

컴퓨터 통신

- (데이터) 통신의 개념
- 유무선 직접 연결 / LAN
- 패킷 네트워킹
- 인터넷 기초

Principles of Computer Networking

Larry Peterson

Department of Computer Science
Princeton University

1장. 기본 개념

☑요구 사항

- 네트워크가 제공해야 하는 것
- 결국, 네트워크에 대한 기능적 정의

□네트워크 구조

- 네트워크를 만드는 방법
- 체계적인 접근이 필수

□성능

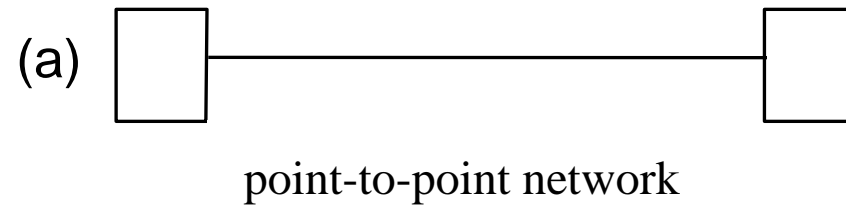
- 네트워크 비교/평가 기준
- 빠른 네트워크란?

연결 (Connectivity)

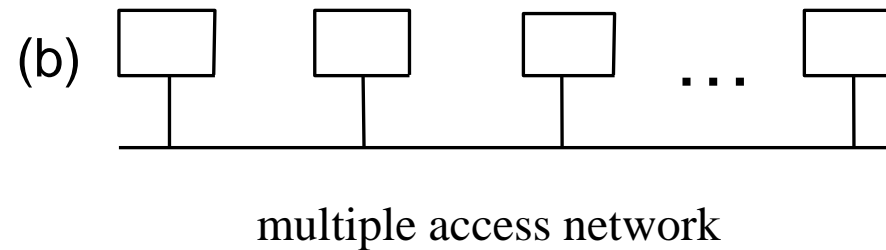
- 통신 첫 단계 – 통신 주체를 연결하는 것
- 따라서, 통신망이 해야 하는 제1업무 : 연결
- 통신 주체
 - 전화기 등 단말기
 - 좀 더 일반적으로는 통신 응용: 예 카톡, 브라우저, ...
 - 당분간은 단말기로 가정
- 연결 방법
 - 선을 연결; 무선 연결도 가능
- 통신망 구성 요소 (물리적 연결 제공을 위한)
 - 노드(nodes): 단말기, 범용 워크스테이션, 라우터 등
 - 링크(links) : 동축케이블, 광케이블, 무선채널 등

연결: 직접 링크 (Direct Links)

- 점대점 연결 (point-to-point)
 - 가장 간단한 네트워크



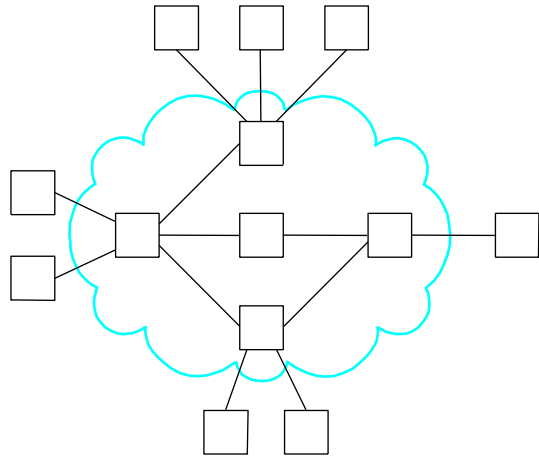
- 다중 접근 (multiple access)



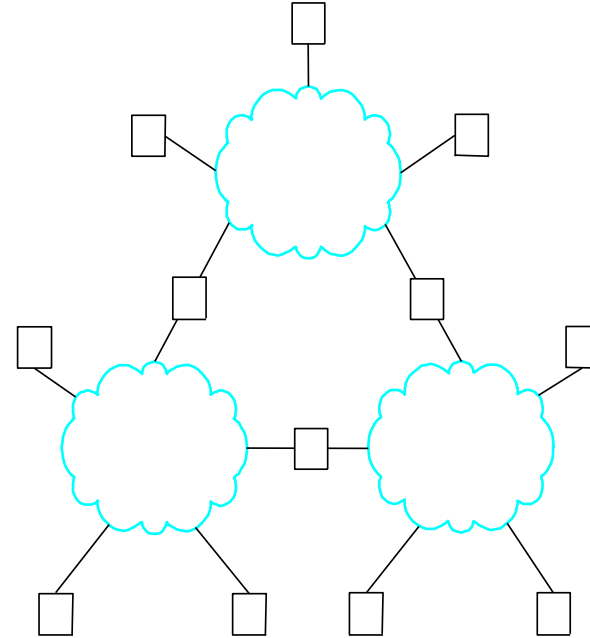
간접 연결 : Switched Networking

- 간접 연결 (Indirect Connectivity): Switched Networks

- 스위칭 네트워크
(switching networks)



- 인터넷워킹(internetworks)
⇒ Network of Networks



간접 연결 방법 : 스위칭 정책

- 회선 스위칭 (circuit switching): 전화 네트워크
 - 스위치가 **사전**에 output link 에 **전용** 회선(용량) 확보
 - **비트스트림**을 중단/간섭 없이 송/수신 (흘려 보냄)
 - 기본적으로 point-to-point 연결
- 패킷 스위칭 (packet switching): 인터넷/우편
 - 데이터를 묶음으로 전송 : 패킷
 - 스위치 동작 : store-**and**-forward
- 컴퓨터 통신에 적합한 것은?
 - 사용자 입장 vs 네트워크 입장?
 - bursty traffic 처리에 적합한 것은?

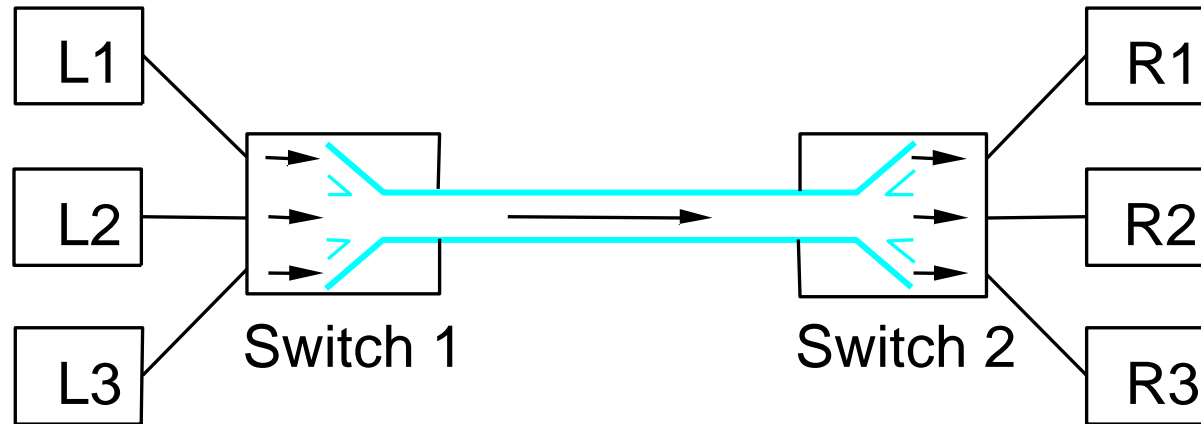
연결: 선결 과제

어드레싱(Addressing) 및 라우팅(Routing)

- 상대방을 지정. 즉, ID 지정
- 주소(address): 노드를 식별하는 바이트열
 - 대개 유일하다.
- 참고: 라우팅
 - 목적지 노드를 향해 메시지를 어떻게 포워딩할지를 주소에 입각해서 결정하는 작업
- 주소의 종류 (즉, 연결 형태)
 - 유니캐스트(unicast): 특정 노드를 지정
 - 방송/브로드캐스트(broadcast): 네트워크의 모든 노드
 - 멀티캐스트(multicast): 네트워크의 일부 노드 집합을 지정

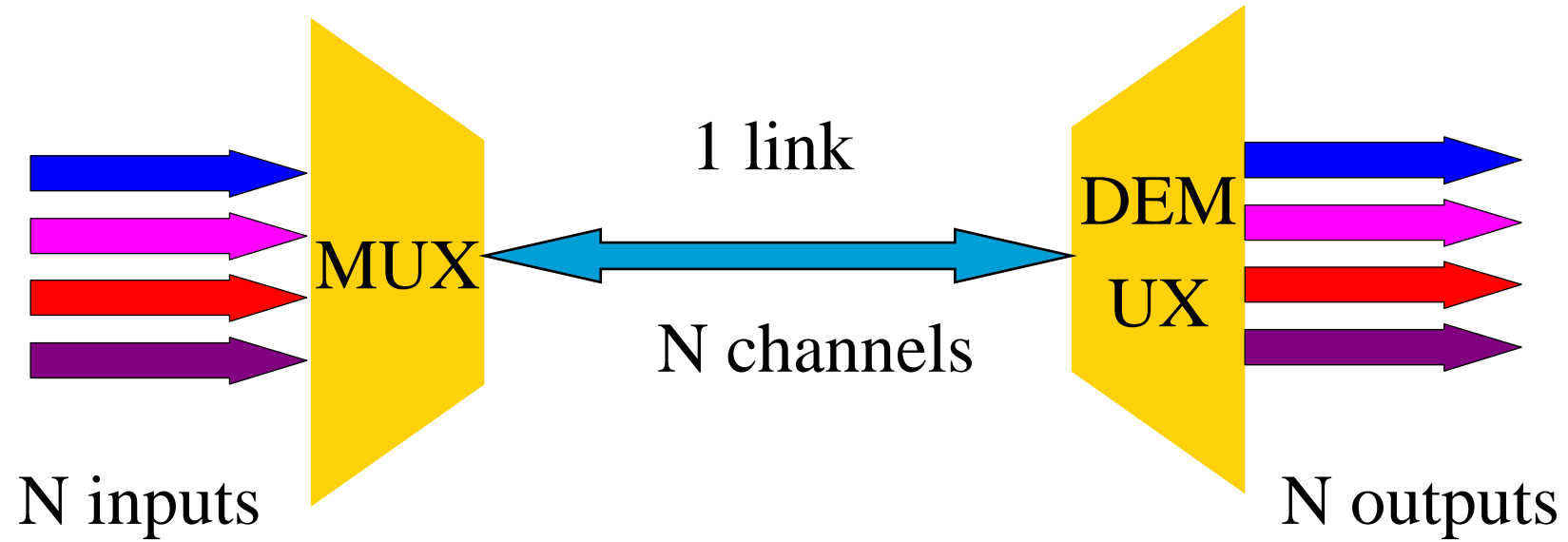
비용 효율적인 자원 공유 (Resource Sharing)

