

chapter 4



랜 - 연결

강의 내용

- 리피터(repeater)의 목적과 사용법
- 여러종류의 허브(hub) 구조와 사용법
- 브리지(bridge)의 구조와 사용처
- 스위치(switch)의 구조
- 라우팅(routing)과 스위칭(switching)의 차이
- 게이트웨이(gateway)의 기능과 필요성

강의 내용

- 네트워크 백본(backbone)의 3가지 계층(layer)
- 인터넷 프로토콜(Internet Protocol)과 IP 주소 클래스의 의미
- DHCP를 통한 IP주소 할당 방법
- 여러가지 다른 종류의 LAN프로토콜과 그 중요성
- 네트워크 관리 프로토콜(SNMP).
- 공유기
- 속제 4

랜 디바이스

- LAN의 구성요소에는 전송매체와 노드 뿐만 아니라 여러가지 디바이스가 있다.
 - LAN에 부착된 간단한 HW일 수도 있다.
 - 디바이스는 노드의 형태를 가질 수도 있다.
 - 노드가 디바이스의 역할을 겸할 수도 있다.
- LAN devices는 다양한 기능을 수행하며 다음과 같은 종류가 있다.
 - 여기서는 랜을 확장하고 연결하는 기능을 갖는 디바이스들을 다룬다.
 - 종류 : 리피터(Repeater), 허브(hub), 브리지(bridge), 스위치(switch), 라우터(router), 게이트웨이(gateway)

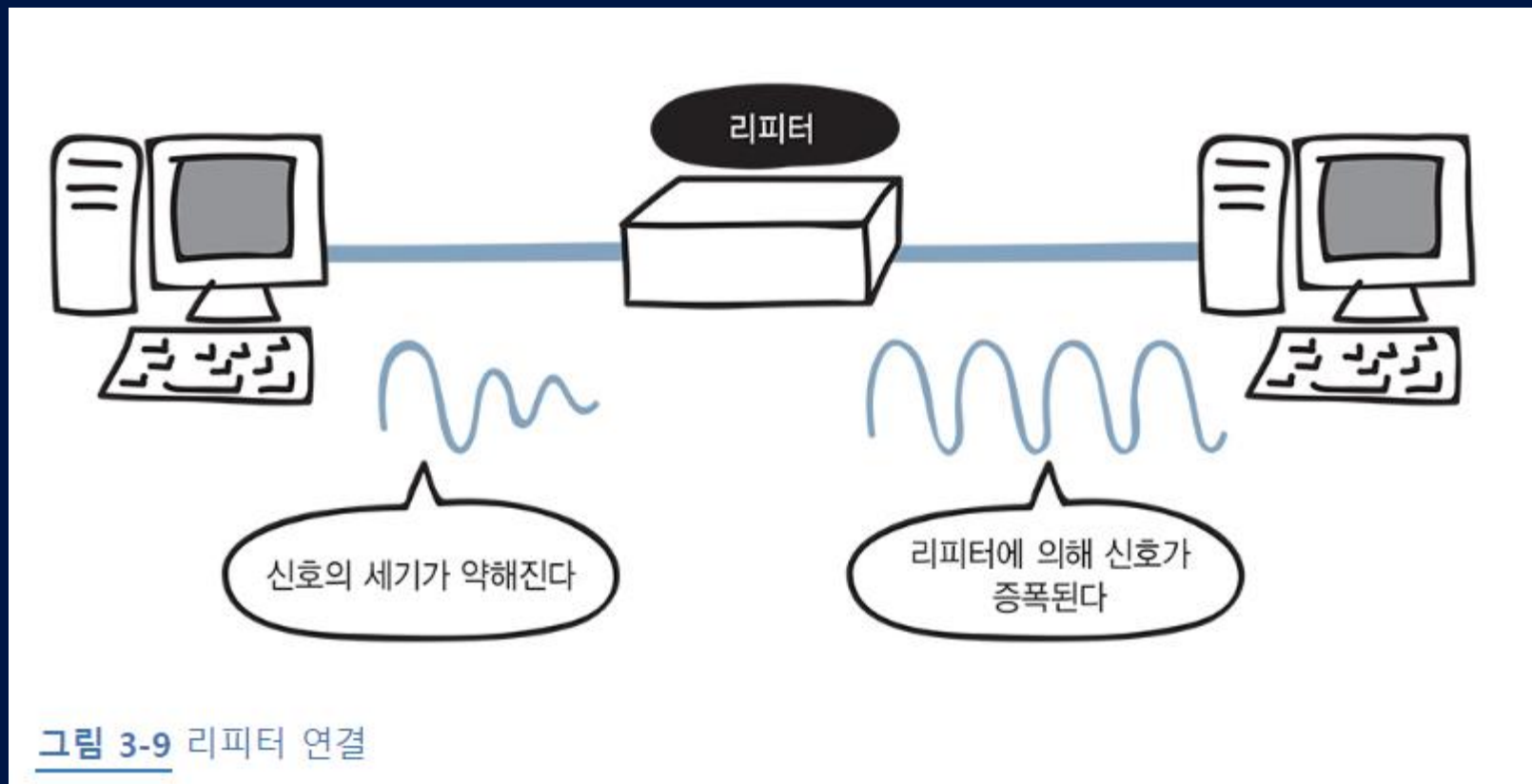
랜 디바이스

● 리피터(Repeater)

- 두개의 랜을 합쳐서 하나의 랜으로 만들어 준다.
 - 두개의 전송 매체를 연결해서 하나로 만든다.
 - 랜의 길이를 연장한다.
- TCP/IP LINK 레이어에 속한다.
 - LINK레이어의 HW부분만 구현하는 경우가 많다.
- 한 쪽의 신호를 받아서 잡음을 제거하고, 약한 신호를 증폭해서 다른 쪽으로 보내는 역할을 한다.
- 랜 디바이스의 기본으로 허브나 스위치도 같은 기능을 갖는다.

랜 디바이스

- 리피터(Repeater)



LAN DEVICES

● 허브(Hub)

- 노드들을 연결하는 장치 이다.
 - 노드 : 컴퓨터(서버, 워크스테이션, 클라이언트), 프린터, 랜 디바이스
 - 1:1 랜들을 모아서 n개의 노드가 연결된 하나의 랜을 구성
- TCP/IP LINK 레이어 장치이다. (OSI Layer 1)
 - 리피터를 여러 개 모아서 연결한 장치이다.
- 허브는 데이터를 건드리지 않고 그냥 복사해서 전체 노드에 보낸다.

