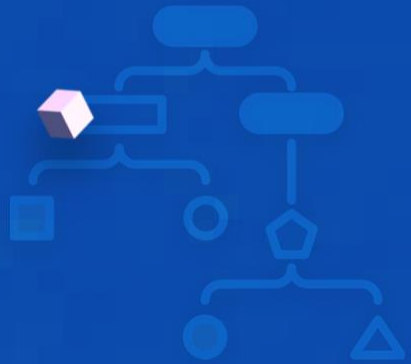
강원혁신플랫폼

리눅스프로그래밍

리눅스(Linux) 네트워크 프로그래밍



Linux에서 IP 주소를 통해 로컬 네트워크의 물리적(MAC) 주소를 구할 때 사용하는 프로토콜 을 무엇이라고 하나요?

ARP
(Address Resolution Protocol)

Linux에서 MAC 주소에서 IP 주소를 얻는 데 사용되는
프로토콜을 무엇이라고 하나요?

RARP
(Reverse Address Resolution Protocol)

학습목표

학습 내용

1. OSI 7계층과 TCP/IP 프로토콜
2. TCP/IP 프로토콜과 주소 체계
3. TCP/IP와 소켓

학습 목표

- OSI 7계층과 TCP/IP 프로토콜에 대해 설명할 수 있다.
- TCP/IP 프로토콜과 주소 체계에 대해 설명할 수 있다
- TCP/IP와 소켓에 대해 설명할 수 있다.

OSI 7계층과 TCP/IP 프로토콜

Linux 네트워크 프로그래밍

Linux 운영 체제에서 네트워크 응용 프로그램을 개발하는 것
다양한 프로그래밍 언어, 라이브러리 및 시스템 호출을 활용하여
네트워크를 통해 통신하는 소프트웨어를 만드는 것

BSD 소켓

네트워크 프로그래밍을 위한
표준화된 인터페이스

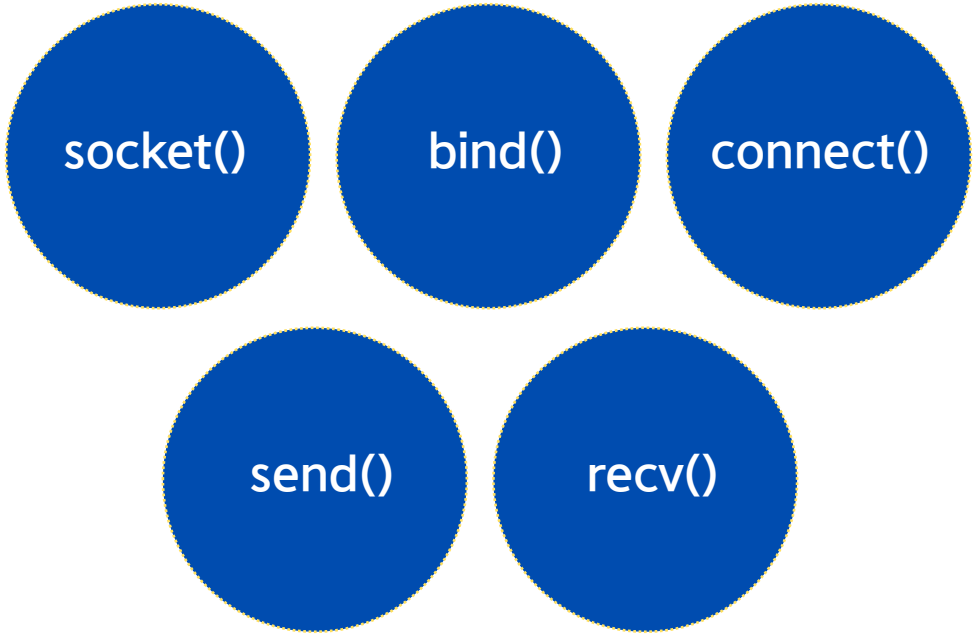
Berkeley Sockets API → 기반 → 소켓 API

BSD 소켓 API

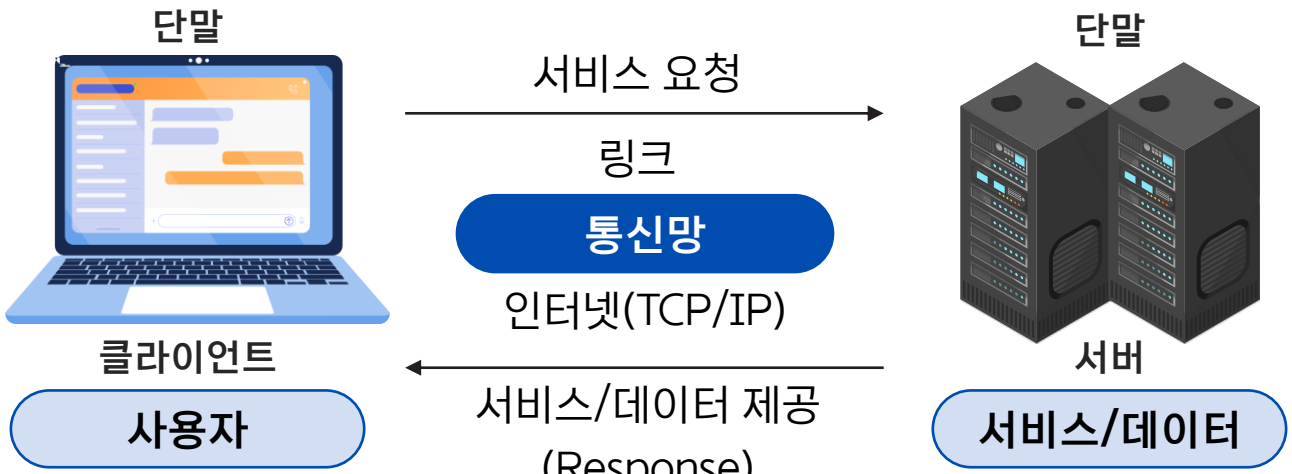
[네트워크 소켓을 통해 생성, 조작 및 통신하기 위한 함수 및 데이터 구조 집합]

