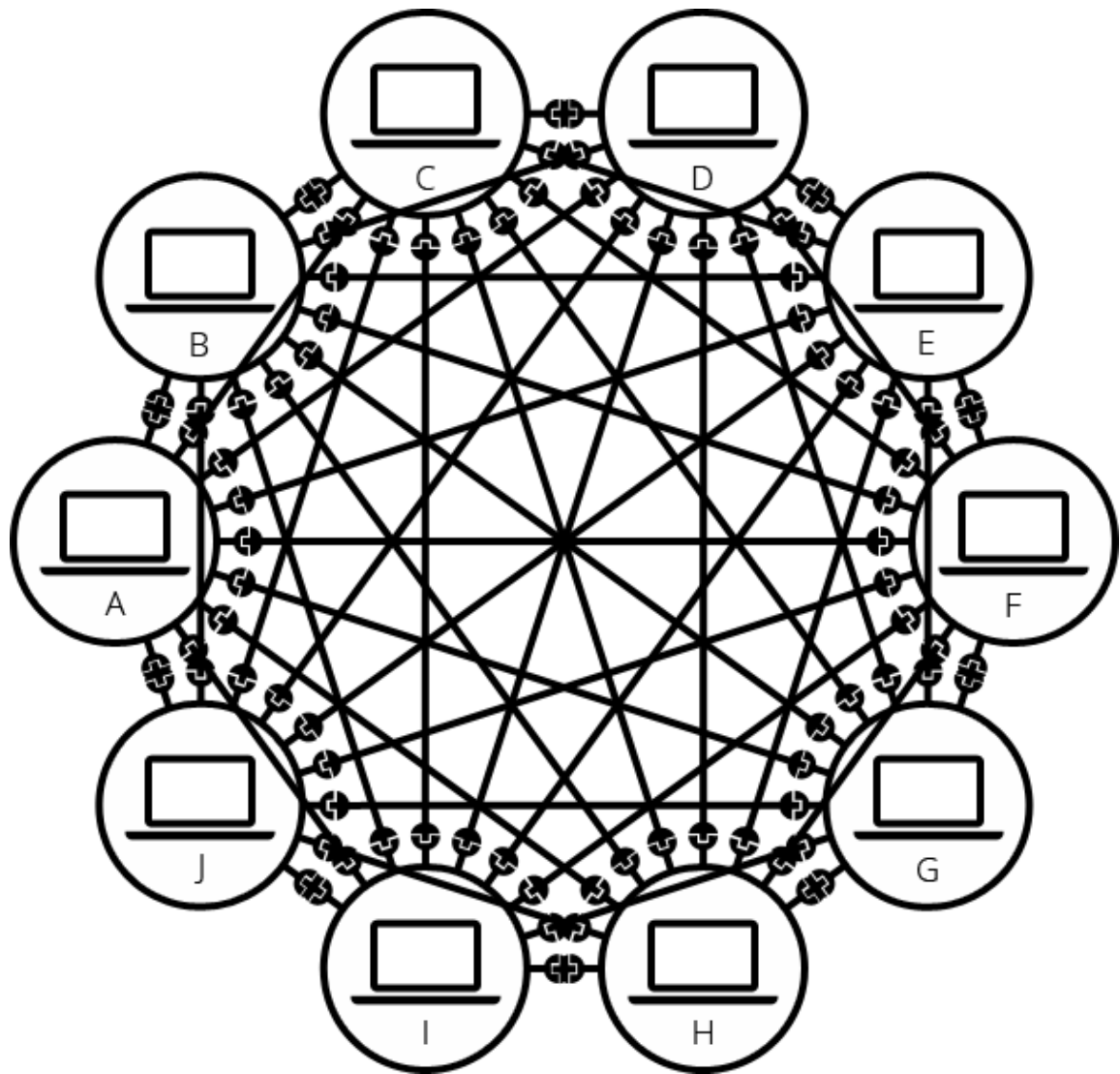


How does the internet work

Internet

- (inter + network)
 - 네트워크란 net+work의 합성어으로써 임의의 연결망을 지칭하는 용어 즉 인터넷에선, 단말들이 서로 통신이 가능하도록 연결해주는 것
- 네트워크와 네트워크를 연결하다.
- 컴퓨터로 연결하여 TCP/IP라는 통신 프로토콜을 이용해 정보를 주고받는 컴퓨터 네트워크
 - 왜 TCP/IP
미국 국방부에서 ARPAnet 을 먼저 사용
알파넷이란 미국국방성 네트워크라고 생각하면됨



| 서로의 네트워크가 연결된 것 = Internet

TCP/IP

정의

TCP/IP는 네트워크 프로토콜 스위트로, 온라인상의 안전하고 효율적인 데이터 전송의 필수 요건, 인터넷에서 사용되는 기본 통신 프로토콜

TCP: Transmission Control Protocol(전송 제어 프로토콜)

IP: Internet Protocol (인터넷 프로토콜)

프로토콜이란?