랜 허브



그림 3-10 허브는 포트가 여러 개다

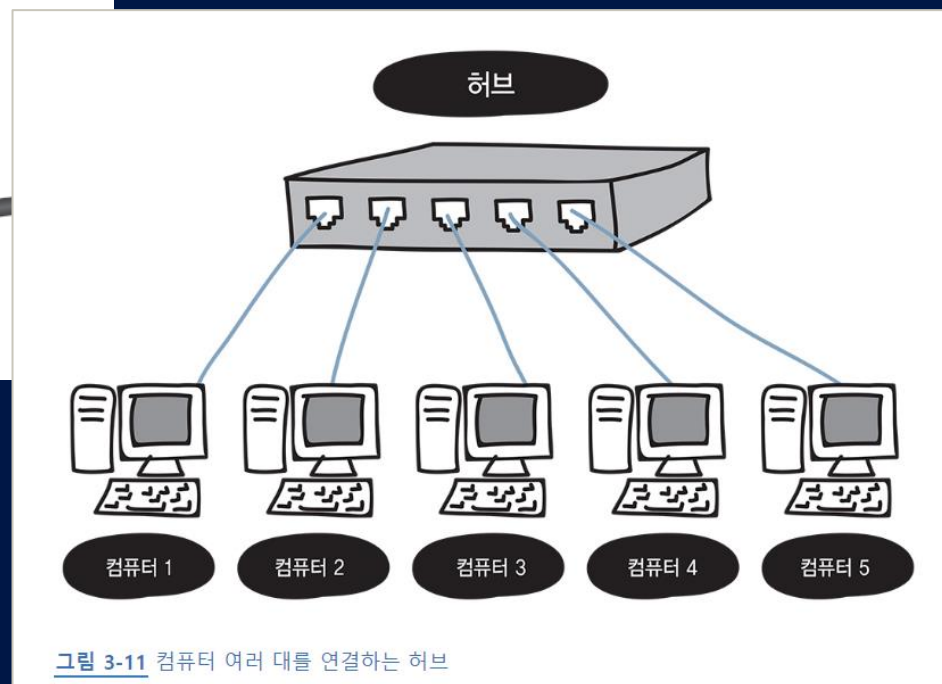


그림 3-11 컴퓨터 여러 대를 연결하는 허브

허브

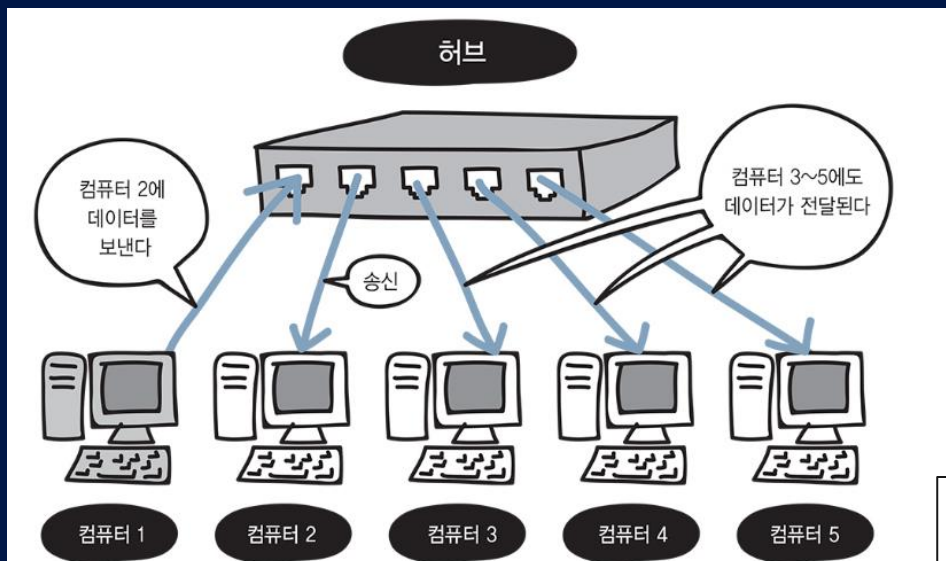


그림 4-2 더미 허브에 데이터를 보내는 경우

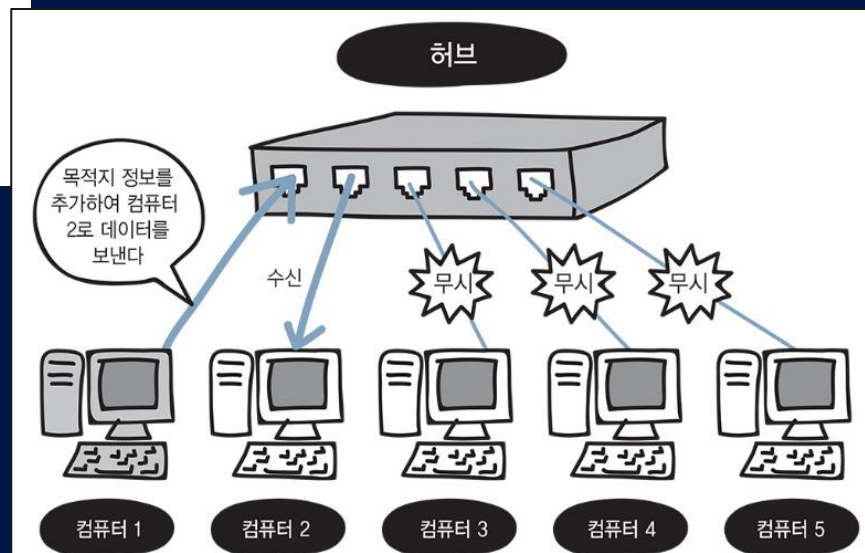


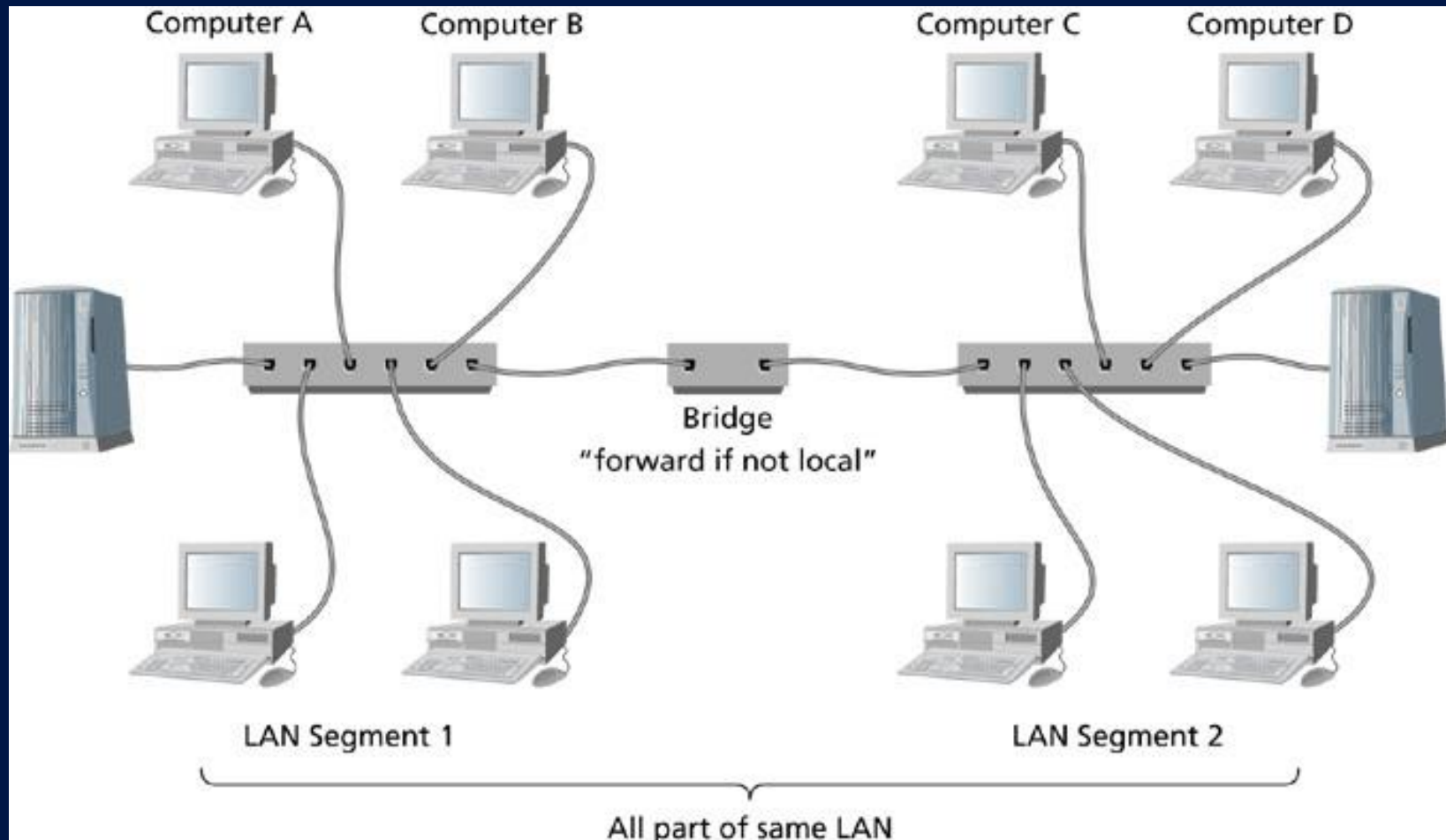
그림 4-3 목적지가 아닌 컴퓨터는 데이터를 무시하는 규칙이 있다

랜 디바이스

- 브리지(Bridge)

- 2개의 LAN 조각(Segment)들을 연결해서 하나의 LAN으로 만들어 주는 디바이스로 네트워크 데이터를 필터링해서 필요한 조각만 전송되도록 한다.
 - MAC 주소로 필터링
- 전체 랜의 성능을 올려준다.
 - CSMA/CD에서 Collision이 줄어든다.
 - 서로 다른 조각에 동시에 다른 데이터가 전송될 수 있다.
- 리피터를 대체한다.

두개의 랜조각을 연결하는 브리지



랜 디바이스

- 브리지
 - 각각의 랜 조각에 속하는 노드들의 주소인 MAC (Media Access Control) 주소를 감지해서 내부 메모리에 저장한다.
 - 저장된 MAC주소를 필터링에 사용한다.

랜 디바이스

● MAC 주소

- 공장에서 붙인 주소, 변경 불가(예외 존재)
 - 개인정보로 활용될 수 있으므로 주의
- 절대로 중복되지 않음
- 48비트

00-23-AE-D9-7A-9A

랜 카드를 만든 제조사 번호 제조사가 붙인 일련번호

그림 4-6 MAC 주소 규칙

```

C:\Users\whnjung>getmac /v

연결 이름      네트워크 어댑터  물리적 주소      전송 이름
-----
이더넷        Marvell AQtion  70-85-C2-33-06-00  미디어 연결 끊김
이더넷 3      Intel(R) Ethern  70-85-C2-33-05-FC  WDevice\Tcpip_{5E6E9D18-D858-4B3A-A077-A1A61E88E37C}
vEthernet (Defa Hyper-V Virtual  00-15-5D-9C-DC-40  WDevice\Tcpip_{F38AEE35-D1EC-4ABF-A91C-6F26E6B41F20}
Bluetooth 네트 Bluetooth Devic 해당 없음        하드웨어가 없음
Wi-Fi 3       Intel(R) Dual B  F4-06-69-D4-8D-66  미디어 연결 끊김
이더넷 5      Intel(R) I211 G  70-85-C2-33-05-FE  미디어 연결 끊김
  
```

랜 디바이스

● 프레임

- LINK 레이어에서 오고 가는 신호의 단위
- MAC주소와 DATA를 포함.
- 트레일러 : 종료 신호, 오류 검사 데이터

목적지 MAC 주소
(6바이트)

출발지 MAC 주소
(6바이트)

유형
(2바이트)

그림 4-7 이더넷 헤더 구조

표 4-1 프로토콜을 식별하는 유형 번호(16진수)

유형 번호	프로토콜
0800	IPv4
0806	ARP
8035	RARP
814C	SNMP over Ethernet
86DD	IPv6

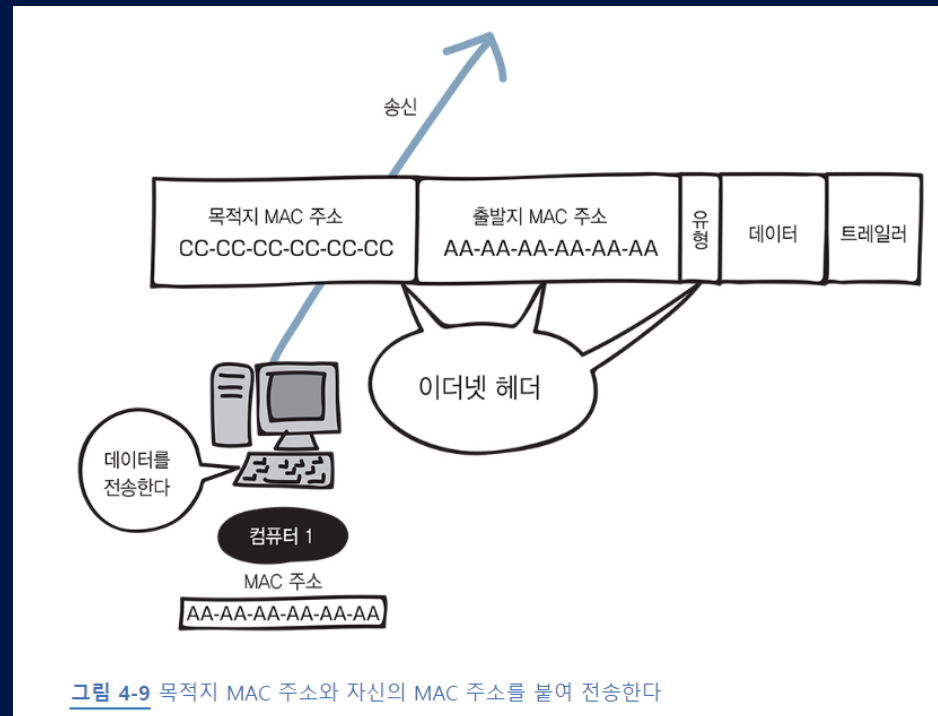


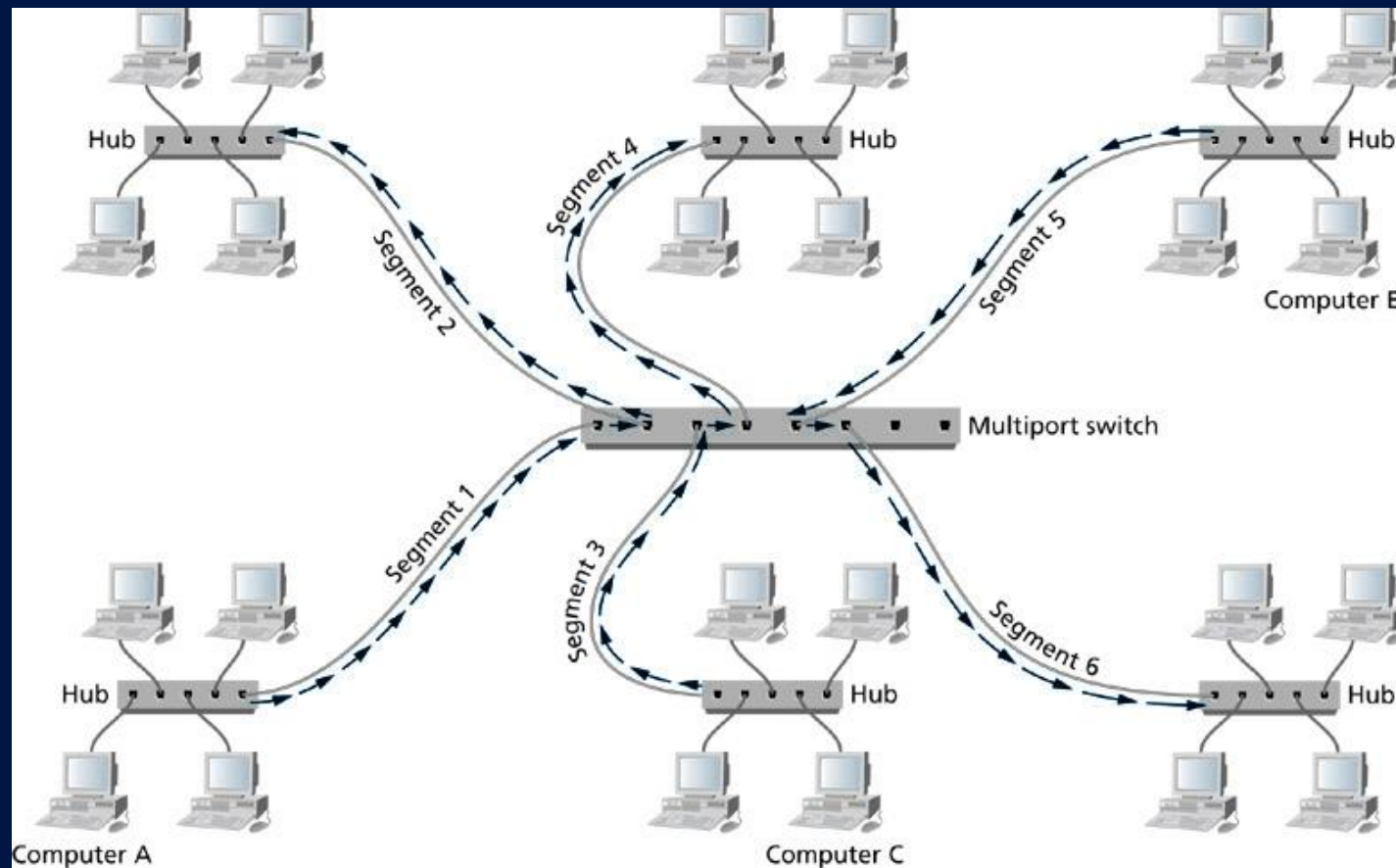
그림 4-9 목적지 MAC 주소와 자신의 MAC 주소를 붙여 전송한다

랜 디바이스

- 스위치(Switch)

