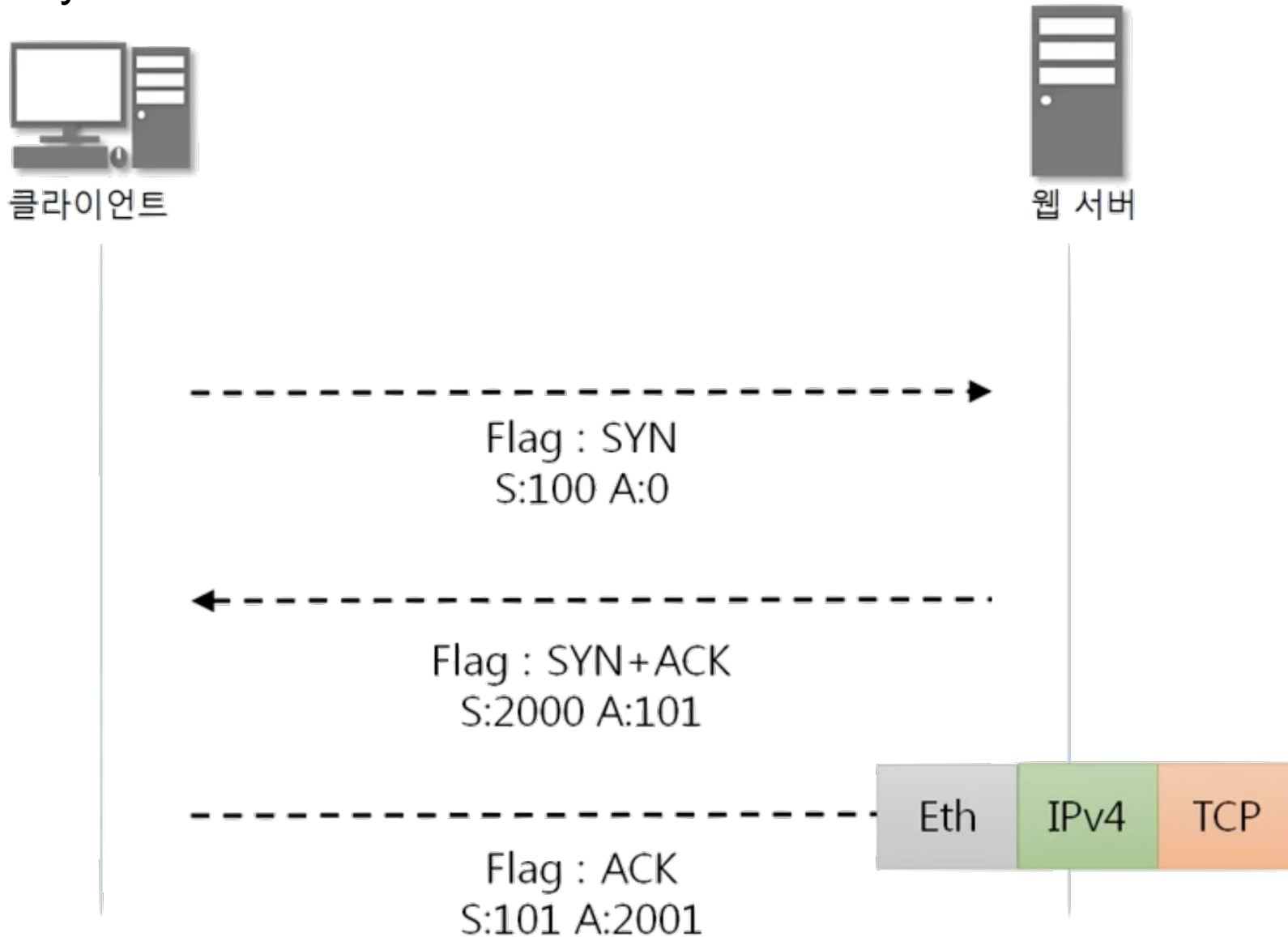
```
        if cv.waitKey(1) == ord('q'):
```

```
            break
```

```
cv.destroyAllWindows()
```


TCP

- ❖ **T**ransmission **C**ontrol **P**rotocol, 신뢰성 있게 에러없이 전송.
 - Using Email, Transfer File etc
 - 3 way-handshake



Try - TCP

❖ refer : <https://www.computerhope.com/unix/nc.htm>

jetbot@jetson-4-3:~\$ nc -l 7777 # **Server** : Protocol default TCP, Port 7777

 # **Waiting** for Message

~\$ **nc** 192.168.0.12 7777 # **Client** : Protocol TCP, ServerIP ...0.12 Port 7777

 # **Waiting** for Message

↵ wireshark filter **tcp.port == 7777 && ip.addr == 192.168.0.12** 입력.