컴퓨터나 원거리 통신 장비 사이에서 메시지를 주고 받는 양식과 규칙의 체계

| 즉 통신 장비끼리의 약속이라 생각하면 된다.

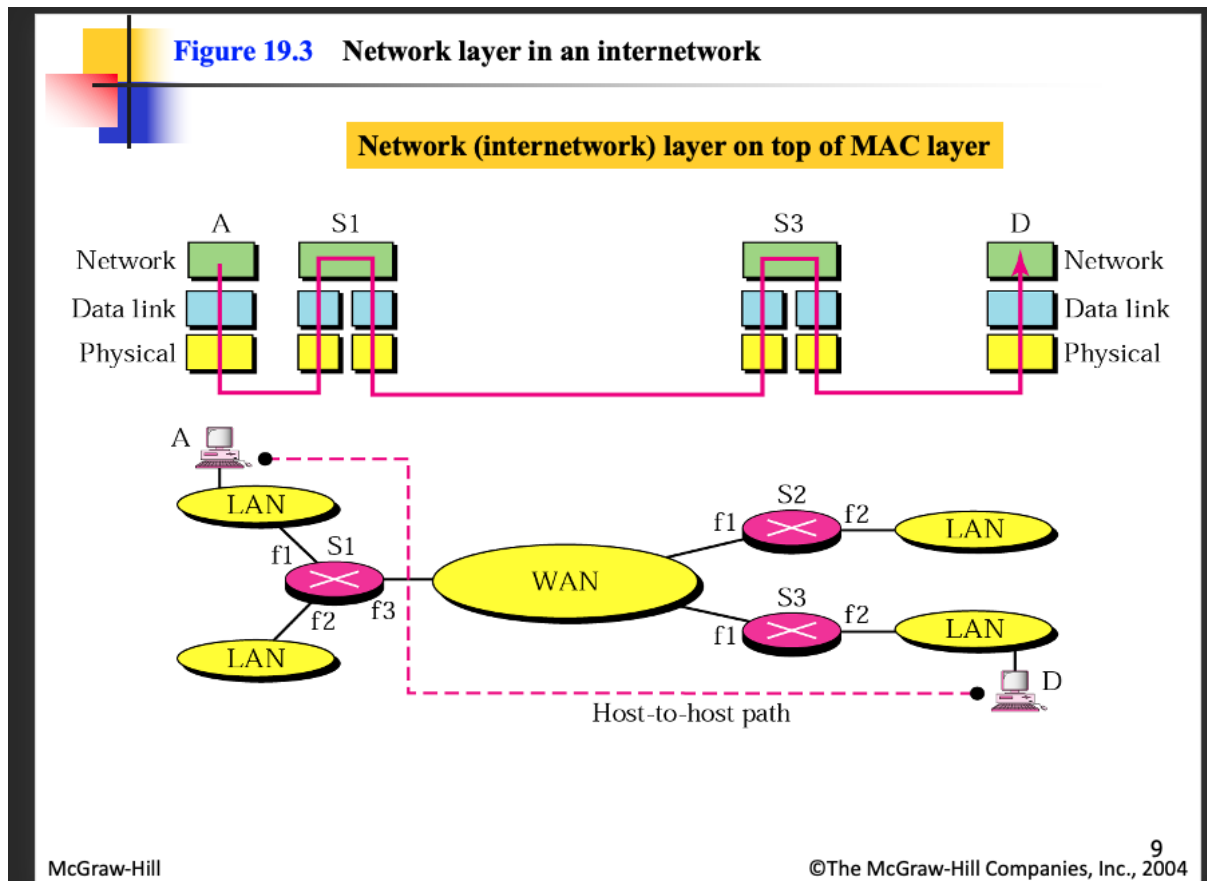
TCP/IP 계층구조 (4계층 모델)