- 여러 사용자들이 네트워크 자원(노드 및 링크)을 공유하도록 하여야 한다. 통신용어로는, 다중화(multiplexing) 지원

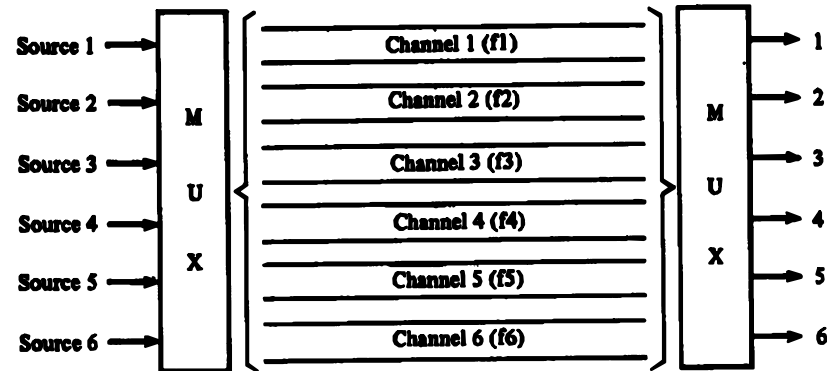


- 흔히 사용되는 다중화 정책
 - 시분할 다중화 (Time-Division Multiplexing: TDM)
 - 주파수분할 다중화 (Frequency-Division Multiplexing: FDM)

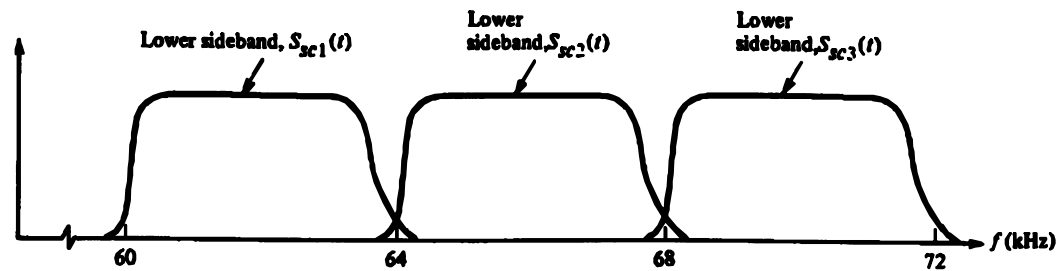
Multiplexing



주파수분할 다중화 (Frequency Division Multiplexing)



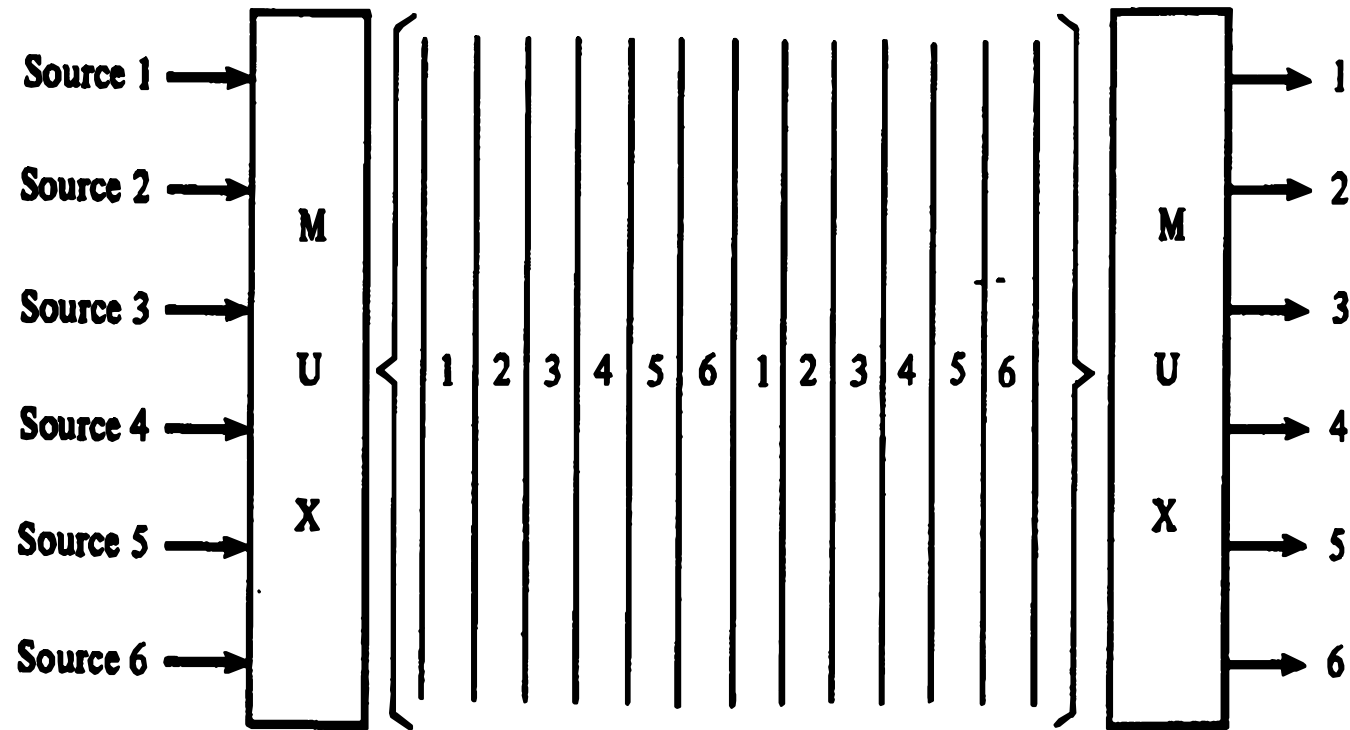
(a) Frequency-Division Multiplexing



(c) Spectrum of composite signal using subcarriers at 64 kHz, 68kHz, and 72 kHz

시분할 다중화 (Time Division Multiplexing)

동기식 (Synchronous) 시분할 다중화



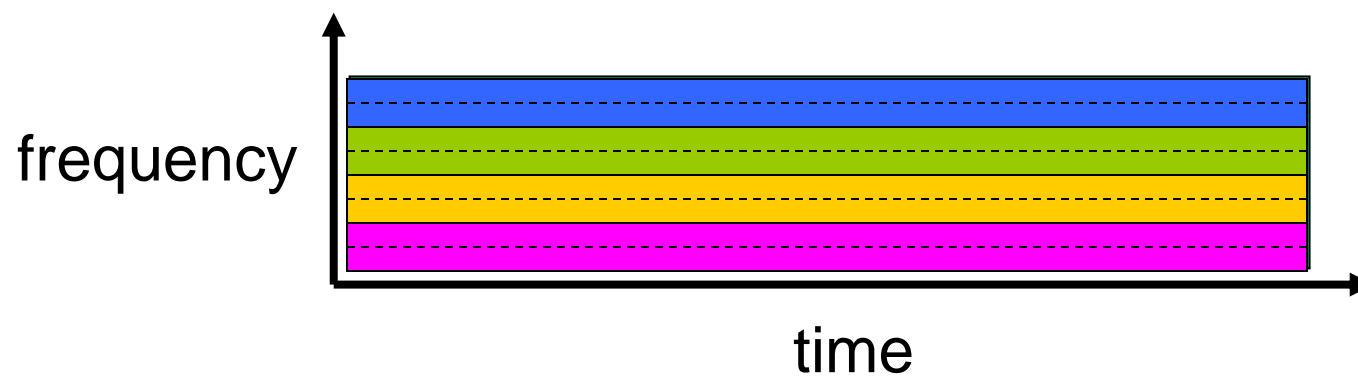
(b) Time-Division Multiplexing

다중화 : FDM and TDM

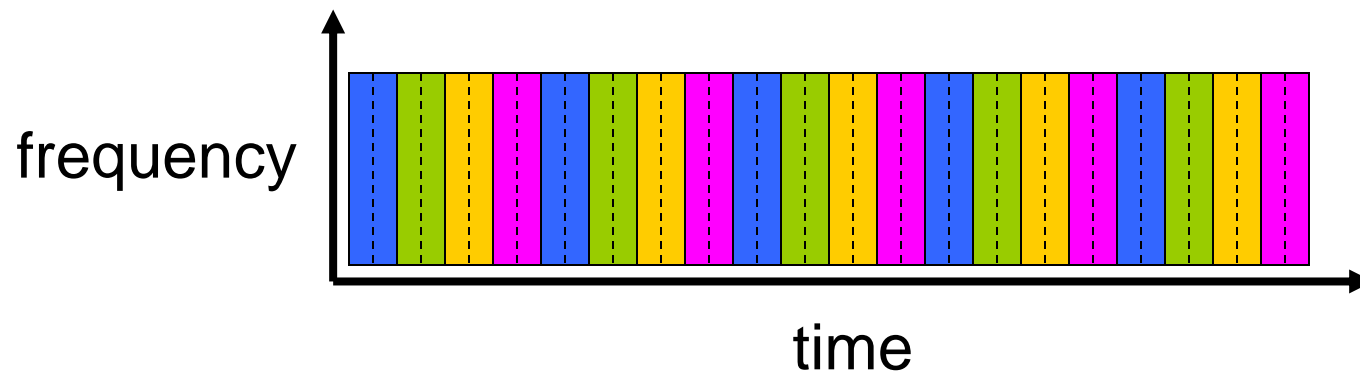
FDM

Example:

4 users

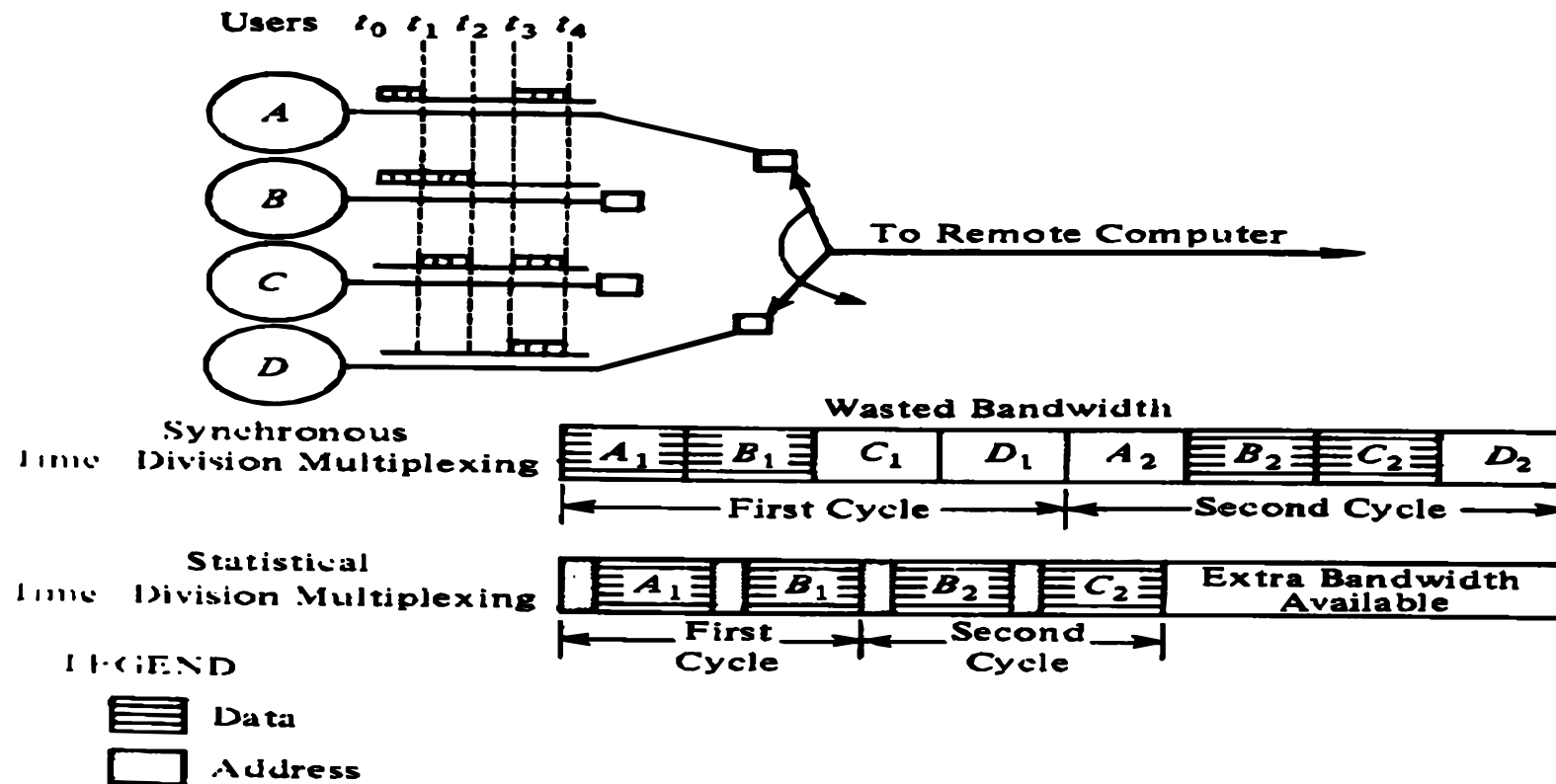


TDM

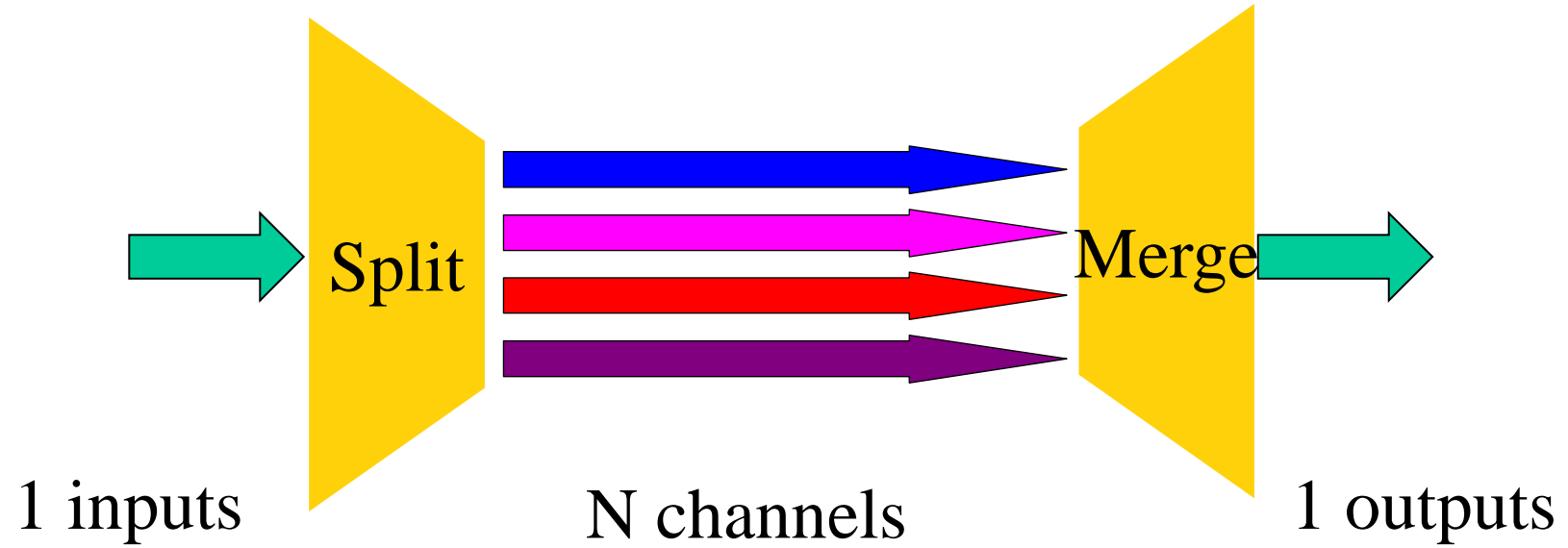


통계적 다중화 (Statistical Multiplexing)

- 시분할 방법의 일종: 고정 분할이 아닌 요구에 따른 분할.
 - 비동기식 다중화
- Demux key/select ?
- 항상 좋은가?

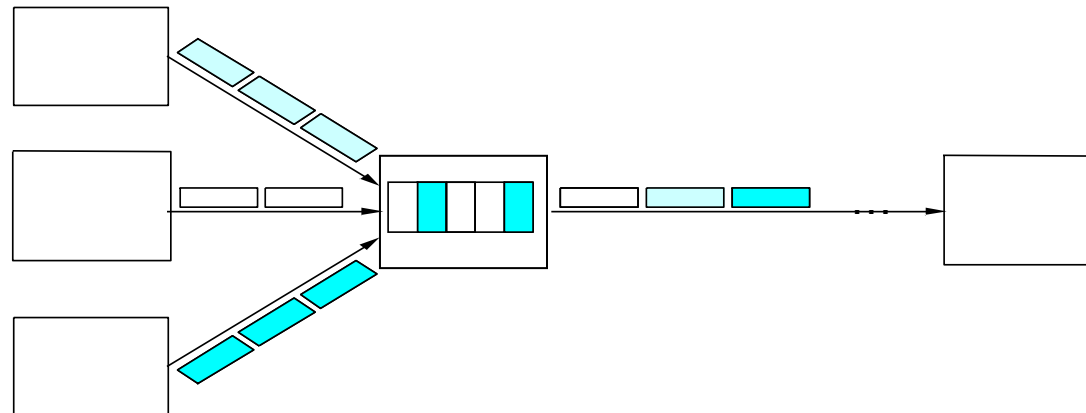


Splitting

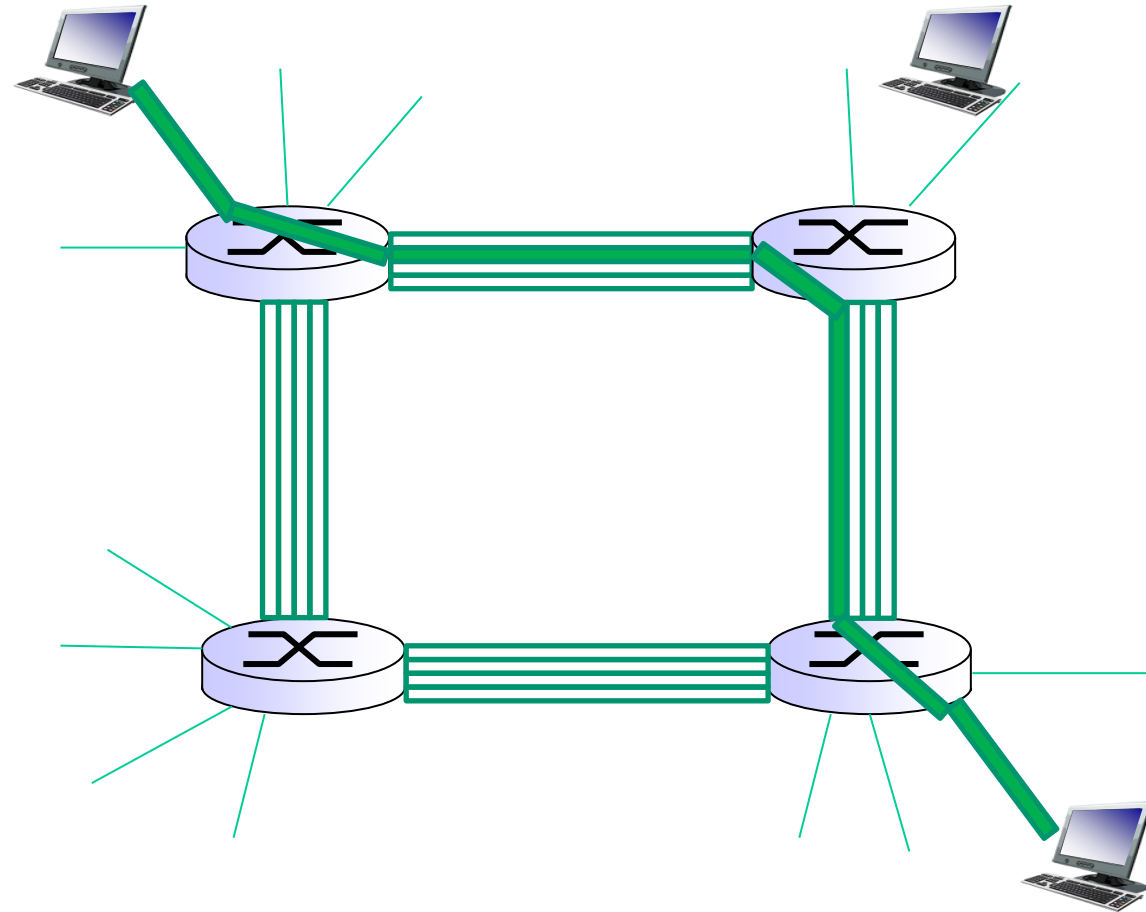


통계적 다중화와 패킷스위칭

- 통계적 다중화는 링크를 공유하는 방법
- 패킷스위칭은 노드가 목적지를 향해 데이터를 전달하는 방법
- 패킷스위칭: 패킷 단위로 링크 사용을 재스케줄링
 - 패킷스위칭의 결과, 링크는 (거의) 통계적다중화
 - 링크에서 통계적다중화를 하려면, 노드에서는 패킷 스위칭 필요
- 다른 출발지/소스(source)로부터의 패킷들이 링크에서 섞이게 됨.
- 링크로 나가기 위해 경쟁하는 패킷들을 저장: 버퍼링(buffering)
- 패킷은 FIFO로 처리되거나, 기타 다른 방식으로 처리
- 버퍼 오버프로우(overflow)를 혼잡(congestion)이라 부름.