 -> selected line > shorted menu > Follow > TCP Stream

 -> menu > Statistics > Flow graph > check box 'Limit to Display filter

~\$ nc 192.168.0.12 7777 # **Client** : input text below

Hello There TCP.

안녕하세요.

jetbot@jetson-4-3:~\$ nc -l 7777 # **Server** : Check below with Message

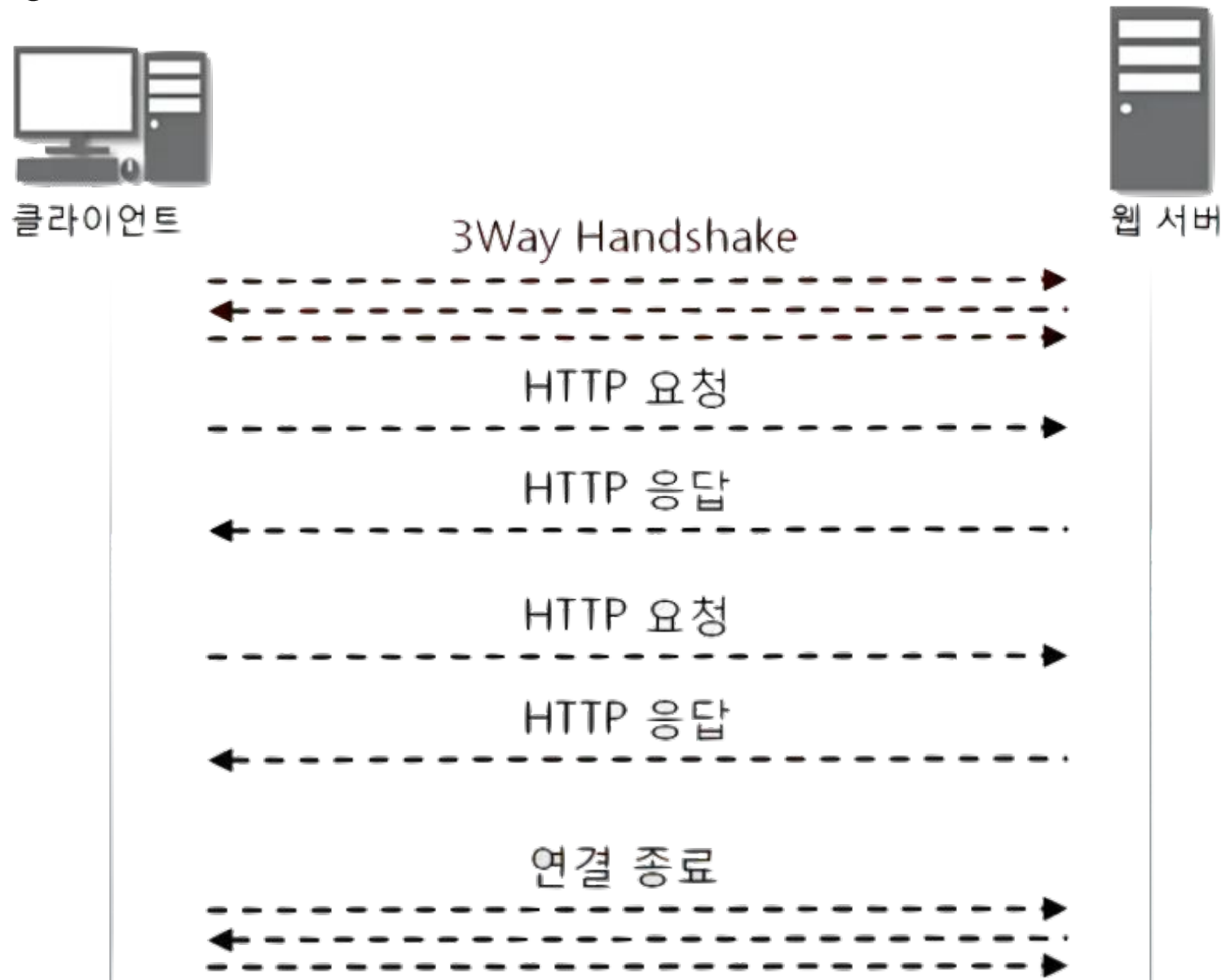
Hello There TCP.

안녕하세요.