- HUB에 브리지 기능 추가
- 스마트 허브 또는 스위칭 허브라고도 불림
- 각각의 포트는 노드 (컴퓨터, 서버, 랜 디바이스) 들과 직접 연결
- 스위치는 각각의 포트를 동시에 읽으면서 데이터를 필요한 포트에 전달한다.
 - 주소를 보고 필요한 노드끼리만 연결
 - 복수의 동시 연결 관리 가능 (한계가 있고, 한계가 높을수록 비싸다.)
 - 브리지가 1:1연결이고 랜을 연결한다면, 포트는 주로 노드들을 연결(LAN연결도 가능) :

랜들을 연결한 스위치



스위치의 노드 관리

MAC 주소 테이블

포트	MAC 주소
①	AA-AA-AA-AA-AA-AA
②	
③	CC-CC-CC-CC-CC-CC
④	
⑤	

컴퓨터 3의
MAC 주소가
등록되어 있다

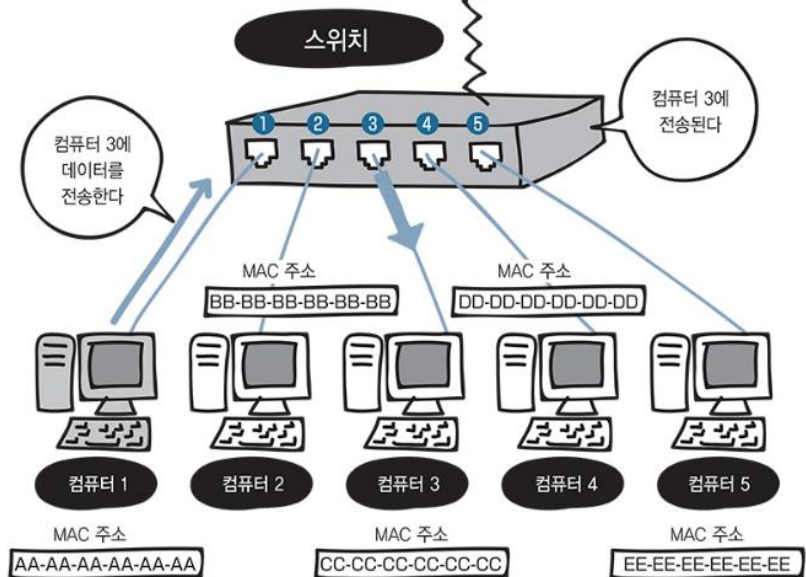


그림 4-14 MAC 주소 필터링

랜 디바이스

- 저장전진형(Store and Forward) 스위치
 - 프레임을 받아서 에러를 검사한 후 에러가 없으면 원하는 포트로 전달하는 방식
 - 손상된 프레임을 걸러내기 때문에 신뢰성이 높음
 - 저장 -> 검사 -> 전달의 순서를 가지므로 다른 형태의 스위치에 비해 딜레이가 있다.
 - 검사를 위해서는 전체 프레임을 다 받아 보아야 한다.

랜 디바이스

- 통과형(Cut Through) 스위치
 - 에러검사를 하지 않는다.
 - 저장전진형 보다 빠르다.
 - 프레임의 주소부분까지만 읽으면 전달을 시작한다.

LAN DEVICES

● 고 레이어 스위치

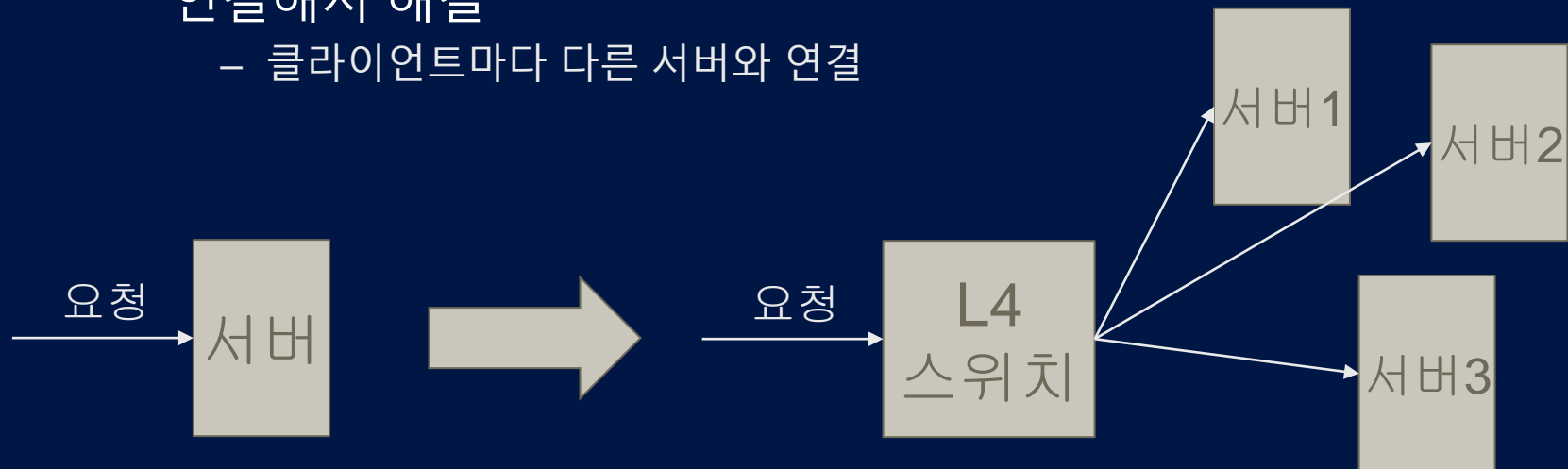
- 일반 스위치들은 TCP/IP LINK 레이어에서 동작.
- 고 레이어 스위치는 TCP/IP Network레이어에서 동작 (여기서 레이어는 OSI 7 계층을 의미)
 - L2 스위치 : 일반 스위치, TCP/IP LINK 레이어에서 동작
 - L3 스위치 : IP 주소를 참조해서 동작, 라우터라고 불림
 - L4 스위치 : IP주소와 Port번호를 함께 참조해서 동작.
 - L7 스위치 : 패킷의 내용을 보고 동작.

LAN DEVICES

● 고 레이어 스위치

– L4 스위치

- IP주소와 Port번호를 함께 참조해서 동작.
- 보통 부하 분산을 위해 사용
 - 예) 웹서버
- 서버 한대에서 처리할 수 없을 만큼 많은 요청이 도착할 경우 => L4스위치를 설치하고 그 뒤에 여러 개의 서버를 LAN에 연결해서 해결
 - 클라이언트마다 다른 서버와 연결



LAN DEVICES

- 고 레이어 스위치

- L4 스위치

- L4스위치에서 설정한 Port번호의 요청들을 여러 컴퓨터에 번갈아 보낸다. 이 때 목적지 IP주소를 변조(NAT)한다.
 - 하나의 주소에서 여러 대의 컴퓨터로 대용량 처리가 가능하다.

- NAT(Network Addresss Translation)

- 패킷을 다른 컴퓨터에서 처리하도록 하게 하기위해 시작 주소나 도착 주소를 바꿔 적는 것.

랜 디바이스

- 라우터(Router)

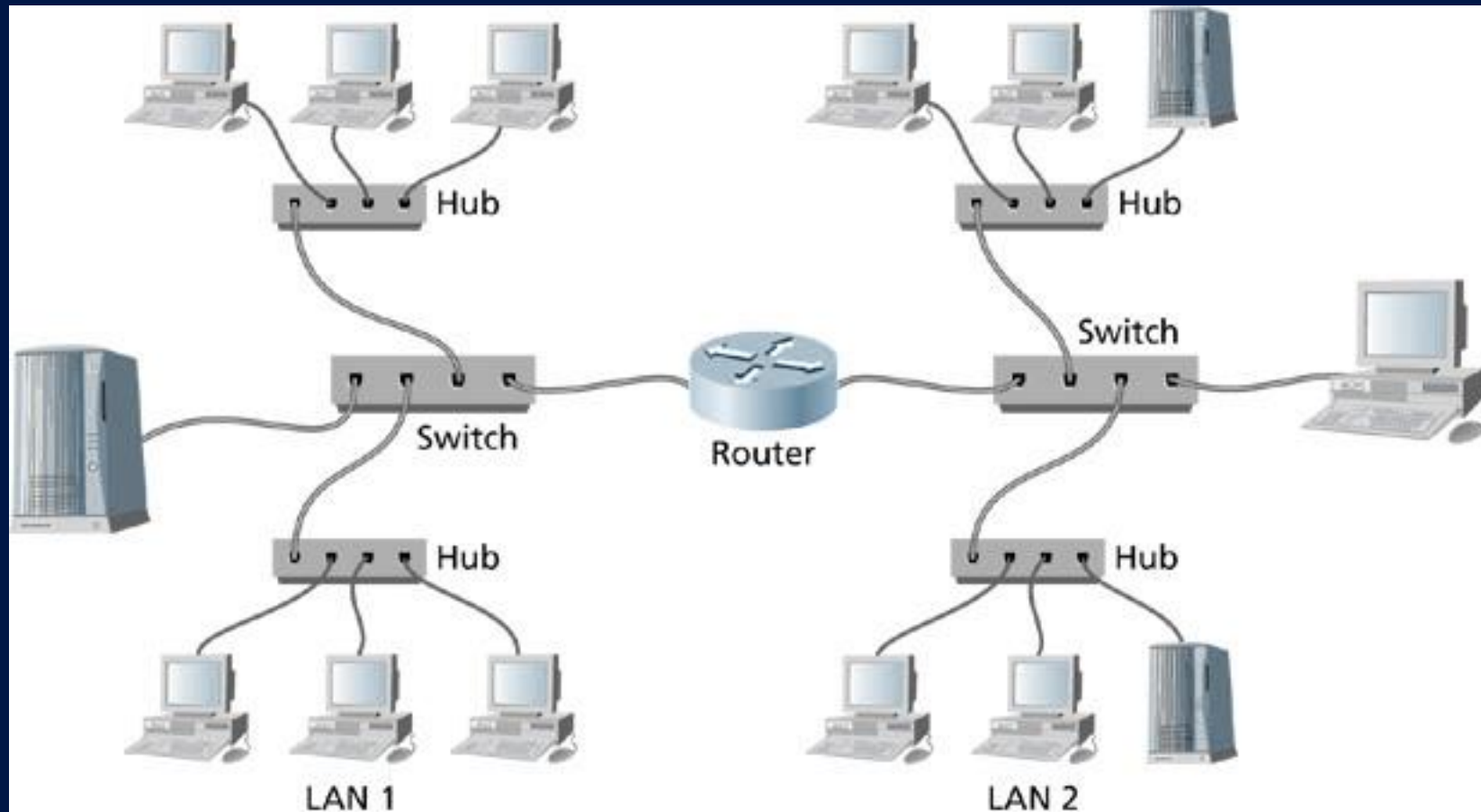
- TCP/IP 인터넷 계층, OSI 3 Layer에서 동작
- L3 Switch라고도 불림
- 서로 다른 랜들을 연결해 준다.
 - 두개의 랜을 하나로 만들어 주는 것과는 다르다.
 - 최소한 연결된 랜의 개수 만큼의 NIC가 필요
 - 하나의 디바이스이지만 연결된 랜의 개수 만큼 다른 논리주소(IP주소)를 갖는다
- 방송(broadcast)이 차단된다.
- MAC주소가 아니라 IP주소로 패킷을 전달한다.
 - MAC 주소 : 물리주소, 공장에서 결정
 - IP 주소 : 논리주소, 네트워크에 연결할 때 결정
- 복수의 경로가 존재할 수 있으며, 그때 그때 최적의 경로를 판단한다.

랜 디바이스

- IP 주소

- 서로 다른 LAN에 연결된 노드들을 찾아갈 수 있게 해주는 주소
- 4 바이트(LAN 주소 + NODE 주소)로 구성
 - LAN 주소는 전세계에서 중복 되지 않아야 함.
 - 중복되면 찾아갈 수 없음
- 같은 LAN주소를 갖는 노드들은 하나의 LAN으로 연결
- 라우터는 다른 LAN주소를 갖는 네트워크를 연결
 - 같은 LAN주소를 갖는다면 오동작을 함.

