

# 컴퓨터 네트워크

## 제2장 기본 개념

## 2.1 회선 구성

## 2.2 전송 기술의 종류와 특성

## 2.3 토폴로지

## 2.4 네트워크

## ❖ 기본 개념

---

통신 장치간의 관계에 대한 기본 사항

- ✓ 회선 구성(Line configuration)
- ✓ 접속형태(Topology)
- ✓ 전송 모드(Transmission mode)
- ✓ 네트워크 분류(Categories of Networks)
- ✓ 네트워크간 네트워크(Internetworks)

## 2.1 회선 구성 (1/16)

### 점대점(point-to-point) 방식

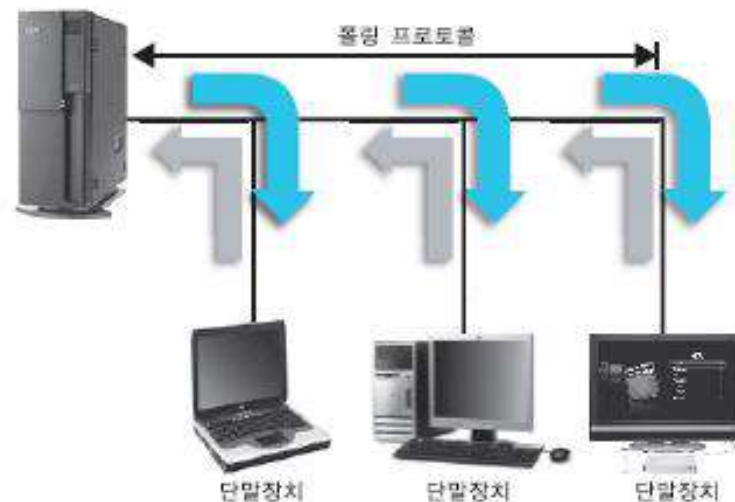
- 메인프레임 형태의 중앙의 컴퓨터와 여러 터미널들이 **독립적인 회선**을 이용하여 **1:1로 연결되는 방식**
- 비지능형(dumb) 터미널을 비동기식으로 **중앙 컴퓨터에 연결할 때 사용**
- TCP/IP 환경에서는 PPP를 사용하여 1:1로 연결



## 2.1 회선 구성 (2/16)

### 다중점(Multi-point) 방식

- 하나의 장치에 연결된 하나의 전용회선을 사용하여 다수개의 장치들을 연결하고 정보를 송수신하는 방식
- 멀티 드롭(Multi-drop)방식 이라고도 함
- 컴퓨터가 폴링하는 시스템에서만 사용 가능(단말에서 서버로 정보 전송)
- 컴퓨터가 방송하는 형태로 모든 터미널에 데이터 전송
- 터미널의 주소 판단 기능과 버퍼가 필요
- 장점 : 데이터 양이 적을 때 효과적, 회선 비용 절감
- 단점 : 회선 고장 시 고장지점 이후 단말장치 운용 불가



## 2.1 회선 구성 (3/16)

데이터 전송을 원하는 10개의 디바이스가 있고, 각 디바이스가 서로 데이터 전송을 하기 위해서는 디바이스 상호간의 연결(connection)이 필요

✓ 10개의 모든 디바이스를 각각 연결하려면? --> 45개의 점대점 (point-to-point) 연결이 필요.

✓ 계산 :  $(10 \times 9) / 2 = 45$ 개 --> 메시(mesh)형 연결

✓ 10개의 디바이스 사이의 거리가 수백 km 떨어져 있는 경우  
--> 45개의 점대점(point-to-point) 연결은 현실적이지 못함.

디바이스 수가 증가할수록 요구되는 연결의 수가 급격히 증가  
--> 비현실적인 방법

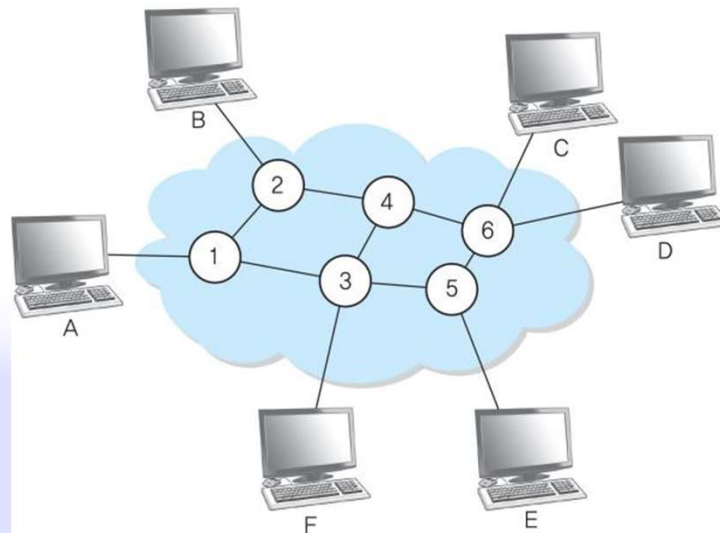
## 2.1 회선 구성 (4/16)

좋은 해결책은 없는가? **교환기술이 해결책 제공**

- ✓ 교환기술은 다수의 디바이스 상호간에 최적의 연결성을 제공
- ✓ 각 디바이스를 데이터통신 네트워크(network)에 연결하는 방법

교환기술과 디바이스간 상호 연결

- ✓ **스테이션**: 컴퓨터는 물론, 전화, 터미널 등 데이터 통신이 가능한 모든 디바이스를 포함
- ✓ 각 스테이션은 데이터 통신 네트워크를 구성하는 **노드(node)**에 연결
- ✓ 네트워크 노드: 라우터, 브릿지 등과 같은 **디바이스에 대한 총칭**으로, 전송측으로부터 목적지까지 데이터의 이동에만 관여.



## 2.1 회선 구성 (5/16)

### ❖ 데이터통신 네트워크의 분류 (전송기술과 구조에 따른)

- 교환 데이터 통신 네트워크(**switched data communication network**)
- 방송 데이터 통신 네트워크(**broadcast data communication network**)

#### 교환 네트워크의 구분

- ✓ 회선교환 네트워크
- ✓ 메시지 교환 네트워크
- ✓ 패킷교환 네트워크

#### ■ 방송 네트워크의 구분

- ✓ 패킷 라디오 네트워크
- ✓ 인공위성 네트워크
- ✓ 지역(local) 네트워크

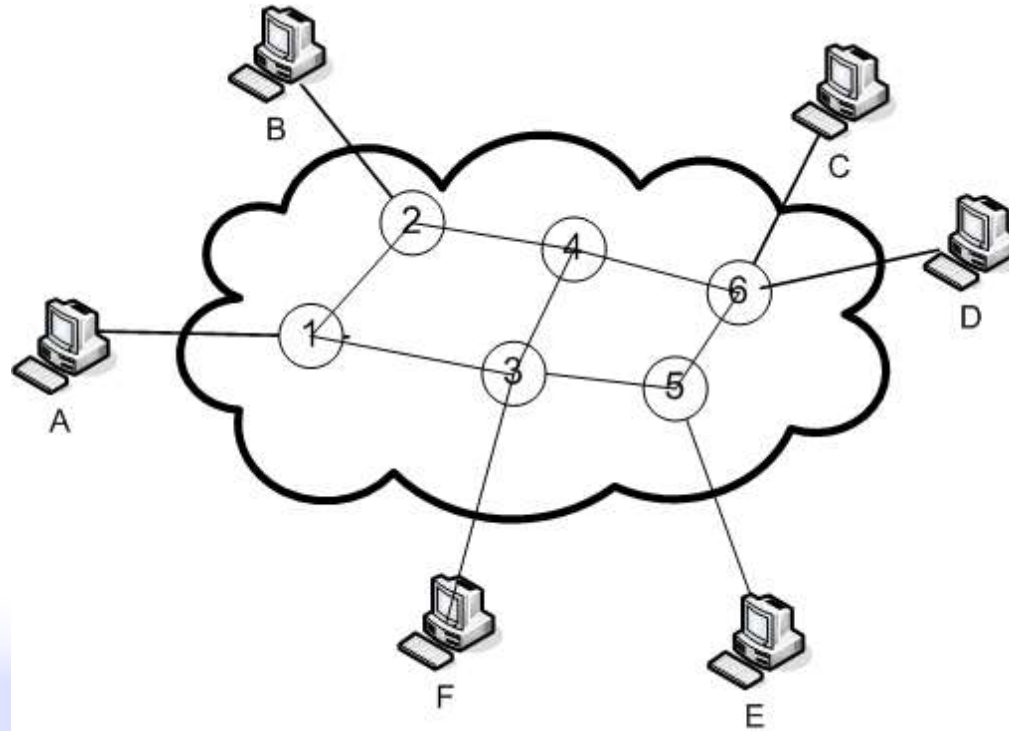




## 2.1 회선 구성 (6/16)교환 네트워크

### 교환 데이터통신 네트워크의 개념

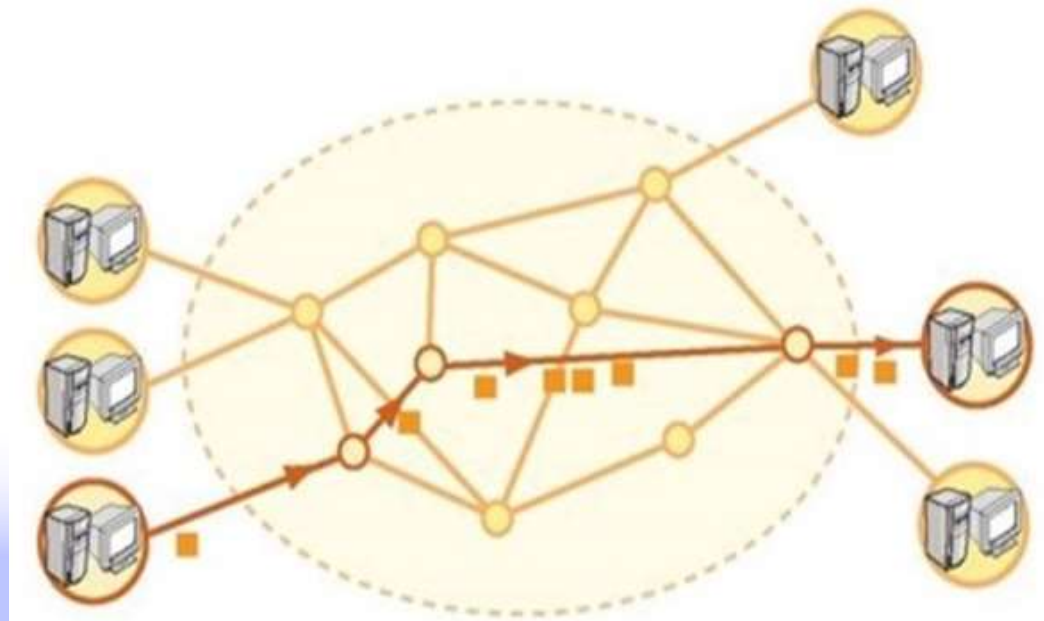
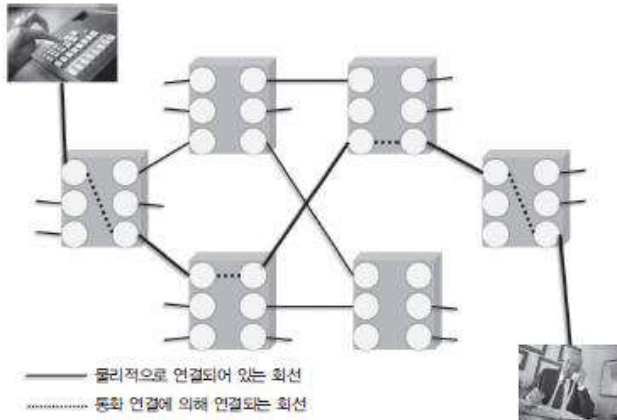
- ✓ 교환 네트워크는 6개의 노드로 구성되고 각 노드는 다수의 데이터링크로 연결
- ✓ 스테이션 A에서 스테이션 E로 데이터를 전송하는 경우
- ✓ 스테이션 A에서 출발한 데이터는 먼저 노드 1에 보내짐 → 노드 1에서는 몇가지의 데이터링크의 조합 가능



## 2.1 회선 구성 (7/16)

### 회선교환방식

- ✓ 대표적인 회선교환(circuit switching) 방식은 **전화 네트워크 시스템**
- ✓ 전송이 이루어지기 전에 먼저 데이터 통신을 위한 **전용 전송로 (dedicated transmission path)**를 설정.
- ✓ 정보 전송 시작할 때 물리적인 연결을 확립하고 **전송이 종료될 때까지 연결 유지**
- ✓ **물리적으로 연결된 회선은 다른 사람과 공유하지 못함**
- ✓ 음성 교환기의 교환방식
- ✓ **POTS(Plain Old Telephone System)**



