

스마트 네트워크 및 실습 (LD02700)

김준철

정보시스템공학과

greensday@sungshin.ac.kr

2주차 강의

	주차	강의 목차
9.4	1	과목소개 / 컴퓨터 네트워크와 인터넷 1
9.11	2	컴퓨터 네트워크와 인터넷 2
9.18	3	컴퓨터 네트워크와 인터넷 3, 애플리케이션 계층 1
9.25	4	애플리케이션 계층 2
휴강 10.2 (추석) 한글날-정상수업	5	트랜스포트 계층 1
10.9	6	트랜스포트 계층 2
10.16	7	트랜스포트 계층 3
10.23	8	중간고사
학교 입시일 - 정상수업, 시험	9	네트워크 계층 1 – data plane1
11.6	10	네트워크 계층 2 – data plane2
11.13	11	네트워크 계층 3 – control plane1
11.20	12	네트워크 계층 4 – control plane2
11.27	13	물리계층, 링크계층
12.4	14	무선이동통신 네트워크
12.11	15	기말고사
12.18		

Smart System Network

Chapter 1

Network and Internet -2

1.1 what is the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

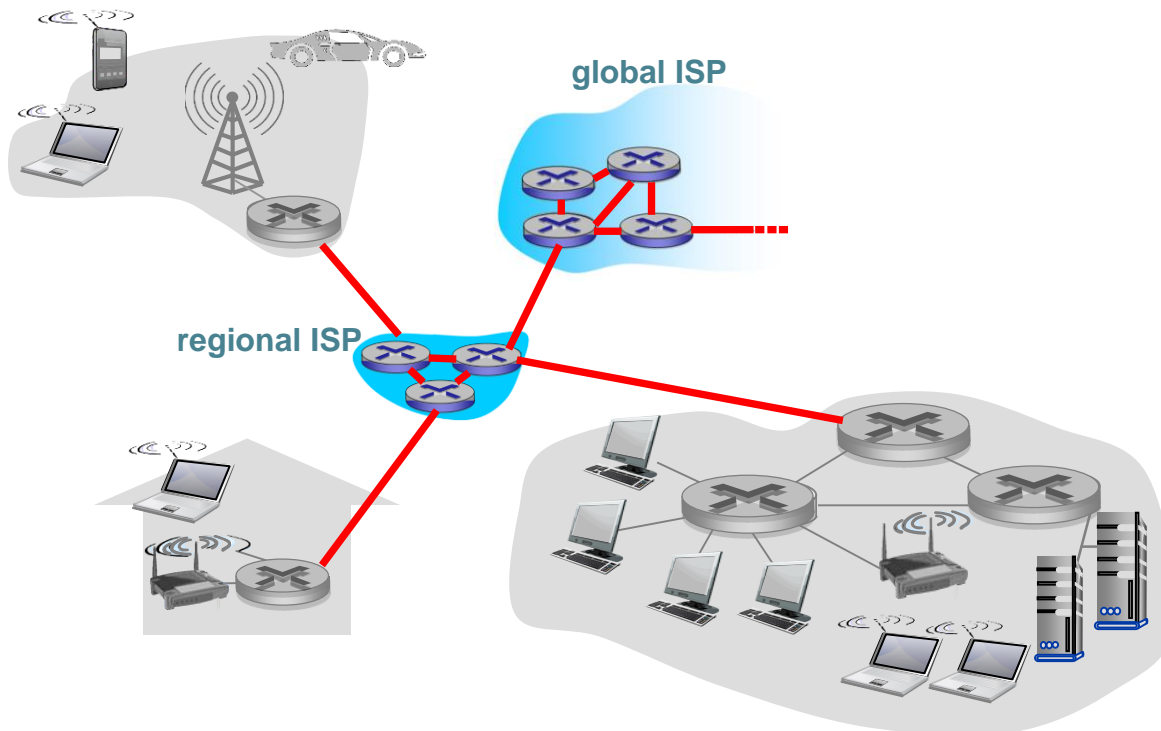
1.5 protocol layers, service models

1.6 networks under attack: security

1.7 history

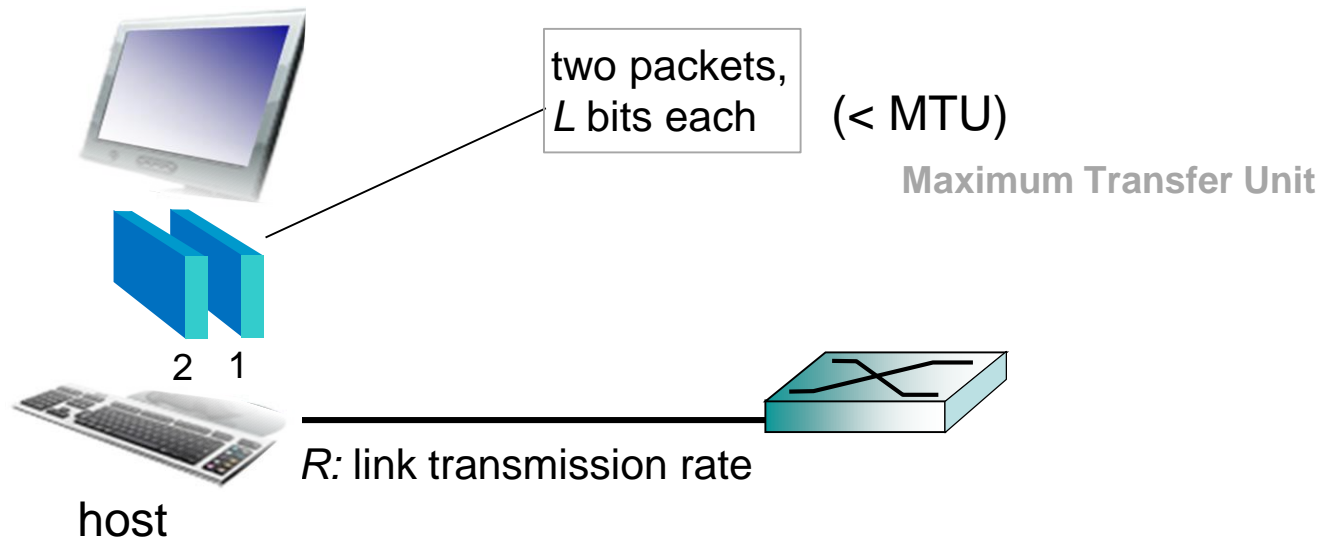
The network core

- The network core: interconnected router들의 mesh



Host : sends packets of data

- Host는 application program을 실행하고 있음
 - App.들은 **message**를 주고 받을 수 있게 host(computer)에 요청
- * host sending function:
 - data - L bits (작은 chunks, **packets**) 길이로 나누어서 access network로 연결되는 link를 통해 전송
 - link transmission rate R (bit per second)로 packet을 전송
(link transmission rate = link capacity = link bandwidth)