라우터로 연결된 두 LAN

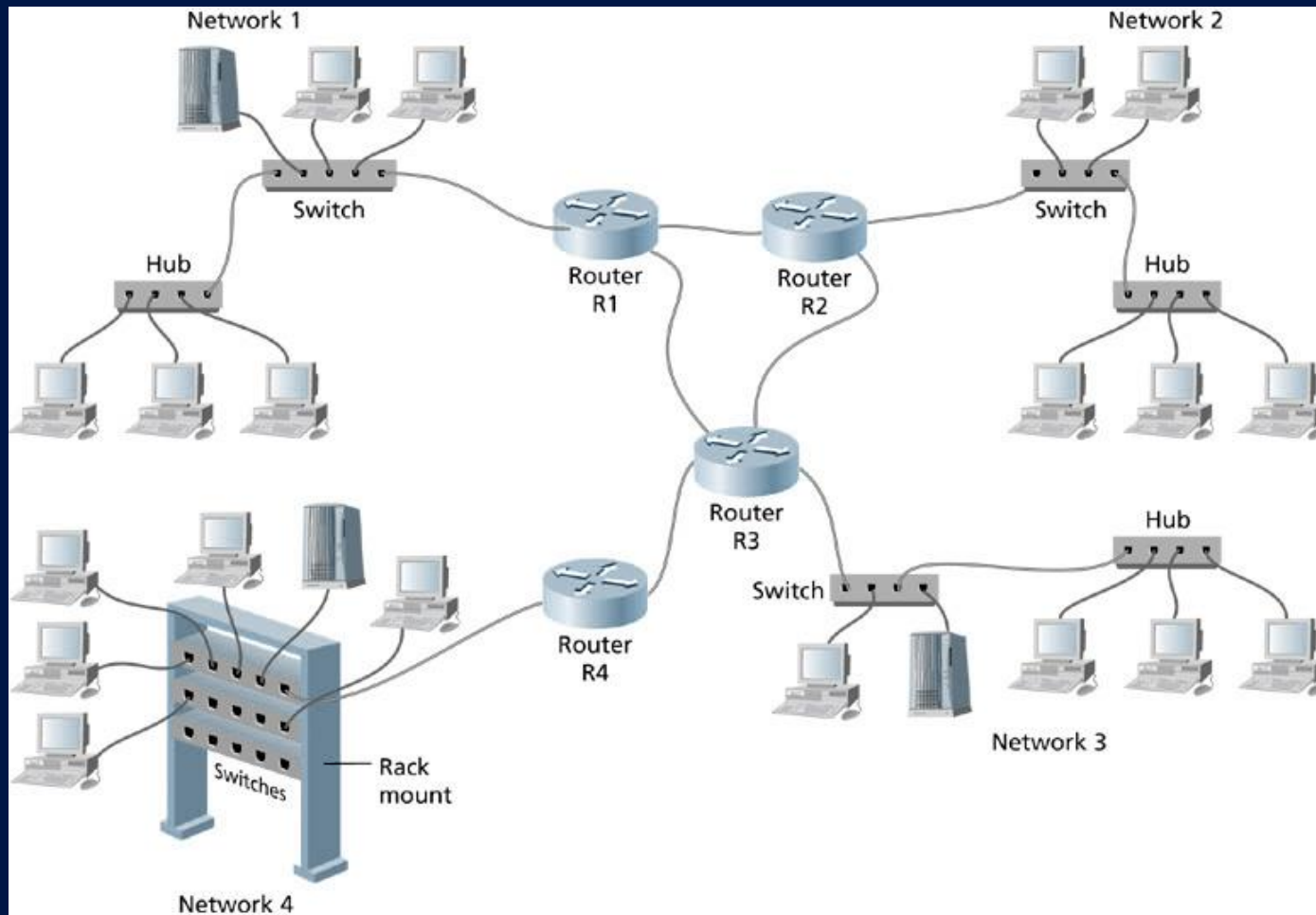


LAN DEVICES

- 라우터

- 전세계의 모든 LAN들은 라우터를 통해서 연결됨
- 하나의 라우터는 여러 개의 Port를 가질 수 있음.
 - 많을 수록 비쌈.
- 라우터는 자신에게 온 패킷의 목적 IP 주소에 대해 알맞은 전송 Port를 판단해야 함.
 - Port별 전송가능 LAN주소 목록 필요.
 - 상황에 맞춰 변경 가능해야 한다. (옵션)
- 라우터로 연결하면 전체전송(방송, broadcasting)을 차단하므로 충돌을 막고 정보 누출을 막아 성능과 보안성이 향상된다.

라우터로 연결된 여러 개의 LAN



LAN DEVICES

- 길찾기

- 정적 길찾기 (Static routing)
 - 관리자가 routing정보 직접 입력
- 동적 길찾기 (Dynamic routing) – 인근 router와 협조 하여 자동적으로 길찾기 정보 생성, 동적 길찾기 알고리즘 필요
 - 라우터의 운영체제에서 관리, 전송 딜레이를 모니터하면서 더 나은 경로를 발견하면 수정
 - 표준 프로토콜을 사용해 라우터끼리 정보 교환.
 - RIP, IGRP, EIGRP, OSPF, BGPV4...

LAN DEVICES

● IP 길 찾기

- 데이터와 IP주소를 묶어서 IP 패킷을 만들어 전달

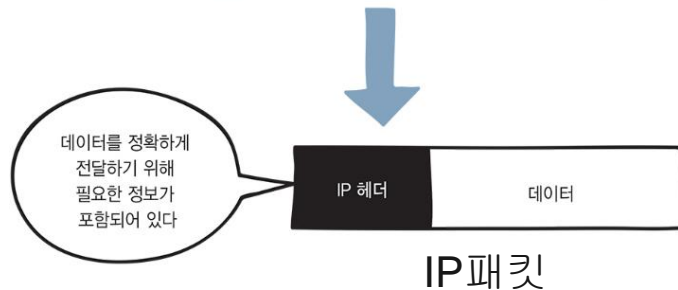
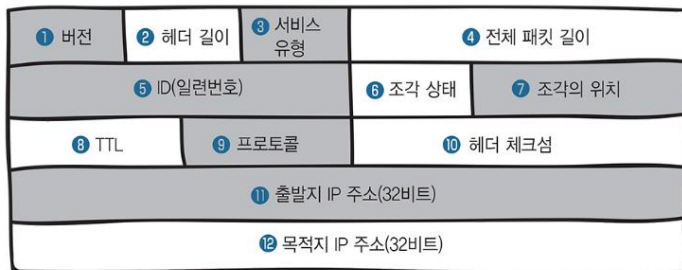


그림 5-5 IP 헤더

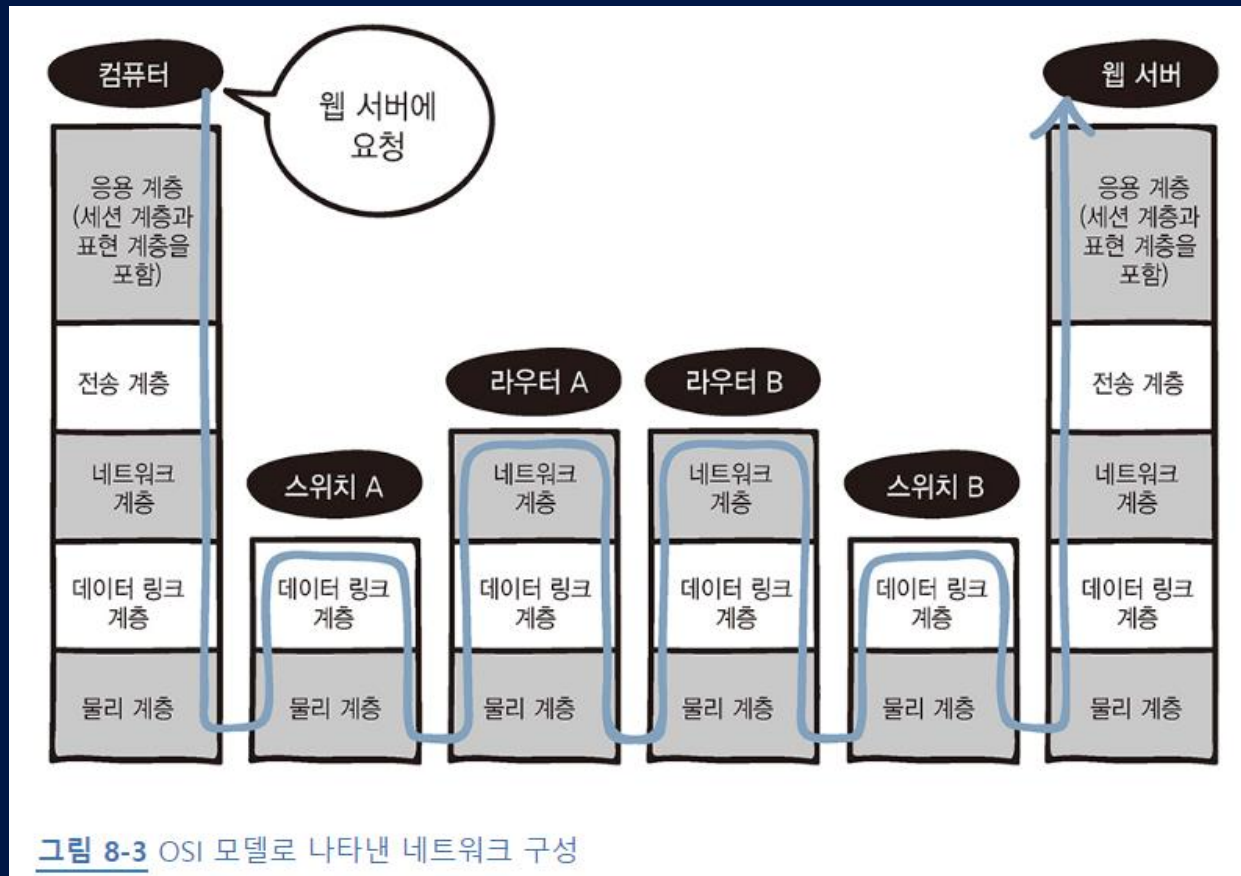


네트워크 계층에서는 캡슐화할 때 그림 5-5와 같은 **IP 헤더**를 붙여요.

지면이 한정되어 있어서 가로로 길게 보여 주지는 못하지만 실제로는
 ① 버전(version), ② 헤더 길이(header length), ③ 서비스 유형(service type), ④ 전체 패킷 길이(total length), ⑤ ID(identification, 일련번호), ⑥ 조각 상태(flags), ⑦ 조각의 위치(fragment offset), ⑧ TTL, ⑨ 프로토콜(protocol), ⑩ 헤더 체크섬(header checksum), ⑪ 출발지 IP 주소(source IP address), ⑫ 목적지 IP 주소(destination IP address) 순서로 헤더 정보가 구성되어 있어요.

LAN DEVICES

- IP 길 찾기

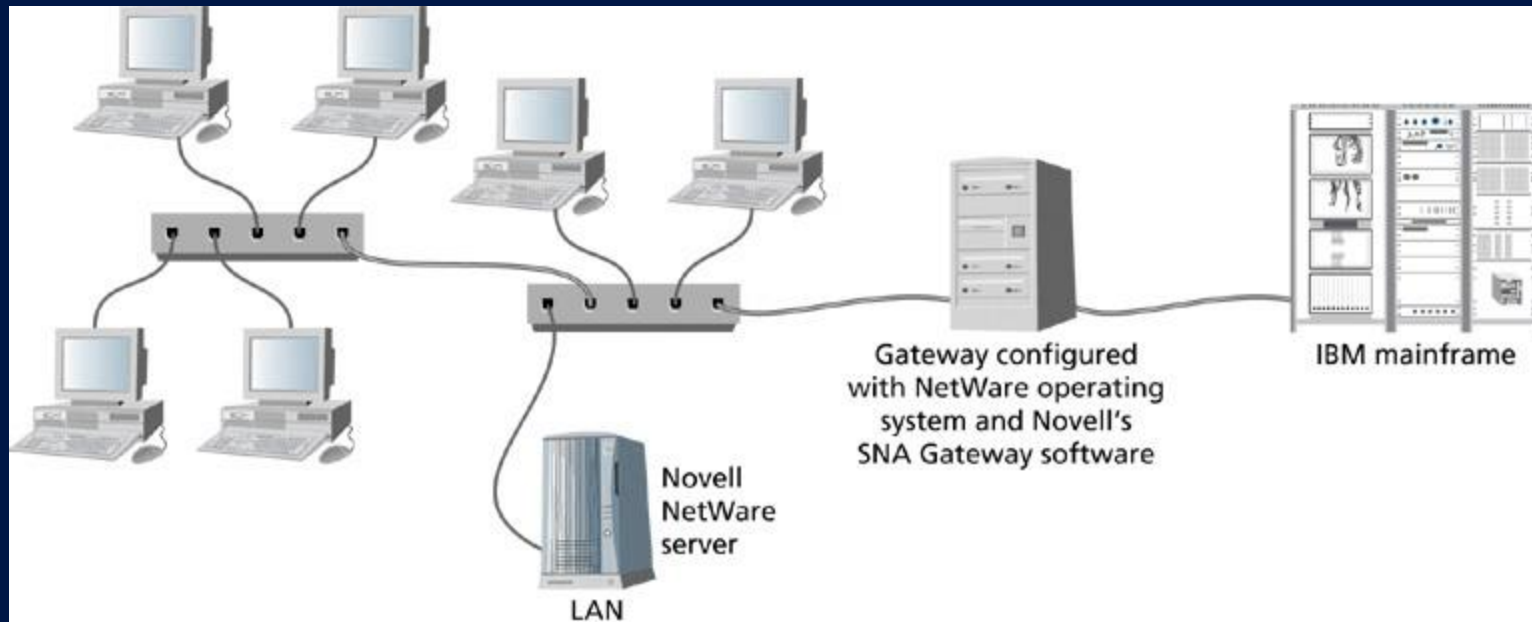


LAN DEVICES

- 게이트웨이 (Gateways)

- 서로 다른 구조를 갖는 네트워크를 연결해 주는 장치. HW나 SW또는 둘 다 사용해 구현
 - 예) 이더넷을 토큰링에 연결, 백본 망에 이더넷 연결
 - 예) email을 프린트해서 할머니에게 전달
 - 예) Web서버를 통한 SMS 전송
- TCP/IP 인터넷 레이어 위의 계층을 통해 구현된다.