주요 프로토콜	TCP/IP 프로토콜 계층모델	OSI 계층모델
TELNET, FTP, SNMP, DHCP 등	Application	Application
		Presentation
		Session
SCTP, TCP, UDP	Transport	Transport
IGMP ICMP IP ARP	Internet (IP 계층)	Network
	Link (Network Access)	Data Link
		Physical

1. 데이터 링크 계층

- 데이터가 원하는 IP주소로 도달
- 네트워크 내의 연결된 기기에 연결되어 있는지 확인
- 원하는 기기의 MAC 주소를 확인
 - MAC 주소란?

media access control address의 약자이며 데이터 링크 계층 통신을 위한 네트워크 인터페이스에 할당된 고유 식별자 (IP주소와 다르게 고정적인 주소라 생각하면 된다.)



2. 인터넷 계층

- ARP(Address Resolution Protocol: IP ↔ MAC)
- IGMP(Inter Group Management Protocol)
- ICMP(Internet Control Message Protocol): 인터넷 제어 메시지 프로토콜
 - Ping 명령어가 ICMP안에 속한다. 따라서 Ping명령어는 TCP/IP 또는 OSI 에서 Internet 또는 네트워크(3계층)에 속한다.
- 네트워크 간 데이터 패킷의 전송을 관리한다.

3. 전송 계층

- 전송을 담당하는 계층
 - TCP/UDP
 - TCP: 연결형 서비스, 신뢰도 높고 속도가 느리다
 - UDP: 비연결형 서비스, 신뢰도가 낮고 속도가 빠르다
4. 응용 계층
- 사용자가 네트워크에 접근할 수 있도록함.
 - 이메일, 원격 파일 접근 및 전송, 공유 데이터베이스 관리등의 서비스를 제공
 - SMTP, HTTP, FTP, DHCP, SNMP
 - HTTP(Hypertext Transfer Protocol)란?
클라이언트와 서버 간 통신을 위한 통신 규칙 세트 또는 프로토콜