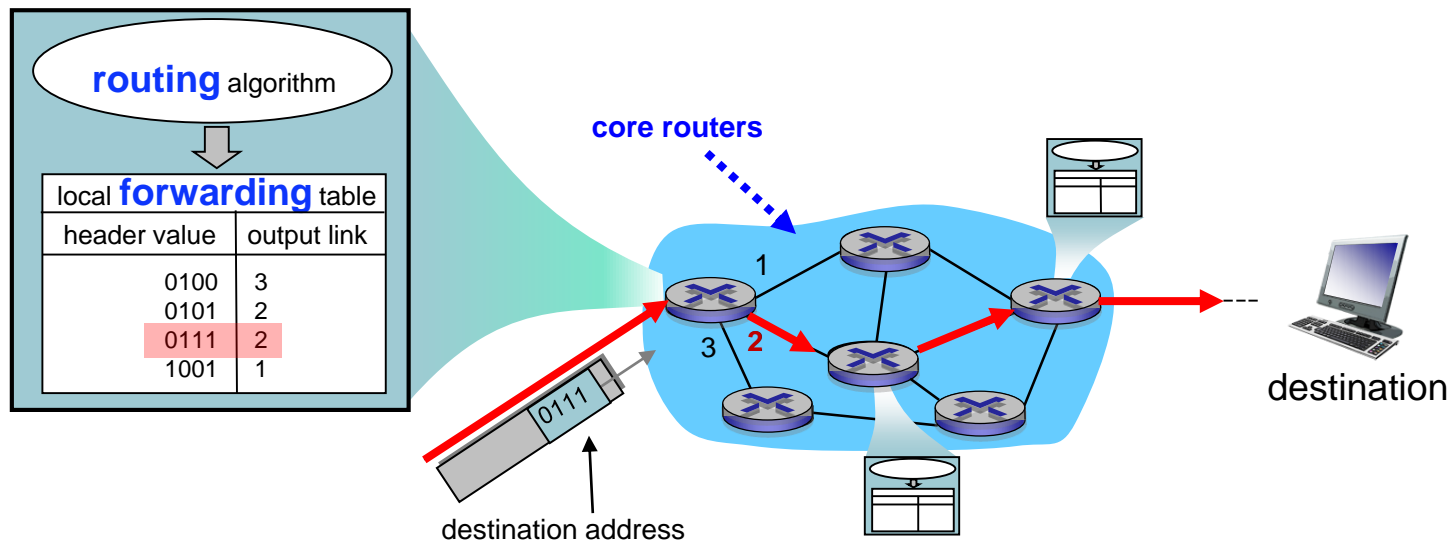
Two key network-core functions

■ Core network의 기능(function)

- source에서 destination으로 packet을 전송할 때, (end host간 packet을 전송 할 때) 하나의 router에서 다음 router로 forwarding을 해주는 역할
- 각 packet은 full link capacity로 전송

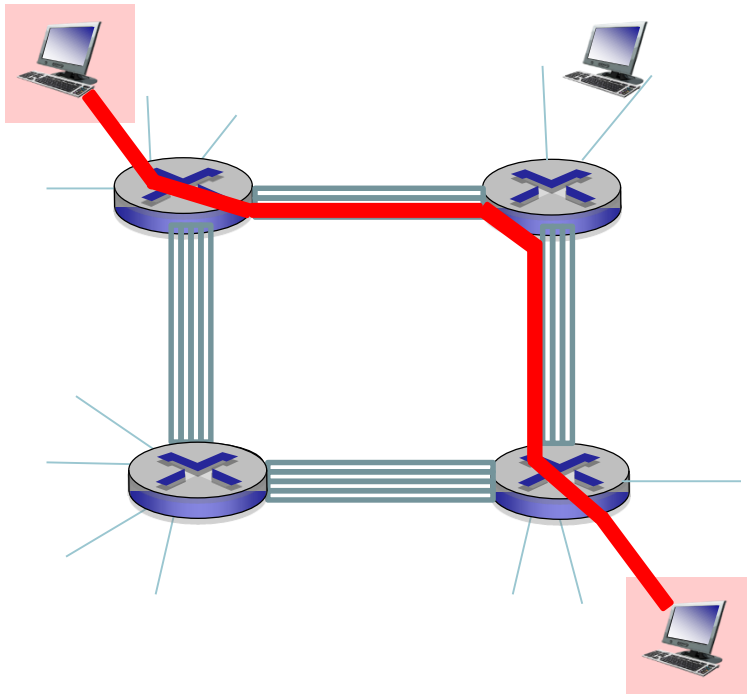
1. routing: routing algorithms을 통해서 source ~ destination의 경로를 계산함

2. forwarding: router의 input에서 받은 packet들을 적절한 output을 선정하여 다음 router로 이동 (forwarding table을 활용함)

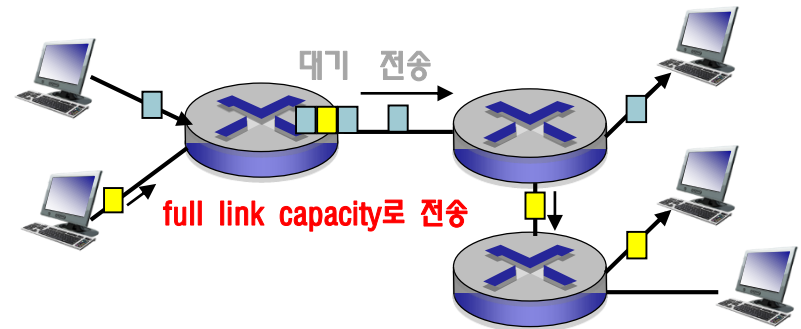


circuit switching vs. packet switching-1

- Circuit switching(회선교환)
 - data를 보내기 전에 path(경로)를 설정 (해당 path에 네트워크 자원을 할당)
 - 설정된 path를 통해 모든 data가 이동
 - traditional telephone networks