BSD 소켓

- Linux BSD 소켓 API



웹 서비스와 네트워크



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

네트워크와 통신의 역사

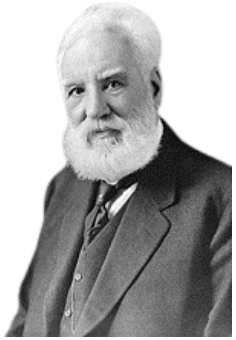
모스 부호
(Morse code)

새뮤얼 모스

1844년

1876년

전화기 특허가
미국의 특허청에 등록



알렉산더 그레이엄 벨

AT&T(American Telephone and
Telegraph Company)가 설립

1885년

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020
위키피디아

네트워크와 통신의 역사

굴리엘모 마르코니

1895년 마르코니의 최초의 무선 통신

1901년 대서양(영국과 미국) 무선 통신

1929년 기계식 TV 방송/BBC

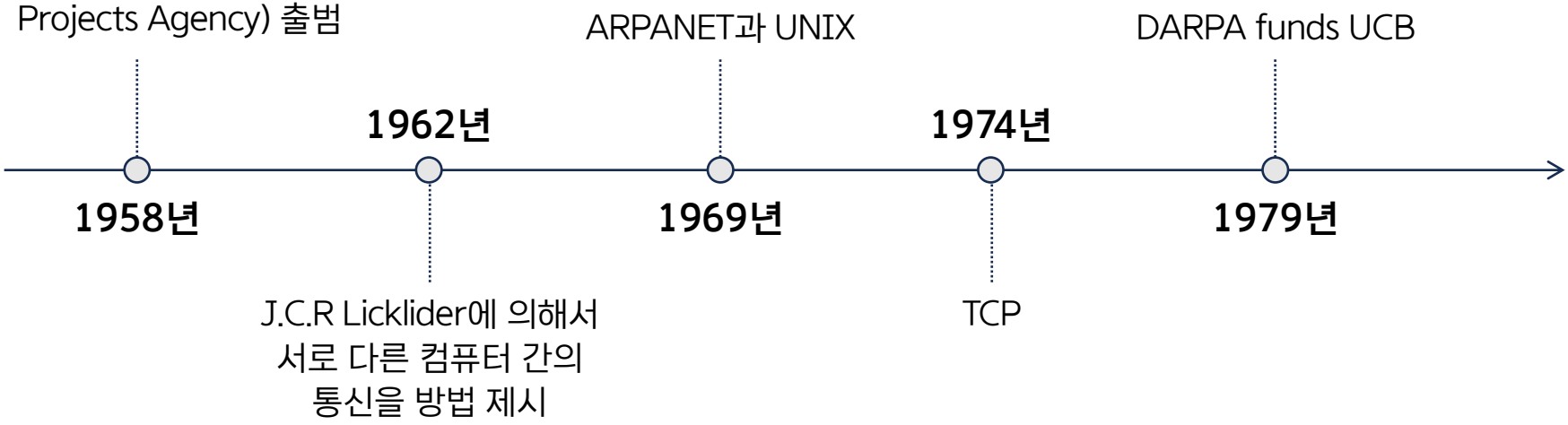
1939년 컬러 TV 방송/영국 도미니언 극장

1957년 Sputnik 발사 성공/소련

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020
위키피디아

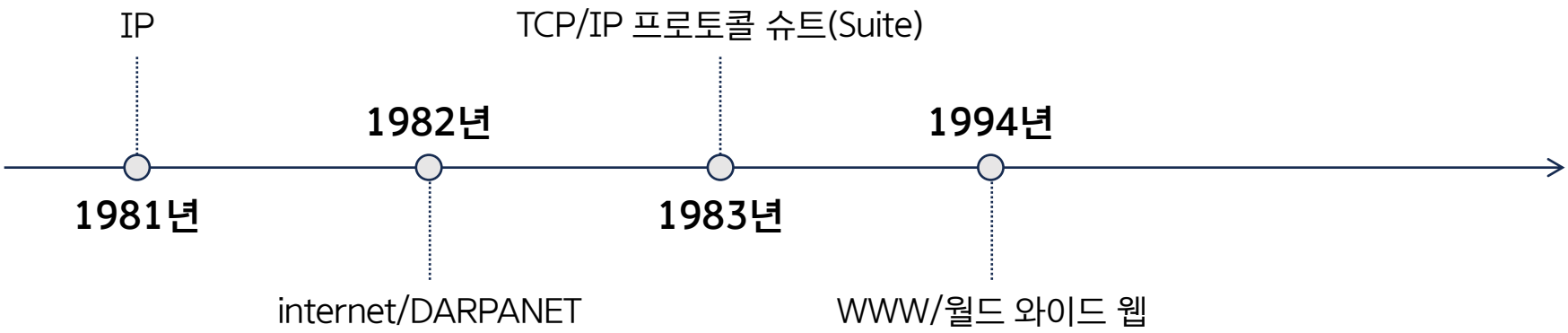
네트워크와 통신의 역사

- 미국 공군 SAGE 시스템
- ARPA(Advanced Research Projects Agency) 출범



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

네트워크와 통신의 역사



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

Internet, internet, protocol

01 02 03

인터넷(Internet)

- 인터넷은 상호 연결된 컴퓨터 네트워크의 글로벌 네트워크
- 전 세계에 걸쳐 있으며 통신 및 정보 교환을 가능하게 하는 공통 프로토콜 세트를 사용하는 방대한 네트워크

Internet, internet, protocol

01 02 03

인터넷(internet)

- “인터넷”(소문자 “i”)이라는 용어는 네트워크의 네트워크
- 상호 연결된 네트워크 집합을 설명하는 데 사용되는 일반적인 용어
- LAN(Local Area Network), 개인 네트워크, 공용 네트워크
- Internet은 가장 잘 알려지고 가장 큰 internet 예

Internet, internet, protocol

01 02 03

프로토콜(protocol)

- 컴퓨터 네트워크의 맥락에서 프로토콜은 데이터가 네트워크를 통해 전송, 수신 및 해석되는 방식을 관리하는 일련의 규칙