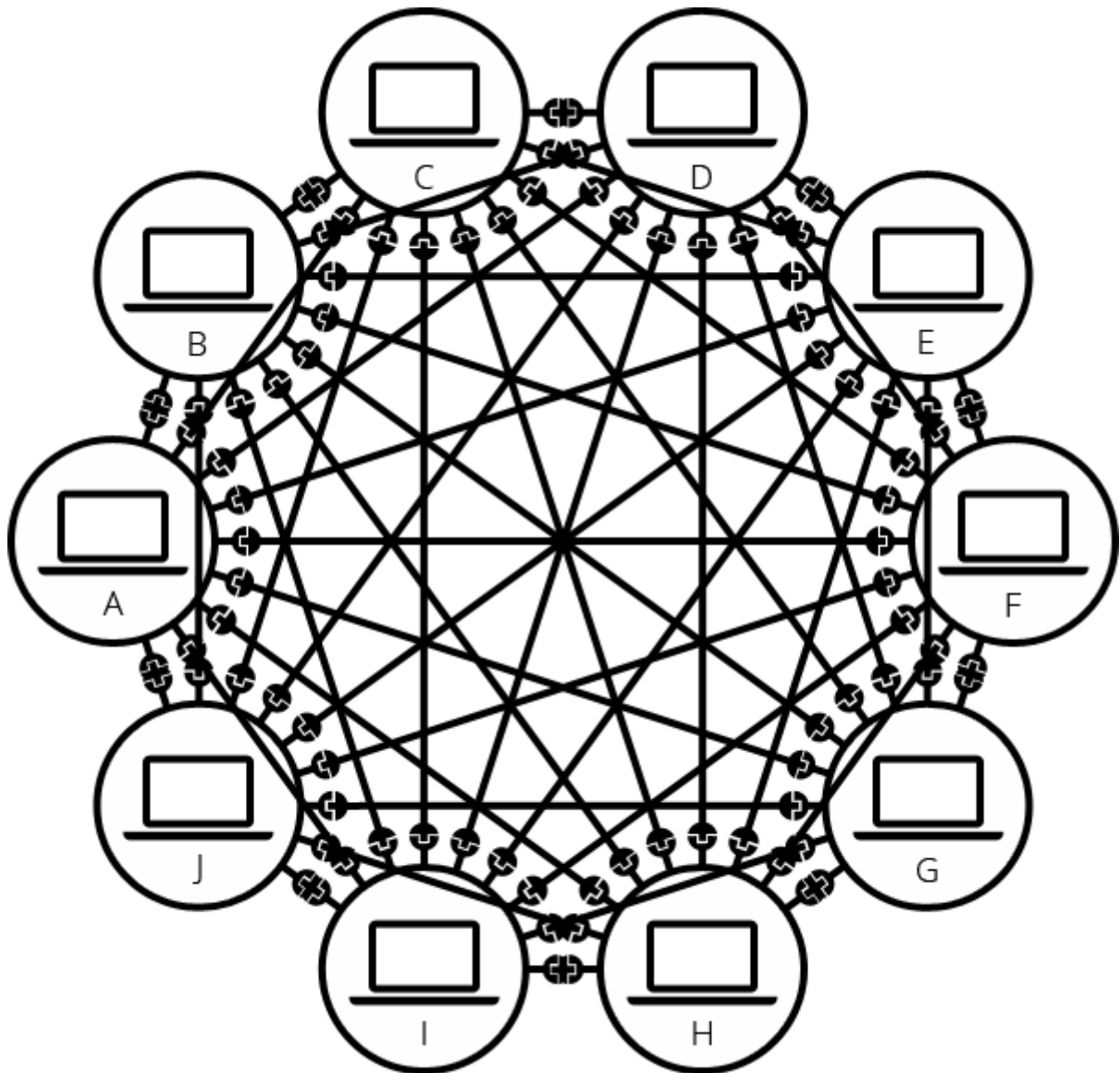
OSI 7계층

1. 물리계층
 - a. 데이터를 전기적인 신호(디지털신호)로 주고받는 기능만함
2. 데이터 링크계층
 - a. 통신의 흐름관리
 - b. 물리주소 부여(맥주소) 에러검출 재전송 흐름제어
3. 네트워크 계층
 - a. 라우터를 통해 경로를 선택 IP경로에 따라 패킷을 전달
4. 전송 계층
 - a. 전송방식(TCP/UDP) 결정하고 전송
5. 세션 계층
6. 표현 계층
 - a. 데이터 표현방식 결정 (JPEF, MPEG, GIF, ASCII)
7. 응용 계층
 - a. 응용서비스 하는데