↵ wireshark check log

HTTP

- ❖ HyperText Transfer Protocol
 - Using WWW etc



Try - HTTP

❖ <https://stackoverflow.com/questions/32341518/how-to-make-an-http-get-request-manually-with-netcat>

↓ run capturing on wireshark

@ www.naver.com

~\$ nc -v naver.com 80

or curl -vv naver.com

found 0 associations

found 1 connections:

1: flags=82<CONNECTED,PREFERRED>

outif en0

src 192.168.0.6 port 49530

dst **125.209.222.141** port 80

rank info not available

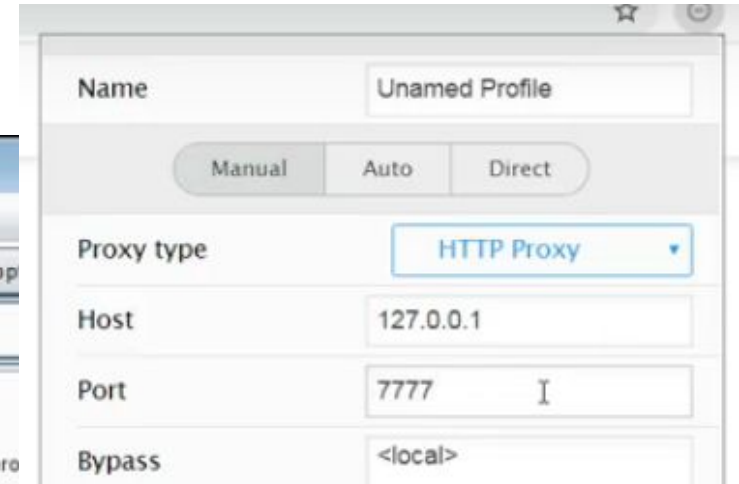
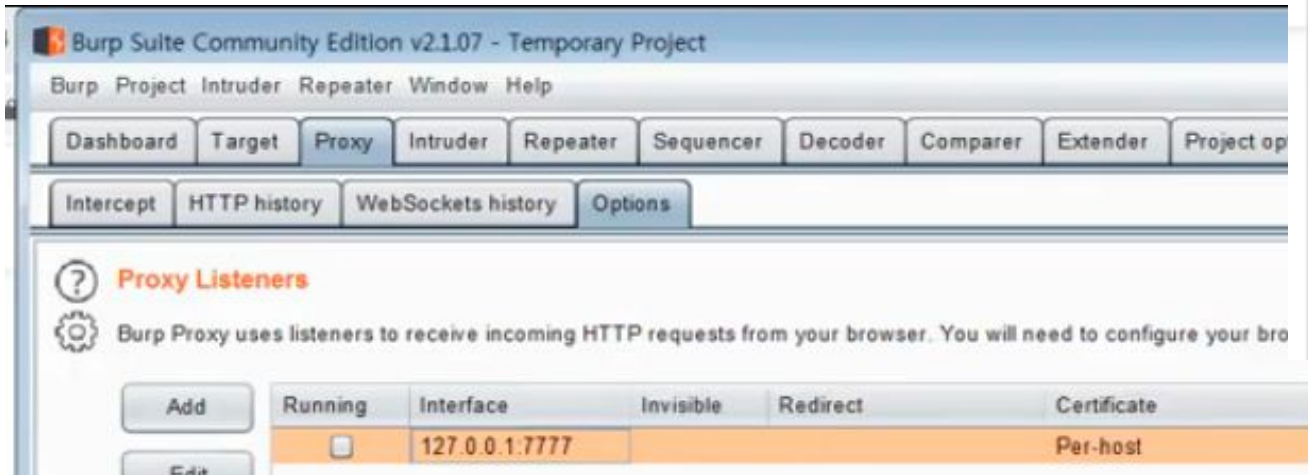
TCP aux info available

↓ wireshark filter **ip.addr == 125.209.222.141** 입력.

↓ wireshark check log

Try - HTTP packet

- ❖ Down : <https://portswigger.net/burp/communitydownload>
- ~\$ sh ./burpsuite_community_linux_v2020_11.sh
- ❖ Plugin Extension on Chrome : <https://chrome.google.com/webstore>
 - install Falcon Proxy > Set Port > Switch On
- ❖ Run Burp Suite And set up like Below



@ <https://blog.naver.com/newings> > 비공개 설정 오픈

- ❖ Change text below

```
var isInitializingBlog = false;  
var rightClickOpenYn = true;  
var isMylogBlocked = false;
```




❖ reference

- <https://www.ssh.com/ssh/key/>
- <https://youtu.be/vBrQ3yzerMg>