- TCP/IP(Transmission Control Protocol/Internet Protocol), HTTP(Hypertext Transfer Protocol), FTP(File Transfer Protocol) 및 SMTP(Simple Mail Transfer Protocol)

OSI 7계층

레벨	계층	내용
7계층	응용 계층	사용자가 네트워크에 접근하기 위해 필요한 서비스(Service) 정의
6계층	표현 계층	네트워크를 통한 데이터를 사용자에게 표현(Presentation)하기 위해 필요한 기능 정의
5계층	세션 계층	포트(Port)의 연결로 통신 장치 간의 상호 작용을 설정 및 유지하고 동기화하기 위해 필요한 기능 정의
4계층	전송 계층	세그먼트(Segment)를 이용해서 신뢰성 있는 통신을 위한 발신지와 목적지 간의 제어와 에러의 관리를 위해 필요한 기능 정의
3계층	네트워크 계층	다중 네트워크 링크에서 패킷(Packet) 단위로 데이터를 목적지에 확실하게 전송하기 위해 필요한 기능 정의
2계층	데이터링크 계층	네트워크에서 디바이스들 사이에 프레임(Frame) 단위로 데이터를 오류 없이 전송하기 위해 필요한 기능 정의
1계층	물리 계층	네트워크 카드나 케이블과 같은 물리적인 매체를 통한 비트(bit) 단위의 데이터 전송을 위해 필요한 기능 정의

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

OSI 7계층과 TCP/IP 프로토콜

<table>
<tr><th colspan="2">OSI 7계층</th><th colspan="3">TCP/IP 프로토콜 계층</th></tr>
<tr><td>Layer 7</td><td>응용(Application) 계층</td><td rowspan="3" colspan="2">Telnet, FTP, HTTP,
SMTP 등</td><td rowspan="3">응용(Application) 계층</td></tr>
<tr><td>Layer 6</td><td>표현(Presentation) 계층</td></tr>
<tr><td>Layer 5</td><td>세션(Session) 계층</td></tr>
<tr><td>Layer 4</td><td>전송(Transport) 계층</td><td>TCP</td><td>UDP</td><td>전송(Transport) 계층</td></tr>
<tr><td>Layer 3</td><td>네트워크(Network) 계층</td><td colspan="2">IP</td><td>인터넷(Internet) 계층</td></tr>
<tr><td>Layer 2</td><td>데이터링크(Datalink) 계층</td><td rowspan="2" colspan="2">네트워크 드라이버, 하드웨어,
ARP, RARP 등</td><td rowspan="2">링크(Link) 계층</td></tr>
<tr><td>Layer 1</td><td>물리(Physical) 계층</td></tr>
</table>