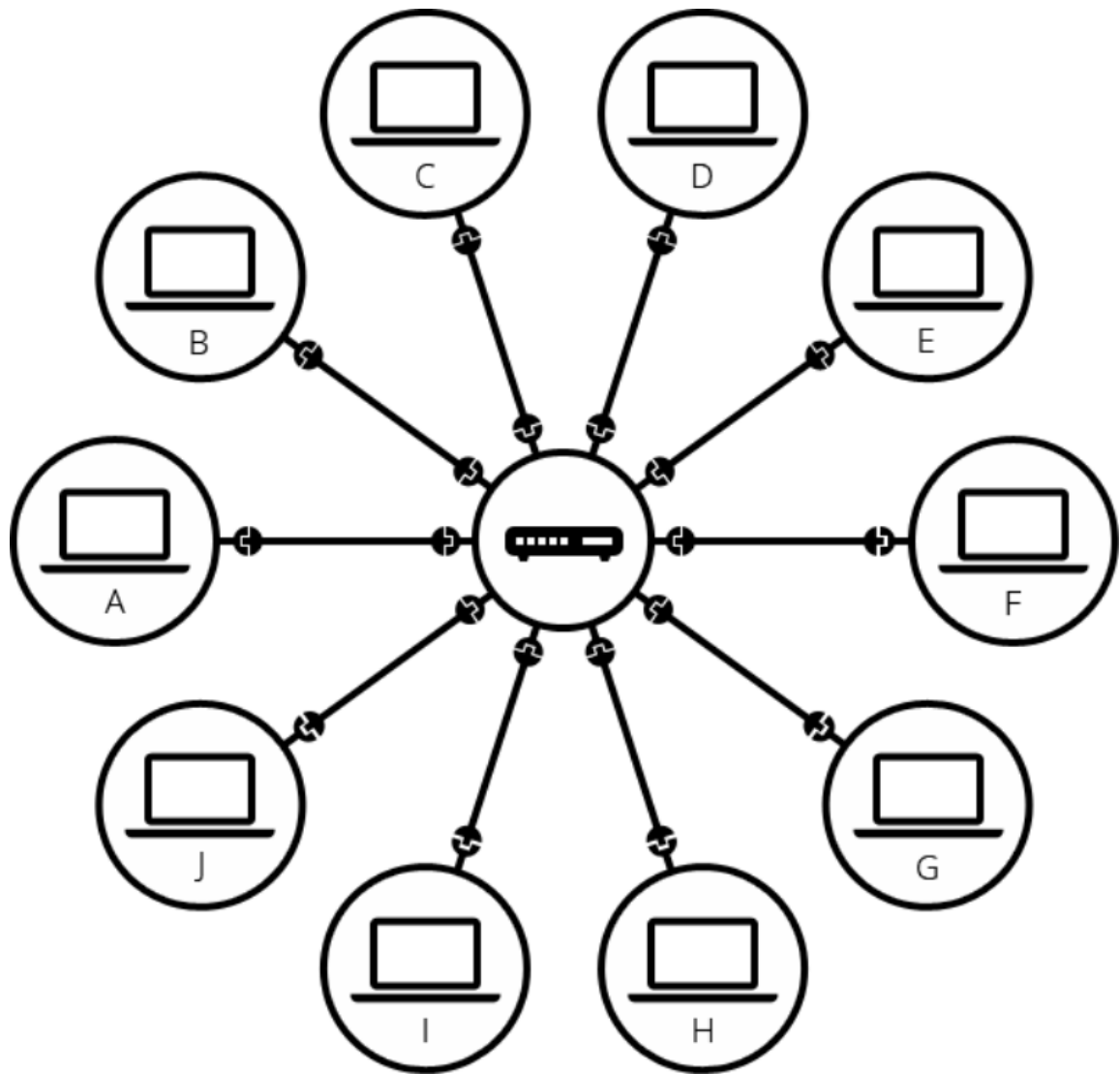
b. HTTP, FTP, POP3, IMAP

인터넷 동작방식의 발전

1. 컴퓨터와 컴퓨터를 케이블 또는 무선 연결



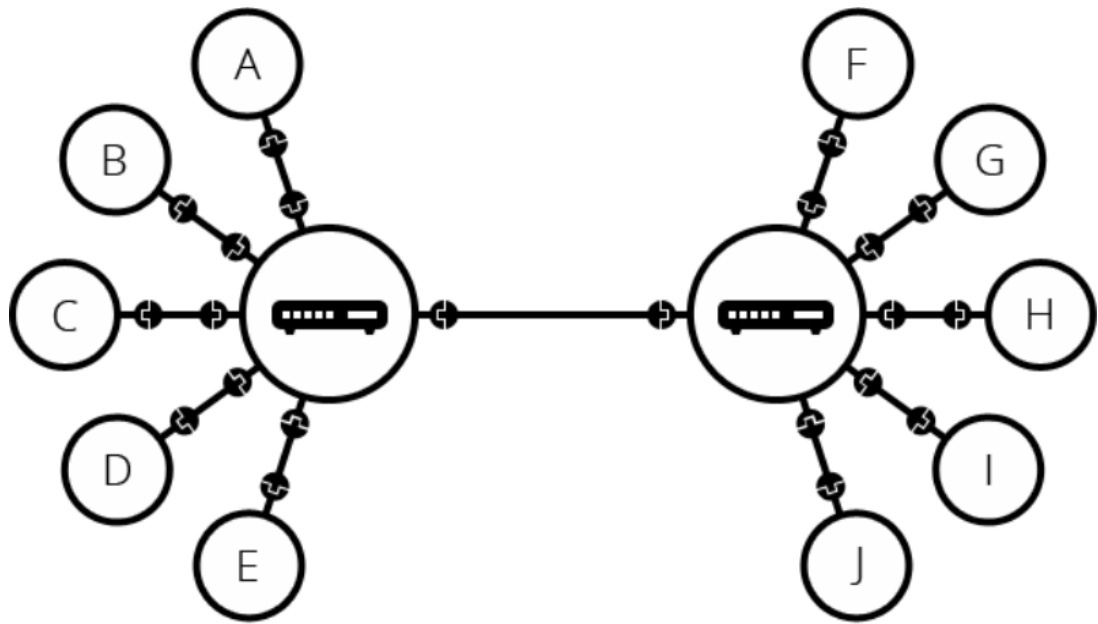
2. 라우터를 이용하여 네트워크 설정



라우터란?

