

컴퓨터 네트워크

# 2장. 네트워크 모델 (1)

## - 계층구조의 개념

# 이번 시간의 학습 목표

- ▶ 계층구조 개념
- ▶ 기본적인 프로토콜 요소
- ▶ 서비스 프리미티브(primitive)

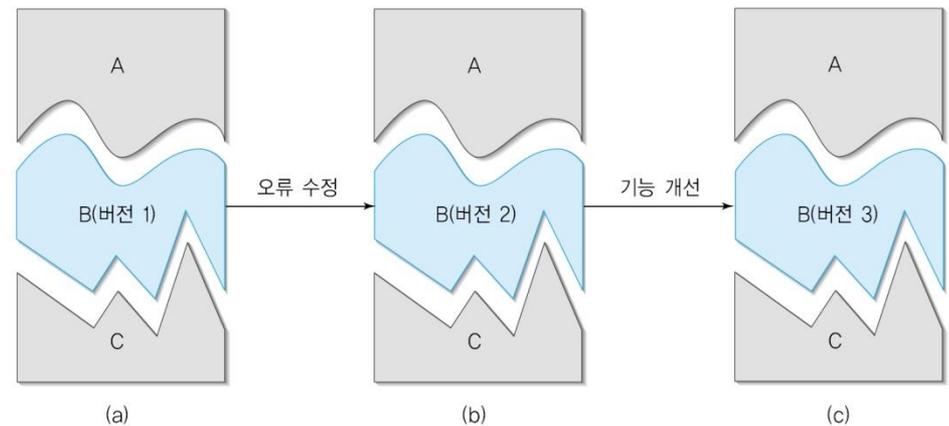
# 모듈화

## ▶ 모듈(module)

- ▶ 독립적으로 동작하면서 상호유기적으로 통합 가능해야 함(적절한 인터페이스 필요)

## ▶ 모듈화의 예

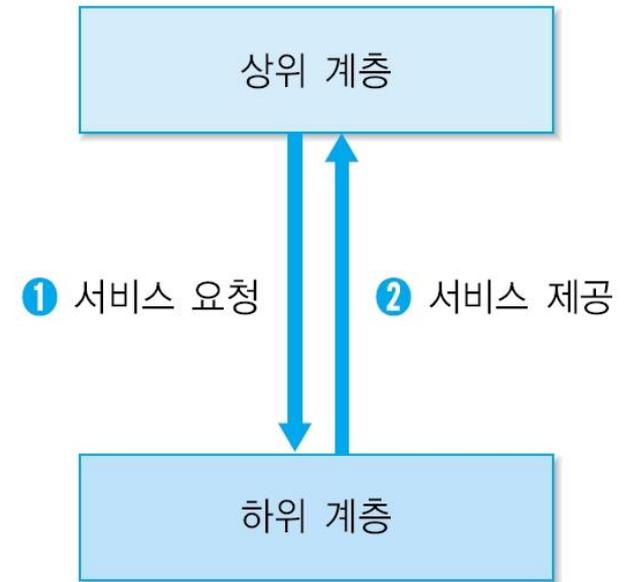
- ▶ 자동차
- ▶ 컴퓨터 하드웨어
- ▶ 소프트웨어의 함수, 객체지향 개념



[그림 2-1] 모듈화

# 계층화된 모듈 구조

- ▶ 상위 계층이 하위 계층에서 제공하는 서비스를 요구하는 방식으로 동작
- ▶ 모듈화된 계층 프로토콜 구조의 장점
  - ▶ 크고 복잡한 시스템을 작고 간단한 시스템들로 재구성 가능
    - ▶ 이해하기 쉽고, 설계 및 구현 용이
  - ▶ 모듈이 연동할 수 있는 인터페이스 제공
  - ▶ 전송 매체 양단의 호스트가 수행하는 프로토콜은 상호 대칭 관계
  - ▶ 각 계층의 오류를 수정하거나 개선하는 경우 전체 시스템의 교체 없이 해당 모듈의 교체로 가능
- ▶ 단점은 없을까?



[그림 2-2] 계층 구조

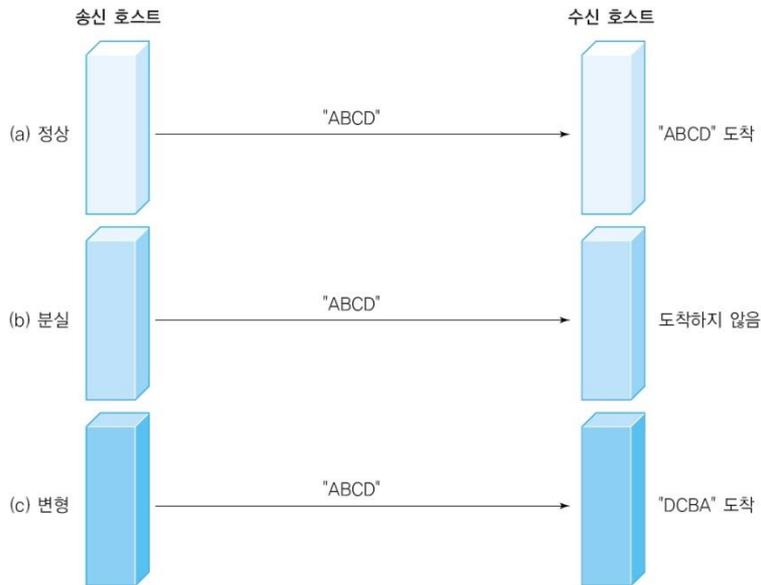
# 프로토콜 설계 시 주요 고려 사항 (1)