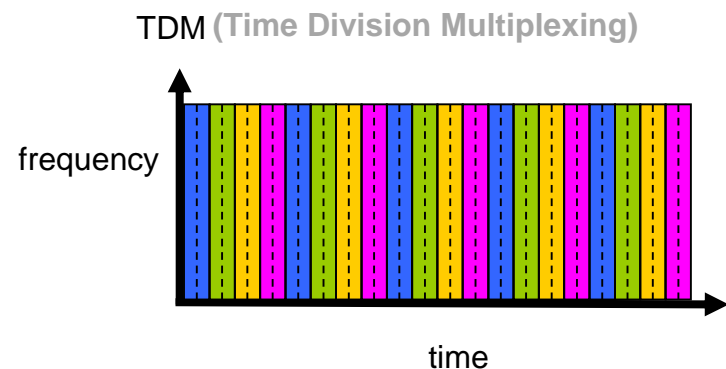
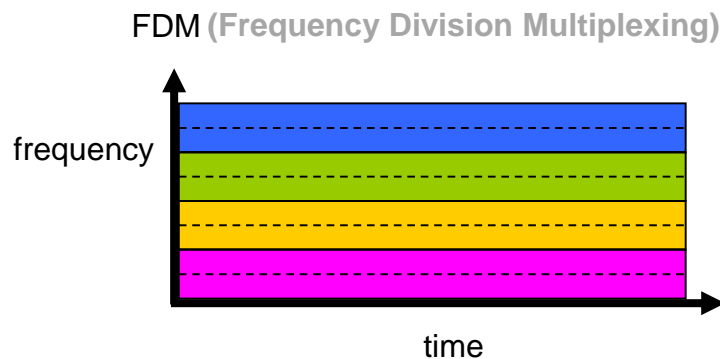
- Packet switching(패킷교환)
 - 모든 data는 작은 packet들로 나뉨 (packet은 각각의 destination address를 가짐)
 - 각각의 packet은 독립적으로 다루어 짐 (handled independently)



Circuit switching

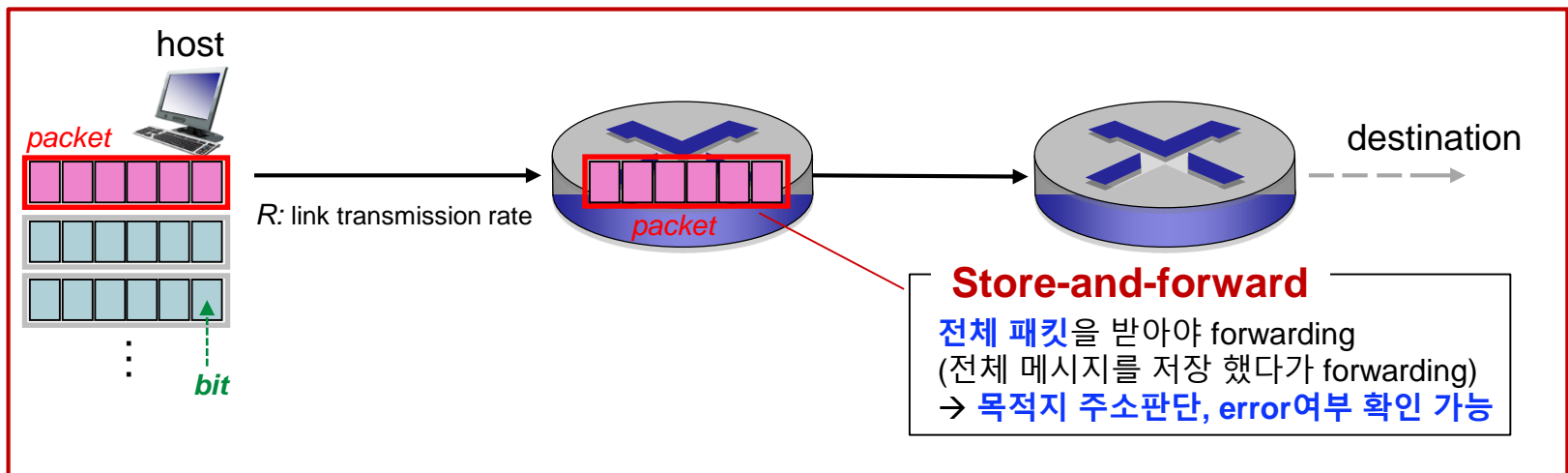
- 각각의 call에 네트워크 자원(resource)이 할당됨
 - Data를 보내기 전 전제 이동 경로가 설정됨 (전제 이동 경로의 resource를 점유)
 - Call setup 과정이 필요함 (resource reservation)
 - 한 사용자가 resource를 점유하면 해당 call에서 data전달이 없어도 다른 사용자는 사용이 불가함 (**no sharing**)
 - 유선 전화망(**telephone networks**)에서 사용되었음
 - Channel allocation methods : FDM, TDM

Example: 4 users ■■■■



Packet switching

- No call setup, no resource reservation
 - full link capacity
 - User의 전송 data 길이가 길면 packet으로 잘라서 전송
 - 각 packet은 목적지 주소가 명시되어야 함
 - Store and forwarding (전체 packet이 router에 도착해야 다음 router로 forward)
 - 각 router는 packet의 각 bit를 저장했다가, 전체 packet이 도착하면 processing 후 다음 router로 전송
 - 다음 router로 packet을 전송 하려고 할 때, 다른 packet이 먼저 나가려고 줄 서 있으면, 대기해야 함



Packet switching

- packet은 link의 **최대전송속도**와 같은 속도로 전송됨 (**full link capacity로 전송**)
- packet을 전송(transmission)하는데 걸리는 시간

packet transmission delay = time needed to transmit L -bit packet into link

(하나의 hop당 지연시간)

$$= \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

- packet 길이 : L bits

- link transmission rate : R (bps, bit per second)



circuit switching vs. packet switching-2

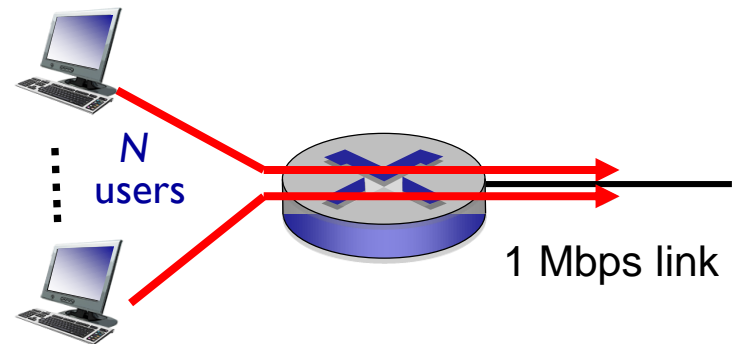
packet switching allows more users to use network!

example:

1 Mb/s link (1000 kb/s link)

35명의 사용자(user)가 있음:

- User 당 **active** 상태에서 100 kb/s 필요
- **active** 시간은 전체시간의 10%



switching 방법에 따라 몇 명의 user가 network를 사용할 수 있는가?

- *circuit-switching*: 10 users
- *packet switching*:
 - 35명의 users중 동시에 10명이 넘게(초과로) active할 확률
→ less than .0004 = 0.04%

packet switching 이 3.5배 더 많은 user가 인터넷을 사용 할 수 있음



Q: what happens if > 35 users ?