## 2.1 회선 구성 (7/16)

### 회선교환방식

#### ✓ 특징

- 전송 중 항상 **동일한 경로**를 경유하여 데이터가 전송된다.
- 점대점 방식의 전송 구조를 갖는다.
- 상대적으로 긴 접속 시간을 필요로 하나 전송 지연은 거의 없다.
- 고정적인 대역폭을 사용한다.
- **속도나 코드의 변환이 불가능하다**
- 회선을 할당받아 사용하기 때문에 안정적이고 실시간 데이터 서비스가 가능.
- **효율성이 떨어짐**. (연결이 설정된 후에는 회선은 다른 사용자가 사용할 수 없음)
- **회선의 설정, 데이터의 이동, 회선의 단절** 등 3가지로 이루어짐

## 2.1 회선 구성 (8/16)

### 회선교환방식의 동작

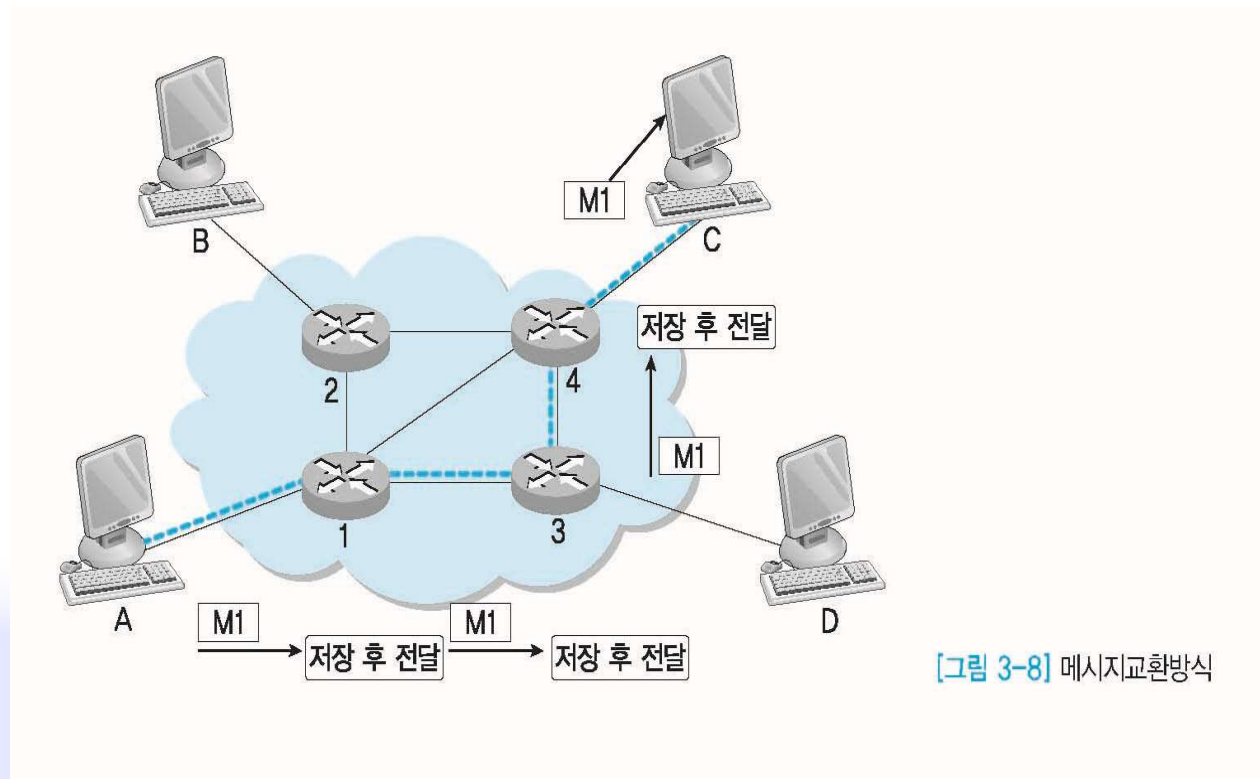
- ✓ 단계 1. **회선 설정(Circuit Establishment)** : 데이터가 전송되기 전에 스테이션 사이에 회선 설정
- ✓ 단계 2. **데이터 전송(Data Transfer)** : 단계 1에서 설정된 <노드 1-노드 3-노드 4-노드 6-노드 5>으로 구성된 전송로를 따라서 스테이션 A에서 스테이션 E로 데이터 전송이 이루어짐.
- ✓ 단계 3. **회선해제 (Circuit Termination)**: 데이터 전송이 완료되면 두 스테이션 중 하나의 스테이션의 동작으로 연결 해제.

회선교환방식은 전화(음성), 센서, 원격측정 등과 같은 비교적 연속적인 데이터 흐름을 갖는 경우에 널리 사용되는 기술

## 2.1 회선 구성 (10/16)

### 메시지 교환방식

- ✓ 전용 전송로의 설정이 불필요
- ✓ 메시지에 **목적지 주소**를 첨부하여 전송
- ✓ 메시지는 노드에서 노드로 네트워크를 통해 이동
- ✓ 각 노드에서 메시지를 수신하게 되면 수신된 메시지를 잠시 저장한 다음, 그 다음 노드로 보냄. ---> **축적 후 전달(store and forward)** 방식



## 2.1 회선 구성 (11/16)

### 메시지 교환방식의 특성

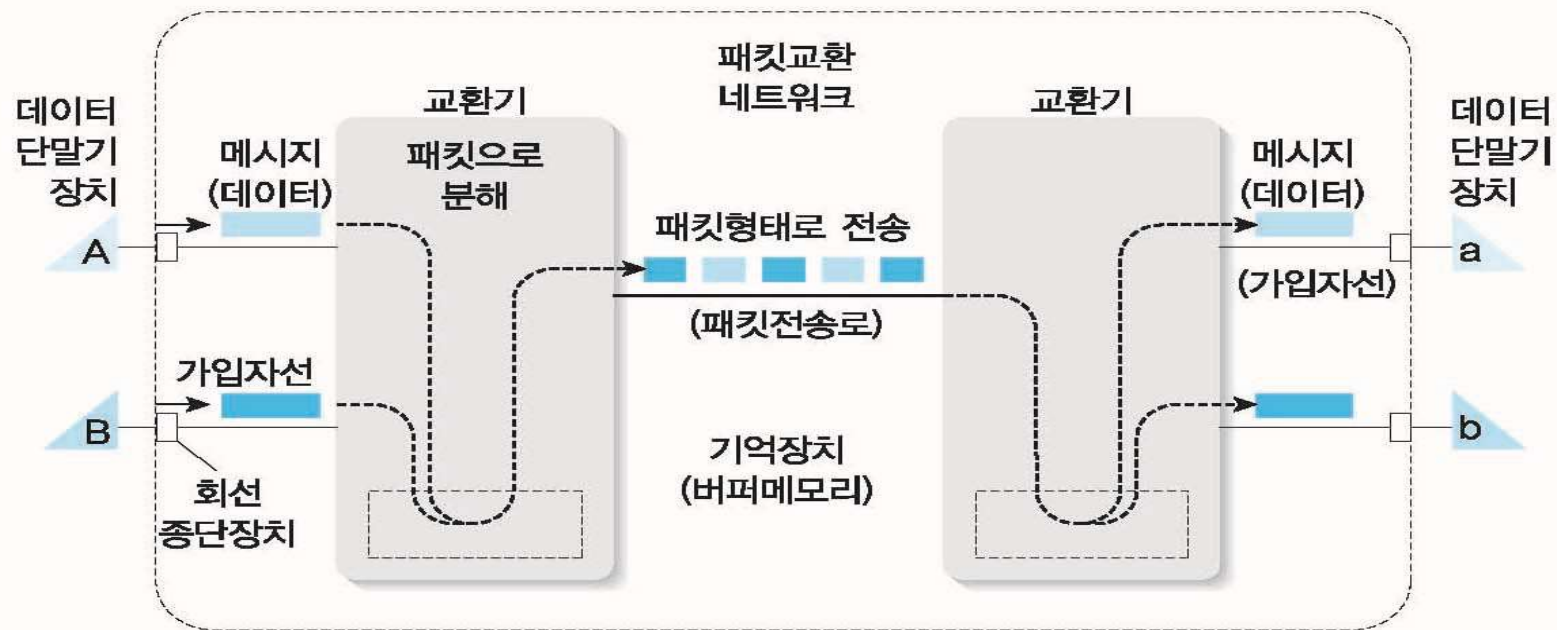
- ✓ 회선교환방식보다 적은 비용으로 네트워크 설계가 가능 (전송로의 효율적이용)
- ✓ 상대 스테이션이 다른 스테이션과 데이터 교환 중이거나 고장 등의 이유로 메시지를 수신할 수 없는 상태일 때에도 메시지 전송 가능
- ✓ 데이터 전송량이 폭주하는 경우에 교환네트워크의 혼란 상태를 피할 수 있음(축적기능)
- ✓ 동일한 내용의 메시지를 동시에 다수의 스테이션에 보낼 수 있음
- ✓ 필요에 따라 우선순위 전송이 가능
- ✓ 메시지의 분실을 방지하기 위해 번호를 부여하거나 전송 날짜, 시간 등을 추가하여 전송가능
- ✓ 코드와 속도가 서로 다른 단말기끼리도 메시지 교환이 가능

## 2.1 회선 구성 (12/16)

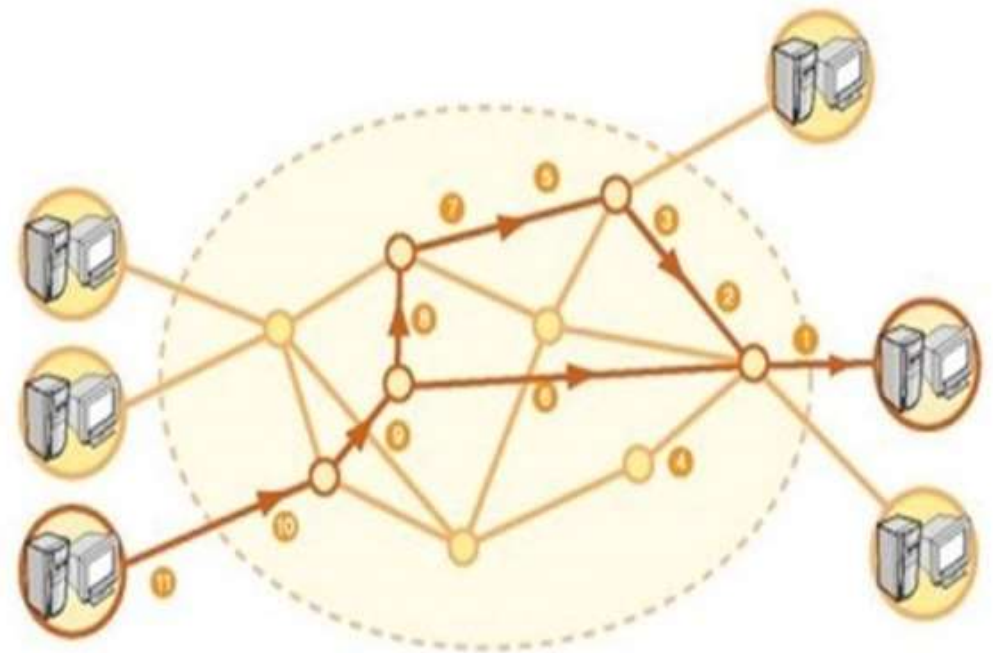
### 패킷교환(packet switching) 방식

- ✓ 메시지 교환방식과 회선교환방식의 장점을 결합하고 단점을 최소화
- ✓ 패킷 형태로 만들어진 데이터를 패킷교환기가 목적지 주소에 따라 적절한 경로(route)를 선택하여 전송하도록 하는 교환방식
- ✓ 노드가 교환기(switch)의 역할을 수행하는 점에서는 메시지교환의 경우와 비슷하나 메시지교환의 경우처럼 노드가 메시지를 축적하지는 않음
  - > 빠른 응답시간이 요구되는 응용에 사용 가능
  - 메시지 교환의 경우와 유사하게 속도와 코드가 서로 다른 디바이스 사이에서도 데이터 교환이 가능
  - 경로변경 방식에 따라 교환기 통신 회선 등의 장애가 발생할 경우에도 대체 경로를 선택할 수 있어 네트워크의 신뢰성 향상

## 2.1 회선 구성 (13/16)



[그림 3-10] 패킷교환방식의 개념도





## 2.1 회선 구성 (13/16)

### ✓ 패킷 교환 방식

- 패킷마다 주소를 삽입
- 노드들이 패킷을 통하여 대역폭을 공유하는 방식
- 패킷의 주소를 보고 최종 목적지까지 패킷을 전달
- 데이터 트래픽이 없을 때 낭비되는 대역폭을 효율적으로 이용
- 물리적인 전송로를 여러 노드가 공유
- 특징
  - 교환기 자체의 비용을 현저하게 낮출 수 있다.
  - 패킷교환기는 컴퓨터 그 자체이며 교환행위는 컴퓨터 메모리의 어떤 부분에 있는 데이터를 다른 메모리 위치로 옮기는 컴퓨터 명령어에 의해 수행되므로 패킷교환 방식은 소프트웨어에 의한 교환이라고 볼 수 있다. (메모리 기반 패킷 스위칭의 경우)

## 2.1 회선 구성 (14/16)

패킷교환방식의 구분 (패킷 스트림(stream)을 처리하는 방법에 따라)

- ✓ 데이터그램(datagram)
- ✓ 가상회선방식(virtual circuit)

데이터그램 패킷교환방식

- ✓ 패킷 스트림을 독립적으로 처리하는 방식으로 **연결 설정 단계가 불필요**하고, 혼잡을 피해 경로구성이 가능하기 때문에 **유통성이 개선**.