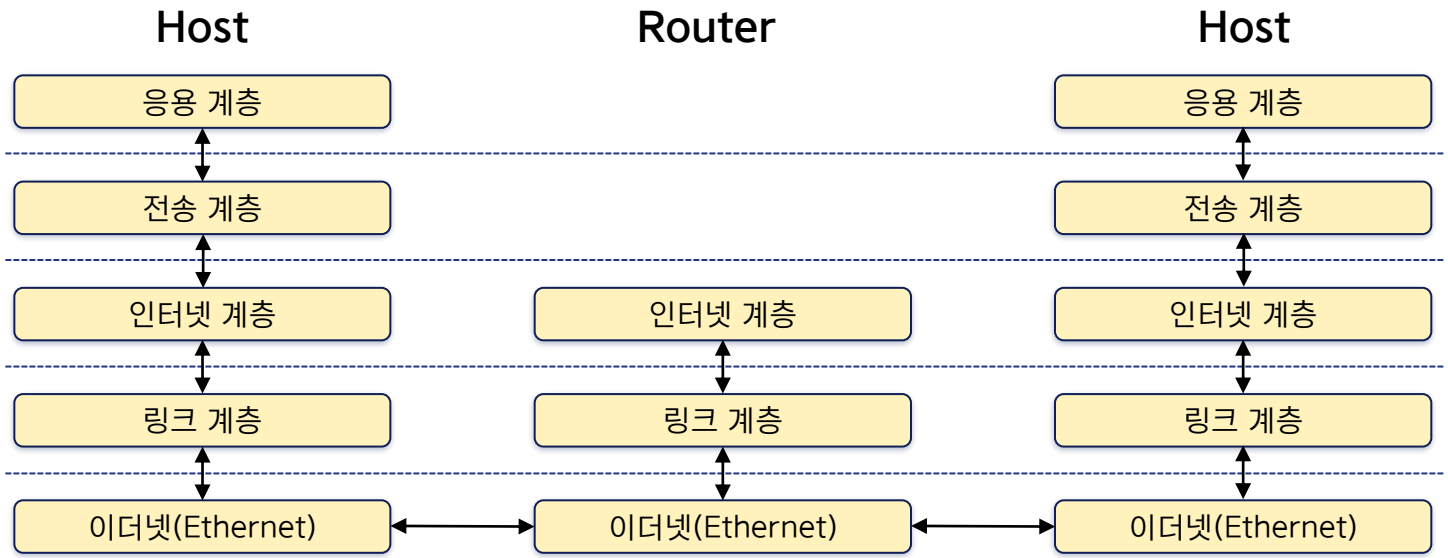
출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

TCP/IP 프로토콜의 계층

계층	내용	주요 프로토콜
응용 계층	사용자가 실제로 사용하는 응용 프로그램을 실행하기 위해 필요한 기능 정의	HTTP, SMTP, POP3, SIP 등
전송 계층	응용 프로그램 간 통신을 위한 기능 정의	TCP, UDP 등
인터넷 계층	IP 주소와 데이터 전송 경로 제어를 위해 필요한 기능 정의	IP(IPv4, IPv6), ICMP 등
링크 계층	네트워크 하드웨어 간의 전송 제어 및 서비스 제공을 위해 필요한 기능 정의	ARP, RARP 등

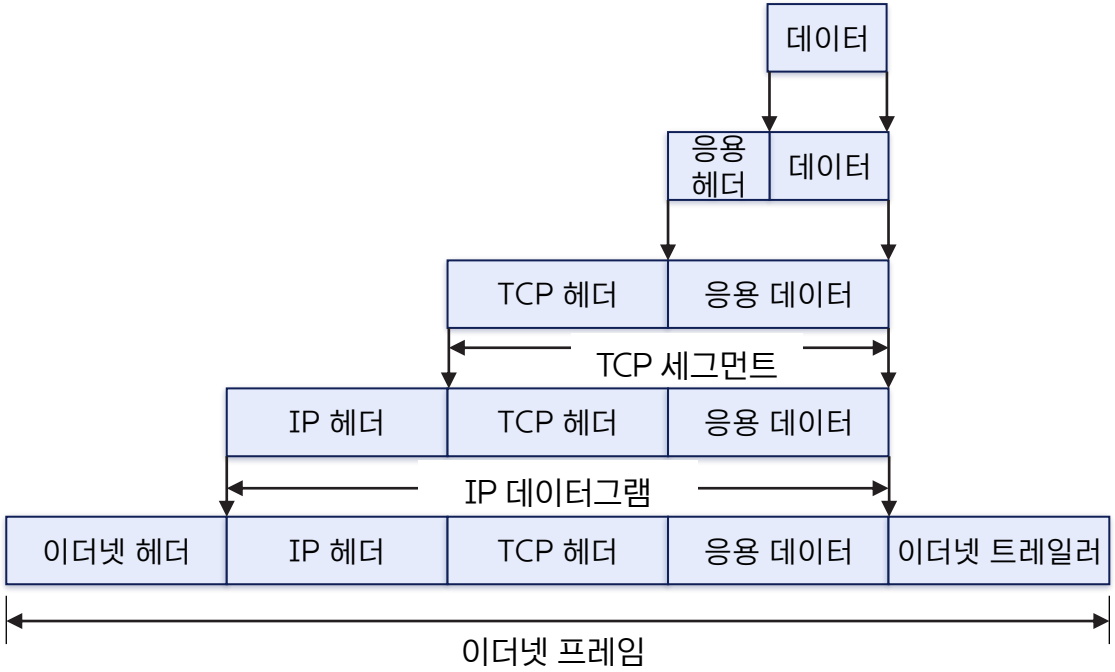
출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