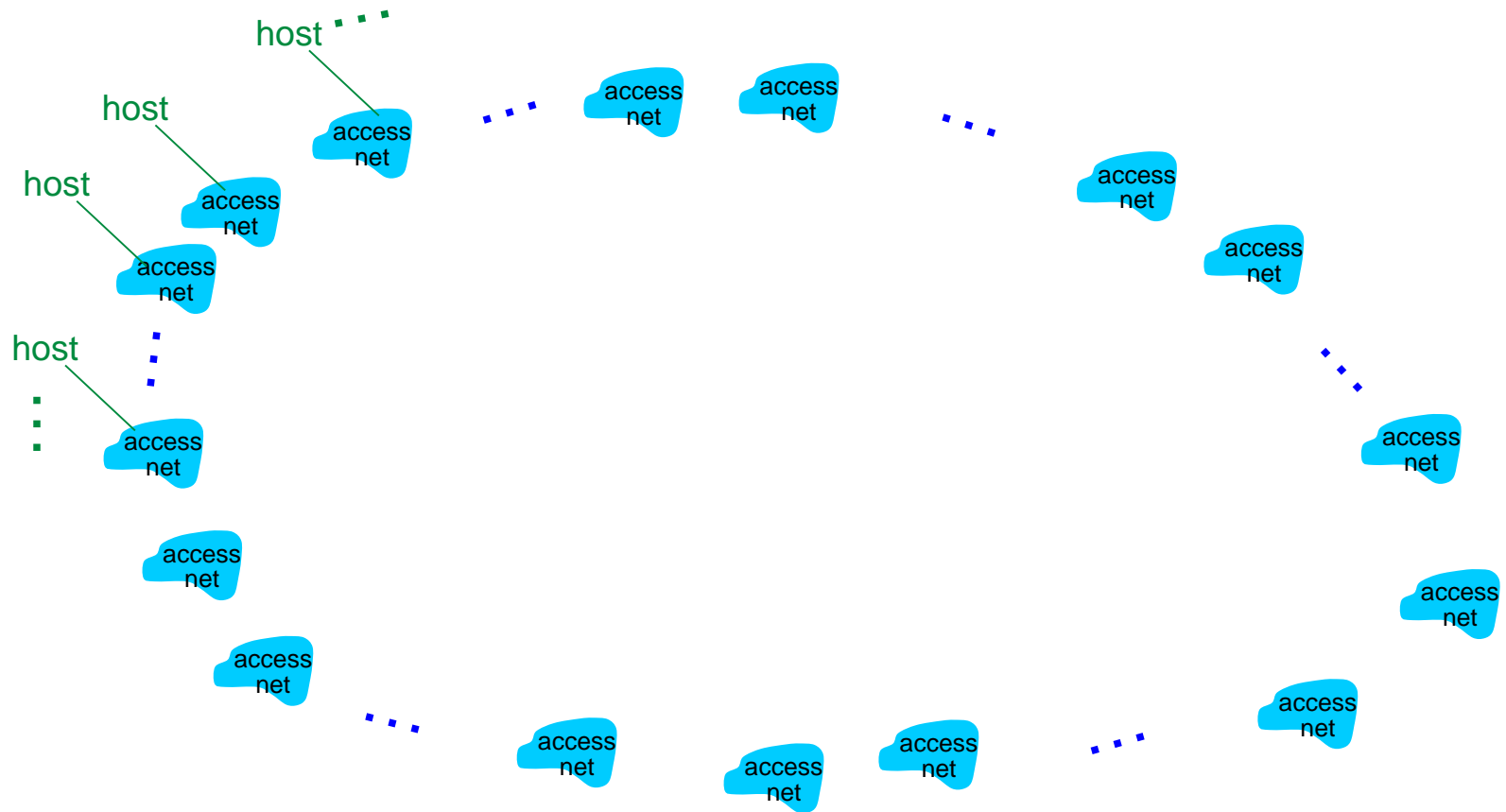
*QOS(Quality Of Service)*가 점점 떨어짐

Internet structure: network of networks-1

- **Access network**들은 서로 연결되어야 함.
 - 연결이 되어야 host들은 서로 packet들을 보낼 수 있음
- **network of networks**는 매우 복잡함
 - Network의 발전은 **economics**과 **national policies**에 의해 주도됨
- End system들은 **ISP**들을 통해서 internet에 연결됨
 - Internet Service Provider interconnect
 - residential, company and university 등은 ISP를 통해서 internet에 연결됨

Internet structure: network of networks-2

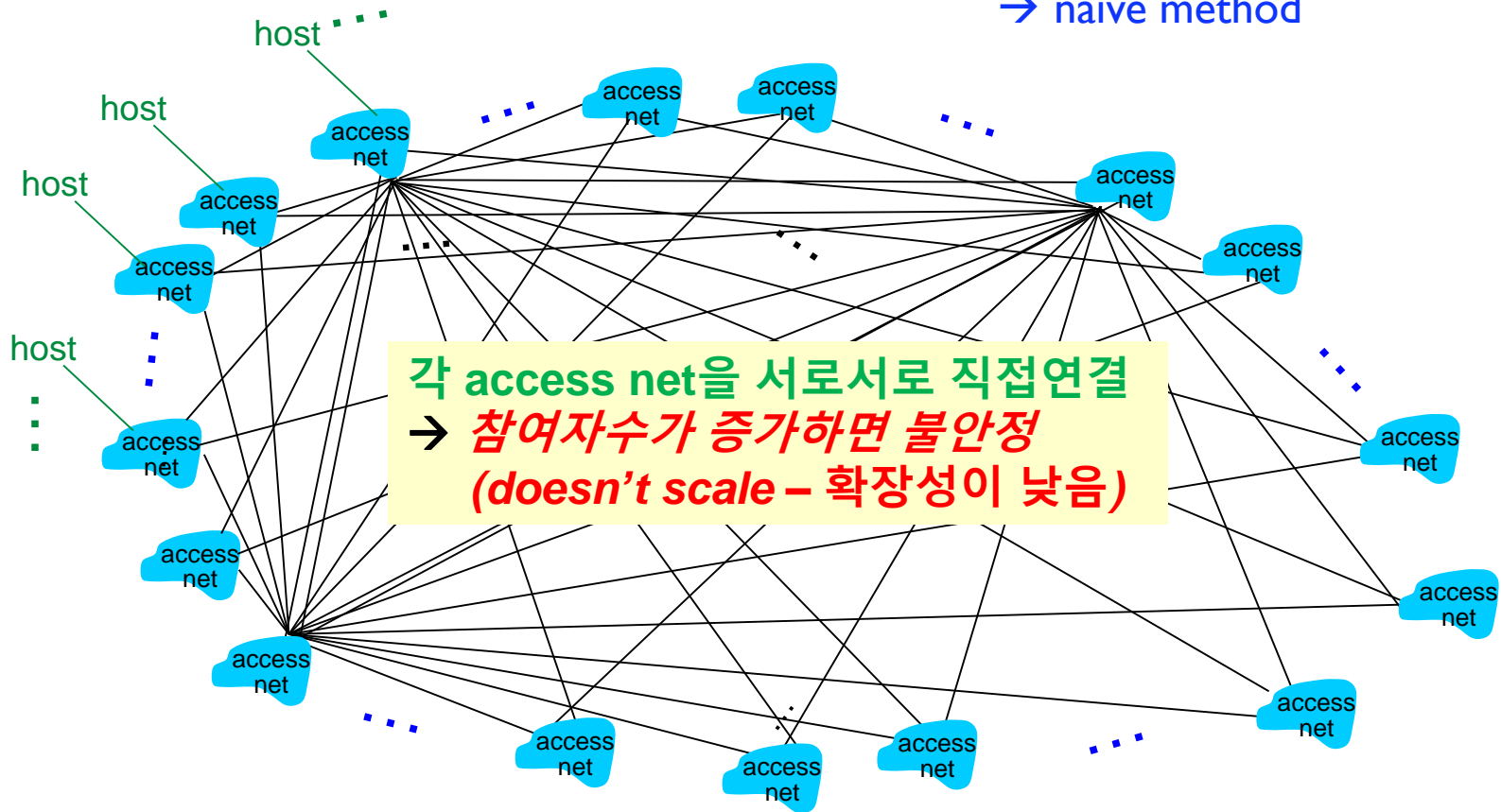
Question: 매우 많은 수의 access net들을 어떻게 서로 연결될 수 있을까?