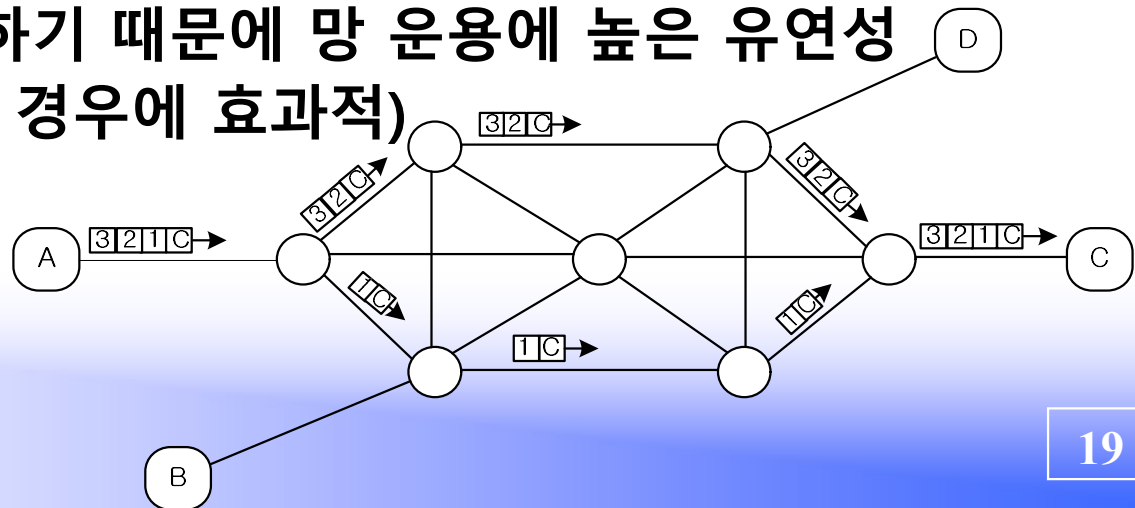
가상회선 패킷교환방식

- ✓ **논리적 연결설정(logical connection)**을 하는 방식으로 **에러제어, 흐름제어**가 가능하여 **신뢰성이 높아짐**.

## 2.1 회선 구성 (15/16)

### ➤ 데이터그램(Datagram) 방식

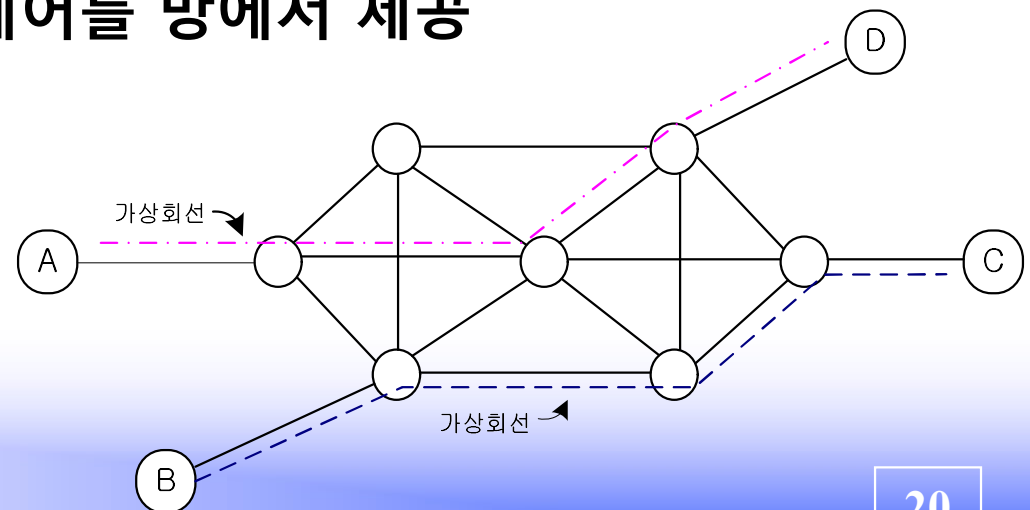
- 컴퓨터통신의 기본 단위
- 그 자체로 모든 것을 완비한 하나의 독립된 메시지
- 패킷마다 주소를 넣어 구성 - 패킷을 독립적으로 취급
- 송신지의 패킷 순서와 수신지의 패킷 순서가 다를 수 있음
- 패킷 손실시 송/수신지에서 복구 제어
- 장점
  - 호 설정 절차가 필요 없음
  - 적은양의 데이터를 전송하는 경우 효과적
  - 노드별로 전송을 하기 때문에 망 운용에 높은 유연성 제공 (오류 발생의 경우에 효과적)



## 2.1 회선 구성 (16/16)

### ➤ 가상회선(Virtual Circuit) 방식

- 전송 시작할 때 두 지점 사이에 **논리적 전송경로 설정**
- 송수신자 주소 대신에 **논리적 전송경로 번호**를 이용하여 스위칭
- **회선교환방식의 회선과 유사한** 기능제공 -> **가상회선**
- 각 패킷은 데이터정보 뿐만 아니라 **가상회선 식별자**를 포함
- 경로 설정과 관련된 결정을 할 필요 없음  
; **각 노드가 패킷에 대한 경로를 알고 있음**
- 장점
  - 패킷의 순서 및 오류 제어를 망에서 제공
  - 패킷을 신속하게 전송



# 교환방식의 특성 비교

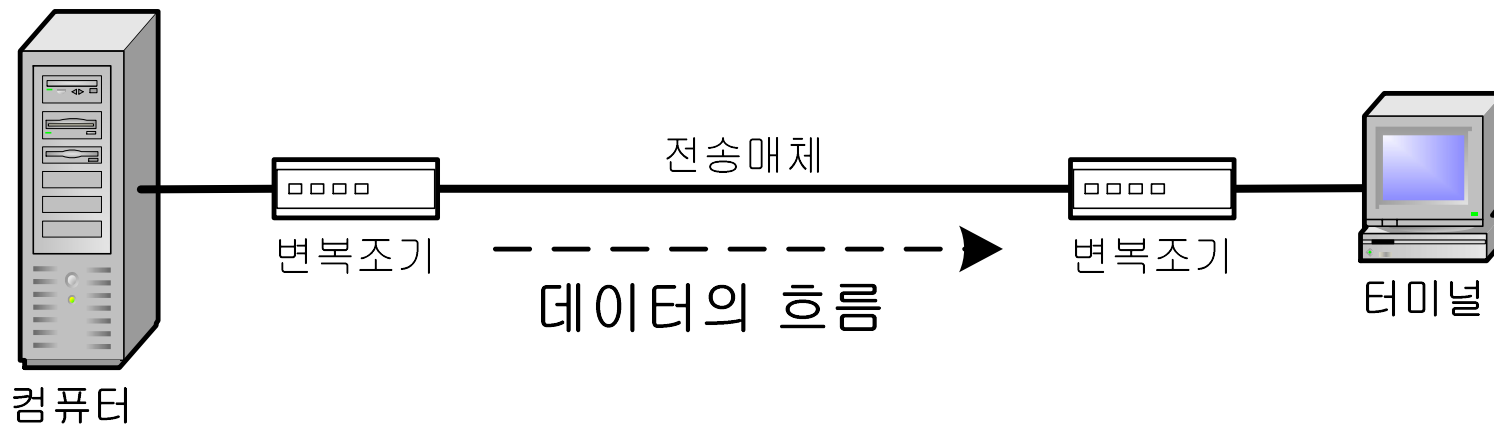
특성	방식	회선교환	메시지 교환	가상회선 패킷교환	데이터그램 패킷교환
전용전송로		유	무	무	무
전송단위		연속적인 데이터	메시지	패킷	패킷
메시지의 저장 여부		저장하지 않음	저장, 필요시 검색	일시적 저장, 검색기능 없음	일시적 저장, 검색기능 없음
이용에 적합한 전송 형태		길이가 긴 연속적 전송	저속 메시지 전송	순간적인 대량 데이터의 고속전송	순간적인 대량 데이터의 고속전송
전송 경로의 형태		동일한 전송경로	메시지마다 경로설정	전체 패킷전송을 위해 경로설정	각 패킷마다 경로설정
지연 시간 영향		연결호출 설정지연, 전송지연은 무시	메시지 전송지연	연결호출 설정지연, 패킷 전송지연	패킷전송지연
과부하시		연결호출 설정중단	메시지 전송지연 증가	연결호출 설정중단; 연결 설정 후에는 패킷전송지연 증가	패킷전송지연 증가
코드 및 속도변환		무	유	유	유
전송데이터와 수신 데이터의 순서일치		일치	불일치	일치	불일치
대역폭		고정	동적사용 가능	동적사용 가능	동적사용 가능
회선 에러발생시		다른 회선 재설정	여러 경로 중 선택	다른 회선 재설정	여러 경로 중 선택
오버헤드		연결설정 후 불필요	메시지마다 필요	각 패킷마다 필요	각 패킷마다 필요
응용분야		실시간 대화형	실시간 대화형 부적합	실시간 대화형	실시간 대화형

## 2.2 전송 기술의 종류와 특성 (1/10)

### 단방향과 양방향 전송

#### ✓ 단방향(simplex) 전송 방식

- 데이터 전송로에서 **한 방향**으로만 데이터가 흐르는 전송 방식
- 원격 측정기(telemeter), **라디오**, **TV** 방송 등
- 데이터는 컴퓨터측에서 제어를 받는 장비측으로 전송



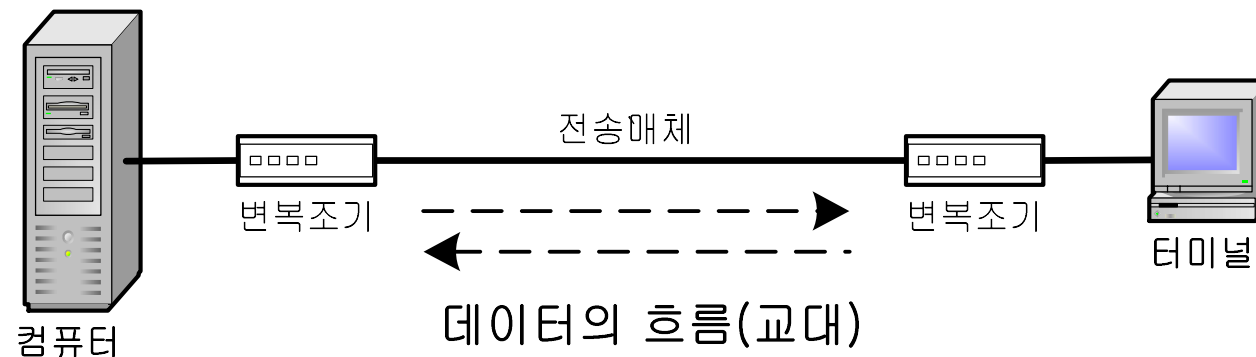
#### ✓ 양방향(duplex) 전송 방식

- 방향의 전환에 의해 데이터의 흐르는 방향을 바꾸어 전송 가능
- 송수신측이 미리 결정되어 있지 않음

## 2.2 전송 기술의 종류와 특성 (2/10)

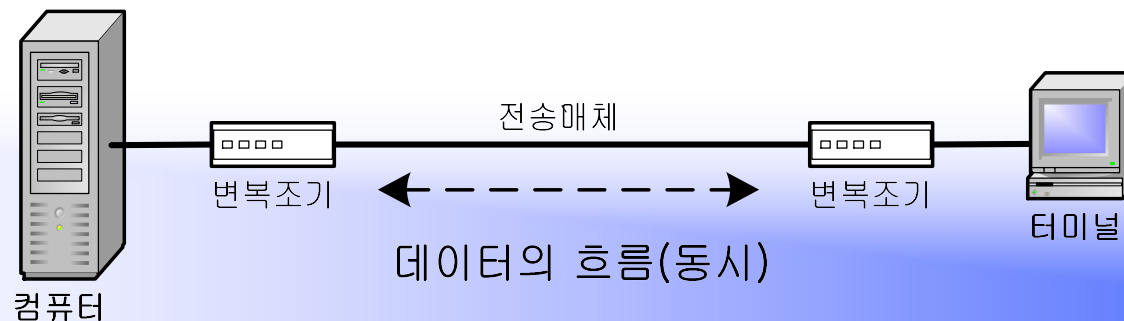
### ➤ 반이중(half duplex) 전송 방식

- 두 장치 간에 교대로 데이터를 교환 (e.g., 무전기)
- 한 순간에는 반드시 **한쪽 방향으로만 전송**



### ➤ 전이중(full duplex) 전송 방식

- 두 장치 간에 동시에 **양방향으로 데이터를 교환**
- 전송 회선의 사용 효율이 높음
- 회선비용이 많이 소요



## 2.2 전송 기술의 종류와 특성 (3/10)

### 아날로그 및 디지털 전송

- 아날로그 데이터 : 연속적으로 변화하는 물리량의 변화값으로부터 획득되는 데이터  
예) 온도, 압력, 전압 등
- 디지털 데이터 : 불연속적인 값을 가지며 임의의 최소값의 정수배를 다루는 데이터  
예) 문자열, 숫자 등
- ✓ **아날로그 전송 방식**
  - 아날로그 신호를 수단으로 전송
  - 아날로그 신호는 음성이나 변조된 디지털 데이터
  - 전송거리 증가에 따른 신호 감쇠 [減衰, attenuation] 현상을 막기 위하여 **증폭기(Amplifier)** 사용