## ▶ 주소 표현

- ▶ 상대방을 구분할 수 있는 방법 필요
  - ▶ 주소 체계
- ▶ 호스트 주소
- ▶ 그룹 주소
  - ▶ 브로드캐스트 (방송 :broadcast)
  - ▶ 멀티캐스트 (multicast)

# 프로토콜 설계 시 주요 고려 사항 (2)

## ▶ 오류 제어(Error control)



[그림 2-3] 가능한 전송 결과 유형

- ▶ 위 그림에서 빠진 유형
  - ▶ 중복 도착

## ▶ 각 경우에 대한 해결 방안은 ?

- ▶ 도착하지 않았다는 것을 누가 어떻게 알 수 있는가?
- ▶ 변형이 일어났다는 사실은 어떻게 알 수 있는가?
- ▶ 동일한 데이터가 도착했다는 것을 어떻게 알 수 있는가?
- ▶ 응답(Acknowledgement)
  - ▶ 응답이 손실되는 경우
- ▶ 타임아웃(Timeout)
  - ▶ 타임아웃 후에 도착하는 경우
- ▶ 재전송(Retransmission)
  - ▶ 재전송이 손실되는 경우
- ▶ 순서번호(Sequence number)

# 프로토콜 설계 시 주요 고려 사항 (3)

## ▶ 흐름제어 (flow control)

- ▶ 송수신 호스트의 데이터 처리 속도 차이에서 데이터 분실 발생
  - ▶ 일반적으로 수신단의 부하가 더 큼
- ▶ 버퍼(buffer)를 이용하여 완화가 가능하나 버퍼 크기에 제한이 있음
- ▶ 송신 호스트의 전송 속도 조절 필요
- ▶ Ready / Not-ready



[그림 2-4] 흐름 제어

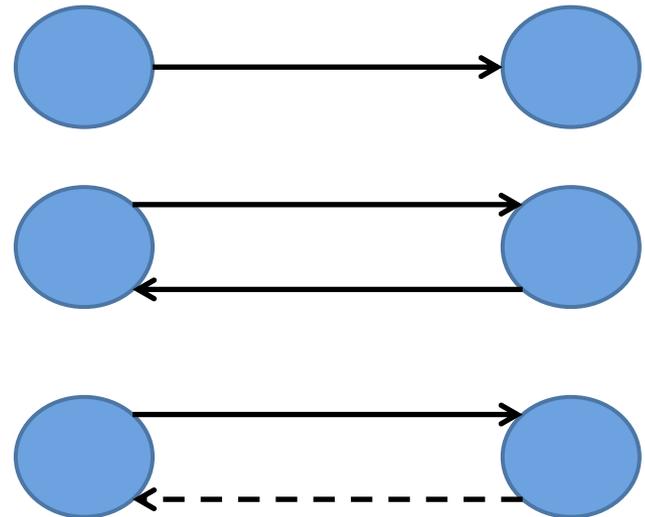
# 프로토콜 설계 시 주요 고려 사항 (4)

## ▶ 데이터 전달 방식

- ▶ 단방향(Simplex)
  - ▶ 한쪽 방향으로만 송신
- ▶ 전이중(Full duplex)
  - ▶ 양쪽이 동시에 송신 가능
- ▶ 반이중(Half duplex)
  - ▶ 양방향 전송이 가능하나 동시에 송신은 불가능
  - ▶ 데이터 전송 시점 관리 필요