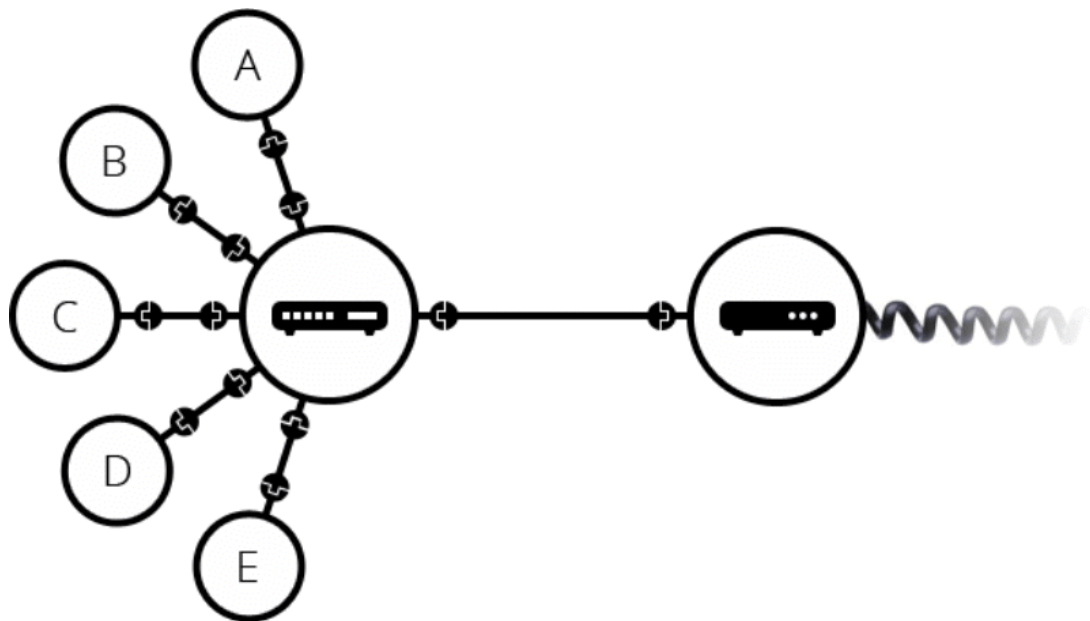
데이터를 원하는 곳에 전달해주는 단순한 역할만 하는 기기 그러나 몇천대로의 확장은 불가능하다.

3. 라우터와 라우터의 연결



컴퓨터와 컴퓨터끼리 연결하듯이 라우터도 라우터끼리 연결이 가능하다.

4. 모뎀을 이용한 네트워크 연결



모뎀이란?

통신회선으로 보내져 오는 아날로그 신호를 디지털 신호로

PC나 스마트폰으로 보내져 오는 디지털 신호를 아날로그 신호로

모뎀을 통해 전화선을 이용을 가능하게 되었다.

이 전화선을 통해 먼 곳에 있는 네트워크에도 연결이 가능하게 됨.

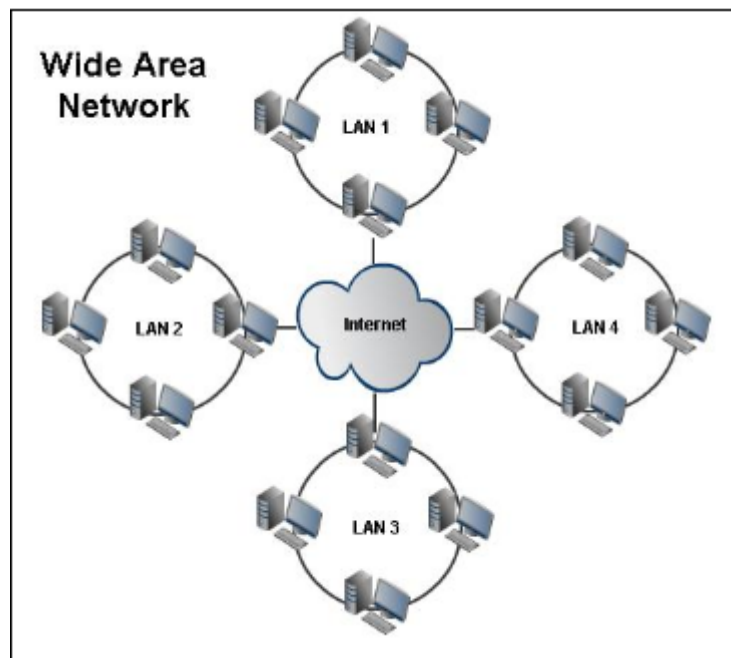
즉 가까운 LAN과 먼 곳에 LAN이 연결이 가능하게 됨.