## 2.2 전송 기술의 종류와 특성 (3/10)

### ✓ 디지털 전송 방식

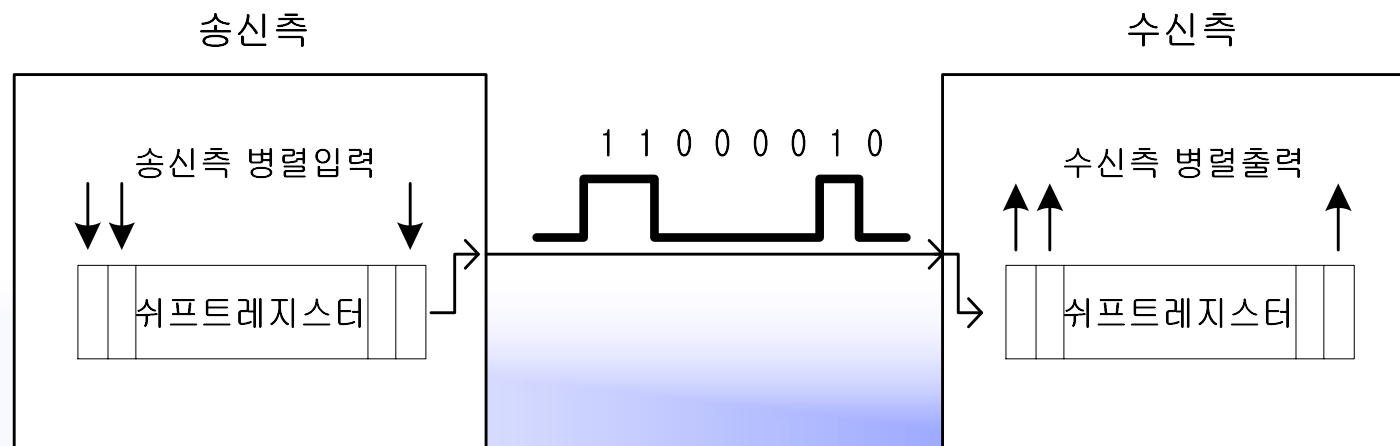
- 디지털 신호를 전송하는 수단
- 제한된 거리에서의 감쇠현상은 없으나 전송거리의 제한을 극복하기 위해서 **리피터(Repeater)** 사용
- **디지털 전송의 장점**
  - 재생이 용이
  - 잡음에 강함
  - 높은 신뢰성 확보
  - 저장이나 조작이 용이
  - IC 사용이 가능
  - 높은 보안성

## 2.2 전송 기술의 종류와 특성 (4/10)

### 직렬 및 병렬 전송

#### ✓ 직렬 전송 방식

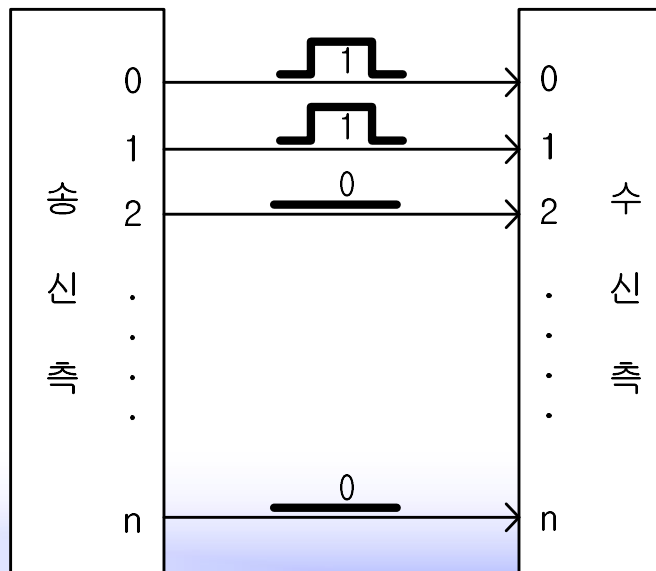
- 한번에 **한 비트씩 순서대로** 데이터 전송
- 쉬프트 레지스터(Shift Register) 사용
  - 직렬 신호 ↔ 병렬 신호
- 문자나 비트들을 구별 할 수 있는 방법 필요
- **원거리 전송에 적합하며, 대부분의 데이터 통신에 사용**
- **구성 비용이 적게 듬**



## 2.2 전송 기술의 종류와 특성 (5/10)

### ✓ 병렬 전송 방식

- 여러 개의 bit 를 그룹으로 한번에 전송
- 패리티 또는 제어비트 전송을 위해 추가적인 전송로 필요
- 컴퓨터와 주변기기 사이의 데이터 전송  
예) 컴퓨터와 프린터 연결
- 전송 속도가 빠르지만, 구성 비용이 많이 듦
- 거리가 멀수록 전송비용이 증가



$n = 8, 16, 32, 64$

## 2.2 전송 기술의 종류와 특성 (6/10)

### 비동기 및 동기 전송

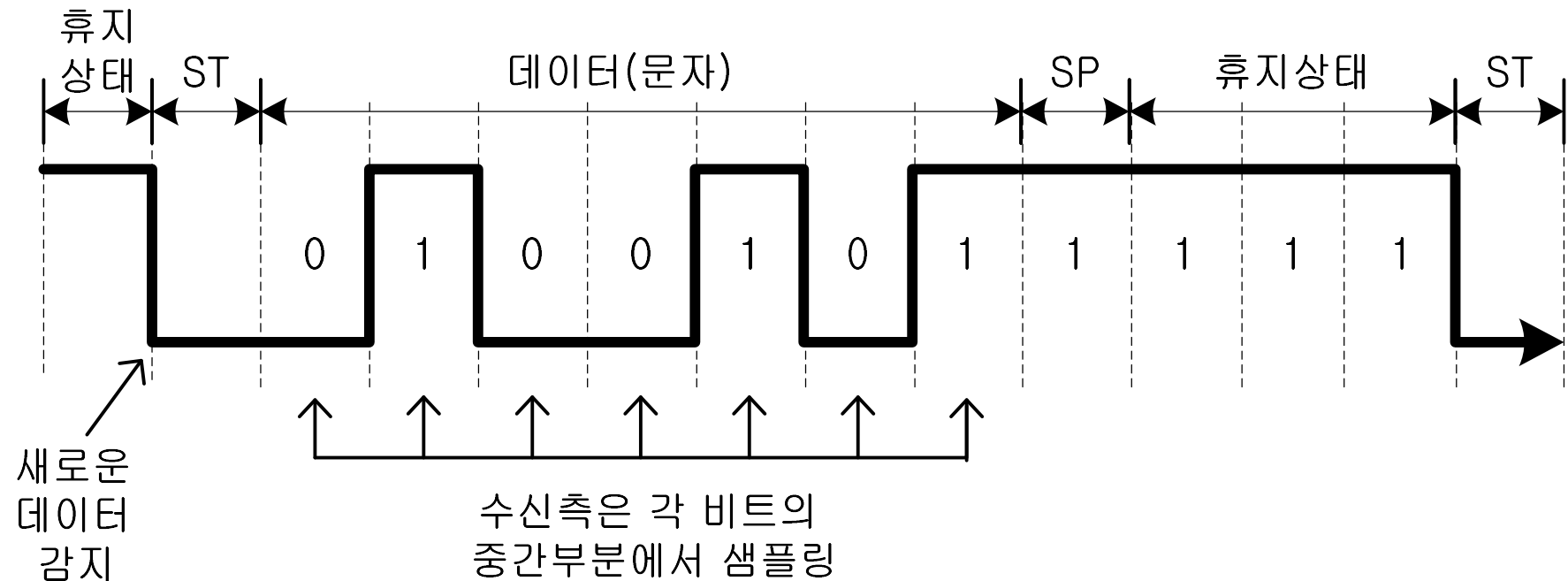
#### ✓ 비동기식 전송 방식 (Asynchronous Transmission)

- 데이터는 짧은 비트열로 나뉘어 전송, 각 전송 비트열 내부에서 동기화 유지
- 비트열 전후에 시작 비트(ST: Start bit)와 정지 비트(SP: Stop bit)를 추가
- 전송할 데이터가 있을 경우, 휴지상태(1상태)의 선로에 시작비트(0상태)를 전송하여 선로를 0상태로 전환
- 정해진 비트 수 만큼 전송후, 정지 비트를 확인하고 종료
- 최근에는 고속 전송에도 사용
- 시작 비트와 정지 비트로 인한 회선 이용효율 저하

## 2.2 전송 기술의 종류와 특성 (6/10)

### 비동기 및 동기 전송

#### ✓ 비동기식 전송 방식 (Asynchronous Transmission)



## 2.2 전송 기술의 종류와 특성 (8/10)

- ✓ 동기식 전송 방식(Synchronous Transmission)
  - 문자 또는 비트들의 데이터 블록 단위로 송수신
  - 데이터 블록의 전후에 프리앰블 (preamble), 포스트앰블 (postamble)의 제어정보 삽입
  - 데이터와 제어정보를 합쳐서 프레임(frame) 이라고 함
  - 전송 효율 및 전송속도가 높음
  
- 비트 전송방식
- 문자 전송방식

## 2.2 전송 기술의 종류와 특성 (8/10)

### ✓ 동기식 전송 방식(Synchronous Transmission)

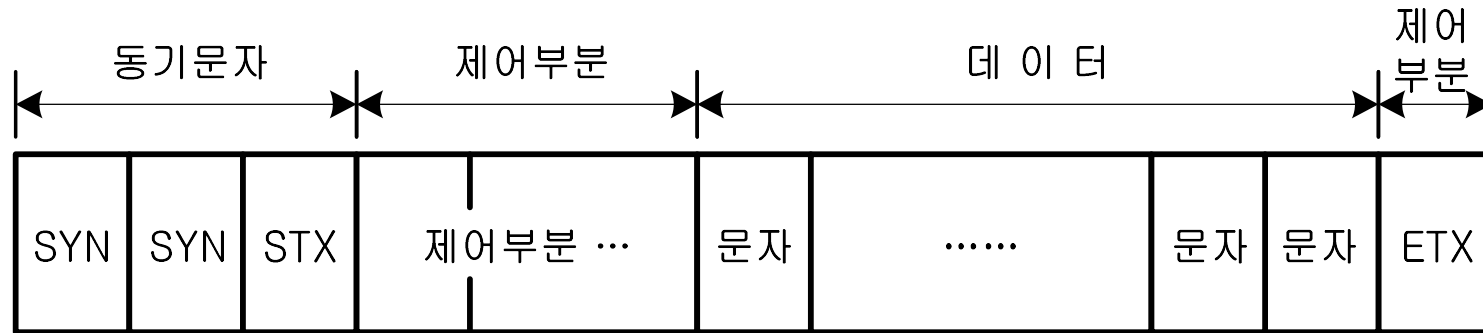
#### ➤ 비트 전송방식

- 데이터 블록을 **플래그**를 사용하여 구분
- 플래그 : 데이터 블록의 전후에 추가되어 **블록의 시작과 끝**을 나타내는 특별한 비트

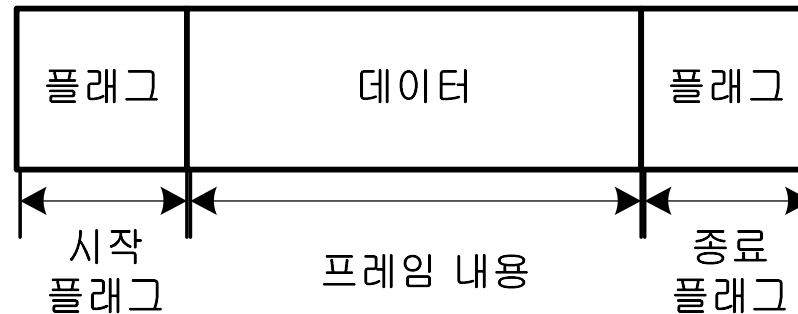
#### ➤ 문자 전송방식

- **특정문자를 이용**하여 동기화 수행, 전송 데이터도 문자 단위로 취급
- 프레임은 동기화 문자를 포함  
예) SYN : 블록의 시작, ETX : 블록의 마지막
- HDLC(High-level Data Link Control)