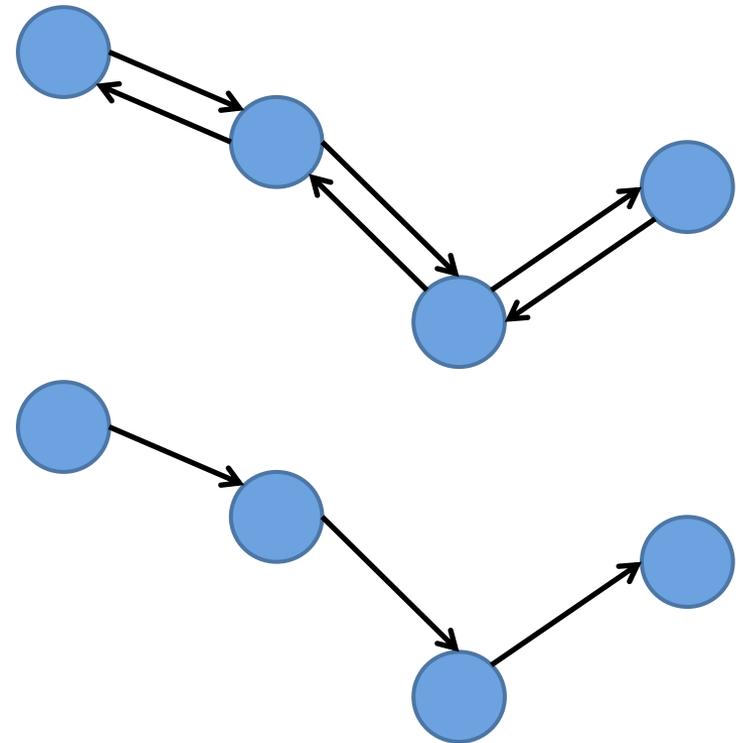
## ▶ 그외

- ▶ 데이터의 전송 우선순위
- ▶ 긴급데이터 처리



# 서비스의 종류

- ▶ 연결형(connection-oriented)
  - ▶ 전달경로를 설정하는 연결 설정 과정 필요
  - ▶ 데이터 전송 단계
  - ▶ 연결 해제 과정 필요
- ▶ 비연결형(connectionless)
  - ▶ 연결 설정 및 해제 과정 없이 바로 데이터 전송



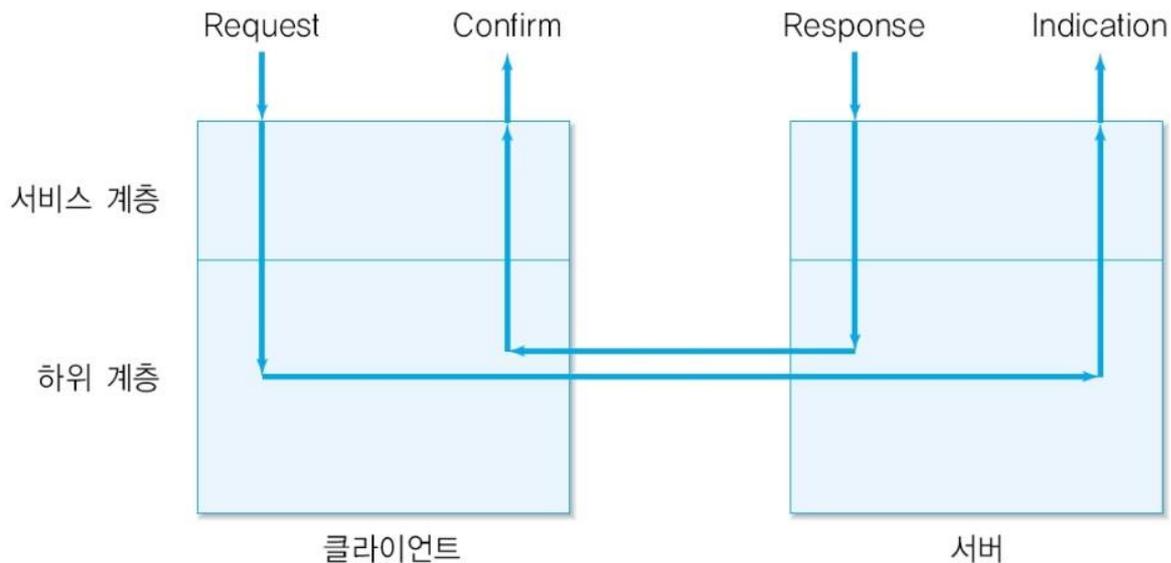
# 서비스 프리미티브 (1)

- ▶ 연결형 서비스에서 사용하는 프리미티브 종류
  - ▶ CONNECT: 연결 설정
  - ▶ DATA: 데이터 전송
  - ▶ DISCONNECT: 연결 해제
- ▶ 비연결형 서비스에서 사용하는 프리미티브 종류
  - ▶ DATA: 데이터 전송

# 서비스 프리미티브 (2)

[표 2-1] 서비스 프리미티브의 기능

프리미티브	의미
Request	클라이언트가 서버에 서비스 요구
Indication	서버에 서비스 요구가 도착했음을 통지
Response	서버가 클라이언트에 서비스 응답
Confirm	클라이언트에 응답이 도착했음을 통지



[그림 2-5] 프리미티브의 동작 원리

# 서비스 프리미티브 (3)

## ▶ 서비스 프리미티브 기능 요약

### ▶ Request

- ▶ 클라이언트에서 발생
- ▶ 하위 계층에 요구
- ▶ CONNECT.request, DATA.request, DISCONNECT.request

### ▶ Indication

- ▶ 서버에서 발생
- ▶ 클라이언트의 Request가 있었음을 상위 계층에 알림
- ▶ CONNECT.indication, DATA.indication, DISCONNECT.Indication

### ▶ Response

- ▶ 서버에서 발생
- ▶ 클라이언트로 응답
- ▶ CONNECT.response, DATA.response, DISCONNECT.response

### ▶ Confirm

- ▶ 서버에서 보낸 응답을 클라이언트로 통보
- ▶ CONNECT.confirm, DATA. confirm, DISCONNECT. confirm

# 질의 / 응답