Internet structure: network of networks-3

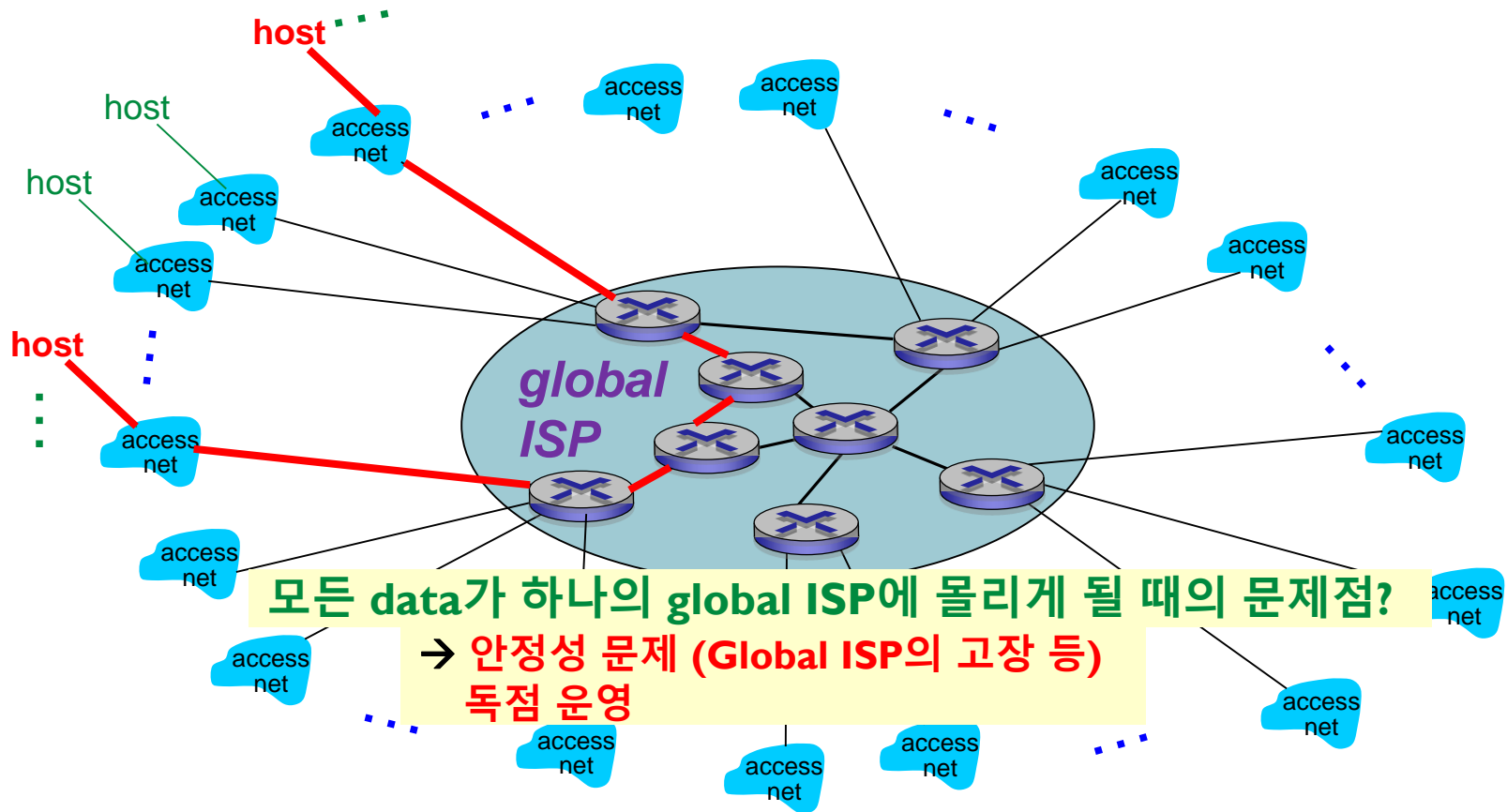
Option: 모든 access net들을 서로서로 연결하면 어떻게 될까?