## 2.2 전송 기술의 종류와 특성 (9/10)



(a) 문자 전송 방식



(b) 비트 전송 방식



# ASCII 코드표

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b>@</b>	96	60	140	&#96;	<b>`</b>
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>	97	61	141	&#97;	<b>a</b>
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>	98	62	142	&#98;	<b>b</b>
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>	99	63	143	&#99;	<b>c</b>
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>\$</b>	68	44	104	&#68;	<b>D</b>	100	64	144	&#100;	<b>d</b>
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>	101	65	145	&#101;	<b>e</b>
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>	102	66	146	&#102;	<b>f</b>
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	<b>'</b>	71	47	107	&#71;	<b>G</b>	103	67	147	&#103;	<b>g</b>
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>	104	68	150	&#104;	<b>h</b>
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>	105	69	151	&#105;	<b>i</b>
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>	106	6A	152	&#106;	<b>j</b>
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>	107	6B	153	&#107;	<b>k</b>
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>	108	6C	154	&#108;	<b>l</b>
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>	109	6D	155	&#109;	<b>m</b>
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>	110	6E	156	&#110;	<b>n</b>
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>	111	6F	157	&#111;	<b>o</b>
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>	112	70	160	&#112;	<b>p</b>
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>	113	71	161	&#113;	<b>q</b>
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>	114	72	162	&#114;	<b>r</b>
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>	115	73	163	&#115;	<b>s</b>
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>	116	74	164	&#116;	<b>t</b>
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>	117	75	165	&#117;	<b>u</b>
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>	118	76	166	&#118;	<b>v</b>
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>	119	77	167	&#119;	<b>w</b>
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>	120	78	170	&#120;	<b>x</b>
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>	121	79	171	&#121;	<b>y</b>
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>	122	7A	172	&#122;	<b>z</b>
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>:</b>	91	5B	133	&#91;	<b>[</b>	123	7B	173	&#123;	<b>{</b>
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>	124	7C	174	&#124;	<b> </b>
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>	125	7D	175	&#125;	<b>}</b>
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>	126	7E	176	&#126;	<b>~</b>
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>	127	7F	177	&#127;	<b>DEL</b>

## 2.2 전송 기술의 종류와 특성

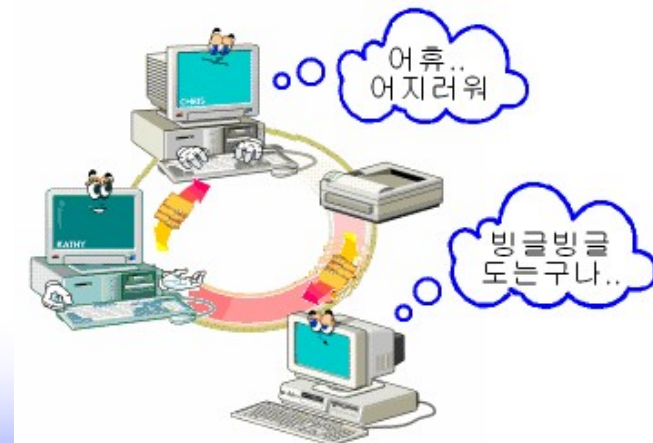
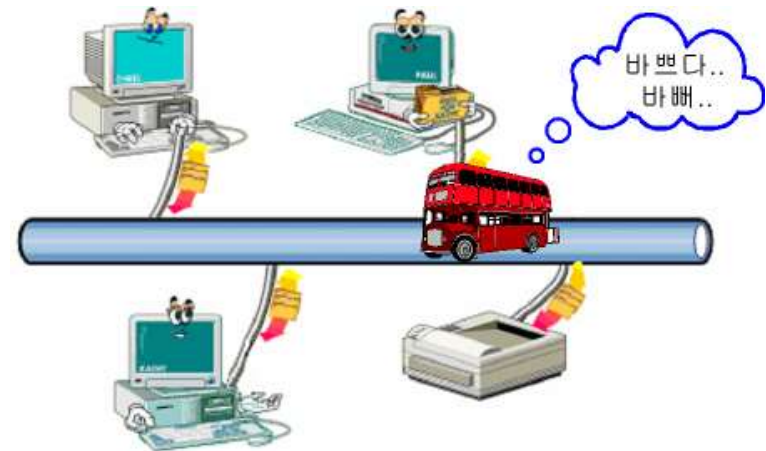
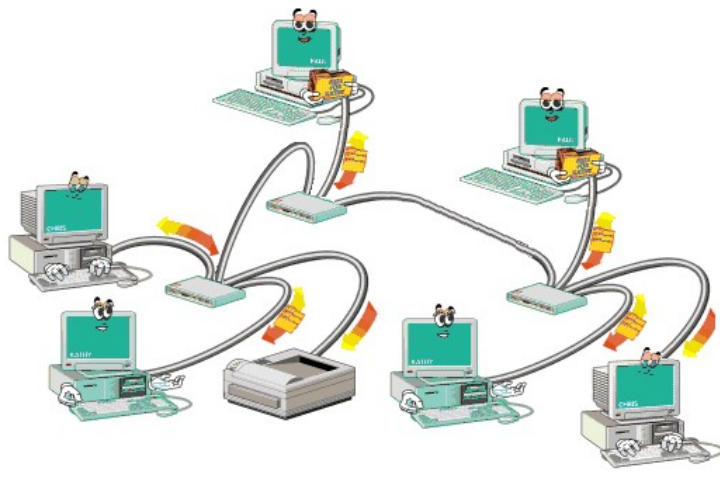
(10/10)

구 분	내 용
비동기식 전송	<ul style="list-style-type: none"> <li>- 전송되는 각 문자는 앞쪽에 1개의 시작비트, 뒤쪽에 1~2개의 정지비트를 갖는다.</li> <li>- 각 글자 사이에는 일정치 않은 시간의 휴지기간이 있을 수 있다.</li> <li>- 글자를 구성하는 각 비트의 길이는 통신 속도에 따라 정해지며 일정하다.</li> <li>- 동기는 글자단위로 이루어지며 송신측과 수신측이 항상 동기 상태에 있을 필요는 없다.</li> </ul>
동기식 전송	<ul style="list-style-type: none"> <li>- 데이터의 앞쪽에 반드시 동기문자가 온다.</li> <li>- 동기문자는 송신측과 수신측이 동기를 이루도록 하는 목적으로 사용된다.</li> <li>- 한 묶음으로 구성하는 글자들 사이에는 휴지간격이 없다.</li> <li>- 타이밍신호는 변복조기, 터미널 등에 의해 공급된다.</li> <li>- 터미널은 반드시 버퍼를 갖고 있어야 한다.</li> </ul>

## 2.3 토폴로지 (1/6)

**토폴로지(Topology)** : 네트워크 상의 컴퓨터의 위치나 컴퓨터 간의 케이블 연결 등의 물리적인 배치

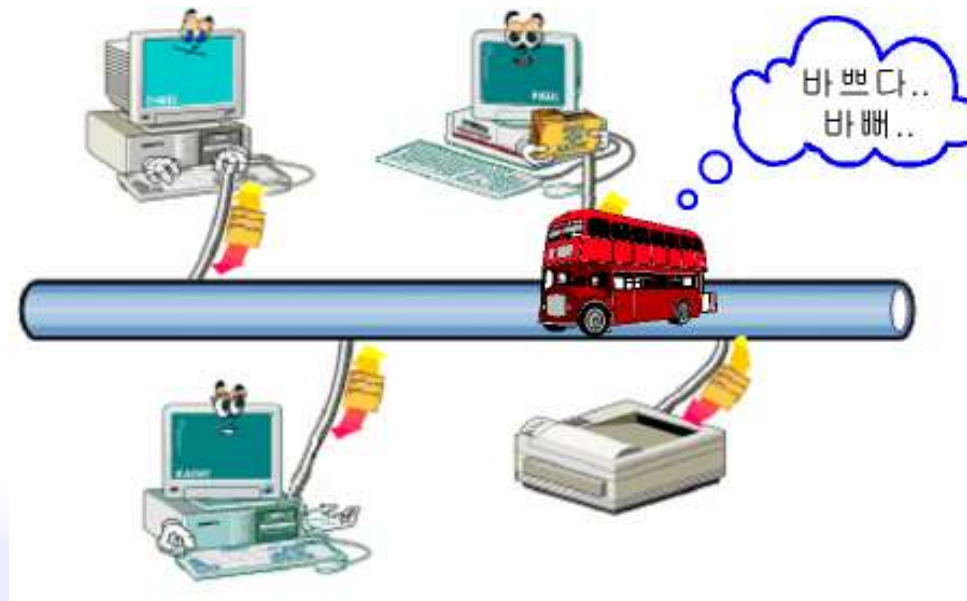
접속형태에 따른 분류 : **버스형, 성형, 원형, 계층형, 그물형(Mesh)** 등



## 2.3 토폴로지 (1/6)

### 버스(Bus) 방식

- ✓ 버스라 불리는 **공통배선**을 모든 노드가 공유
  - ✓ 근거리 통신망(LAN)의 일반적 방식
  - ✓ 케이블링에 소요되는 비용의 최소화
  - ✓ 특정 노드의 상태에 따라 네트워크 형태가 변하지 않음
- ∴ **브로드캐스팅 방식**



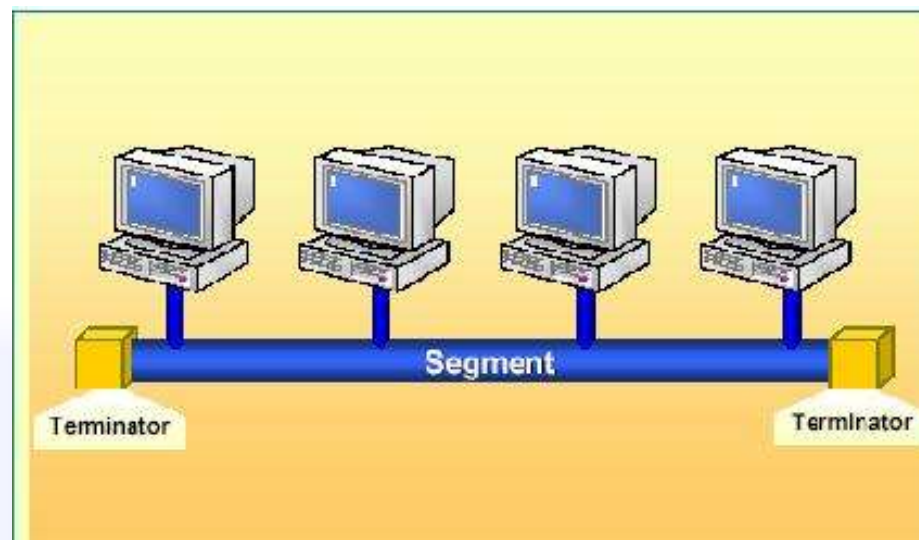
## 2.3 토폴로지 (2/6)

### ✓ 장점

- 네트워크 구성이 간단, 작은 네트워크에 유용, 사용이 용이
- 관리가 용이하고 새로운 노드의 추가가 용이.