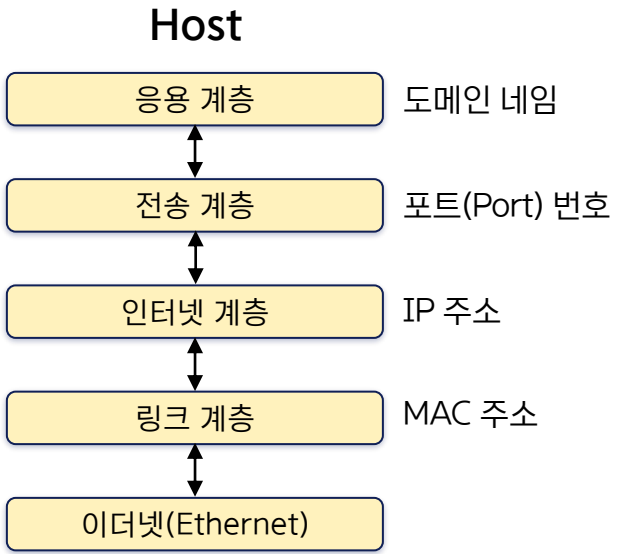
TCP/IP에서의 데이터 통신



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

TCP/IP 프로토콜과 주소 체계





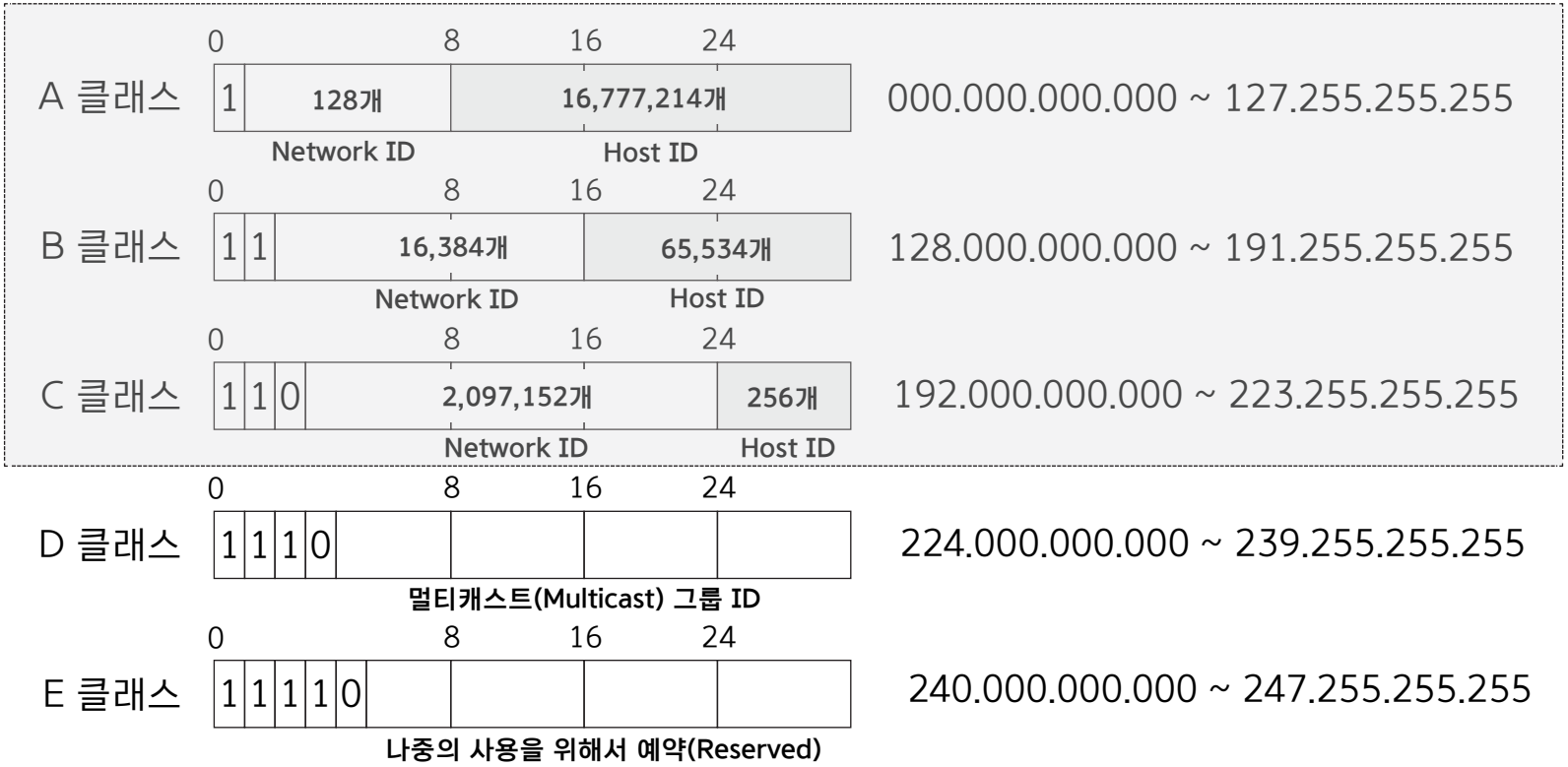
이더넷의 최대 전송 단위(MTU: Maximun Transfer Unit)는 1500바이트

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

OUI(Organizational Unique Identifier)			UAA(Universal Administered Address)		
1st Octet	2nd Octet	3rd Octet	4th Octet	5th Octet	6th Octet
1	2	3	4	5	6

