



컴퓨터 네트워크 기본 1

📅 강의날짜	@2022/09/19
🕒 작성일시	@2022년 9월 19일 오후 5:53
🕒 편집일시	@2022년 9월 27일 오전 12:39
▼ 분야	네트워크
▼ 공부유형	스터디 그룹
☑ 복습	<input type="checkbox"/>
☰ 태그	

Chapter 1: Introduction

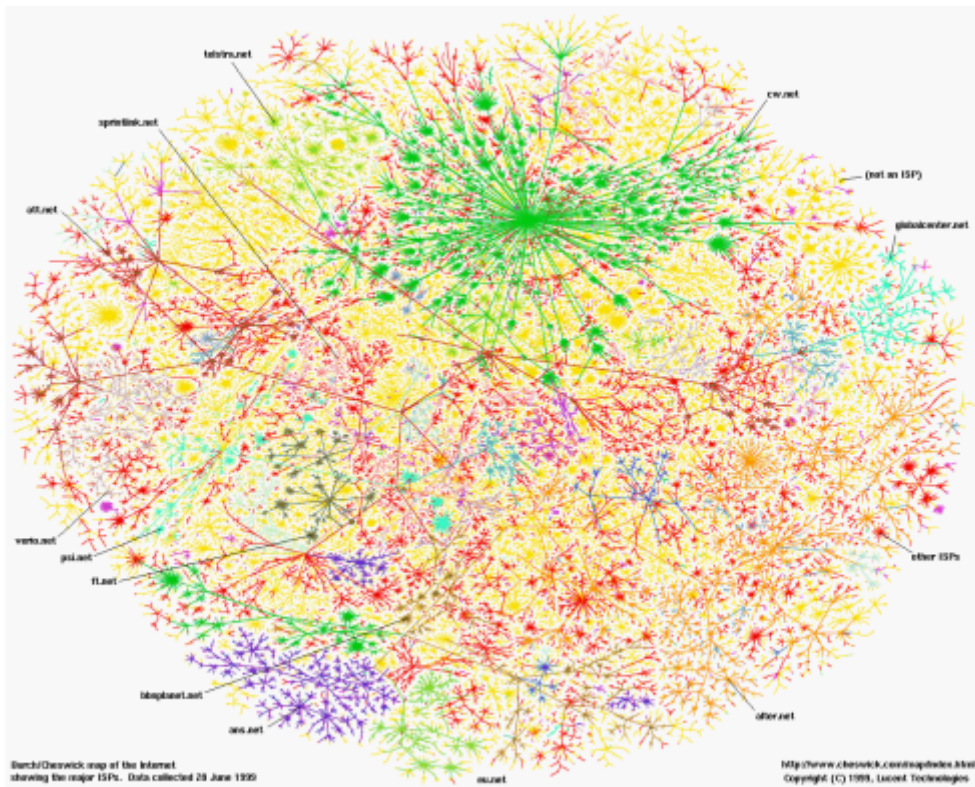
Our goal:

- ❑ get “feel” and terminology
- ❑ more depth, detail *later* in course
- ❑ approach:
 - use Internet as example

Overview:

- ❑ what’s the Internet
- ❑ what’s a protocol?
- ❑ network edge
- ❑ access net, physical media
- ❑ network core
- ❑ Internet/ISP structure
- ❑ performance: loss, delay
- ❑ protocol layers, service models
- ❑ network modeling

Introduction 1-1

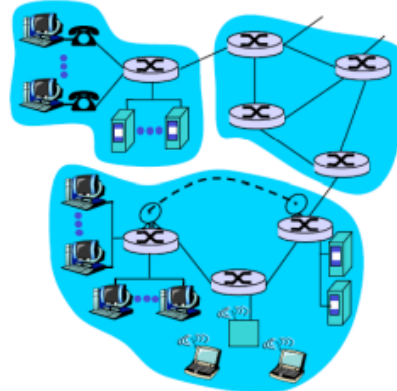


Introduction 1-9

- 인터넷의 개념적인 모습
- 가장자리 : **네트워크 엣지** 우리, 웹 서버
 - edge에는 컴퓨터만 있는 것이 아니고, 네트워크에 들어가기 위한 ‘입구’로서 역할 하는 것이 필요함 : **라우터**
- 가운데 : **네트워크 코어** 라우터 메시지를 전달 받아서 목적지를 향해 전송하는
 - 가장자리에 있는 컴퓨터들끼리 연결이 되어야함
 - 그 연결을 제공해 주는 것이 **네트워크 코어**
- 네트워크 코어로 들어가기 위한 라우터들이 존재
 - 이러한 라우터들이 또 하나의 네트워크를 구성하기도 함 : **access network**
 - access network를 연결시켜주는 다리 연결을 해주는 라우터들의 집합 : **core network**
 - core network 에는 라우터들만 존재