LAN이란?

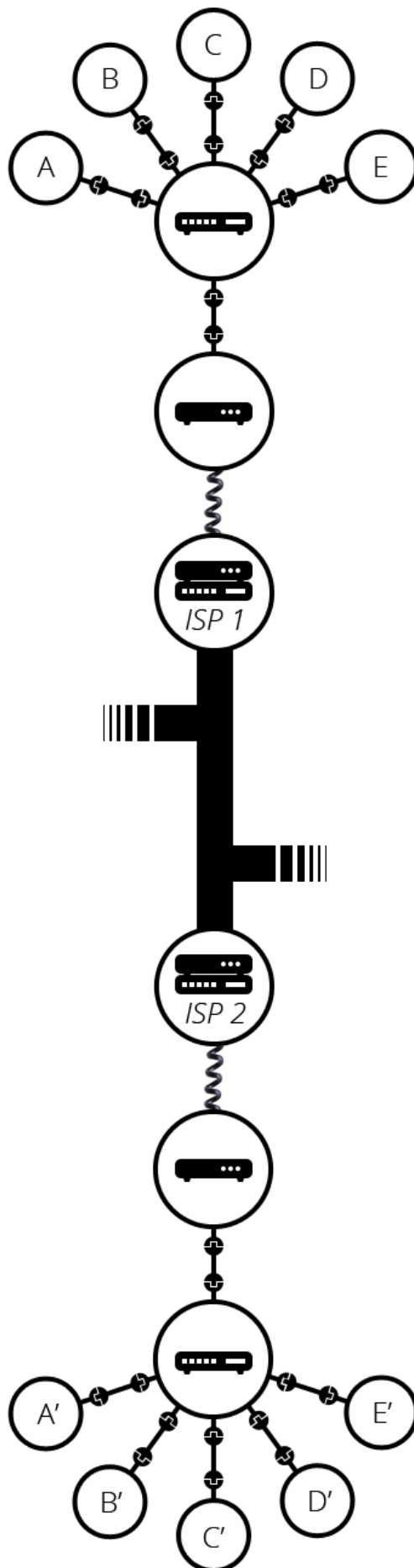
Local Area Network에서 작은 공간에서 연결되는 네트워크, 근거리 통신망

이 때 이 LAN들을 연결해주는 사람이 ISP(Inter Service Provider)

그리고 이렇게 연결된 네트워크를 WAN(Wide Area Network) 광역 통신망이라 한다.

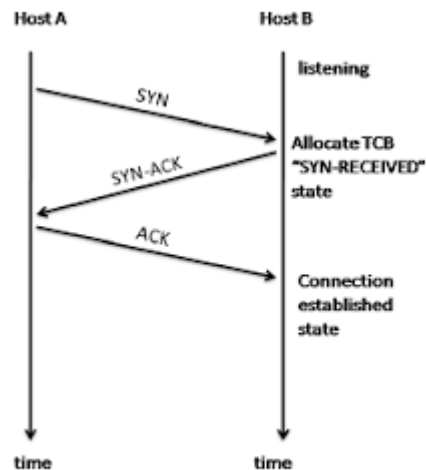


5. ISP가 네트워크와 네트워크를 연결



TCP 전송방식

- 3 Way HandShake



- SYN: 클라이언트가 서버에 SYN를 전송한다. 클라이언트는 세그먼트의 시퀀스 번호를 임의의 값으로 설정한다.
- SYN-ACK: 응답으로 서버는 SYN-ACK로 응답을 한다. 수신된 시퀀스 보다 많은 숫자로 설정 된다.
- ACK: 마지막으로 클라이언트가 서버에 다시 ACK를 보낸다.

TCP VS UDP

1. TCP

- a. 연결지향형
- b. 순서 보장, 느림
- c. HTTP, 메일, 파일

2. UDP

- a. 비연결지향형
- b. 순서 보장 X, 빠름

DNS

(Domain Name Service)

도메인 네임 시스템(Domain Name System, **DNS**)은 호스트의 도메인 이름을 호스트의 네트워크 주소로 바꾸거나 그 반대의 변환을 수행할 수 있도록 하기 위해 개발되었다. 특정 컴퓨터(또는 네트워크로 연결된 임의의 장치)의 주소를 찾기 위해, 사람이 이해하기 쉬운 도메인 이름을 숫자로 된 식별 번호(IP 주소)로 변환해 준다.