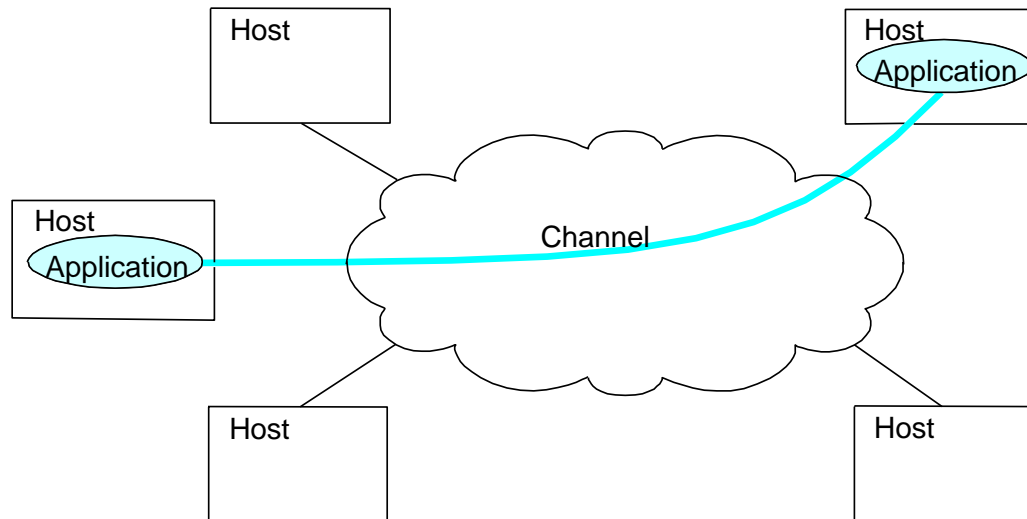


!참고! : 서킷스위칭 ⇒ 다중화 ?



통신 서비스 제공

- 통신의 주체는 응용 프로그램
- 따라서, 네트워크는 응용 프로그램이 원활히 통신할 수 있는 기능을 제공
 - 통신기술을 기반으로 응용 프로그램이 요구하는 기능을 구현/제공
 - 즉, **호스트** 간의 연결을 **프로세스** 간의 통신 형태로 변환



네트워크는 프로세스와 프로세스간의 채널(channel)을 지원한다.

통신 서비스 : 통신 장애 극복

- 네트워크가 정상적으로 동작하지 않는 경우:
 - 비트 수준 오류 (전자기 간섭/방해)
 - 패킷 수준 오류 (혼잡)
 - 링크/노드 고장
 - 메시지의 지연
 - 메시지의 순서가 바뀌어 전달 (out-of-order)
 - 제삼자의 도청

통신 기술의 핵심은 응용이 예상하는 것과 통신 기술이 제공하는 것 사이의 거리를 메우는 일이다.

1장. 기본 개념

□요구 사항

- 네트워크가 제공해야 하는 것
- 결국, 네트워크에 대한 기능적 정의

☑네트워크 구조

- 네트워크를 만드는 방법
- 체계적인 접근이 필수 : 계층화에 기초한 표준

□성능

- 네트워크 비교/평가 기준
- 빠른 네트워크란?

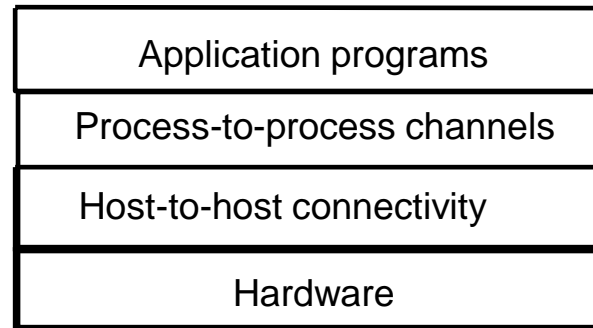
프로토콜 (Protocol)

- 통신에 사용되는 약속
 - 예) 수신호, 언어
 - 양쪽이 같아야 함. 즉, 반드시 대칭 관계
- 다양한 컴퓨터통신 시스템/응용
 - ⇒ 프로토콜의 복잡화
 - ⇒ 불명확한 해석
 - ⇒ 변경 등 관리의 어려움
 - ⇒ 새로운 프로토콜이 필요할 때마다 반복
- 복잡성을 해결하는 구조적인 기법이 필요

계층화 (Layering)

- 복잡한 문제를 한번에 풀 수 없다.
 - 복잡한 내용/문제를 숨겨서 문제를 단순화 → 추상화(abstractions)
 - 추상화된 문제/내용의 해결 → 추상화를 recursive하게 적용

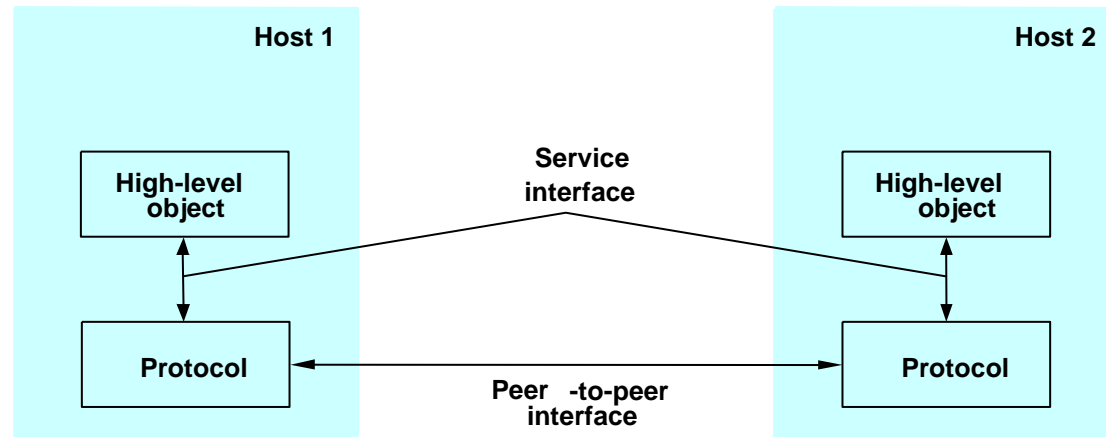
- 추상화는 자연히 계층화를 유도



- 통신 프로토콜 ← 여러 계층으로 정의
- 각 계층은 하나의 기능을 하는 부품/개체로서 다른 프로토콜에서 재사용 가능

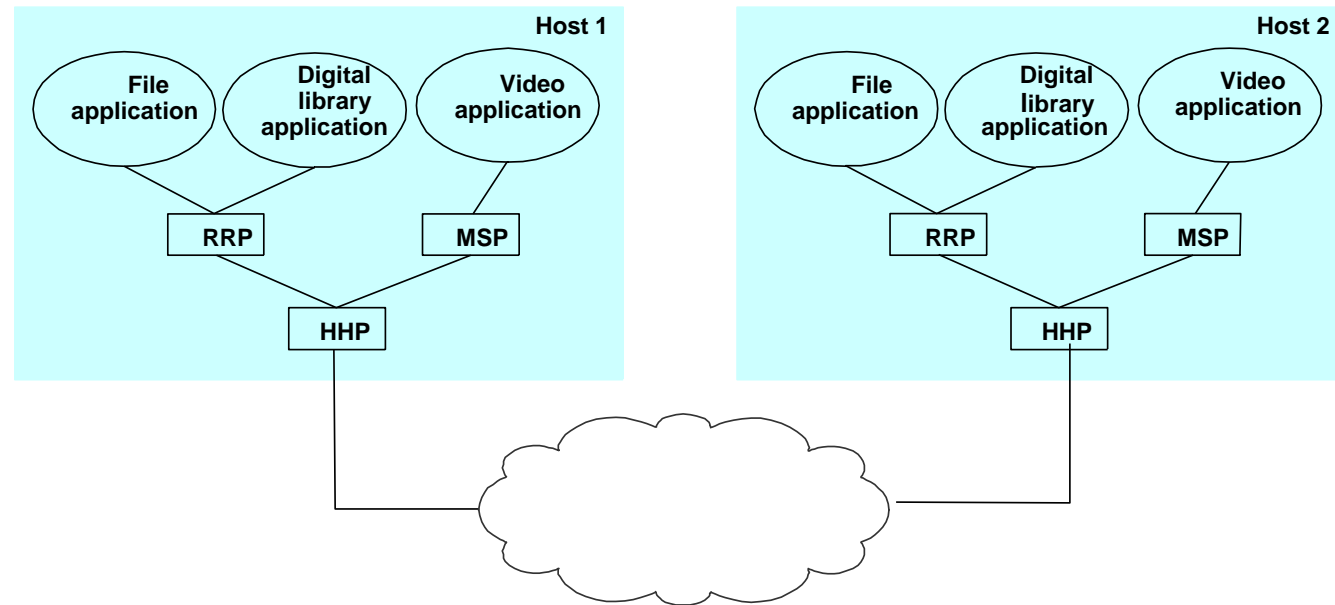
프로토콜 계층/개체

- 프로토콜 계층/개체
 - (전체) 프로토콜을 구성하는 계층, 즉, 프로토콜의 구성 요소
 - 자체도 프로토콜이라고도 부름.
- 각 프로토콜 개체는 두 개의 다른 인터페이스를 갖는다.
 - 서비스 인터페이스(service interface): 해당 프로토콜의 작업을 정의
 - 동료 인터페이스(peer-to-peer interface): 동료 간에 교환되는 메시지를 정의



(전체) 프로토콜 정의: 프로토콜 그래프

- 프로토콜 **그래프** (또는 프로토콜 **스택**)
 - 프로토콜의 모음(collection)과 그들 사이의 의존관계(dependency)
 - 동료간의 통신은 대개 간접적으로 이루어진다.
 - 실제 통신은 하위 계층을 사용하여 (즉, 위임하여) 이루어진다.
 - 하드웨어 수준에서만 동료 간이 직접적임.