A closer look at network structure:

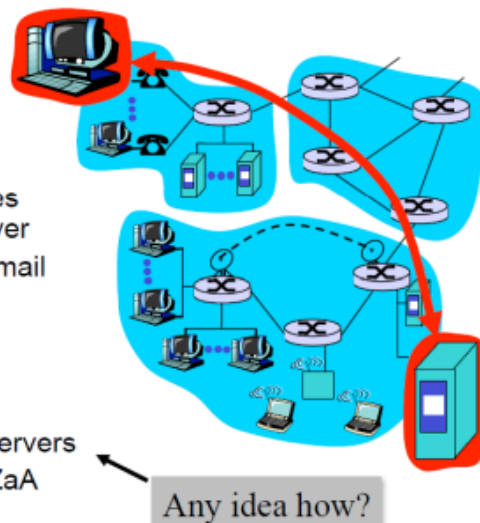
- **network edge:**
applications and hosts
- **network core:**
 - routers
 - network of networks
- **access networks, physical media:**
communication links



Introduction 1-10

The network edge:

- **end systems (hosts):**
 - run application programs
 - e.g. Web, email
- **client/server model**
 - client host requests, receives service from always-on server
 - e.g. Web browser/server; email client/server
- **peer-peer model:**
 - minimal use of dedicated servers
 - e.g. Skype, BitTorrent, KaZaA



Introduction 1-11

- 네트워크 엣지

Network edge: connection-oriented service

Goal: data transfer between end systems

- **Connection:** prepare for data transfer ahead of time
 - Request / Respond
 - **set up "state"** in two communicating hosts
- TCP - Transmission Control Protocol
 - Internet's connection-oriented service

TCP service [RFC 793]

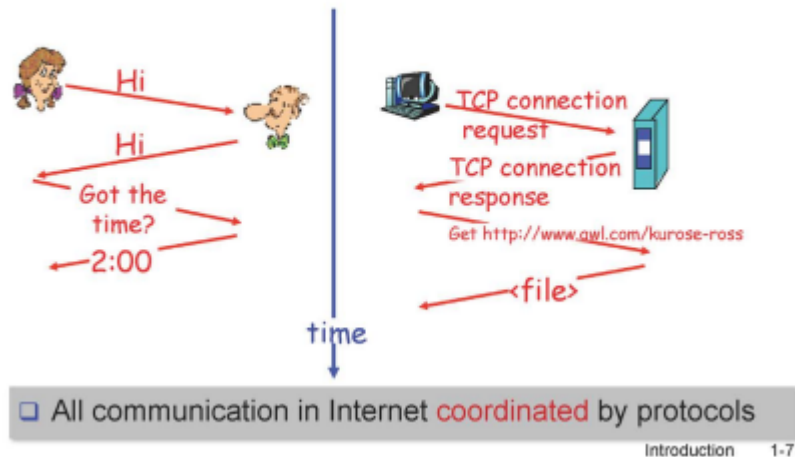
- **reliable, in-order** byte-stream data transfer
 - loss: acknowledgements and retransmissions
- **flow control:**
 - sender won't overwhelm receiver
- **congestion control:**
 - senders "slow down sending rate" when network congested

Introduction 1-12

- 인터넷이 제공하는 기본적인 데이터 통신 서비스 2가지
- TCP : 대부분의 경우에 사용 / 유실되지 않음 / 비쌈
 - reliable, in-order : 순서를 지켜서 전송
 - flow control : 무조건 리시버의 속도에 맞추어 전송
 - congestion control : 네트워크가 혼잡할 때 발신자는 전송 속도를 낮춤
- UDP : 몇개 유실되어도 상관없을때
 - reliable 하지 않음 → 신뢰성이 필요없음
 - ex) real time voice 음성통화
- packet : 편지봉투
 - Host : 데이터를 전송하는 기능
 - Packet : 전송하려는 데이터를 chunk라는 것으로 자른다
 - ⇒ 전송하는 데이터를 일정한 크기의 데이터로 자른 것
 - 서버 상에서 주고받는 데이터는 모두 packet

What's a protocol?