→ naive method



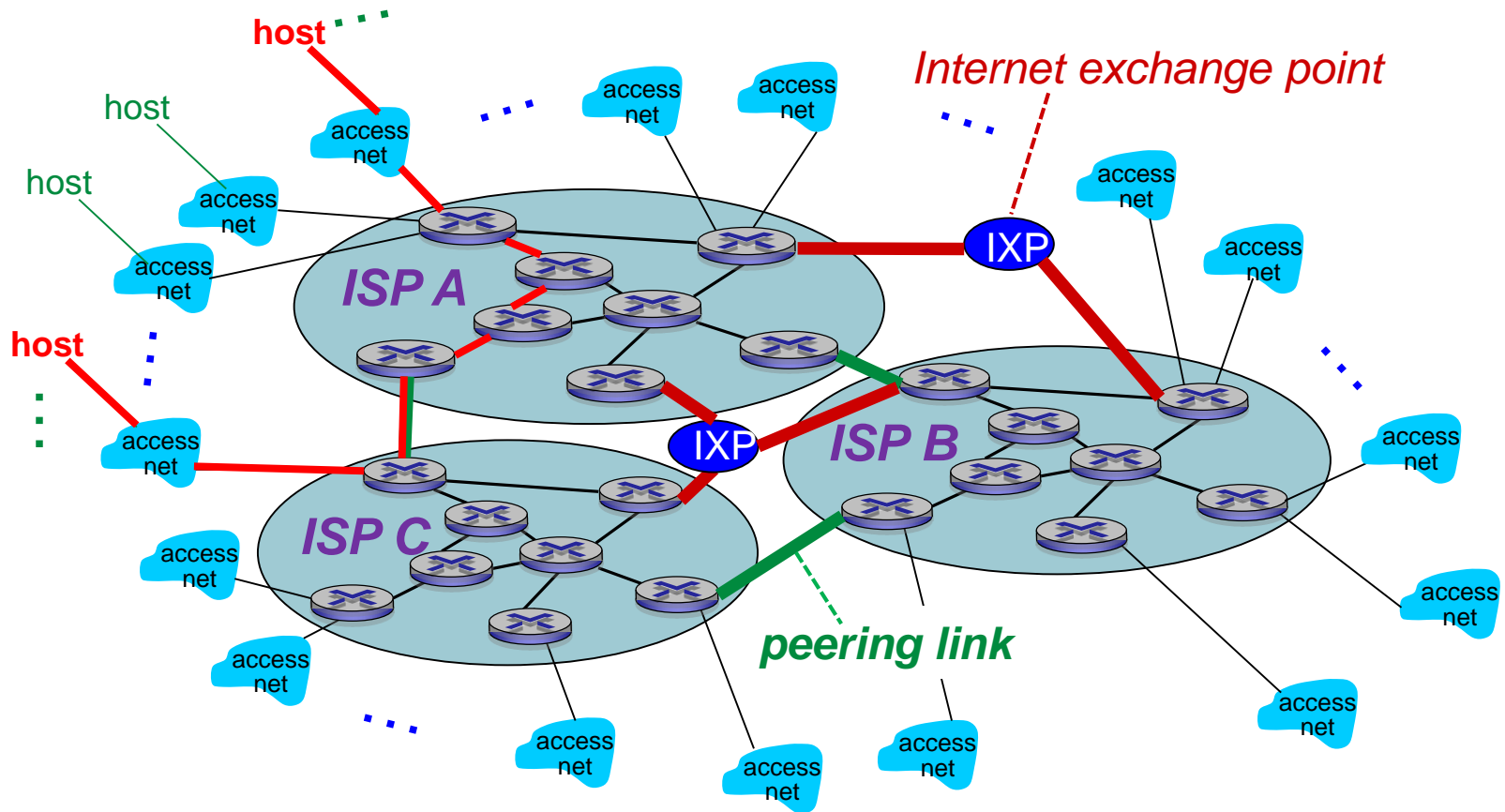
Internet structure: network of networks-4

Option: 각각의 access net을 global ISP에 연결하면 어떻게 될까?



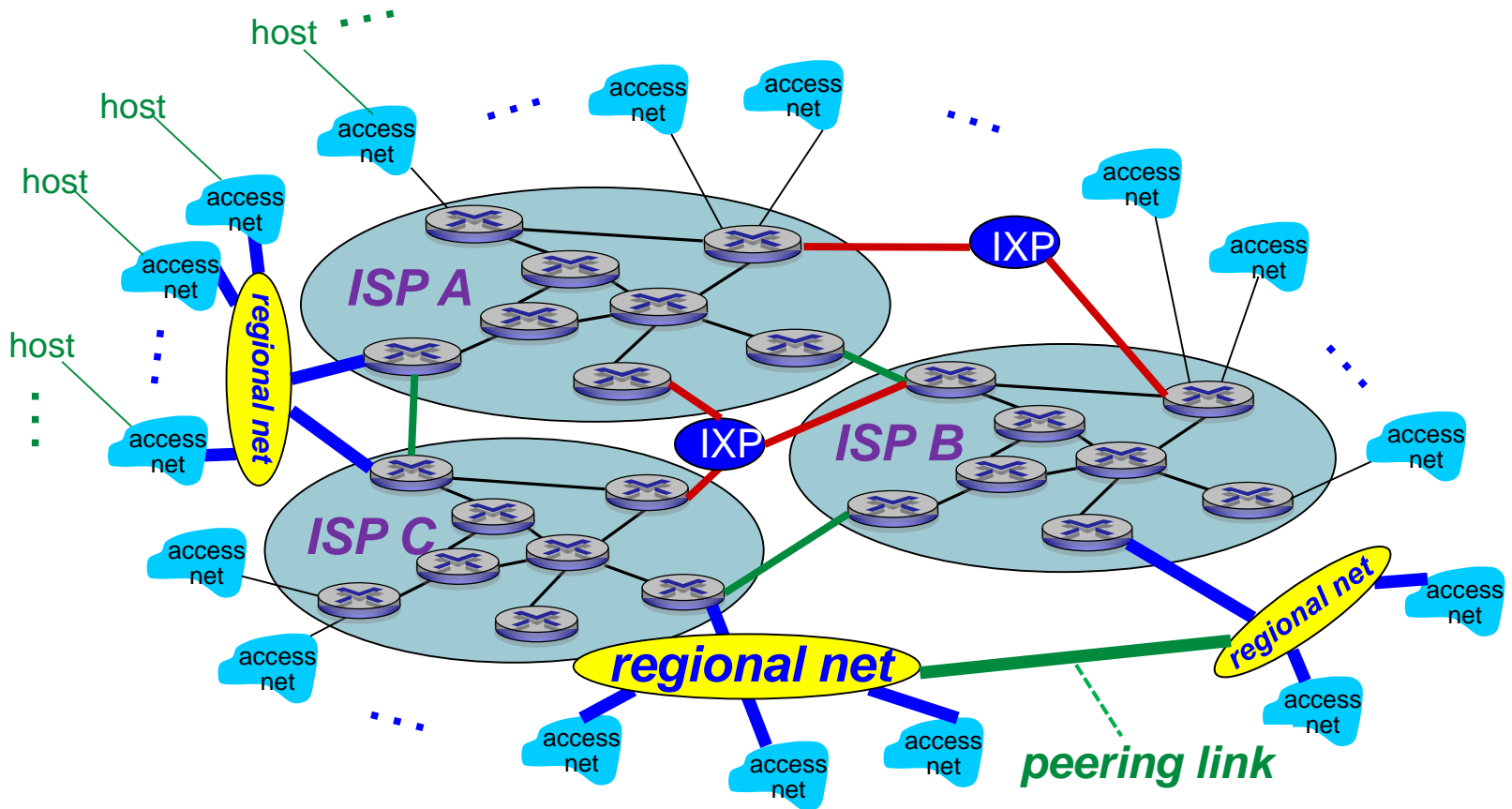
Internet structure: network of networks-5