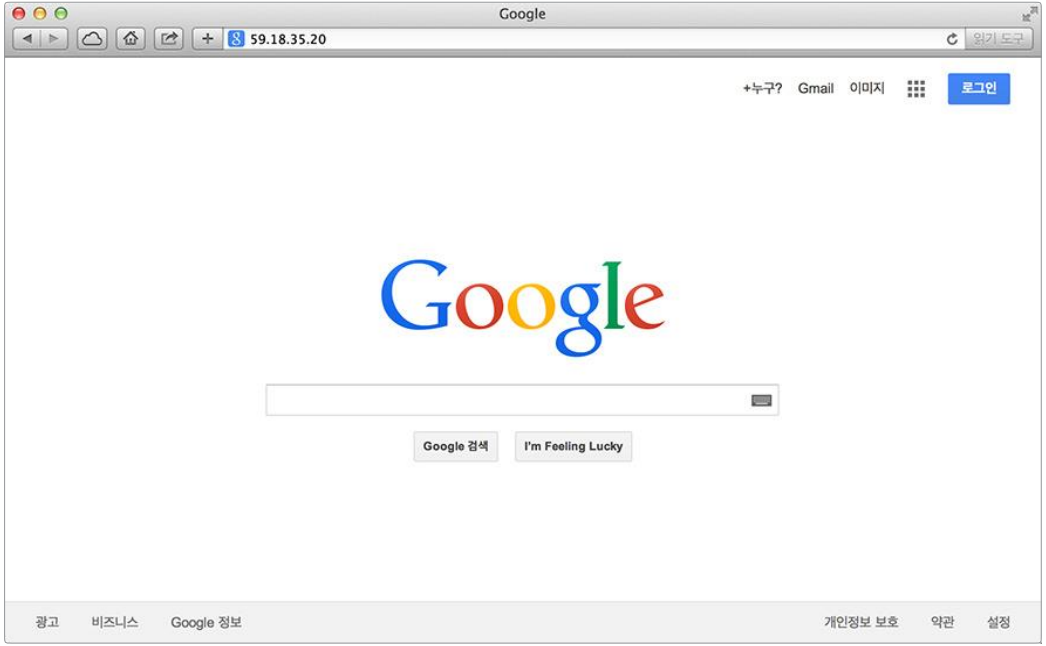
출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

IPv4의 주소 구분



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

IP 주소 142.251.46.174 로 접속한 구글의 홈페이지



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

nslookup / dig


```
$ nslookup google.com
Server:         172.18.80.1
Address:        172.18.80.1#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.26.238
Name:   google.com
Address: 2404:6800:4004:801::200e
```

nslookup / dig


```
$ dig google.com
; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> google.com
… 중간 생략

;; ANSWER SECTION:
google.com.             0       IN      A       172.217.26.238

;; Query time: 9 msec
;; SERVER: 172.18.80.1#53(172.18.80.1) (UDP)

;; WHEN: Mon Jul 10 20:00:00 KST 2023
;; MSG SIZE  rcvd: 54
```

유닉스의 포트 번호

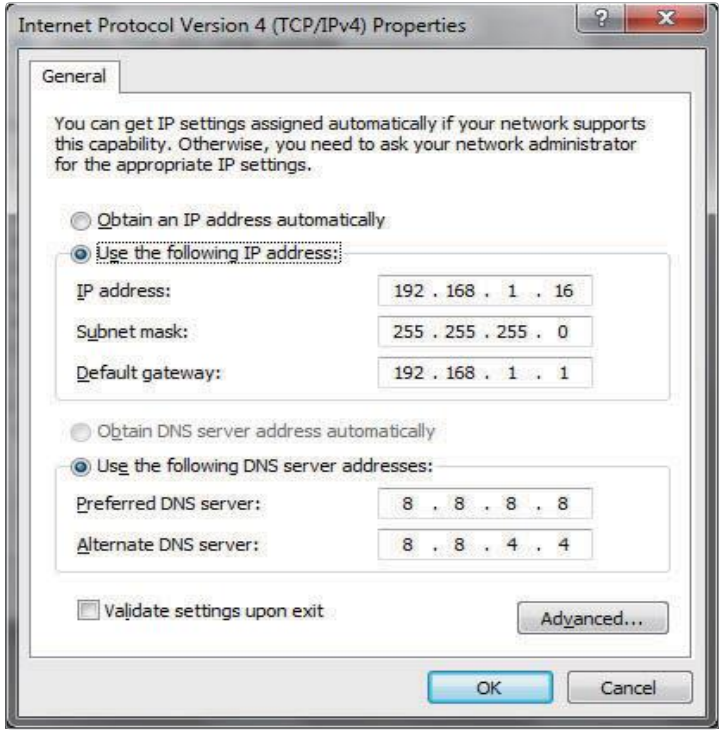

```
$ sed -n '28,32p' /etc/services
time            37/udp          timserver
whois           43/tcp          nicname
tacacs          49/tcp                          # Login Host Protocol (TACACS)
tacacs          49/udp
domain          53/tcp                          # Domain Name Server
```