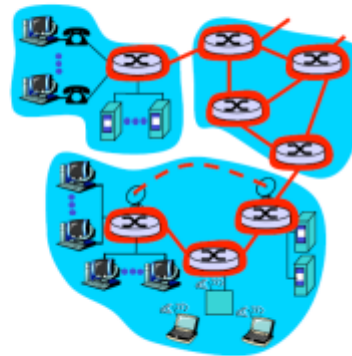
a human protocol and a computer network protocol:



- 프로토콜 : 정말 중요한 메시지를 주고받기 위한 서로의 준비동작

The Network Core

- ☐ mesh of interconnected routers
- ☐ the fundamental question: how is data transferred through net?
 - **circuit switching**: dedicated circuit per call: telephone net
 - **packet-switching**: data sent thru net in discrete "chunks"

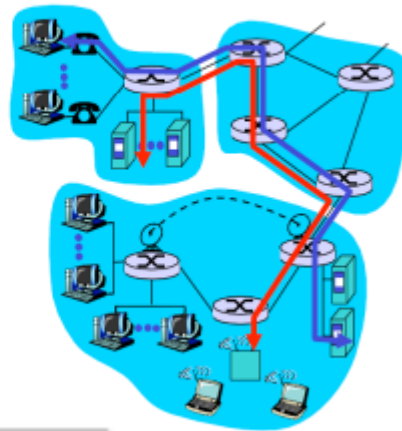


Introduction 1-15

Network Core: Circuit Switching

End-end resources reserved for "call"

- ❑ link bandwidth, switch capacity
- ❑ dedicated resources: no sharing
- ❑ circuit-like (guaranteed) performance
- ❑ call setup required

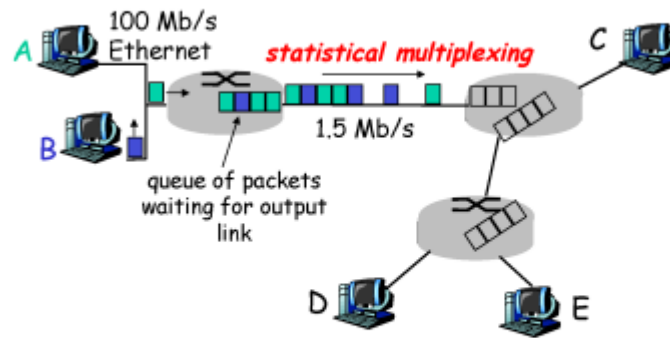


Analogy: When president travels, a CS path set up.

Introduction 1-16

- **switching** : 어디로 갈 지 길을 바꿔주는(정해주는) 역할을 하는 것
- circuit switching : 출발지부터 도착지까지 가는 길을 다 예약해놓음
 - 경로를 설정해놨는데 사용을 안한다면 낭비임
 - 계속해서 데이터를 주고받아야함 ex) 통화
- 1. 공유하지 않아 독립적이다.
- 2. 경로를 미리 지정한다.
- 3. delay와 loss가 완화된다. (라우터에 도착하는 패킷은 미리 지정된 경로로 빠르게 전송되기 때문에)
- 4. Bandwidth를 고정하면 사용자의 수가 정해져있어서 resource가 없으면 연결이 안 된다.

Packet Switching: Statistical Multiplexing



Sequence of A & B packets does not have fixed pattern, shared on demand ➔ **statistical multiplexing**.

TDM: each host gets same slot in revolving TDM frame.

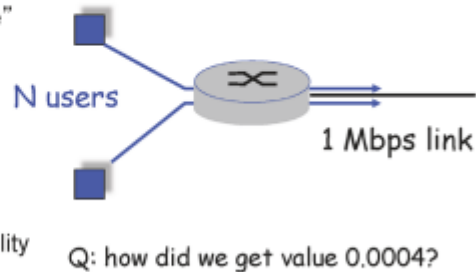
Introduction 1-17

- packet-switching : 패킷단위로 어디로 갈지 길을 정해주는(switching)것
 - 초기 인터넷 디자이너 선택
 - store and forward
 - 라우터가 패킷 하나를 받아서 잠깐 저장(store)후, 어디로 보낼지 판단하고 보내는 것(forward) 이 과정에서 걸리는 시간은 지연시간(delay time)이라고 함
1. 라우터에 들어오는 패킷의 header에는 목적지 주소가 있다.
 2. 이 주소에 대한 경로 정보를 미리 정하는 routing algorithm에 의해 각 라우터에 local forwarding table이 만들어진다.
 3. 이후 라우터에 패킷이 왔을 때, 패킷의 목적지 주소와 table의 header value를 보고 해당하는 output link로 보낸다.