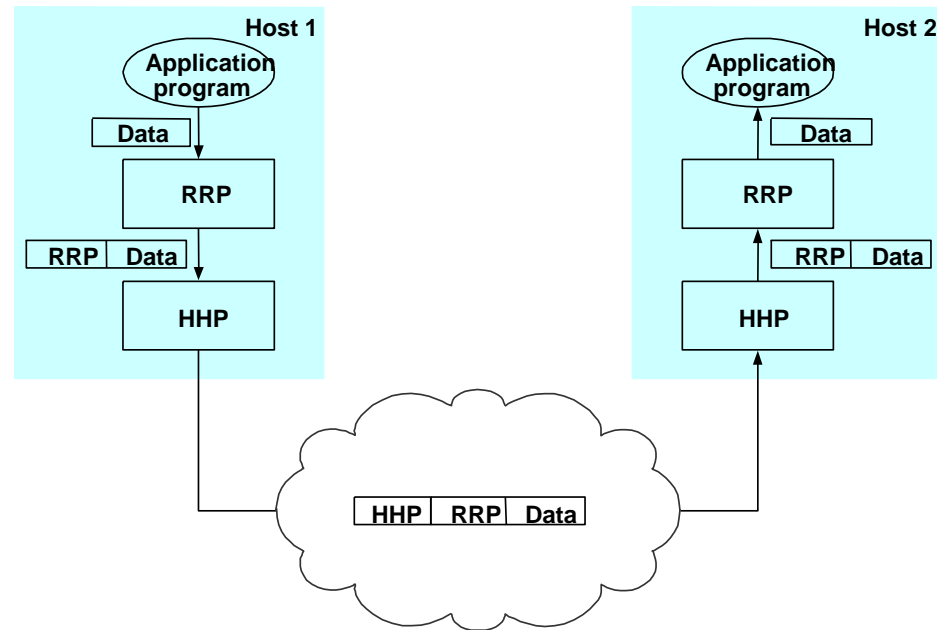


하위 프로토콜 공유

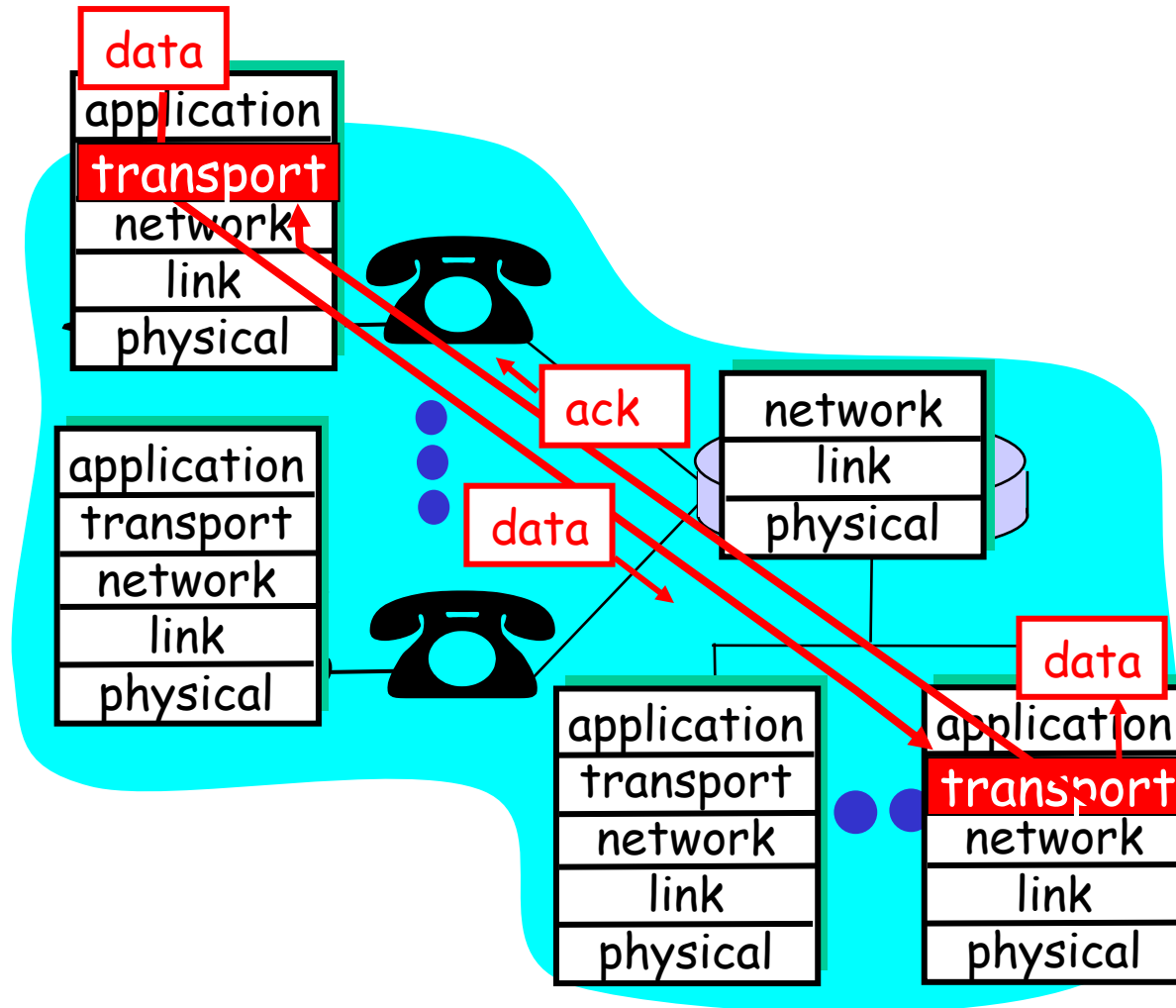
⇒ 다중화(Multiplexing) 및 역 다중화 (Demultiplexing) -- demux key

(계층적) 프로토콜: 동작 원칙

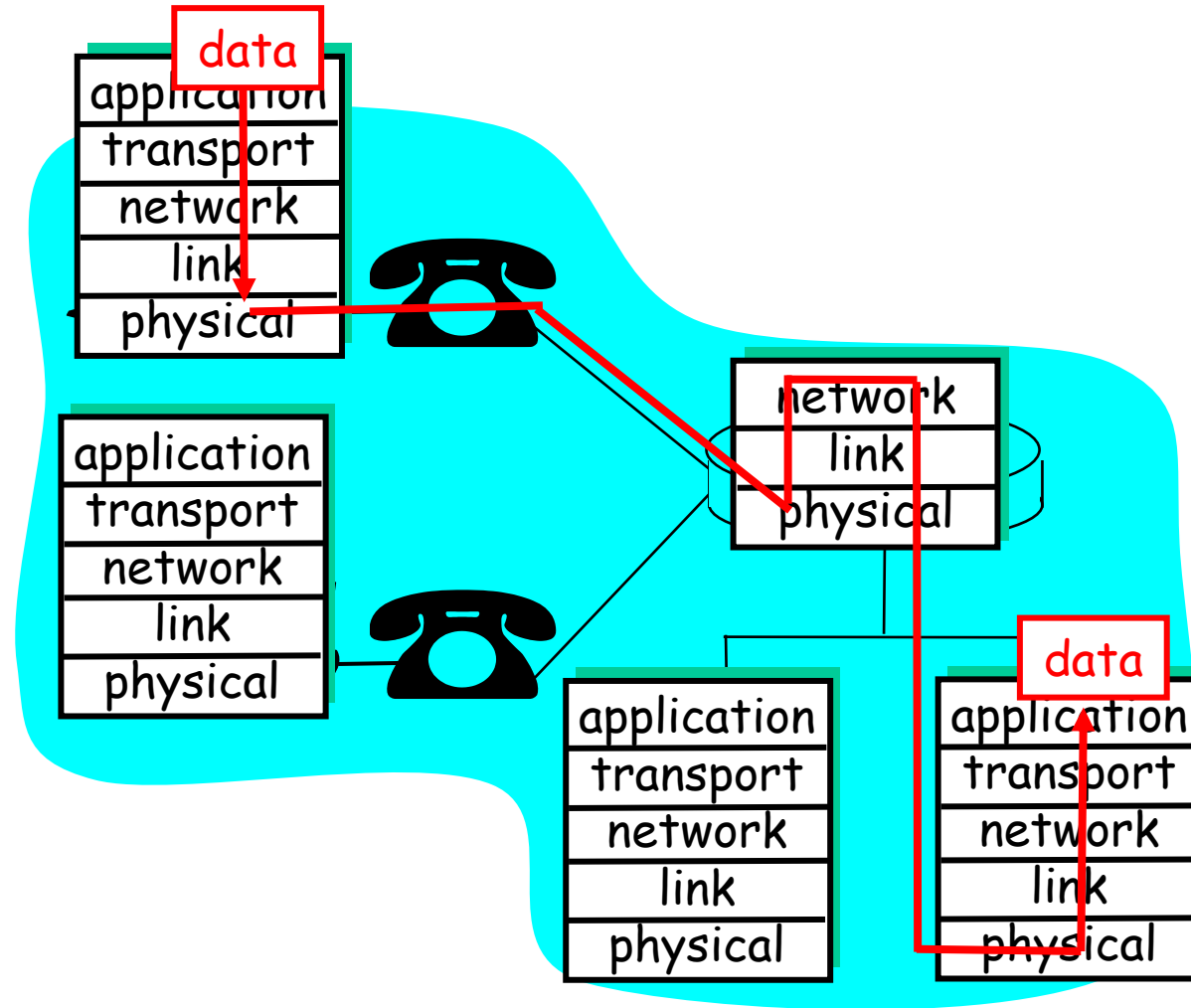
- 포장/캡슐화(Encapsulation) -- 헤더(header)/바디(body)