유닉스의 포트 번호

포트 번호	내용	비고
0번 ~ 1023번	잘 알려진 포트(well-known port)	ECHO: 7, FTP: 21, telnet: 23, WWW: 80 등
1024번 ~ 49151번	등록된 포트(registered port)	
49152번 ~ 65535번	동적 포트(dynamic port)	

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

도메인 네임 서버와 기본 게이트웨이의 설정



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

arp, rarp, traceroute

01 02 03

ARP(Address Resolution Protocol)

- IP 주소를 로컬 네트워크의 물리적(MAC) 주소에 매핑하는 데 사용되는 프로토콜

arp, rarp, traceroute

RARP(Reverse Address Resolution Protocol)

- MAC 주소에서 IP 주소를 얻는 데 사용되는 프로토콜

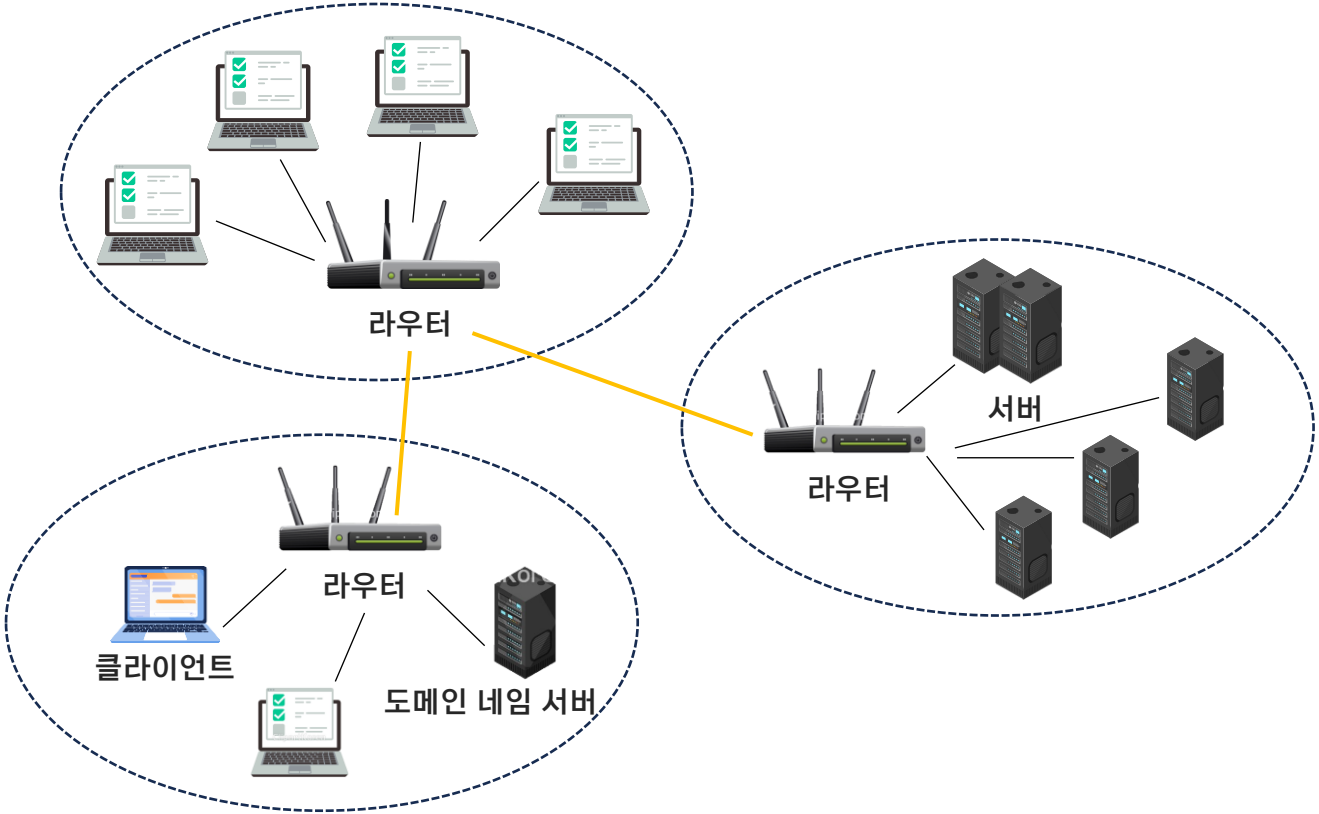
arp, rarp, traceroute

Traceroute

- 소스에서 목적지까지의 경로 패킷을 추적하는 데 사용되는 네트워크 진단 도구

```
$ sudo apt install net-tools
$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
ATwin.mshome.net         ether   00:15:5d:45:d5:6d   C                     eth0
$
```