Packet switching versus circuit switching

Packet switching allows more users to use network!

- 1 Mb/s link
- each user:
 - 100 kb/s when "active"
 - active 10% of time
- circuit-switching:
 - 10 users
- packet switching:
 - with 35 users, probability > 10 active less than .0004

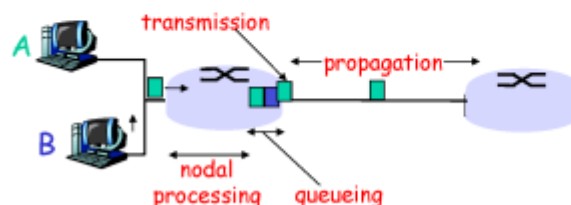


Introduction 1-19

- 1Mbps : 크면 클수록 좋음
 - circuit switching : 최대 10명의 user까지 지원가능
 - packet switching : 몇 명까지 가능한지 그런 개념이 없음 제약이 없음
 - 우리는 사실상 클릭보다 보는 시간이 많음 따라서 우리가 이용하는 인터넷 방식에는 packet switching이 더 알맞음
 - 보다 많은 사람에게 제공 가능

Four sources of packet delay

- 1. nodal processing:
 - check bit errors
 - determine output link
- 2. queueing
 - time waiting at output link for transmission
 - depends on congestion level of router



Introduction 1-32

1. **nodal processing** : bit error 체크 / 패키지 검사(프로세싱 딜레이) 라우터에 내보냄

- a. 라우터 성능 개선
- 2. **queueing delay** : 나가는 속도보다 들어오는 속도가 빠르면 bucket이나 queue같은 임시 저장장치가 필요함
 - a. 제일 골치 아픔 : 개선 불가능
 - b. 추석 때 하이패스 같은 것
 - c. 사람들이 몰려서 queue가 터지면 유실이 생길 수 있음
- 3. **transmission delay** : 패키지 : 비트의 집합 / 첫번째비트가 나가는 순간부터 마지막 비트가 나가는 순간까지 딜레이
 - a. 케이블 공사
- 4. **propagation delay** : 물리적인 선을 따라 신호가 가는데 걸리는 시간

Delay in packet-switched networks

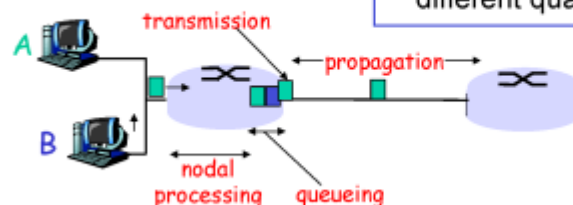
3. Transmission delay:

- R = link bandwidth (bps)
- L = packet length (bits)
- time to send bits into link = L/R

4. Propagation delay:

- d = length of physical link
- s = propagation speed in medium ($\sim 2 \times 10^8$ m/sec)
- propagation delay = d/s

Note: s and R are very different quantities!



Introduction 1-33

Packet-switching보다 Circuit-switching가 나은 점

Circuit-switching는 경로를 지정하여 일정한 전송률을 보장

전화통화 같은 품질보장이 되어야 하고 계속해서 데이터를 주고받아야 하는 서비스에선, 가변적인 Packet-switching보다 일정한 Circuit-switching를 이용하는 것이 좋다.

1. Queueing delay가 완화된다
 - a. source와 destination의 경로를 정해버리면 바로바로 보낼 수 있기에 queueing delay가 상당히 완화
2. 장시간 긴 데이터를 보낼 때 더 낫다

- a. **Packet-switching**는 여러 **sources**에서 데이터가 패킷으로 쪼개져 날라가기 때문에, **장시간 긴 데이터를 보낸다면 많은 패킷을 보내므로**, 데이터 전송이 지연되거나 전송을 완전히 보장받을 수 없다.

3. 제한된 시간에 빠르게 보낼 때 더 낫다.

- a. 위와 같은 이유로, 패킷같은 일정한 전송을 보장받지 못하는 상황에서, 시간을 제한한다면 신뢰있는 전송을 보장받지 못할 것이다.

Circuit-switching보다 Packet-switching가 나은 점

- 주고받는 데이터를 패킷단위로 자르고 그 패킷들을 섞어서 보냄으로써 데이터가 전달되는 길을 효율적으로 이용
1. link의 낭비가 줄어든다
 2. 더 많은 사용자가 사용가능하다