Layering: *logical* communication



Layering: physical communication

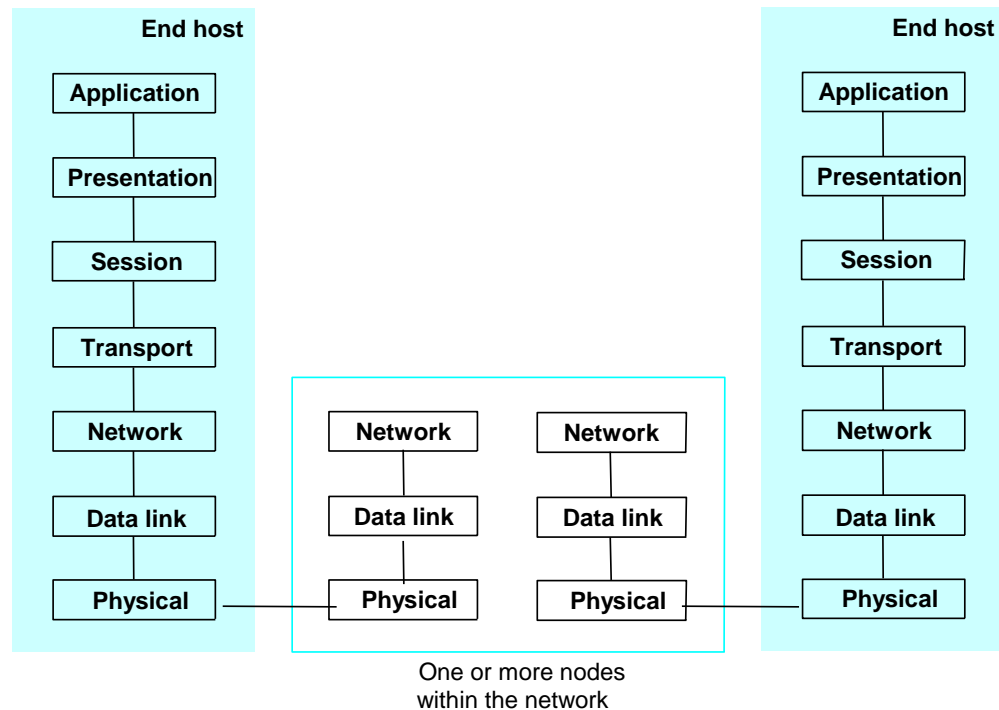


추상화/계층화 개념 정리

- 추상화 : “ ... 기능은 있다고 가정하자”
⇒ “... 기능이 하위 계층에서 제공된다”고 가정
⇒ “ ... 기능” 문제로부터 해방
- 계층화 : “ 여러 계층으로 나누어 각 계층이 해당 기능을 담당한다”
⇒ 하위 계층이 제공하는 기능을 (정해진 서비스 인터페이스에 따라) 이용해서,
⇒ 주어진 기능을 상위 계층에게 제공

표준 구조 (Standard Architectures) (1)

- Open Systems Interconnect (OSI) Architecture
 - International Standards Organization (ISO)
 - International Telecommunications Union (ITU): formerly CCITT
 - “X dot” series: X.25, X.400, X.500
 - 참조 모델(Reference Model) : 통신문제를 나누어 생각하는 틀

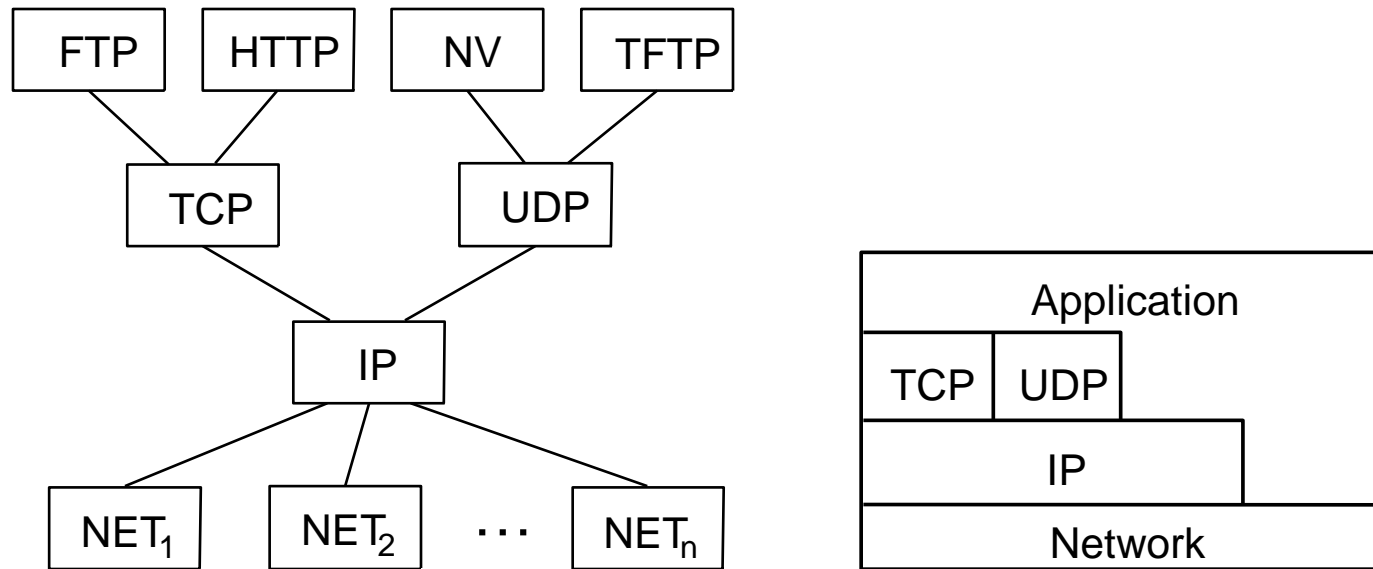


7 계층 (기능) 정의

- 응용 계층 : 응용 자체와 관련된 사항만.
- 프리젠테이션 계층 : 데이터 표현 방법과 관련된 사항
- 세션 계층 : 대화 패턴과 관련된 사항 담당
- 전송 계층 : 통신 응용 사이의 신뢰성 있는 메시지 교환
 - end-to-end reliable data transfer (process간 channel 제공)
- 네트워크 계층 : 네트워크를 통해 연결된 호스트(단말) 사이의 데이터 (패킷) 교환
- 링크 계층 : 하나의 링크로 연결된 노드 사이의 비트묶음 (프레임) 교환
- 물리 계층 : 물리적인 신호 교환과 관련된 사항 담당

표준 구조 (Standard Architectures) (2)

- 인터넷(Internet) 구조
 - Internet Engineering Task Force (IETF)



- 응용 및 응용 프로토콜 (FTP, HTTP)
- 특징
 - 계층화를 그대로 따르지 않음.
 - 모래시계 형태의 모양
 - 설계와 구현을 병행해서 진행

OSI (7계층) 모델에 대해 설명하시오.

- 컴퓨터네트워크 설계/구현의 구조 체계 표준
 - 실제 구현 X, 문제인식/설계의 참조 모델
- 통신에 필요한 모든 기능/기술 요소를 7 계층의 계층 구조로 배분
 - 하위 계층 서비스를 추상화하고 상위 계층 해결
- 각 계층 설명
 - 1 물리계층: 직접 연결된 노드 사이의 비트 전송
 - 2 링크 계층: 직접 연결 노드 사이의 프레임(비트 묶음)의 전송
 - 3 네트워크 계층: 스위치로 간접 연결된 호스트 사이의 패킷 전송
 - 4 트랜스포트 계층: 종단간 (end-to-end) 신뢰성 있는 전송
 - 종단: 응용이 연결되는 지점

1장. 기본 개념

□요구 사항

- 네트워크가 제공해야 하는 것
- 결국, 네트워크에 대한 기능적 정의

□네트워크 구조

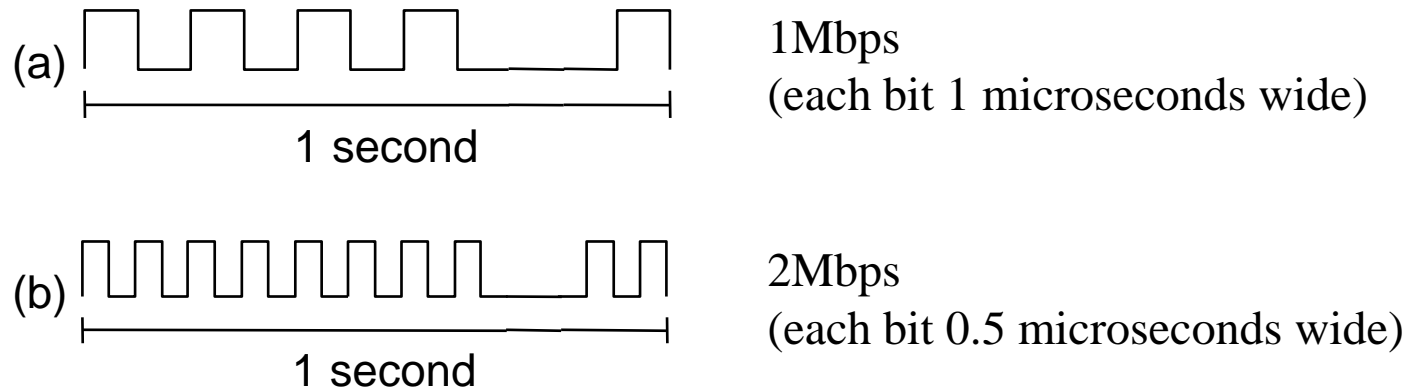
- 네트워크를 만드는 방법
- 체계적인 접근이 필수 : 계층 구조

☑성능

- 네트워크 비교/평가 기준
- 빠른 네트워크란?