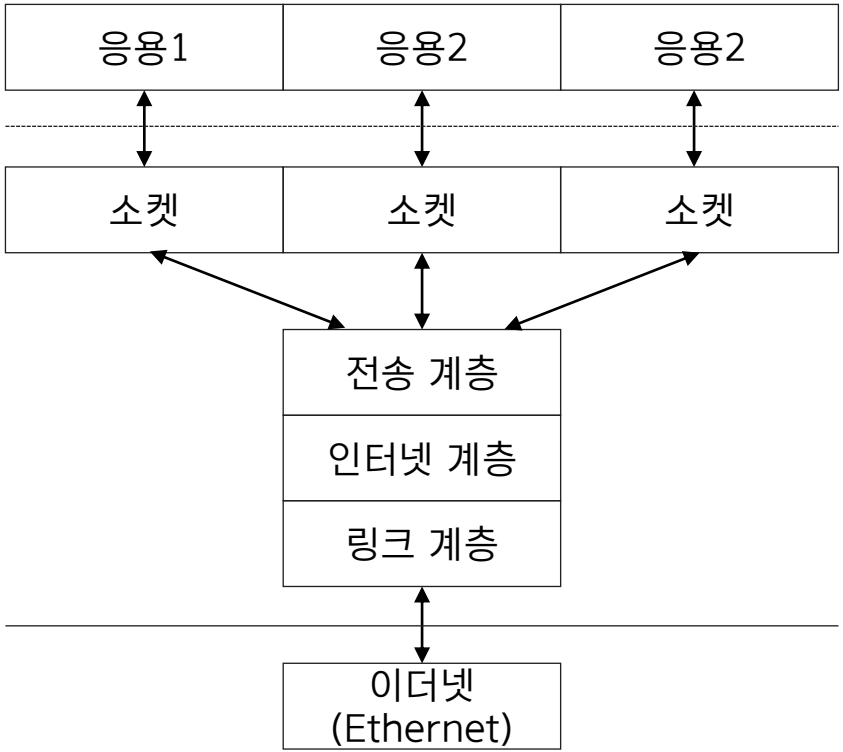
TCP/IP LAN과 인터넷



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

TCP/IP와 소켓

TCP/IP와 소켓



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

BSD 소켓 함수

함수	내용	비고
socket( )	사용하고자 하는 통신 프로토콜 지정	
socketpair( )	연결을 위한 한 쌍의 소켓을 지정함	IPC
bind( )	서버에서 소켓에 이름을 부여함	서버
listen( )	클라이언트의 소켓들을 위한 큐(Queue)를 생성함	
accept( )	서버에서 클라이언트의 접속 대기	
connect( )	클라이언트에서 서버로 접속	클라이언트

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

BSD 소켓 함수

함수	내용	비고
write( ) / send( )	데이터를 상대방에게 전송	TCP
sendto( )	데이터를 UDP 패킷으로 상대방에게 전송	UDP
read( ) / recv( )	데이터를 상대방으로부터 받음	TCP
recvfrom( )	데이터를 UDP 패킷으로 상대방으로부터 받음	UDP
close( )	소켓의 연결 종료	
shutdown( )	선택적으로 소켓의 연결을 종료	

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

int socketpair(int domain, int type, int protocol, int sv[2]);

- socketpair는 Linux를 포함한 Unix 계열 운영 체제에서 사용할 수 있는 시스템 호출
- 동일한 시스템 내의 프로세스 간 통신에 사용할 수 있는 한 쌍의 연결된 소켓 작성
- 두 개의 소켓으로 구성되며 하나는 읽기용이고 다른 하나는 쓰기용임
- 한 소켓에 쓰여진 모든 데이터는 다른 소켓에서 읽을 수 있음
- socketpair 함수는 일반적으로 부모 프로세스와 자식 프로세스 간 또는 동일한 프로세스 내의 다른 스레드 간에 통신에 사용
- 프로세스 간 통신(IPC)을 달성하는 간단하고 효율적인 방법임

따라하기 sockpair.c


```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <wait.h>
#include <sys/socket.h>

int main(int argc, char **argv)
{
    int ret, sock_fd[2];
    int status;
    char buf[ ] = "Hello World", line[BUFSIZ];
    pid_t pid;
```

따라하기 sockpair.c


```
ret = socketpair(AF_LOCAL, SOCK_STREAM, 0, sock_fd); /* 한 쌍의 소켓을 생성 */
if(ret == -1) {
    perror("socketpair()");
    return -1;
}

printf("socket 1 : %d\n", sock_fd[0]);          /* 각 소켓의 디스크립트 번호를 출력 */
printf("socket 2 : %d\n", sock_fd[1]);

if((pid = fork()) < 0) {                        /* fork( ) 함수 실행 에러 시의 처리 */
    perror("fork()");
} else if(pid == 0) {                           /* 자식 프로세스일 때의 처리 */
```

```
    write(sock_fd[0], buf, strlen(buf) + 1);        /* 부모 프로세스로 데이터 보내기 */
        printf("Data send : %s\n", buf);
        close(sock_fd[0]);                          /* 소켓 닫기 */
    } else {                                    /* 부모 프로세스일 때의 처리 */
        wait(&status);                          /* 자식 프로세스의 종료 대기 */
        read(sock_fd[1], line, BUFSIZ);             /* 자식 프로세스에서 온 데이터 읽기 */
        printf("Received data : %s\n", line);
        close(sock_fd[1]);                          /* 소켓 닫기 */
    }

    return 0;
}
```

따라하기 sockpair.c

실행결과

```
$ gcc -o socketpair socketpair.c
$ ./socketpair
socket 1 : 3
socket 2 : 4
Data send : Hello World
Received data : Hello World
```

요약정리

01 • OSI 7계층과 TCP/IP 프로토콜

02 • TCP/IP 프로토콜과 주소 체계

03 • TCP/IP와 소켓