DNS를 사용하는이유

인터넷은 서버를 유일하게 구분할 수 있는 IP 주소를 사용하게 되는데 이를 일일이 외우지 않아도 DNS 를 사용하여 호스트의 도메인 이름을 호스트의 네트워크 주소로 바꾸거나 그 반대의 변환을 용이하게 하기 때문에 사용

HTTP

하이퍼텍스트를 전송하는 규약

하이퍼텍스트란?

한 문서에서 다른문서로 접근할 수 있는 텍스트

HTTP REQUEST - GET & POST

-GET: 서버에 데이터를 전달할 때 URL Query를 사용함 → 내용을 보기편함 (데이터를 받는 용도로 적합)

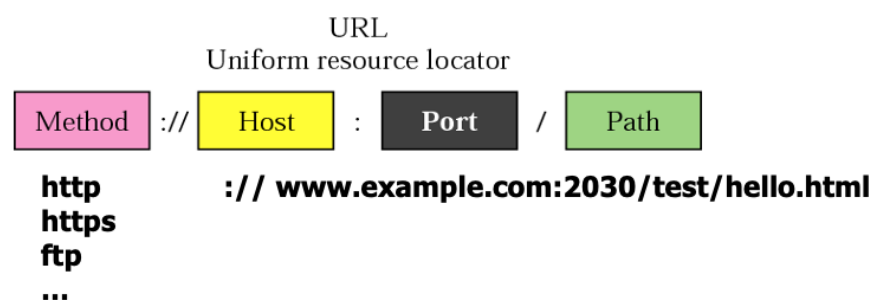
-POST: 데이터를 Header에 넣어서 전송 → 볼 순 있으나, 직접 헤더를 열어봐야됨 (DB 건들일 때 적합)

GET

```
app.get("/names", (req, res) => {  
  const names = table.map(item => {  
    return { name: item.name };  
  });  
  res.status(200).json(names);  
});
```

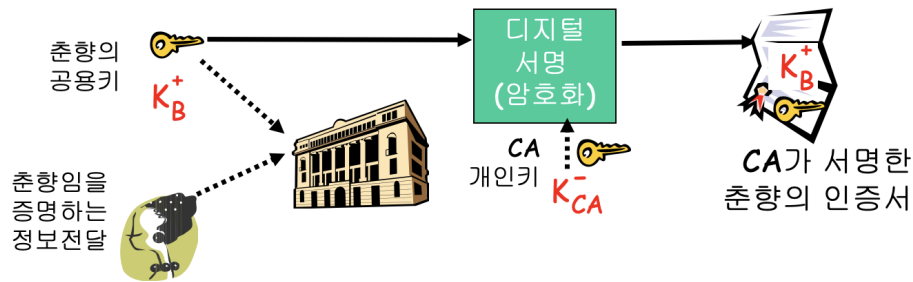
POST

```
app.use(express.json());  
app.post("/add", (req, res) => {  
  const item = req.body;  
  if(item.name && item.alias) {  
    table.push(item);  
    res.sendStatus(200);  
  } else {  
    res.sendStatus(400);  
  }  
});
```



CA 공용키 검증 방식(1)

- 각 주체(사람, 라우터)는 자신의 공용키를 **CA**에 등록
 - 각 주체는 **CA**에게 "자신에 대한 인증"을 받음
 - **CA**는 "이것이 이 사람의 공용키임"을 의미하는 인증서를 저장



20

PGP의 공용키 검증 방식 (1)

□ "Web of Trust"

- 우리의 신뢰 구조와 비슷: 누가 누구를 신뢰하느냐는 "자신"의 몫
 - 자신이 맞다고 "확인"한 것은 믿음
 - 자신이 신뢰하는 사람이 틀림없다고 한 것은 믿음
- 각 키는 **validity**와 **trust** 항목이 있음
 - **Validity**: 그 키가 그 사용자임을 말하는 것
 - **Trust**: 그 사용자를 얼마나 신뢰하느냐를 말하는 것
 - Trust level로 신뢰 정도를 표현

42

DNS