성능 (Performance): 대역폭

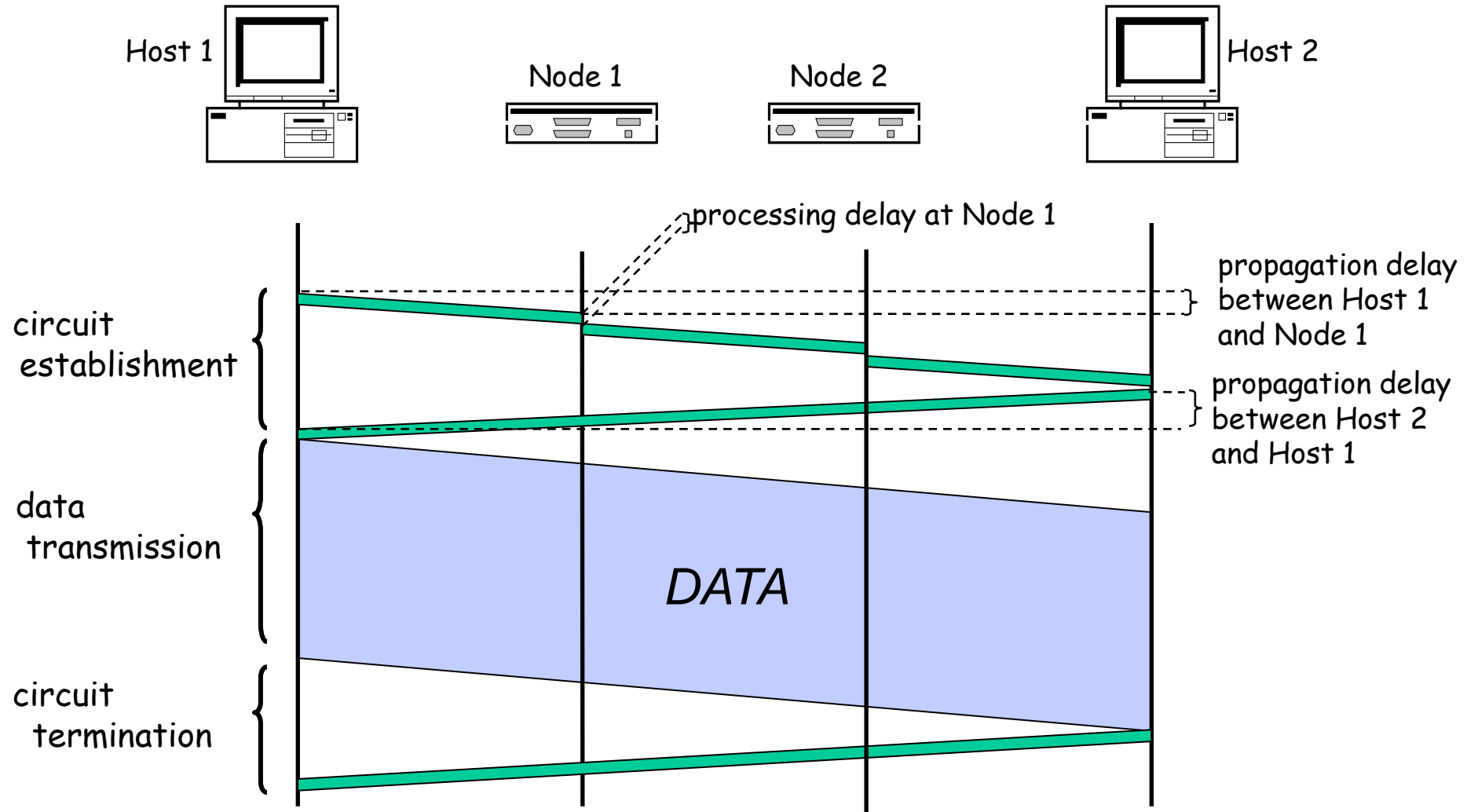
- 대역폭 (Bandwidth) -- 처리량/처리속도 (throughput)
 - 단위 시간당 전송될 수 있는 데이터의 양
 - 예: 10Mbps (bits-per-second)
 - 링크간 대 종단간
 - 표기 방법
 - $\text{KB} = 2^{10} \text{ bytes}$
 - $\text{Mbps} = 10^6 \text{ bits per second}$
 - 대역폭은 비트 폭(bit width)과 관련이 있다.
 - 속도가 높아지면 비트폭은 작아진다. 즉, 비트 사이가 좁아진다.



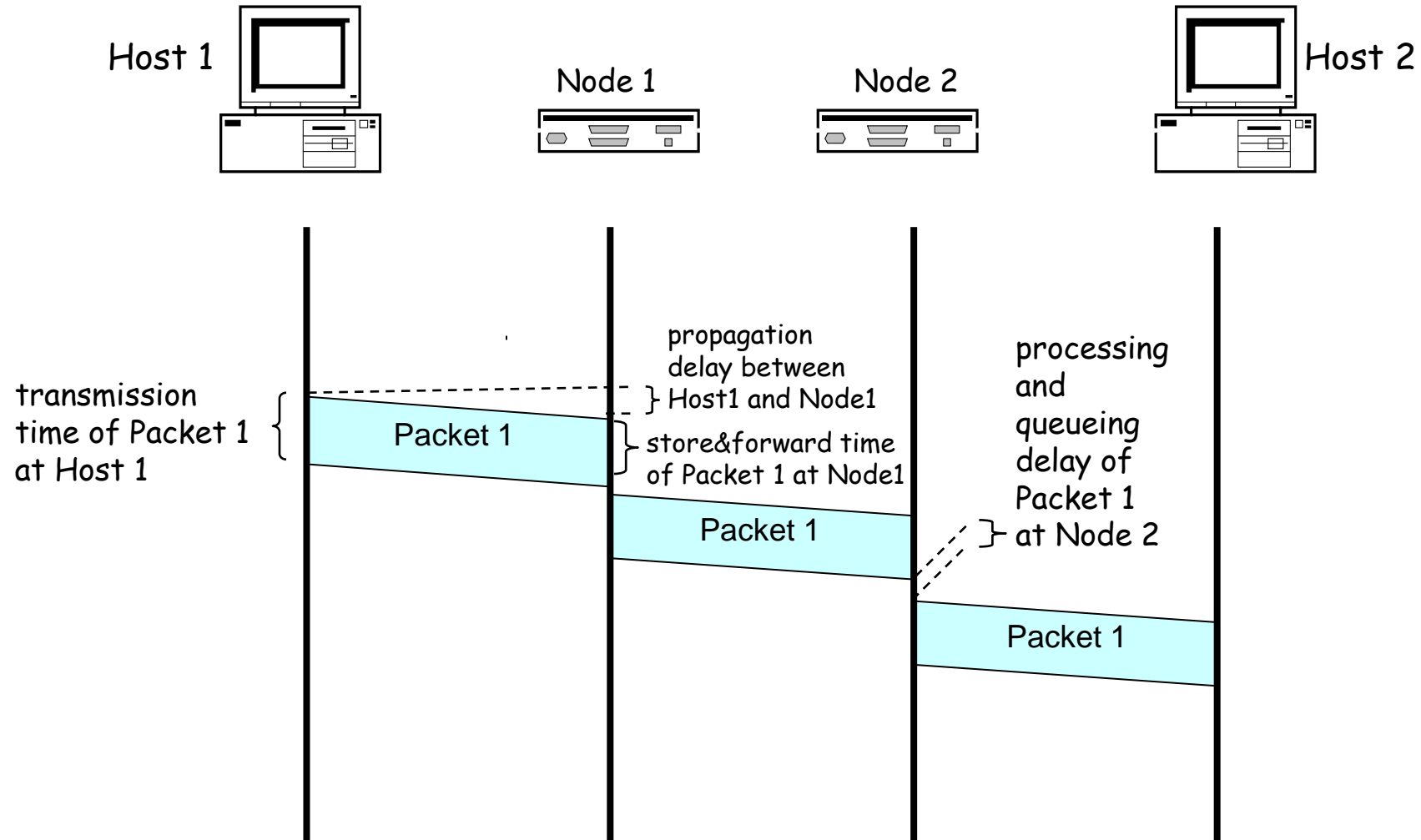
성능 : 소요시간/지연시간

- 소요시간(Latency) / 지연시간(delay)
- A 지점에서 B 지점으로 메시지를 송신하는데 걸리는 시간
 - 예: 24 milliseconds (ms)
- 때로 왕복지연시간(round-trip time: RTT)이 중요할 수 있다.
- 시간 소요 항목
 - 소요시간 = Propagation + Transmit + Queue [+ 스위칭 시간]
 - Propagation(전파지연시간) = Distance / SpeedOfLight
 - Transmit(전송시간) = Size / Bandwidth
- 광속(Speed of light) -- 전파의 속도
 - 3.0×10^8 m/s in a vacuum, 2.3×10^8 in a cable, 2.0×10^8 in a fiber
- 직접 링크에서는 큐잉 지연(queueing delay)은 없음.