게이트웨이를 통한 메인프레임 연결



LAN 백본(BACKBONES)

- Network Backbones (백본, 중추)
 - 여러 개의 LAN을 연결하는 고속 연결 링크
 - 여러 LAN이 공유하므로 고속이 필요
 - 예) 간단한 백본



랜 백본 - 설계

- 필요 특성

- Fault tolerance(고장 감내성)

- 일부 장치가 고장나도 동작.
 - SPOF 회피, 중복 및 우회로 필요.

- Load balancing(부하 균등화)

- Fault tolerance와 대역폭 확장을 위해 복수의 연결선 제공
 - 복수의 연결선을 균일하게 사용하게 해주는 기능.

랜 백본 - 구현

- 초고속 전송 구조 필요
 - 중거리 전송도 필요.
- 예) FDDI, 광통신 네트워크, ATM, 10 Gigabit Ethernet, and 100 Gig Ethernet.

랜 프로토콜

- **프로토콜(Protocol)**은 서비스나 장치들이 정보를 어떻게 교환할지 정하는 규칙이다.
 - 한국인들 사이의 프로토콜 => 한국어
- **프로토콜의 규칙**
 - 데이터를 어떻게 묶을 것인가 (패킷 포맷)
 - 데이터를 어떻게 네트워크 장치에 실을 것인가
 - 데이터를 어떻게 전달할 것인가
 - 받은 데이터에서 어떻게 원래 데이터를 꺼낼 것인가
- **프로토콜의 요소**
 - 데이터의 의미 (포맷)
 - 데이터의 전달 순서

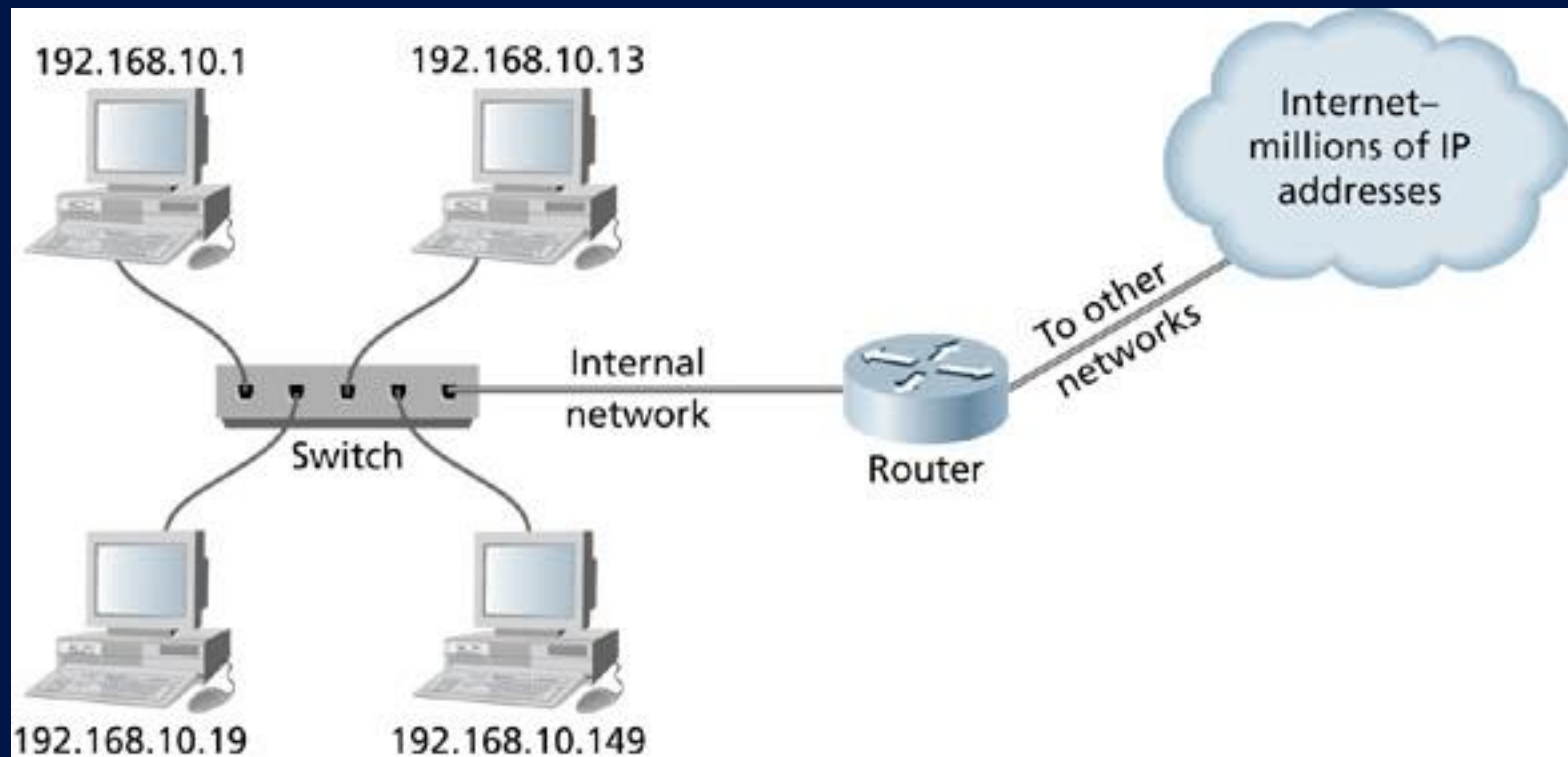
LAN PROTOCOLS

- 통신(Communication) 프로토콜
 - 이를 통해 출발지에서 목적지까지 데이터를 전송한다.
 - 정보 교환의 기본 구성 요소.
 - 예) 인터넷 프로토콜(The Internet Protocol)

LAN PROTOCOLS

- 인터넷 프로토콜(The Internet Protocol)
 - LAN과 NODE를 찾아내게 해주는 주소 정책을 포함한다.
 - 주소만 알면 전세계의 어디에 있는 노드와도 정보를 교환할 수 있게 해준다.

인터넷 프로토콜과 인터넷



LAN PROTOCOLS

- IP주소

- 현재 IPv4 프로토콜을 많이 사용
 - IPv6로 이전해야 하는데 계속 발목을 잡고 있음.
- IPv4는 네 개의 8비트 옥텟(Octet)으로 이루어진 32비트 2진수 주소를 사용한다.

32비트 IP 주소



```
C:\> 명령 프롬프트
C:\Users\whnjung> ipconfig

Windows IP 구성

이더넷 어댑터 이더넷 4:

    연결별 DNS 접미사. . . . . : mshome.net
    링크-로컬 IPv6 주소 . . . . . : fe80::20b4:1262:d71e:b998%7
    IPv4 주소 . . . . . : 172.18.12.156
    서브넷 마스크 . . . . . : 255.255.240.0
    기본 게이트웨이 . . . . . : 172.18.0.1
```

LAN PROTOCOLS

- IP주소
 - ‘.’으로 구분된 4개의 십진수로 구성된 주소를 많이 사용
 - 예) 192.168.1.5
 - 예) 216.239.32.10 (www.google.com)
 - 각 숫자는 8비트이기 때문에 0에서 255까지의 값을 갖는다.
 - 0과 255는 특별하게 취급되므로 잘 사용되지 않는다.

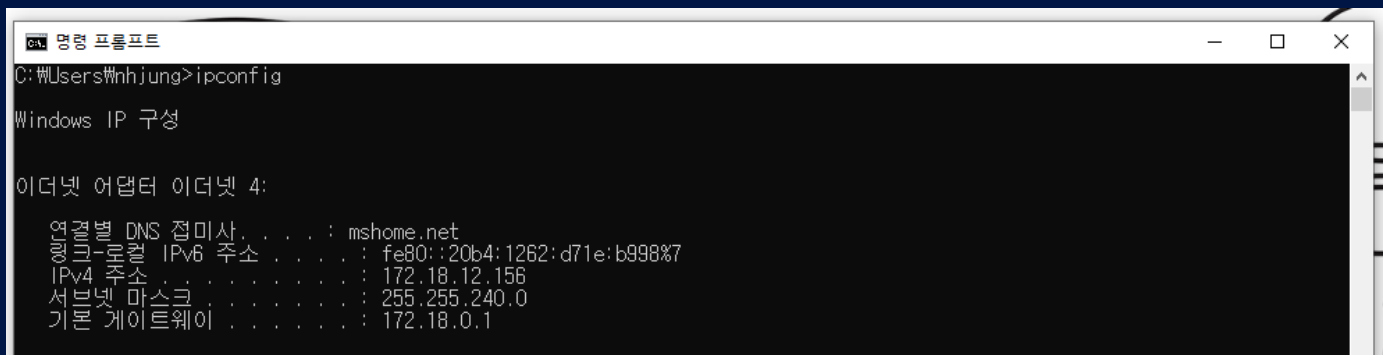
십진수와 2진수 표현

192								.	168								.	10								.	149								IP address
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^n power
128	64	32	16	8	4	2	1		128	64	32	16	8	4	2	1		128	64	32	16	8	4	2	1		128	64	32	16	8	4	2	1	Decimal equivalent
1	1	0	0	0	0	0	0		1	0	1	0	1	0	0	0		0	0	0	0	1	0	1	0		1	0	0	1	0	1	0	1	Binary representation of an IP address

LAN PROTOCOLS

- IP 주소

- IP 주소는 네트워크 주소와 노드 주소로 나뉘어 있다.
 - 보통 앞부분이 네트워크 주소, 뒷부분이 노드 주소
 - 이것을 구분 해주는 값을 서브넷 마스크(subnet mask)라고 부른다.
- 서브넷 마스크도 32비트 2진수이다.
 - 1인 비트가 네트워크 주소를 0인 비트가 노드 주소를 뜻한다.
 - 255.255.240.0 -> FF.FF.F0.00 -> 11111111 11111111
11110000 00000000 => 위의 20비트가 랜주소, 12비트가
노드주소



```
cmd 명령 프롬프트
C:\Users\Wnhjung>ipconfig

Windows IP 구성

이더넷 어댑터 이더넷 4:

    연결별 DNS 접미사 . . . . . : mshome.net
    링크-로컬 IPv6 주소 . . . . . : fe80::20b4:1262:d71e:b998%7
    IPv4 주소 . . . . . : 172.18.12.156
    서브넷 마스크 . . . . . : 255.255.240.0
    기본 게이트웨이 . . . . . : 172.18.0.1
```

랜 주소 노드주소 구분

Class C address

IP address	205								.	122								.	10								.	5								Dotted decimal
Subnet mask	255								.	255								.	255								.	0								Dotted decimal
	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰		2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰		2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰		2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	
	128	64	32	16	8	4	2	1		128	64	32	16	8	4	2	1		128	64	32	16	8	4	2	1		128	64	32	16	8	4	2	1	
IP address	1	1	0	0	1	1	0	1		0	1	1	1	1	0	1	0		0	0	0	0	1	0	1	0		0	0	0	0	0	1	0	1	Binary format
Subnet mask	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		0	0	0	0	0	0	0	0	Binary format
Network address																Node address																				

랜 프로토콜

- 랜 주소에는 5단계 등급이 있다.
 - **Class A** – 첫 비트가 0인 주소. 1.0.0.0부터 126.255.255.255까지가 여기에 속한다.
 - 첫번째 옥텟이 네트워크 주소이다. (총 126개)
 - 총 126개가 있다.
 - 서브넷 마스크는 255.0.0.0이다.
 - **Class B** – 첫 두 비트가 10인 주소. 128.0.0.0부터 191.255.255.255까지가 여기에 속한다.
 - 앞의 두개의 옥텟이 네트워크 주소이다 (총 $126 * 256$ 개)
 - 서브넷 마스크는 255.255.0.0이다.
 - **Class C** – 첫 세 비트가 110인 주소. 192부터 223이 여기에 속한다.
 - 앞의 세개의 옥텟이 네트워크 주소이다. (총 $126 * 256 * 256$ 개)
 - 서브넷 마스크는 255.255.255.0이다.