### ✓ 단점

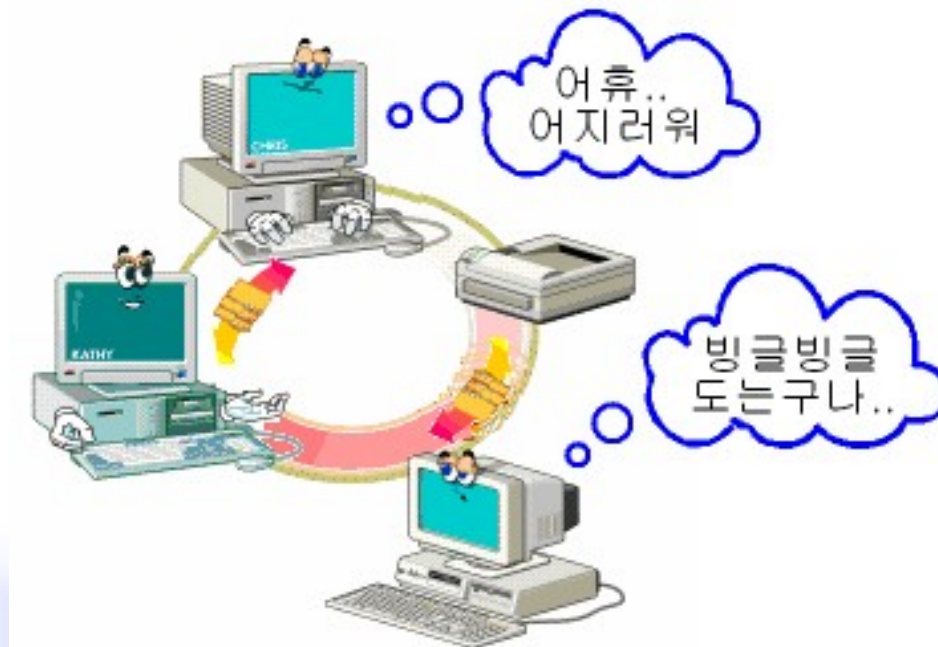
- 통신 채널이 단 한 개이므로 고장 시 네트워크 전체가 동작을 하지 않으므로 잉여 채널이 필요.
- 네트워크 트래픽이 많을 경우 네트워크 효율이 떨어짐.
- 브로트캐스트등으로 인한 잦은 컴퓨터 인터럽트로 호스트의 성능을 떨어트리고 네트워크 대역폭을 낭비할 수 있음.



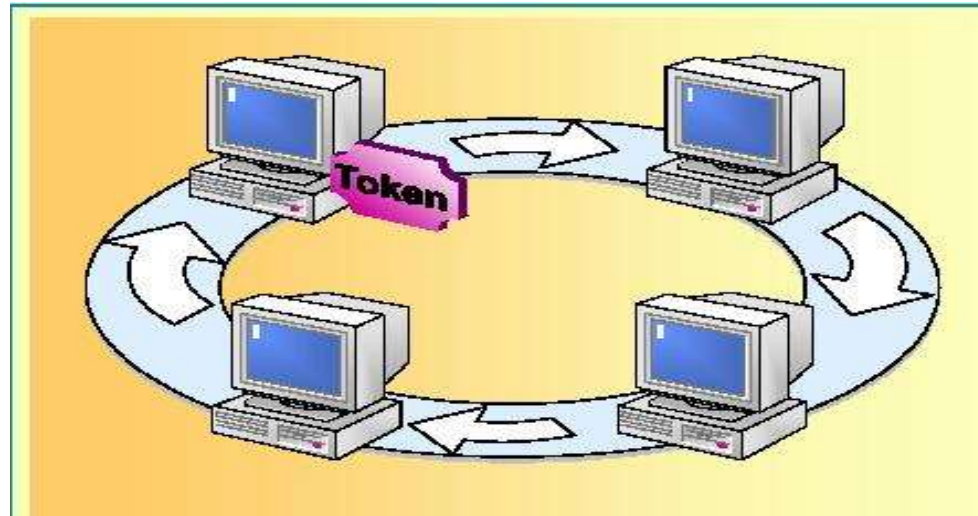
## 2.3 토폴로지 (2/6)

### 링(Ring) 방식

- ✓ 데이터의 흐름이 **한 방향**
- ✓ 수신된 데이터가 자신의 것이면 네트워크에서 삭제
- ✓ 그렇지 않은 경우 인접 노드로 데이터를 중계
- ✓ 만일 다른 호스트가 수신을 하지 못한 경우, 송신자는 해당 데이터를 제거해야 함



## 2.3 토폴로지 (3/6)



### ✓ 장점

- 병목 현상이 드물다.
- 분산 제어와 검사, 회복 등이 쉽다.

### ✓ 단점

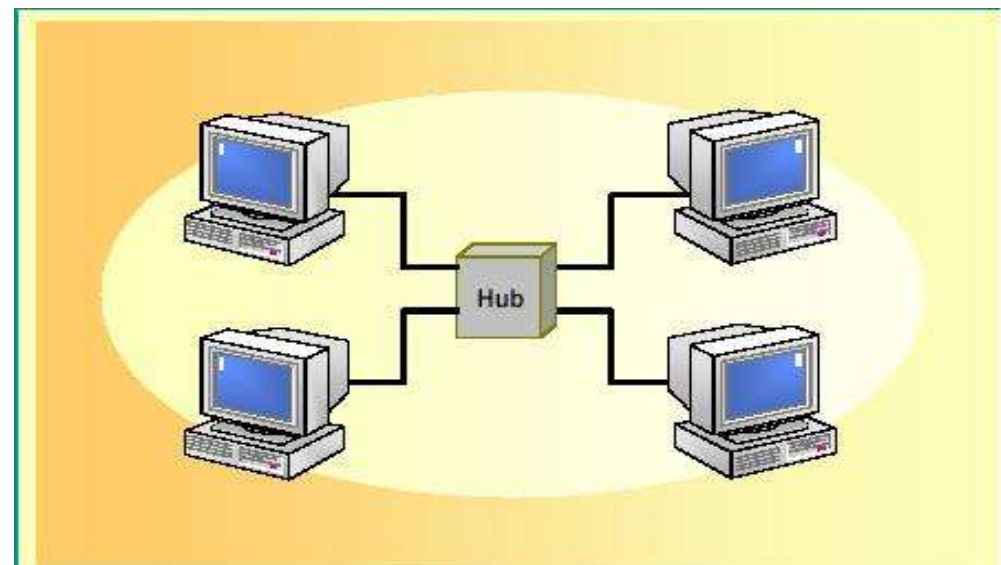
- 새로운 네트워크에 대한 확장이나 구조의 변경이 비교적 어렵다.
- 네트워크상의 어떤 노드라도 문제가 발생하면 네트워크 전체가 통신 불능상태에 빠질 수 있다.
- 다중 링 형태로 구성하는 것이 일반적.



## 2.3 토폴로지 (4/6)

### 성형(Star) 방식

- ✓ 중앙 제어 노드가 통신상의 제어에 대한 권한과 책임
- ✓ 분산 처리 능력이 제한



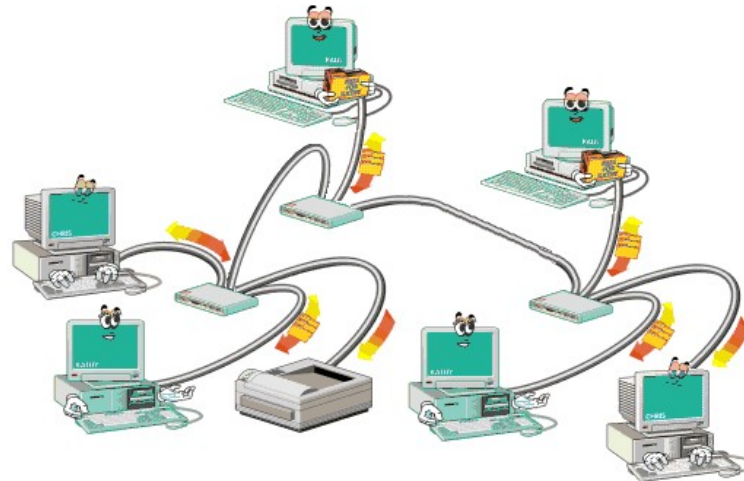
- ✓ 장점
  - 고장의 발견과 수리가 쉽고, 노드의 증설, 이전이 쉽다.
- ✓ 단점
  - 잠재적 병목성을 가지며 중앙 지역 고장에 취약.
  - 중앙 제어 노드에 문제가 발생하면 네트워크 전체가 통신 불능 상태에 빠지게 됨.



## 2.3 토폴로지 (5/6)

### 트리(Tree) 방식

- ✓ 다수의 버스 방식을 허브(스위치)를 이용하여 **트리처럼 연결**
- ✓ 제어와 오류 해결을 각각의 허브에서 수행
- ✓ 허브로 구성되는 경우와 스위치로 구성되는 경우 다르게 동작

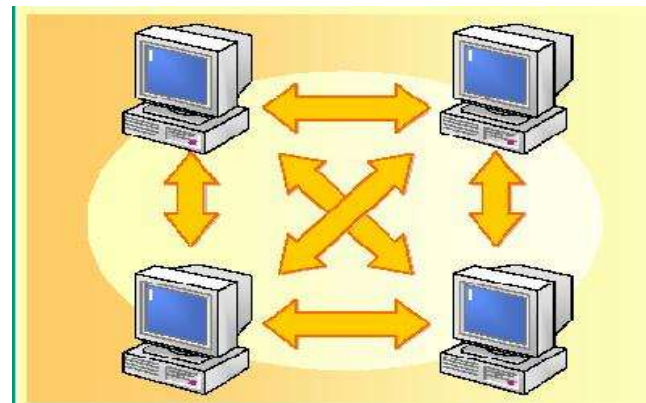
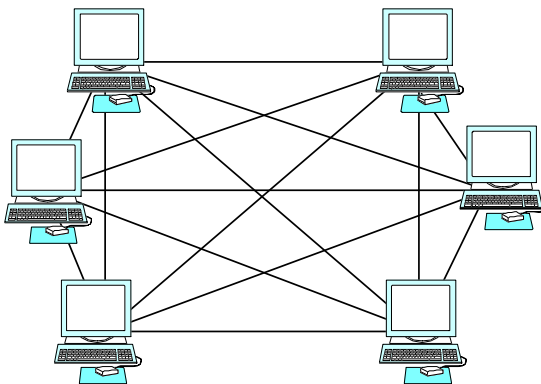


- ✓ 장점
  - 제어가 간단하여 **관리 및 확장이 용이.**
- ✓ 단점
  - **중앙 지점에서 병목 현상**이 발생할 수 있음.
  - 중앙 지점의 고장 발생시 대체 방법이 없을 경우 네트워크가 마비 또는 분할될 수 있음.

## 2.3 토폴로지 (6/6)

### 그물형(Mesh) 방식

- ✓ 중앙의 제어 노드에 의한 중계 대신에 각 노드간 점대점 방식으로 직접 연결
- ✓ 완전 그물형(full mesh)과 부분 그물형(partial mesh)으로 나뉨
- ✓ 장애발생시 대체경로로 전달 가능
- ✓ 링형과 더불어 네트워크 백본을 구성하는 방식



- ✓ 장점
  - 고장의 발견이 쉬움.
  - 한 노드의 고장 시 네트워크의 다른 트래픽에 미치는 영향을 최소화할 수 있음
- ✓ 단점
  - 선로 구축 비용이 많이 듦.
  - 선로 설치 및 설정 과정이 상대적으로 오래 걸리고 어려움.

## 2.4 네트워크 (1/15)

### 네트워크 정의

- ✓ 네트워크 : 통신 회선에 의해서 서로 연결되어 있는 노드와 링크의 집합

### 네트워크 구성 요소

- ✓ 네트워크 케이블

- 노드 간을 연결시키는 매개체
- 동축케이블, 트위스티드 페어, 광섬유, 무선 등

- ✓ 네트워크 인터페이스 카드(NIC: Network Interface Card)

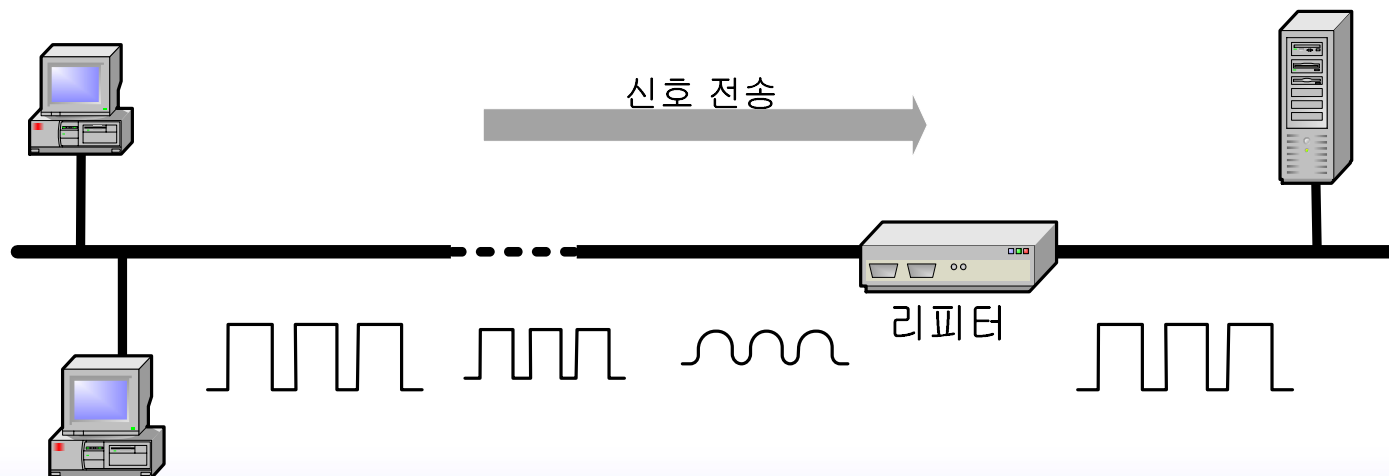
- 네트워크 전송매체와 노드 간을 연결시키는 인터페이스
- 전송매체 제어방식에 따라 이더넷(Ethernet), 토큰링(Token Ring) 등 여러가지 형태가 존재
  - 컨트롤칩 : 데이터의 입출력과 기본적인 기능을 관장
  - 통신전용칩 : 통신기능을 관장
  - 버퍼 : 데이터를 저장
  - MAC주소 (media access control address) ; 48 비트