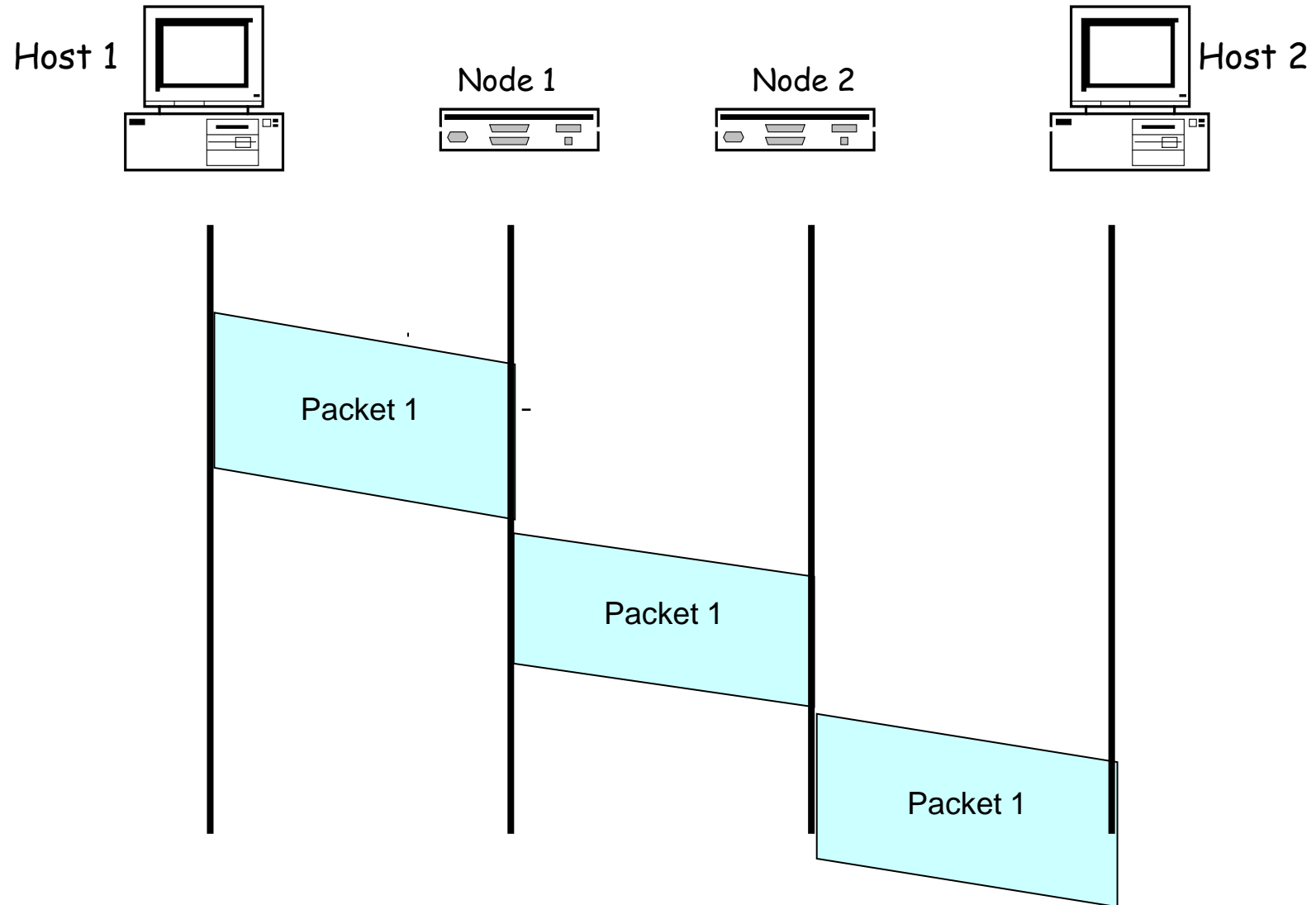
Timing in Circuit Switching



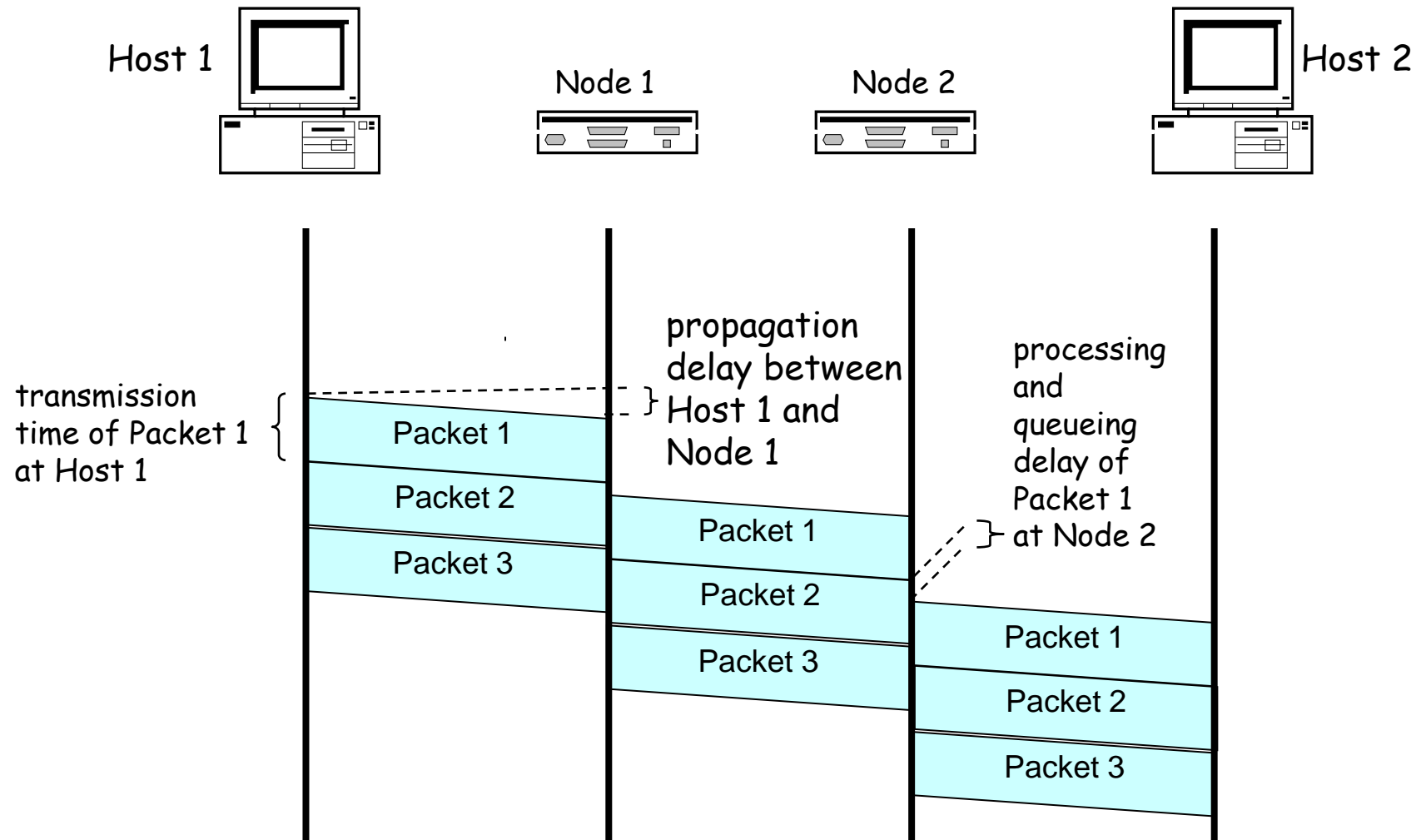
Timing of Packet Switching



Timing of Packet Switching



Packet Segmentation : Pipelining



성능 (Performance) (3)

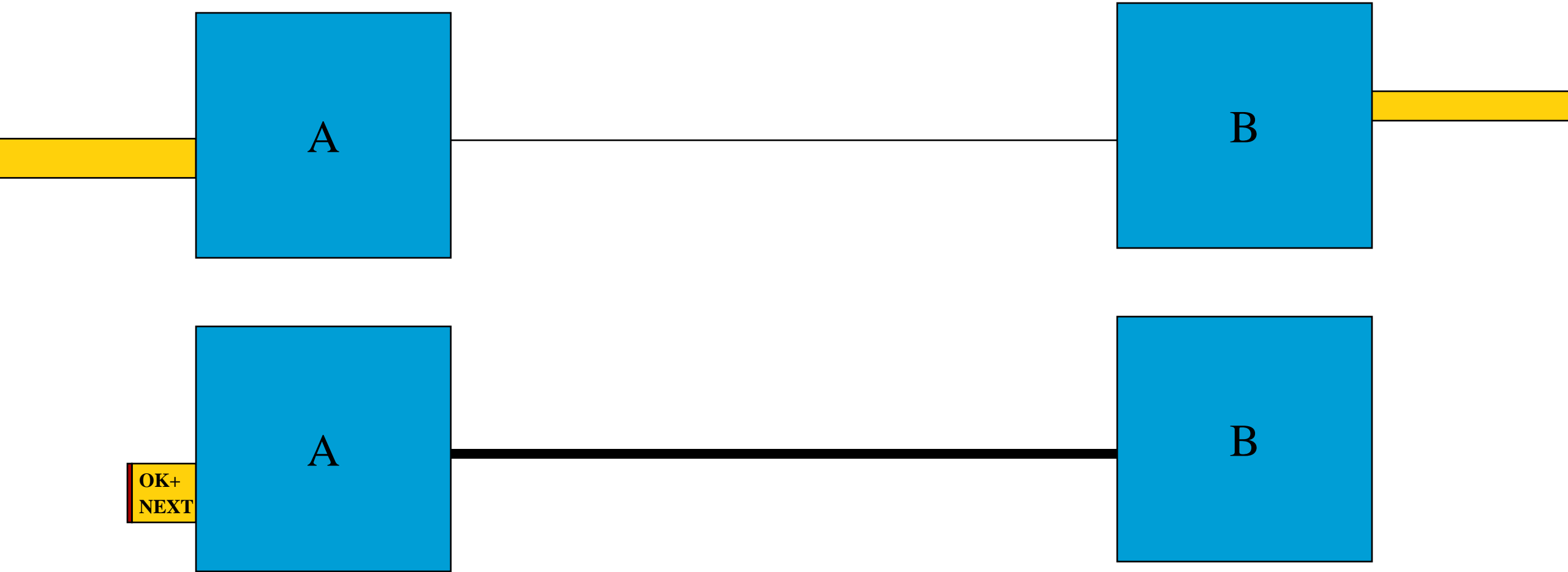
- 대역폭과 소요시간의 상대적 중요성
 - 작은 메시지 (예, 1 byte): 소요시간 중요
1ms vs 100ms dominates 1Mbps vs 100Mbps
 - 큰 메시지 (예, 25 MB): 대역폭 중요
1Mbps vs 100Mbps dominates 1ms vs 100ms
- 무한대 대역폭
 - 총 소요시간이 중요
 - $\text{Throughput} = \text{TransferSize} / \text{TransferTime}$ (전송완료시간)
 - 회선 스위칭: $\text{TransferTime} = \text{RTT} + (1/\text{Bandwidth}) \times \text{TransferSize}$
 - 패킷 스위칭: 여러 요소에 영향. 과제로.
 - 1-MB *file* to 1-Gbps link as 1-KB *packet* to 1-Mbps link
 - 데이터도 같이 증가!

지연시간 x 대역폭 (Delay x Bandwidth Product)

- “in flight” or “in the pipe” 데이터 양
- 비트로 나타내는 링크의 길이 (bit length)
 - 링크의 부피
 - 대역폭이 반영된 지연시간
- 예: $100\text{ms} \times 45\text{Mbps} = 560\text{KB}$



Frames



성능: 기타 사항

- 응용의 요구사항
 - 대역폭 요구: 버스트(burst) 대 최고속도(peak rate)
 - 지터(jitter): 소요시간의 변이 (패킷 사이의 차이)
- 기타 주의 사항
 - 만일 (메시지) 길이가 1이라면 대역폭은 의미가 없음.
 - 프로세스 간의 소요시간에는 소프트웨어 처리 부하도 포함됨.
 - 거리가 짧으면 소프트웨어 부하가 주요소가 된다.

통신 성능을 높이는 방법?

- 처리량 (대역폭)을 높이는 방법
- 소요시간 (지연시간)을 줄이는 방법
- 두 방법이 같은 것이 아님!
 - 대역폭을 높이면, 소요시간이 일부분 개선.
 - 그러나 소요시간은 다른 요소 포함
 - 또한, 데이터 양이 작으면 대역폭 확장은 효과 미미
- 실제 생활 문제로 대체해서 이해
 - 두 지점 사이에 탑승자/물류 이송의 성능 문제

대역폭을 높이는 방법?

- 한번에 보내는 양을 높인다.
 - 대형 운송차, 2층 버스, ...
 - 데이터를 신호화하는 작업의 효율 증대
- 고속 도로/차선을 만들어 사용한다.
 - 고주파 채널 사용
- 채널을 넓힌다.
 - 많은 차선을 확보하고, 많은 차량으로 동시에
 - 넓은 대역 채널 사용
 - 여러 통신 채널을 병렬 사용

소요시간을 줄이는 방법?

- 신호 (전파/광파)의 속도를 높일 수는 없다.
- 지역 처리를 높여서 실질적인 거리를 줄인다.
 - Caching, ...
- 트래픽을 조절해서 큐잉 대기 시간을 줄인다.