LAN PROTOCOLS

- 주소 등급
 - **Class D** – IP multicast에 사용. 첫 네 비트가 1110. 224부터 239사용
 - **Class E** – 정의된 네트워크 내에서 전체전송에 사용 240에서 255까지 사용.

Network and Node Portions of IP Addresses

Class A address

95	122	140	89
255	0	0	0
2 ⁷ 2 ⁶ 2 ⁵ 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰	2 ⁷ 2 ⁶ 2 ⁵ 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰	2 ⁷ 2 ⁶ 2 ⁵ 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰	2 ⁷ 2 ⁶ 2 ⁵ 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰
128 64 32 16 8 4 2 1	128 64 32 16 8 4 2 1	128 64 32 16 8 4 2 1	128 64 32 16 8 4 2 1
0 1 0 1 1 1 1 1	0 1 1 1 1 0 1 0	1 0 0 0 1 1 0 0	0 1 0 1 1 0 0 1
1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
<div> <div>Network address</div> <div>Node address</div> </div>			

Class B address

162	242	129	118
255	255	0	0
2 ⁷ 2 ⁶ 2 ⁵ 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰	2 ⁷ 2 ⁶ 2 ⁵ 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰	2 ⁷ 2 ⁶ 2 ⁵ 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰	2 ⁷ 2 ⁶ 2 ⁵ 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰
128 64 32 16 8 4 2 1	128 64 32 16 8 4 2 1	128 64 32 16 8 4 2 1	128 64 32 16 8 4 2 1
1 0 1 0 0 0 1 0	1 1 1 1 0 0 1 0	1 0 0 0 0 0 0 1	0 1 1 1 0 1 1 0
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
<div> <div>Network address</div> <div>Node address</div> </div>			

LAN PROTOCOLS(2021)

- IP 주소 설정 Assigning IP Addresses
 - 수동 미리 값을 수동으로 정해 놓고 설정
 - 자동 노드주소를 자동으로 할당해주는 프로토콜인 DHCP(Dynamic Host Configuration Protocol) 서버 프로그램을 통해 구현. DHCP는 IP주소 자동 할당 프로토콜
- 보통 허브나 스위치에서 해 줌

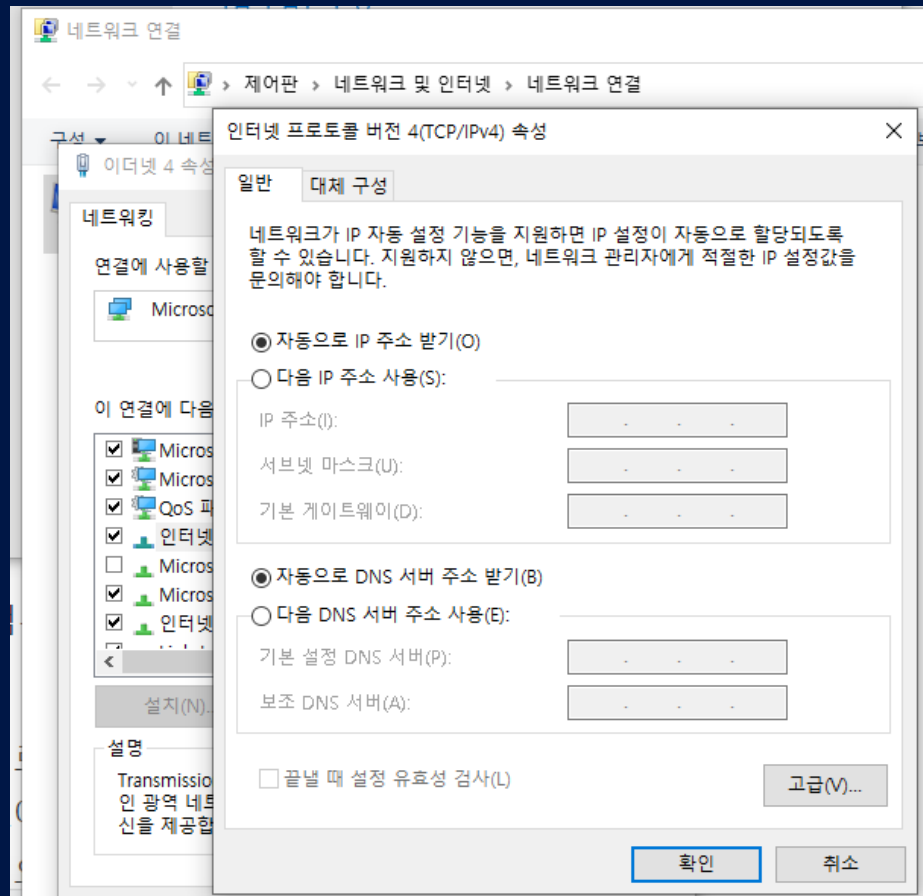
LAN PROTOCOLS

- 수동 IP 설정

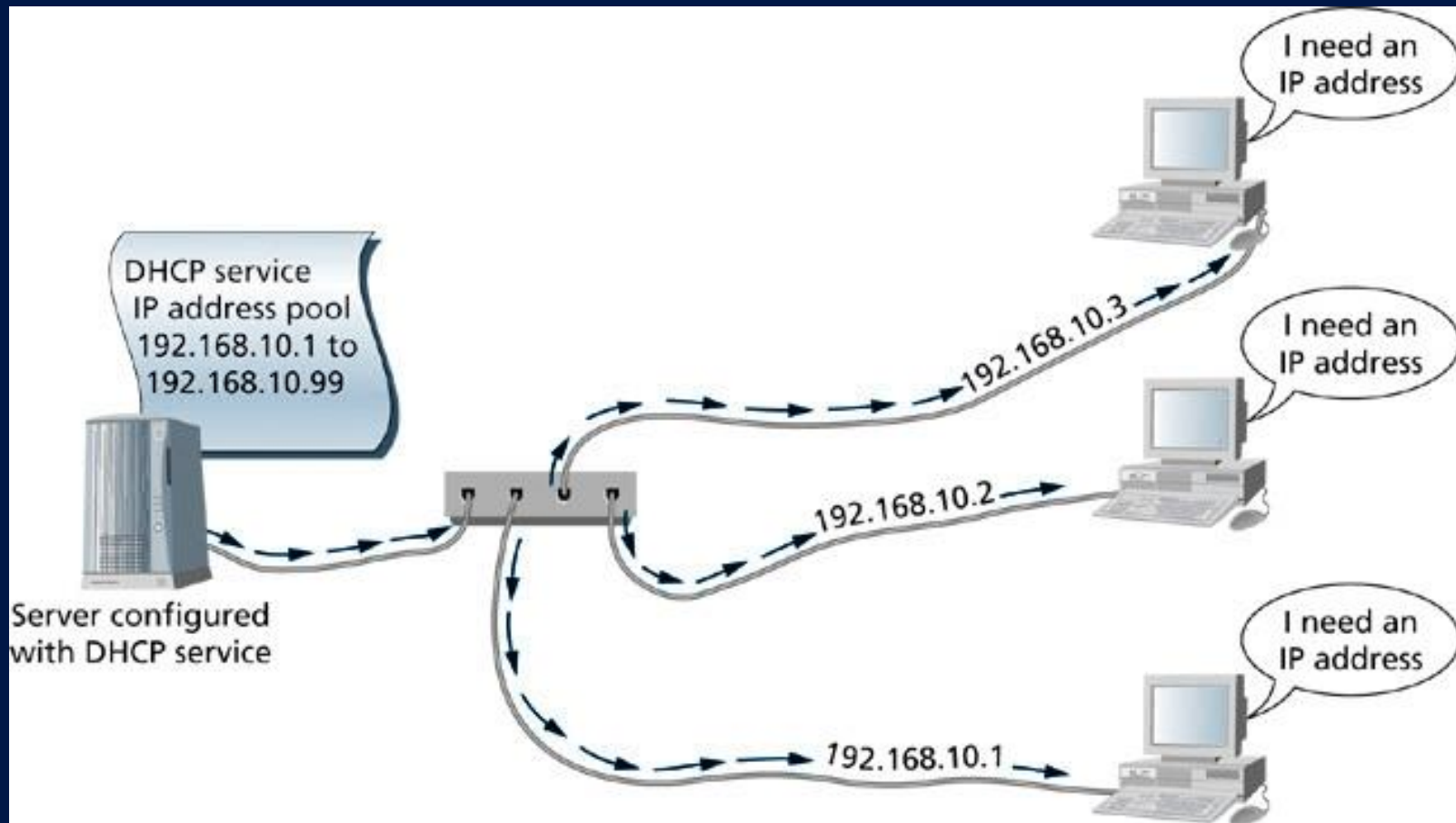
1. 노드가 설치될 LAN의 정보를 얻는다.
 - 랜주소, 랜마스크, 게이트웨이 주소
 - DNS 서버 주소 (다른 챕터에서 설명)
 - 노드의 IP 주소
2. 노드를 랜에 연결한다.
3. 노드의 IP 주소, 랜마스크, 게이트웨이 주소를 설정한다.
4. 노드의 DNS 서버 주소를 설정한다.

LAN PROTOCOLS

● IP 주소 설정 Assigning IP Addresses



DHCP 서버



LAN PROTOCOLS

- 네트워크 관리
 - **SNMP(Simple Network Management Protocol)**은 표준 네트워크 관리 프로토콜로서 Application Layer에서 구축된다.
 - Manager와 Agent로 구분된다.
 - Manager : 정보를 수집 관리
 - Agent : 모든 노드에 존재하면서 Manager에게 정보 전송
 - 정보 : 보안, 네트워크 사용량, 노드간 전송 속도, 노드의 상태, 노드의 타입 (브리지, 허브, 스위치, 라우터)

부록 : 공유기

● 예) IPTIME, 넷기어, TP-LINK

예) 다나와



1 EFM ipTIME AX2004M 유무선공유기

유무선 / 기가비트 / AX1800(Wi-Fi 6) / 최고무선속도: 1201Mbps / 라우터형 / 전원플러그형 / 미디엄텍 MT7621 / 듀얼 코어 / 880 MHz 이상 / RAM: 256MB / FLASH: 128MB / WAN: 1포트(기가비트) / LAN: 4포트(기가비트) / USB 3.0: 1개 / 듀얼 밴드 / 안테나: 4개 / Mesh / MU-MIMO / IPTV 지원 / DDNS / DLNA / WOL / 모바일 관리 어플 / VPN / QoS / 빔포밍 / 멀티 SSID / 모바일 UI / OFDMA / 컨트롤러 / 무선 에이전트 / 유선 에이전트 / 무상 2년

88,000원 ☐ 404물 ☐

관련기사 와이파이6 시대를 저력하게 대비해보자, EFM ipTIME AX2004M 유무선공유기

사용기 가성비 넘치는 기가와이파이 공유기 ipTIME AX2004M

등록월 2021.02 | 상품익권 74건 | 관심상품



2 EFM ipTIME A3004T 유무선공유기

유무선 / 기가비트 / AC1300(Wi-Fi 5) / 최고무선속도: 867Mbps / 라우터형 / 전원플러그형 / 미디엄텍 MT7621 / 듀얼 코어 / 880 MHz / RAM: 256MB / FLASH: 128MB / WAN: 1포트(기가비트) / LAN: 4포트(기가비트) / USB 3.0: 1개 / 듀얼 밴드 / 안테나: 4개 / Mesh / MU-MIMO / IPTV 지원 / DDNS / DLNA / WOL / 모바일 관리 어플 / VPN / QoS / 빔포밍 / 멀티 SSID / 모바일 UI / 컨트롤러 / 무선 에이전트 / 유선 에이전트 / 무상 2년

65,900원 ☐ 688물 ☐

관련기사 [넷알못 사전] 'VPN'이 무엇인가요?