경쟁업체(competitors)들의 발생 : 경쟁 업체의 network과도 연결이 되어야 함



Internet structure: network of networks-6

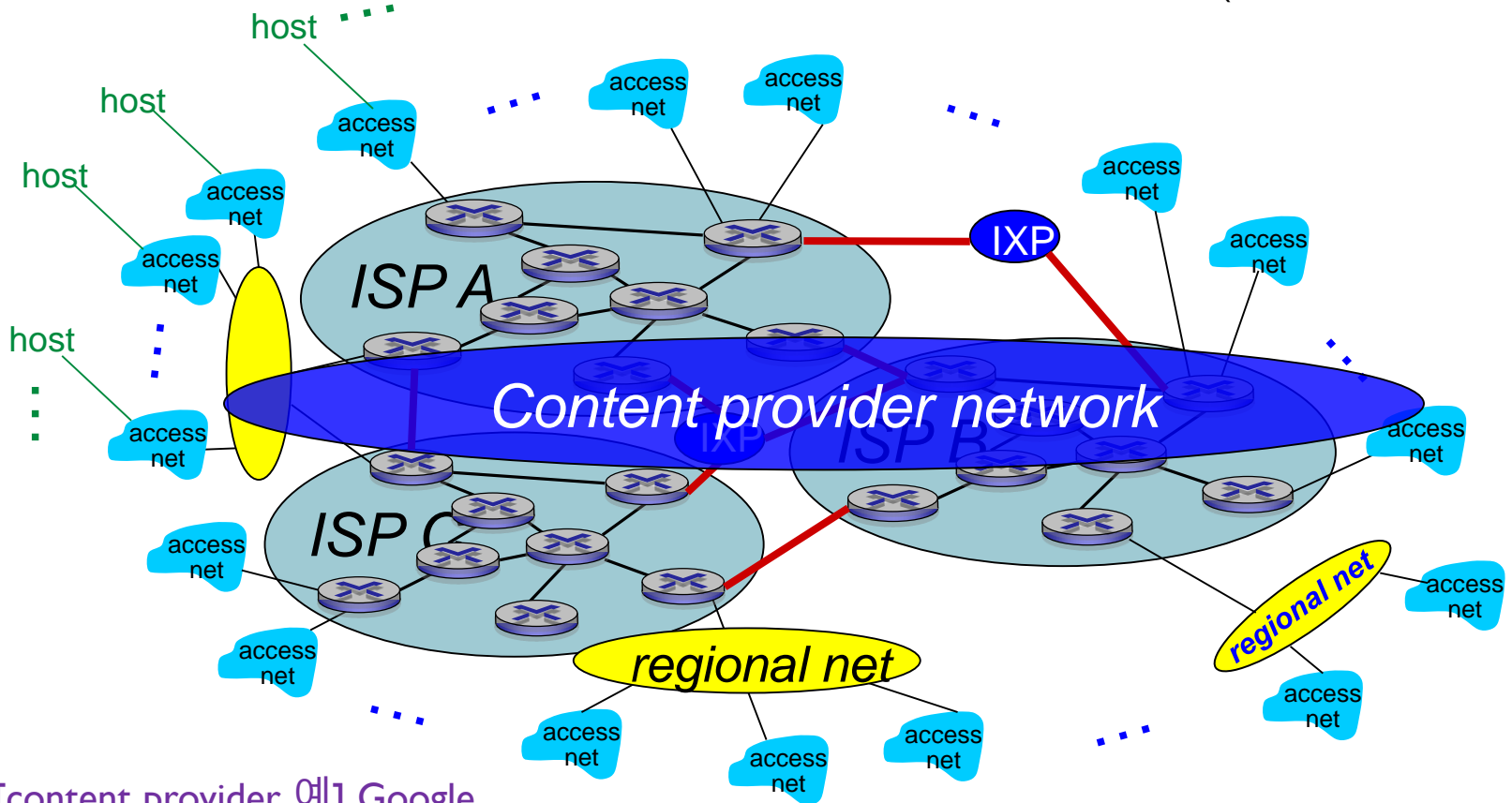
지역에서 access nets 과 ISPs간의 연결 서비스를 하는 regional ISP(network)업체도 발생