## 2.4 네트워크 (3/15)

### ◆ 네트워크 장비

#### ➤ 리피터(Repeater)

- 전송거리에 따른 **신호감쇠를 보상하기 위해** 신호를 수신, 증폭하여 매체의 다음 구간으로 재전송 시키는 장치
- 근거리통신망 내에서 세그먼트들을 서로 연결  
; 신호를 먼거리까지 연장 가능
- OSI 계층의 물리 계층에서 동작



## 2.4 네트워크 (2/15)

### ➤ 허브(Hub); 멀티포트 리피트

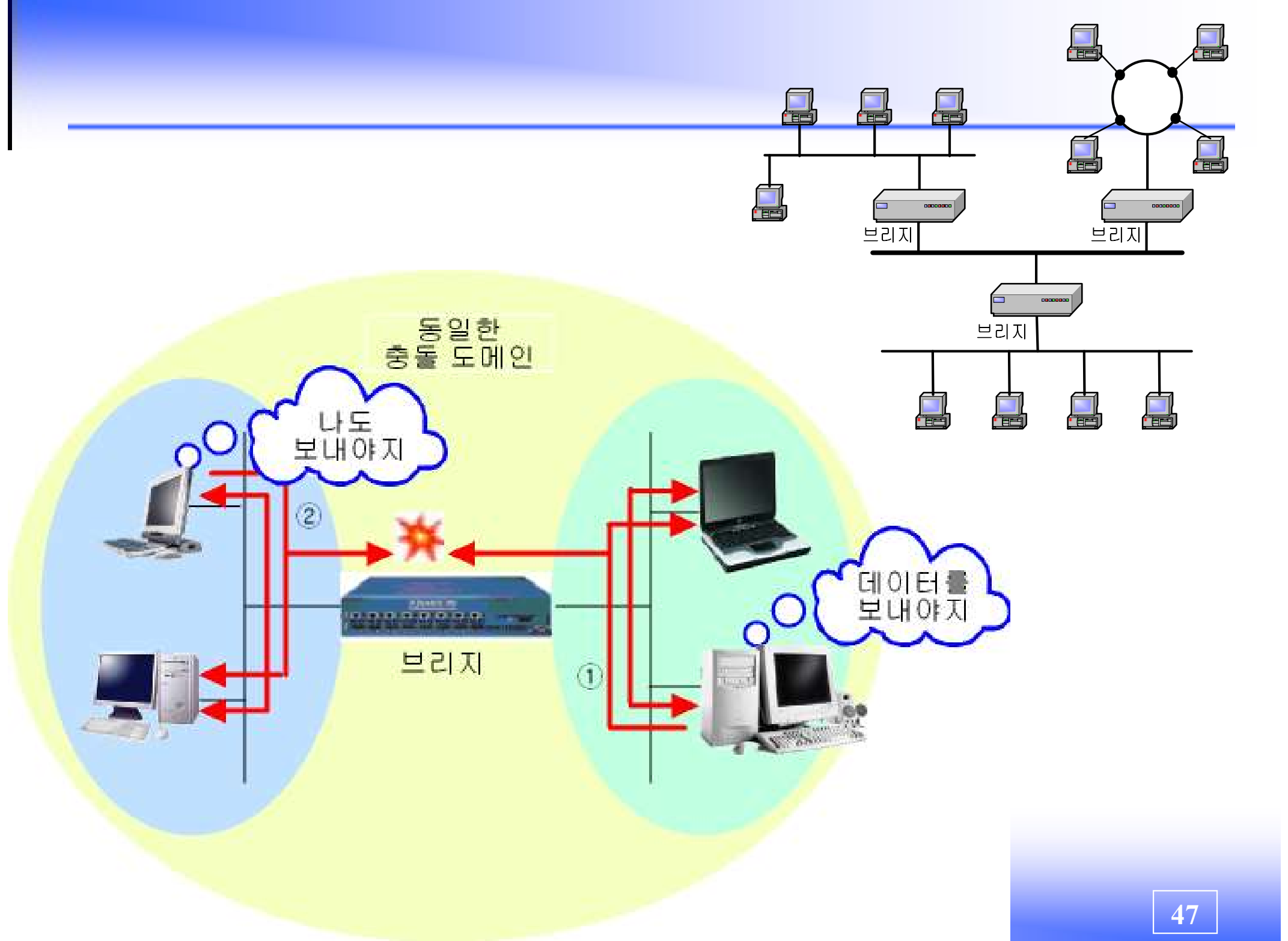
- 집중화장비(concentrator)라고 부르기도 함
- 연결된 장치들은 네트워크 공유
- OSI 계층의 물리 계층에서 동작
- 하나의 버스에 접속된 것처럼 동작
  - ✓ Dummy Hub
  - ✓ Switching Hub
  - ✓ Stackable Hub
  - ✓ MAC 주소(Media Access Control Address)



## 2.4 네트워크 (4/15)

### ➤ 브리지(Bridge)

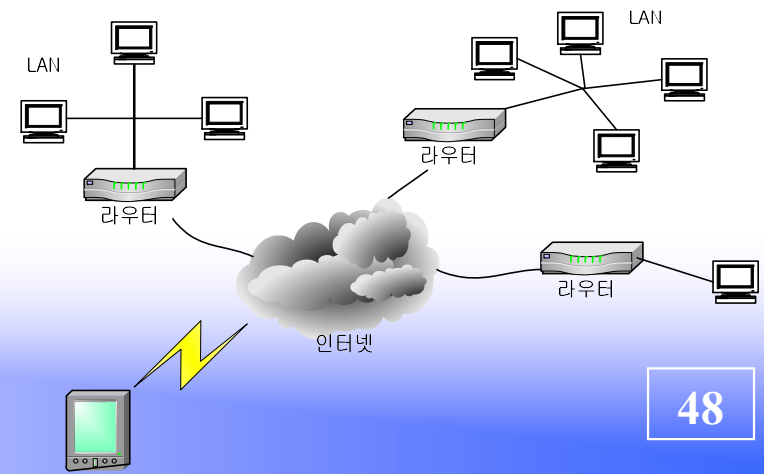
- 매체를 공유하는 근거리 통신망에서 하나의 장비가 데이터를 보내고 있을 때 또 다른 장비가 데이터를 보내면 충돌이 발생.(이 영역을 **충돌 도메인(Collision Domain)**)
- 네트워크에 장비들의 수가 늘어나면, 즉 충돌 도메인이 커지면 충돌이 발생할 확률도 높아지게 되고 **통신 속도와 효율이 저하**되게 됨
- 따라서 네트워크를 확장하기 위해 **충돌 도메인을 나누어 줄 수 있는 장비가 필요**
- 브리지는 데이터 링크 계층에서 동작하는 장비로 데이터 링크에서 사용하는 MAC(Media Access Control)이라는 네트워크 장비에 고정되어 있는 유일무이한 주소, 즉 하드웨어 주소를 기반으로 전송할 포트를 결정
- 매체 접근 제어(MAC) 방식이 같거나 다른 LAN간의 상호 접속  
예) 이더넷과 토큰링 네트워크를 연결
- **동일한 통신 프로토콜을 사용하는 두 네트워크 세그먼트 사이에서 패킷을 연결하고 전달하는 장치**
- OSI 참조 모델의 2계층인 데이터 링크 계층에서 작동



## 2.4 네트워크 (6/15)

### ➤ 라우터(Router)

- 동일한 네트워크 프로토콜을 사용하는 네트워크 세그먼트들을 연결하는 장비
- 네트워크 주소(IP 주소)를 기반으로 목적지까지의 경로 선택
- 라우팅 테이블에 따라 효율적인 경로를 선택하여 패킷 전송
- 흐름제어 및 서브 네트워크 구성 관리 기능
- 라우터는 네트워크 세그먼트를 연결하는 기능
- 특히 브로드캐스트 패킷을 차단하는 기능을 제공.
- 세그먼트에서 발생한 브로드캐스트 패킷은 다른 세그먼트로 전달되지 않는다.
- 네트워크 계층에서 동작
- 다른 LAN간 연결
  - 프로토콜 구조가 다른 네트워크 간의 연결





## 2.4 네트워크 (8/15)

### ➤ 게이트웨이(Gateway)

- 컴퓨터 네트워크에서 서로다른 통신망,프로토콜을 사용하는 네트워크간의 통신을 가능하게 하는 컴퓨터,소프트웨어를 통칭
- 넓은 의미로는 종류가 다른 네트워크 간의 통로의 역할을 하는 장치
- 게이트웨이는 OSI 참조모델의 전계층을 인식하여 전송방식이 다른 통신망도 흡수하여, 서로 다른 기종간의 접속도 가능
- 2개 이상의 다른 종류 혹은 같은 종류의 네트워크를 상호 접속
- 라우터와 혼용하여 사용
- 다른 네트워크로의 입구를 나타내는 네트워크 장비
- 프로토콜 구조가 다른 네트워크 연결 (e.g., PSTN과 데이터 네트워크)  
; 프로토콜 변환기능으로 네트워크 내에서 병목 현상을 발생하기도 함
- 프록시 서버(Proxy Server)나 방화벽(Firewall)의 기능을 수행하기도 함
- OSI 계층의 모든 계층에 걸쳐 동작

## 2.4 네트워크 (9/15)

- ✓ 네트워크 운영체제(NOS: Network Operating System)
  - 네트워크를 관리하고 제어하는 시스템 소프트웨어
  - 예) Window NT와 유닉스(리눅스)
  - 특징
    - 하나 이상의 업체가 만든 H/W 환경에서 동작할 수 있다.
    - 하나 이상의 같지 않은 H/W LAN을 같은 NOS하에서 연결 가능하다.
    - 네트워크 보안 기능과 사용자의 파일 접근 권한을 관리한다.
    - 다수의 서버를 지원하며 사용자가 접속한 서버의 종류와 무관할 수 있는 투명성(transparency)있는 환경을 제공한다.
    - 다중 사용자 환경에서 프로그램 및 파일에 대한 보안 기능을 제공한다.

## 2.4 네트워크 (10/15)

### 네트워크 구성 방식

- ✓ **피어투피어(Peer-to-Peer) 방식**
  - 네트워크에 연결된 각각의 노드가 동등하게 클라이언트 혹은 서버로 동작
  - 동등한 수평적 관계
  
- ✓ **클라이언트 서버(Client/Server) 방식**
  - 클라이언트 : 서비스 요구자, 서버 : 서비스 제공자
  - 대개 서버는 공유를 위한 자료(Database)를 가지고 있어서 클라이언트의 요청 시 자료를 전송
  - 대표적인 예로는 월드와이드웹.
    - 웹 서버가 서버 역할
    - 사용자가 쓰는 웹 브라우저가 클라이언트 프로그램

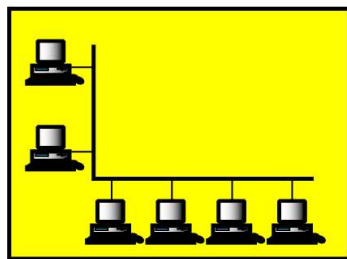
## 2.4 네트워크 (11/15)

Peer-to-Peer 방식		Client-Server 방식	
장 점	단 점	장 점	단 점
<ul style="list-style-type: none"> <li>* 서버쪽의 H/W나 S/W 에 대한 특별한 투자 가 필요 없다</li> <li>* 설치가 용이하다.</li> <li>* 네트워크 관리자를 필요로 하지 않는다.</li> <li>* 작업의 수행에 있어서 다른 컴퓨터에 대한 의존이 덜하다.</li> <li>* 비용이 저렴하다.</li> </ul>	<ul style="list-style-type: none"> <li>* 자원공유를 위한 컴퓨터의 추가적인 부하가 있다.</li> <li>* 많은 컴퓨터의 접속 및 제어가 불가능하다.</li> <li>* 데이터 보관에 대한 중앙 매체가 없다.</li> <li>* 보안에 취약하며 일관성이 없다.</li> <li>* 중앙 관리가 불가능하다.</li> </ul>	<ul style="list-style-type: none"> <li>* H/W 및 S/W를 서버에서 공통으로 사용할 수 있어서 비용이 절감된다.</li> <li>* 중앙 집중식 보안으로 보안이 강력하다.</li> <li>* 중앙에서 데이터에 대한 보관을 담당한다.</li> <li>* 장비의 공유가 가능하다.</li> <li>* 하나의 네트워크와 계정으로 도메인 내의 자원을 사용할 수 있다.</li> <li>* 많은 수의 사용자를 관리할 수 있다.</li> <li>* 사용자의 자원공유 작업을 없애준다.</li> </ul>	<ul style="list-style-type: none"> <li>* 고가의 전용 H/W, S/W가 필요하다.</li> <li>* 특정한 중앙 네트워크 관리자가 필요하다.</li> </ul>