사용기 가정용 와이파이 공유기 ipTIME 마이피타임 A3004T

등록월 2021.01 | 상품익권 117건 | 브랜드로그 | 관심상품



3 ASUS RT-AX56U 유무선공유기

유무선 / 기가비트 / AX1800(Wi-Fi 6) / 최고무선속도: 1201Mbps / 라우터형 / 전원플러그형 / 브로드컴 BCM6755 / 쿼드 코어 / 1.5 GHz / RAM: 512MB / FLASH: 256MB / WAN: 1포트(기가비트) / LAN: 4포트(기가비트) / USB 3.0: 1개 / USB 2.0: 1개 / 듀얼 밴드 / 안테나: 2개 / Mesh / MU-MIMO / IPv6 / IPTV 지원 / DDNS / DLNA / WOL / 모바일 관리 어플 / VPN / QoS / 자동 QoS / 빔포밍 / 멀티 SSID / 모바일 UI / OFDMA / 자녀보호기능 / 컨트롤러 / 무선 에이전트 / 유선 에이전트 / 무상 3년 / TWT

■ 오늘 그만 보기

141,000원 ☐ 293물 ☐

기획전 공유기 구매가이드

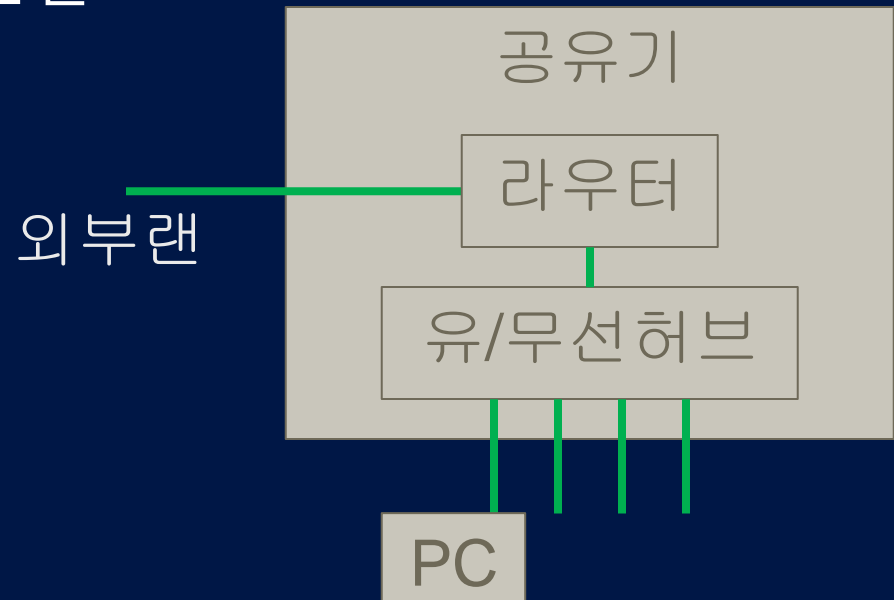
부록 : 공유기

- 핵심기능

- Router : 외부 네트워크 연결
 - 외부 전용 연결 포트 존재
- HUB : 보통 4포트
- 무선 AP

- 보조기능

- DHCP
- 보안 (무선 보안)
- L3/L4 Switch
- 편의기능) 포트포워딩, 네트워크 저장장치, 원격 프린트...



부록 : 공유기

- 공유기 : Router

- Bridge가 아니다!

- bridge로 동작하게 할 수도 있다. (대개 리피터로 동작)
 - DHCP끄고, 내부 연결 포트만 사용

- 내부 연결은 독자적인 LAN으로 연결

- LAN 주소 할당??? : 표준 주소가 있음
 - 사설망 (Private network)이라는 표준 주소가 있고, 이 주소는 내부적으로만 사용하고 외부에 노출되지 않도록 되어 있다.
 - 목록 : 10.x.x.x, 172.16.x.x ~ 172.31.x.x, **192.168.x.x**
 - 따라서 외부에서 공유기 안쪽의 컴퓨터의 접속은 불가능하다.
 - 전세계의 수 많은 192.168.1.5중 어느 것????

부록 : 공유기

- 공유기 : Router : 외부에서 접속
 - 외부에서 접속 가능한 것은 공유기 자체
 - 공유기의 외부 포트가 할당 받은 주소
 - 공유기 관리 메뉴에서 확인 가능

The screenshot shows the web management interface of an ipTIME T3008 router. The left sidebar contains a menu with options like '기본 설정' (Basic Settings), '시스템 요약 정보' (System Summary Information), '인터넷 설정 정보' (Internet Setting Information), and '펌웨어 업그레이드' (Firmware Upgrade). The main area displays '시스템 요약 정보' (System Summary Information) with a table of network status.

인터넷 정보			
인터넷 연결 상태	인터넷에 정상적으로 연결됨		
인터넷 연결 방식	동적 IP 연결	외부 IP 주소	175.208.1
인터넷 연결 시간	178 일 15 시간 36 분 20 초		
내부 네트워크 정보			
내부 IP주소	192.168.0.1		
DHCP 서버 상태	DHCP 서버 동작 중		
동적 IP 할당 범위	192.168.0.2 - 192.168.0.254		
기타 정보			

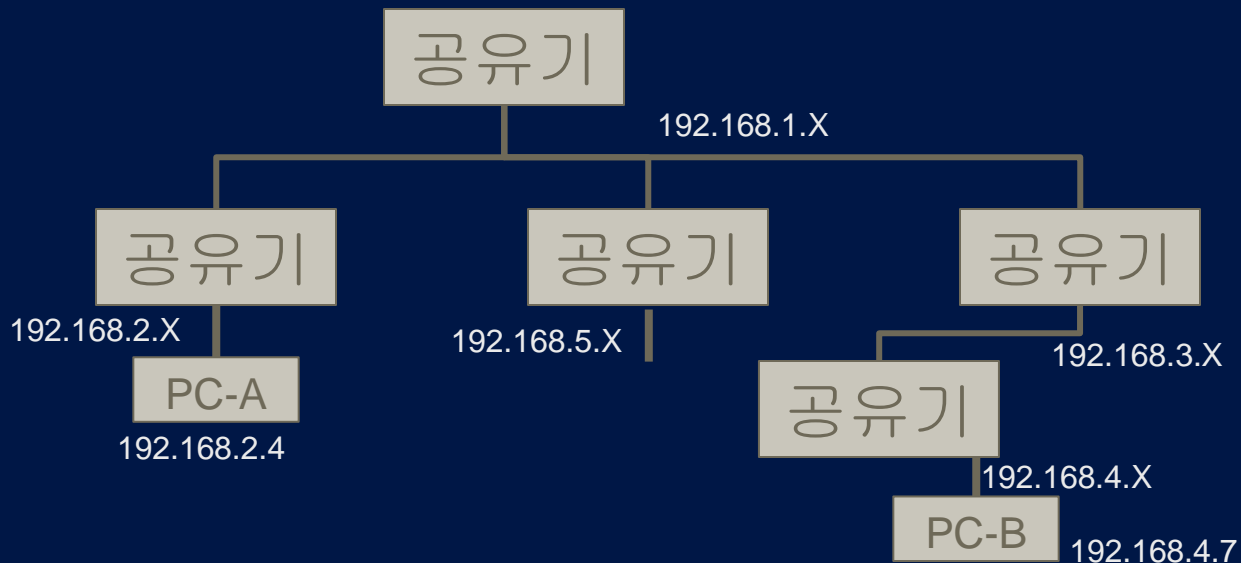
부록 : 공유기

- 공유기 : Router : 외부에서 접속
 - 공유기의 외부 IP주소도 사설망일 때??
 - 외부 공유기의 설정 변경 : 접근 가능?
 - 또는 외부 도움 필요 (예:사설 DDNS, VPN)
 - 아니면 공유기에서 설정 가능
 - **DMZ** : 외부에서 오는 접속을 특정 노드에 몰아주기
 - **포트 포워딩** : 외부에서 오는 접속을 종류 별로 해당 노드에 연결시키기 (보통 포트번호 사용)

부록 : 공유기

- 공유기 : Router

- 라우팅 테이블 필요 없음.
 - 모든 패킷이 외부 포트로 나감.
- 공유기 아래에 공유기? 그 아래에 또 공유기?
 - 라우팅 테이블 기능이 있는 공유기 필요.
 - 실제로는 아래의 공유기들을 그냥 허브로 사용하면 됨 (속도 저하의 주범)



부록 : 공유기

- 공유기 : Router

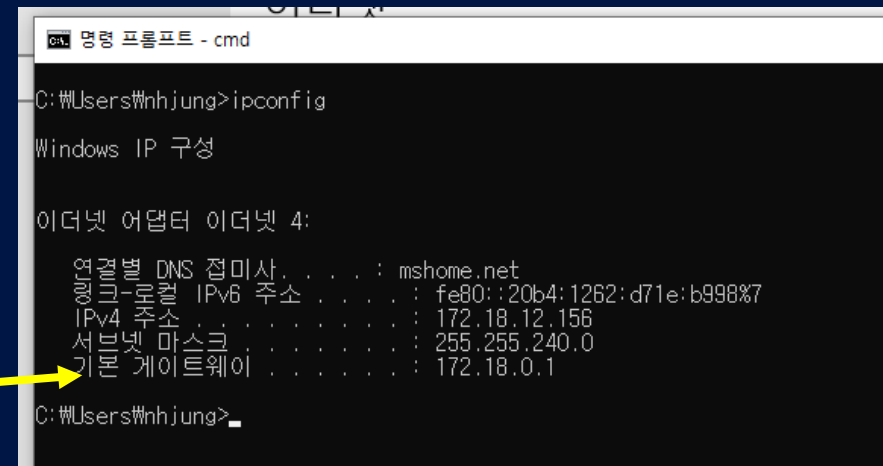
- 외부에서 보았을 때 내 주소가 192.168이 아니던데???
- 접속 시작은 내가 했더라도 밖에 서버에서 데이터를 보내야 함.
- 이때는 공유기 주소로 보냄
 - 그렇게 하기 위해서 내가 보내는 모든 IP 패킷의 출발지 IP주소를 공유기 IP주소로 변경.
- 공유기는 패킷 전달만 하는 것이 아니라 연결도 관리함. 연결된 패킷이 도착할 경우 연결된 NODE로 패킷을 전송 (여기서도 목적지 주소 바꿔치기)

부록 : 공유기

- 공유기 : Router
 - 이 무슨 삽질인가????
 - 전 세계적으로 LAN주소가 모자람
 - 어쩔 수 없이 사용하는 편법
 - 모든 노드의 IP주소 필요 => IP주소(공유기주소) 하나하나를 LAN주소로 사용
 - => IP+포트번호를 LAN주소로 사용 (포트 번호도 바꿔치기)
 - 하지만 이도 한계에 부딪침
 - 공유기 위에 공유기
 - 계속 변하는 외부 IP주소
 - 해결책 : **IPv6**

부록 : 공유기

- 공유기 : Router : 노드 세팅
 - 노드에서 공유기를 통해 패킷을 외부로 보내야 한다.
 - 패킷의 목적지가 같은 LAN에 없을 경우 일단 공유기로 보내야 한다.
 - 노드에서 패킷을 전달할 라우터로 공유기를 지정해야 한다.
 - DHCP에서 알아서 한다.
 - 수동으로 할 수도 있다.



```
C:\Users\wnhjung>ipconfig

Windows IP 구성

이더넷 어댑터 이더넷 4:

    연결별 DNS 접미사. . . . . : mshome.net
    링크-로컬 IPv6 주소 . . . . . : fe80::20b4:1262:d71e:b998%7
    IPv4 주소 . . . . . : 172.18.12.156
    서브넷 마스크 . . . . . : 255.255.240.0
    기본 게이트웨이 . . . . . : 172.18.0.1

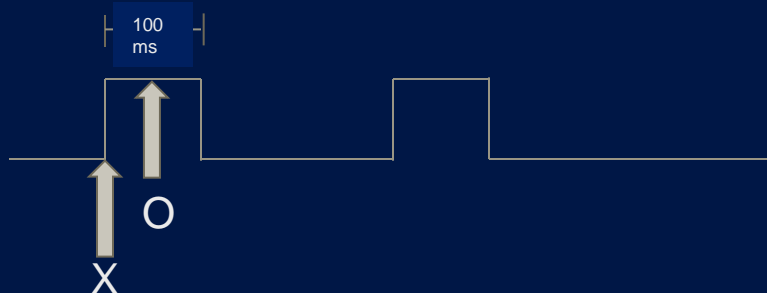
C:\Users\wnhjung>
```

부록 : 공유기

- 공유기 : 무선 AP
 - HUB의 연장. 유선 연결과 다를 것 없다.
 - 단지 연결 매체가 전파.
 - 보안 중요 (암호화) => 나중에

속제 1 설명

- 목표
 - 1 대 1, 단방향 채널 구현하기
- 아이디어 예시
 - 정확한 시간 간격에 맞춰 비트들을 전송한다.
 - 아주 정확하기는 어려우므로 적절한 간격을 사용한다.
 - 8비트 데이터만 전송하면 되므로 오류의 누적이 심하지 않을 것이다.

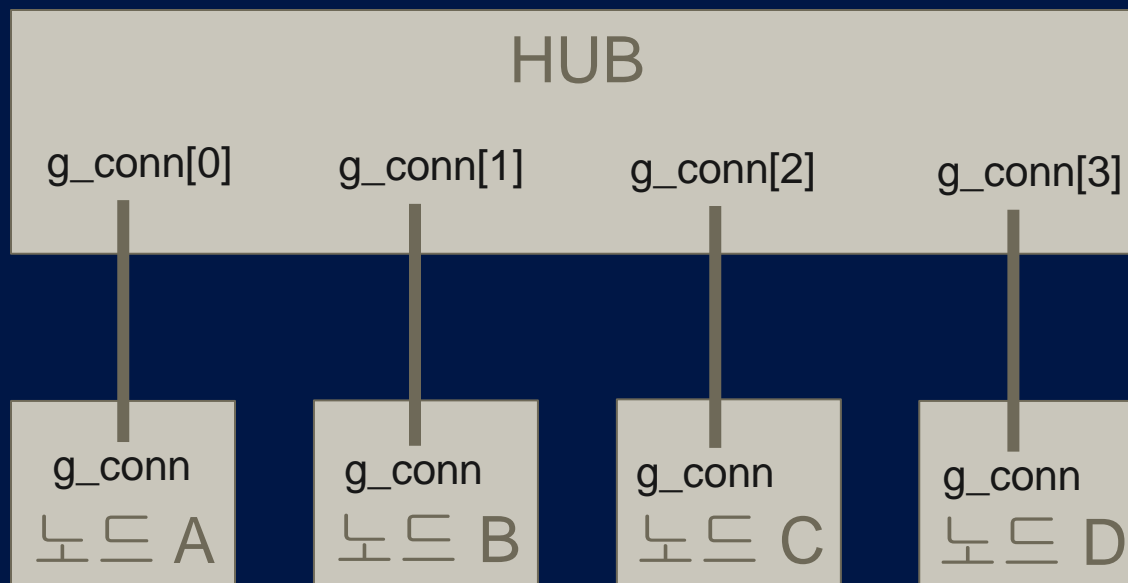


숙제 4 (1/10)

- SWITCHING HUB 구현하기
- 허브 (HUB)와 임의의 4개의 노드 (A, B, C, D) 사이의 통신 구현
 - 허브에는 4개의 NIC가 있고 각각 하나의 노드에 연결되어 있다.
 - 예) node A는 g_conn[0]와 연결
- 샘플 프로그램을 수정하여 임의의 방향의 통신을 구현하라.
 - 예) C -> B, D -> A
 - 허브와 노드 사이에는 충돌이 발생할 수 있다.

속제 4 (2/10)

- 연결 구조



속제 4 (3/10)

- 샘플 프로그램 : `node.cpp`
 - `do_node(char node_id)` : 노드의 동작을 컨트롤 하는 프로그램
 - `node_id` : 'A', 'B', 'C', 'D' 중의 하나로 노드를 구분한다.
 - `do_node_NIC(char node_id, CONN & g_conn)` : 노드의 NIC를 컨트롤 하는 프로그램
 - `g_conn`이 전역 변수가 아니라 매개변수로 들어온다..
 - 샘플에서의 동작
 - `do_node()` : 명령을 받아서 `do_node_NIC`에 전달한다.
 - `do_node_NIC()` : `g_conn`을 감시하면서 동시에 `do_node()`에서 전달된 명령을 수행한다.

속제 4 (4/10)

- 샘플 프로그램 : hub.cpp
 - do_hub() : hub의 동작을 컨트롤 하는 프로그램
 - 실제 구현에서는 NIC사이의 데이터 전송에 전념해야 한다.
 - do_hub_NIC0(CONN & g_conn) : hub의 0번 NIC를 컨트롤 하는 프로그램
 - 0번 부터 3번 까지 4개의 함수가 존재한다.
 - 컨트롤해야 하는 NIC를 파라미터로 넘겨 준다.
 - 샘플에서의 동작
 - do_hub() : 명령을 받아서 do_node_NICx에 전달한다.
 - do_hub_NIC0() : g_conn[0]를 감시하면서 동시에 do_hub()에서 전달된 명령을 수행한다.

속제 4 (5/10)

- 샘플 프로그램 동작

- HUB

- 모든 Port에 연결된 g_conn의 상태를 볼 수 있다.
 - 각각의 Port에 연결된 g_conn의 상태를 바꿀 수 있다.
 - 예) A1 => Port A를 True로, C0 => Port C를 False로
 - 각각의 Port에 연결된 g_conn의 상태가 바뀌면 알려 준다.

- NODE

- 연결된 g_conn의 상태를 볼 수 있다.
 - 연결된 g_conn의 상태를 바꿀 수 있다.
 - g_conn의 상태가 바뀌면 알려 준다.

숙제 4 (6/10)

- 필수 구현 기능
 - 메시지 전송
 - 문자열을 원하는 노드에 전송한다.
 - 노드가 연결되지 않았을 경우 충돌이 없으므로 전송되었다고 판단한다.
 - 충돌 처리
 - 데이터가 전송 되다가 충돌로 전송이 실패되었을 경우 다시 전송 (숙제 #3에서 구현)
 - 버퍼링
 - HUB의 경우 메시지 전송 중에 새로운 메시지가 도착할 수 있다.
 - 예) A에서 B로 메시지를 2번 전송 : 첫번째 메시지를 노드 B로 전송하고 있는데, A에서 두번째 메시지를 전송할 수 있다. => 저장해 놓았다가 전송
 - 예) A와 B가 거의 동시에 C에게 메시지를 전송하는 경우 => 나중 메시지를 저장했다가 전송
 - 동시전송
 - 예) A에서 B로 전송과 C에서 D로의 전송을 동시에 수행

숙제 4 (7/10)

- 힌트

- 메시지 전송 & 충돌 처리
 - HUB와 노드의 연결은 숙제 #3을 사용하면 됨
 - HUB는 받은 메시지를 해당 NIC에 전송하면 됨
 - do_hub()에서 수행, do_hub()이 전송만으로 바빠서 다른기능을 할 수 없다. (예:사용자 명령 수행 X)
- 재 전송 & 버퍼링
 - HUB는 보낼 메시지를 전송이 완료될 때 까지 Queue에 보관할 필요가 있음
- 동시 전송
 - NIC가 독립적을 실행되므로 자연스럽게 구현됨
 - 다른 NIC의 데이터를 건드리지 않아야 함.
 - Queue가 NIC마다 따로 존재해야 한다.

숙제 4 (8/10)

- 제출 내용
 - node.cpp, hub.cpp 파일
 - 구현한 방법에 대한 설명
 - 실행 스크린 샷
- 제출 방법
 - eclass에 제출

숙제 4 (9/10)

● 구현 예

```

C:\WINDOWS\system32\cmd.exe
Hub의 Port A가 활성화되었습니다.
Hub의 Port B가 활성화되었습니다.
Hub의 Port C가 활성화되었습니다.
Hub의 Port D가 활성화되었습니다.
허브이면 (H), 노드이면 연결된 허브의 포트 ID (A, B, C 또는 D)를 입력하십시오. : A
Hello World, I am node A.
Enter destination node with a message to send : DHello 3
You entered [DHello 3]
Enter destination node with a message to send : BHello 7
You entered [BHello 7]
Enter destination node with a message to send :

C:\depot\Projects\Lecture\BASIC_NET\HW04_SOLUTION\Debug\HW04_SOLUTION.exe
Hub의 Port A가 활성화되었습니다.
Hub의 Port B가 활성화되었습니다.
Hub의 Port C가 활성화되었습니다.
Hub의 Port D가 활성화되었습니다.
허브이면 (H), 노드이면 연결된 허브의 포트 ID (A, B, C 또는 D)를 입력하십시오. : B
Hello World, I am node B.
Enter destination node with a message to send : DHello 2
You entered [DHello 2]
Enter destination node with a message to send : DHello 4
You entered [DHello 4]
Enter destination node with a message to send :
NODE A sent [Hello 7]
Enter destination node with a message to send :

C:\WINDOWS\system32\cmd.exe
Hub의 Port A가 활성화되었습니다.
Hub의 Port B가 활성화되었습니다.
Hub의 Port C가 활성화되었습니다.
Hub의 Port D가 활성화되었습니다.
허브이면 (H), 노드이면 연결된 허브의 포트 ID (A, B, C 또는 D)를 입력하십시오. : C
Hello World, I am node C.
Enter destination node with a message to send : DHello 1
You entered [DHello 1]
Enter destination node with a message to send : DHello 5
You entered [DHello 5]
Enter destination node with a message to send :
NODE D sent [Hello 8]
Enter destination node with a message to send :

C:\WINDOWS\system32\cmd.exe
허브이면 (H), 노드이면 연결된 허브의 포트 ID (A, B, C 또는 D)를 입력하십시오. : D
Hello World, I am node D.
Enter destination node with a message to send :
NODE B sent [Hello 2]
Enter destination node with a message to send :
NODE C sent [Hello 1]
Enter destination node with a message to send :
NODE A sent [Hello 3]
Enter destination node with a message to send :
NODE B sent [Hello 4]
Enter destination node with a message to send :
NODE C sent [Hello 5]
Enter destination node with a message to send :
NODE A sent [Hello 7]
Enter destination node with a message to send :
NODE D sent [Hello 8]
Enter destination node with a message to send :
CHello 8
You entered [CHello 8]
Enter destination node with a message to send :
C:\depot\Projects\Lecture\BASIC_NET\HW04_SOLUTION\Debug\HW04_SOLUTION.exe
Hub의 Port A가 활성화되었습니다.
Hub의 Port B가 활성화되었습니다.
Hub의 Port C가 활성화되었습니다.
Hub의 Port D가 활성화되었습니다.
허브이면 (H), 노드이면 연결된 허브의 포트 ID (A, B, C 또는 D)를 입력하십시오. : H
Hello World, I the HUB.
NODE B sent [Hello 2] to NODE D
Send Message from Port B, to Port D : Hello 2
NODE C sent [Hello 1] to NODE D
Send Message from Port C, to Port D : Hello 1
NODE A sent [Hello 3] to NODE D
Send Message from Port A, to Port D : Hello 3
NODE B sent [Hello 4] to NODE D
NODE C sent [Hello 5] to NODE D
Send Message from Port B, to Port D : Hello 4
Send Message from Port C, to Port D : Hello 5
NODE D sent [Hello 8] to NODE C
Send Message from Port D, to Port C : Hello 8
NODE A sent [Hello 7] to NODE B
Send Message from Port A, to Port B : Hello 7
  
```

숙제 4 (10/10)

- 주의 사항

- 전송속도가 너무 빠르면 NIC함수에서 g_conn을 읽을 때 오류가 발생하기 쉬움
 - 특히 이번 과제는 NIC함수를 동시에 여러 개 수행시키므로 오류에 더 취약함.
- std::queue를 사용한다면 NIC별로 queue를 따로 두고 NIC에서만 접근하거나, 노드나 허브에서만 접근해야 함
 - 여러 개의 동시 실행객체에서 접근하면 Data Race 오류