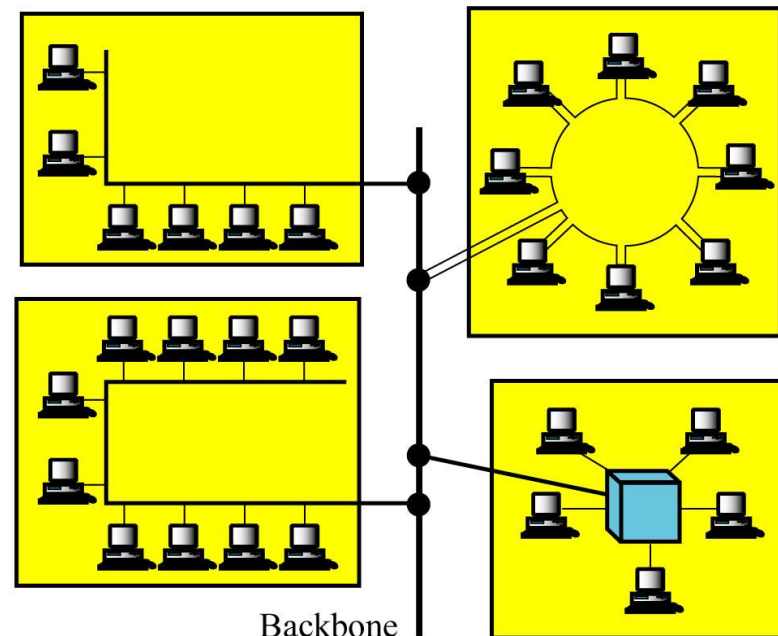
## 2.4 네트워크 (12/15)

### 근거리통신망(LAN: Local Area Network)

- ✓ 좁은 지역(약 50km) 내의 통신회선으로 연결된 PC, 메인프레임, 워크스테이션 등의 네트워크 집합
- ✓ 근거리통신망 내의 정보기기, 소프트웨어, DB등을 공유
- ✓ 통신 속도는 보통 10~100Mbps이며, 1Gbps와 10Gbps급 으로 바뀌어가는 추세



a. Single building LAN

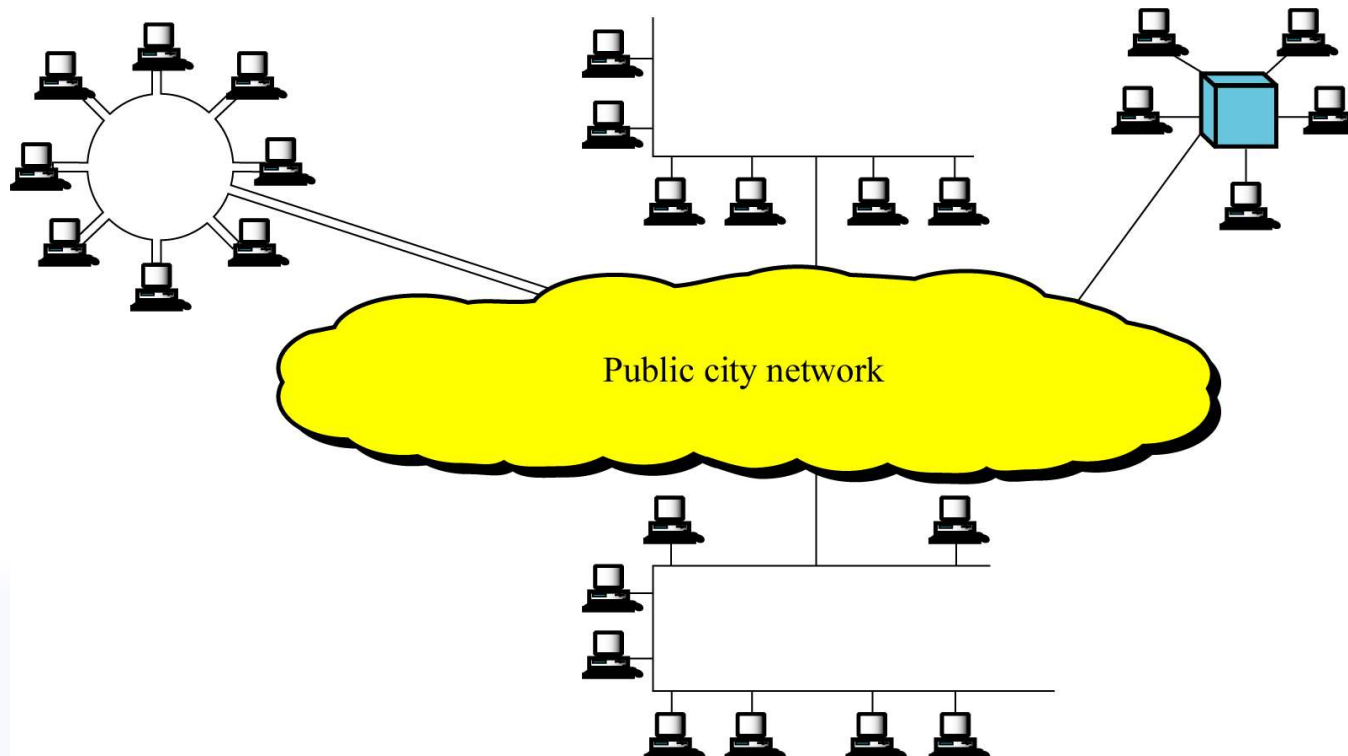


b. Multiple building LAN

## 2.4 네트워크 (13/15)

### 도시권통신망(MAN: Metropolitan Area Network)

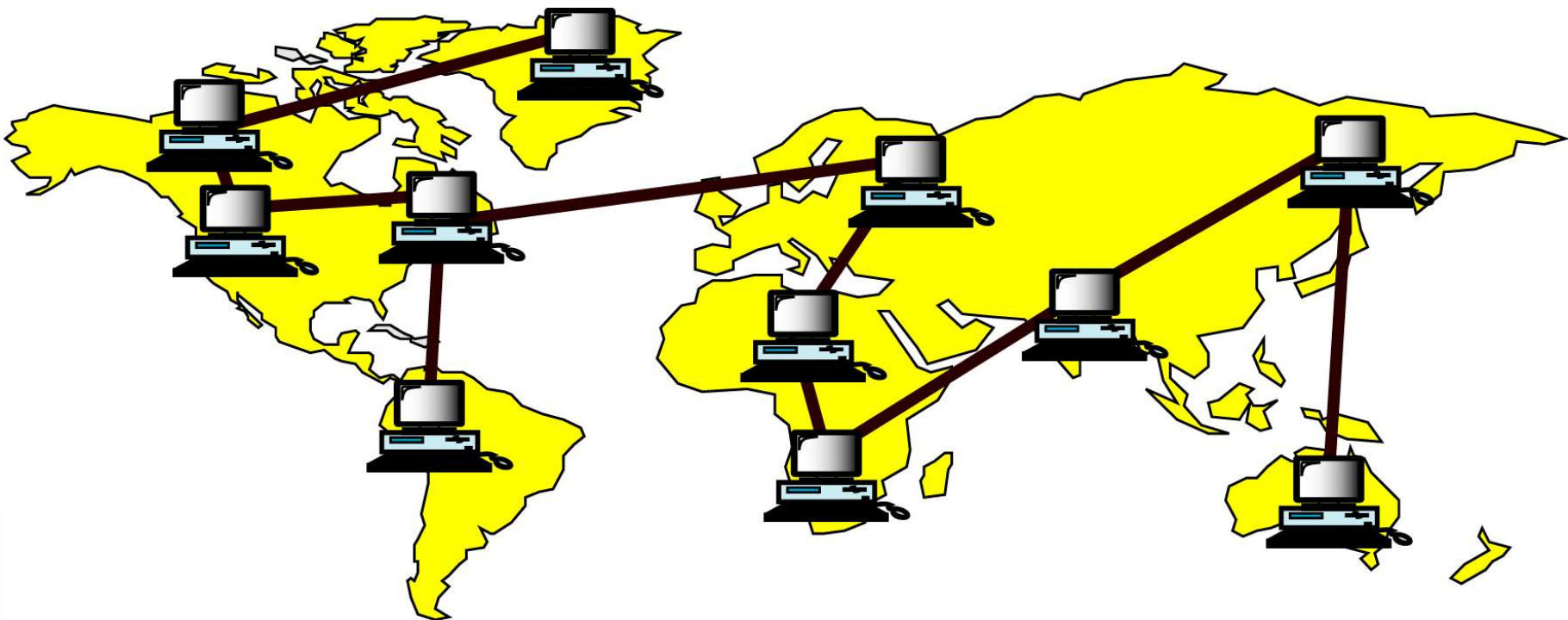
- ✓ 기업, 가정, 학교 등을 망라한 1개 도시 정도의 지역을 연결한 정보 통신망
- ✓ 데이터, 음성, 화상을 종합적으로 전송
- ✓ 전송 매체로는 주로 광섬유를 사용하며, 대용량 고속 전송 지원



## 2.4 네트워크 (14/15)

### 광역통신망(WAN: Wide Area Network)

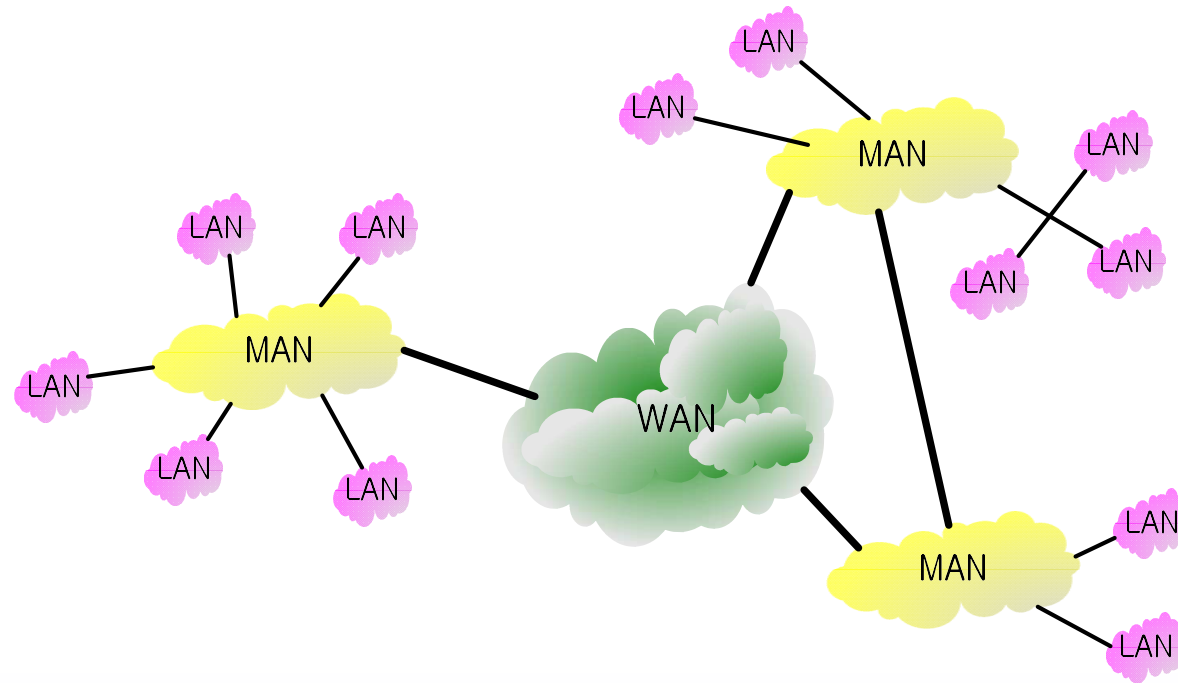
- ✓ 국가, 대륙 등과 같은 넓은 지역을 연결하는 네트워크
- ✓ 장거리 지역을 연결하는 백본(backbone) 네트워크
- ✓ 공공망까지 포함하는 사설망 혹은 임차한 망



## 2.4 네트워크 (15/15)

### 인터넷워크(Internetwork)

- ✓ 두 개 이상의 네트워크를 연결
- ✓ 인터넷워킹(internetworking) : 네트워크간 하드웨어나 소프트웨어 모두를 연결시키는 방법론
- ✓ '네트워크들의 네트워크(A Network of Networks)'



### 비교

- ✓ **Internet** : 상호 연결된 네트워크
- ✓ **Internet** : 전세계적인 특정 네트워크