regional net
: SK Broadband, LG U+ 등

Internet structure: network of networks-7

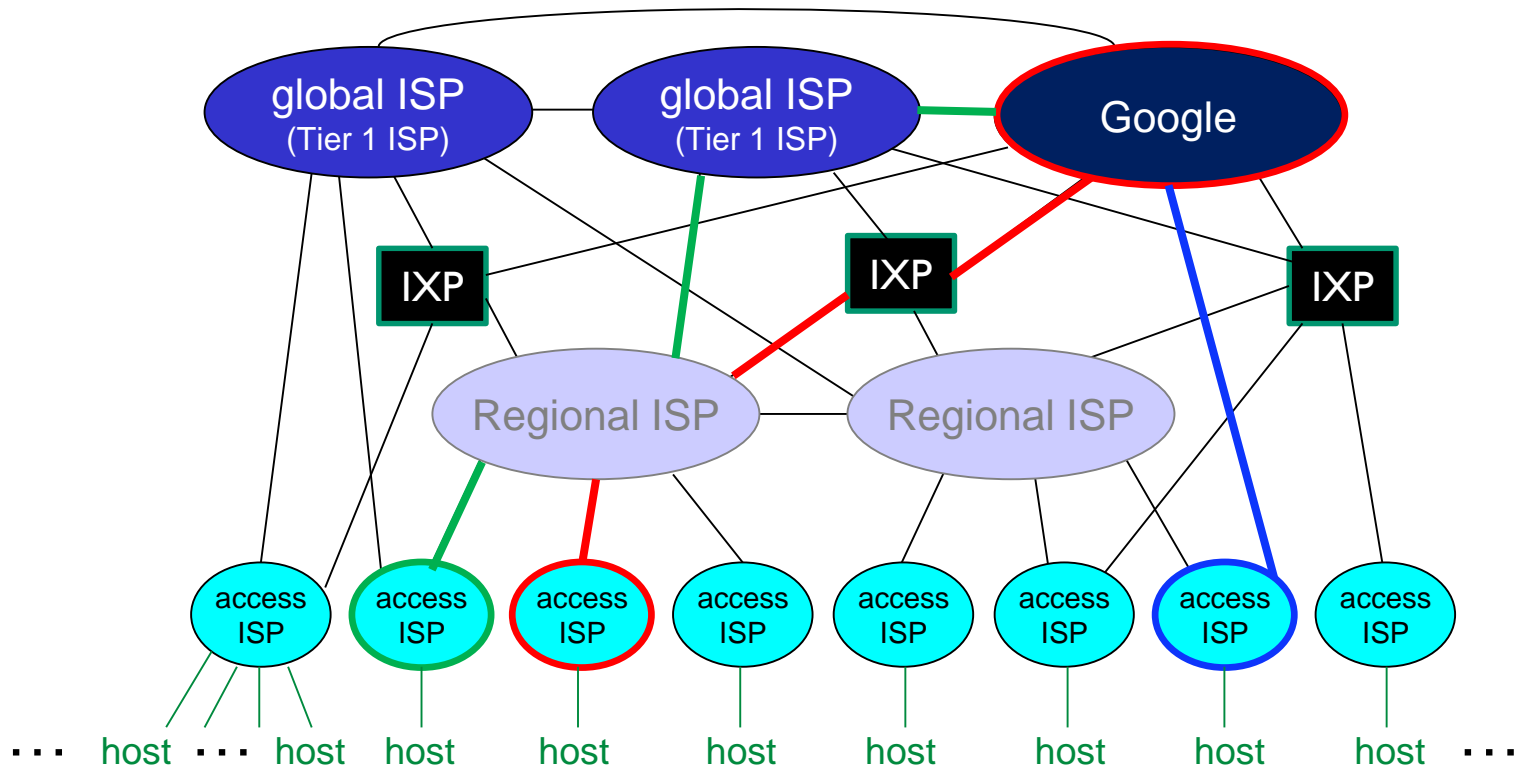
content provider networks (Google, Microsoft, Akamai 등) : services, content 등을 제공
→ 전 세계 각지에 **data center**를 두고 그들의 network로 운영 (**데이터 동기화** 필요)



[content provider 예] Google

- 데이터 전송량이 엄청나게 많음(데이터 동기화 등)
- 상업용 네트워크를 사용 하게 되면 엄청난 비용이 발생

Internet structure: network of networks-8

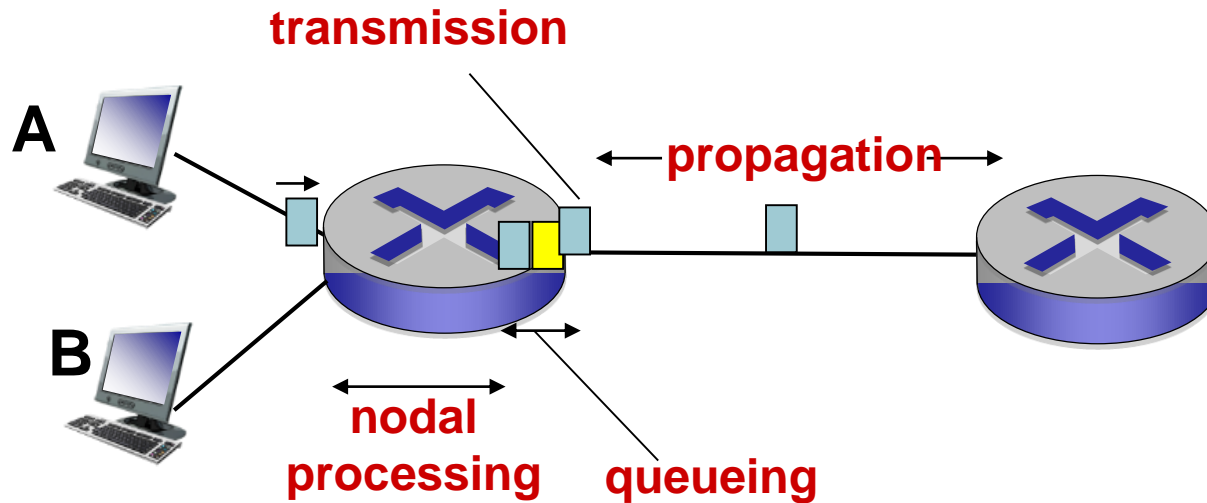


Evaluation Metrics – delay, packet loss, through

(성능)평가척도

- Delay (지연시간)
 - Source에서 destination까지 packet을 전달하는데 걸리는 평균 시간
- Packet loss (손실률)
 - Data를 작은 packet들로 나누어 전송할 때, 전달 중 분실되는 packet이 얼마나 되는 지로 결정됨 (PLR, packet loss rate)
cf. PDR (packet delivery rate)
- Throughput(전송률)
 - 단위 시간(unit time) 동안 전달할 수 있는 data의 이동 총량(traffic의 총량)

4 sources of packet delay



* packet이 하나의 hop을 지나갈 때 걸리는 delay

$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

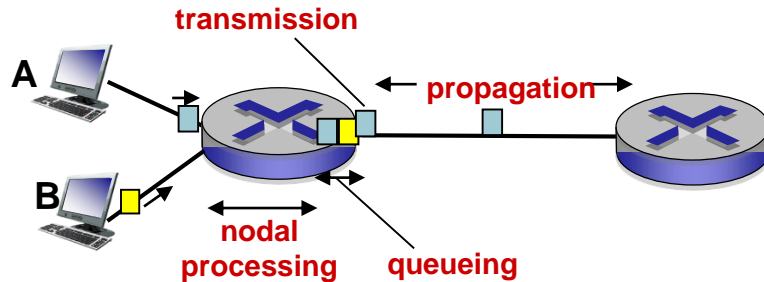
d_{proc} : nodal processing delay

d_{queue} : queueing delay

d_{trans} : transmission delay

d_{prop} : propagation delay

Four sources of packet delay



$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

1. d_{proc} : nodal processing delay (성능 좋은 router를 사용해서 개선)

- destination(목적지) 확인과 bit error 검사하는데 걸리는 시간(delay)
: packet이 destination(주소)으로 가기 위해서 output link를 결정

2. d_{queue} : queueing delay (network 사용자 수에 따라 달라짐) → 가변적인 delay

- router를 나가려고 할 때, 먼저 도착한 packet이 있으면, queue를 통하여 기다리는 대기시간 (delay)

3. d_{trans} : transmission delay (output link의 Bandwidth가 결정) (전송 지연)

- packet의 첫번째 bit부터 마지막 bit까지 router를 나갈 때 까지의 시간 delay ($d_{\text{trans}} = L/R$)

L: packet length (bits)
R: link bandwidth (bps)

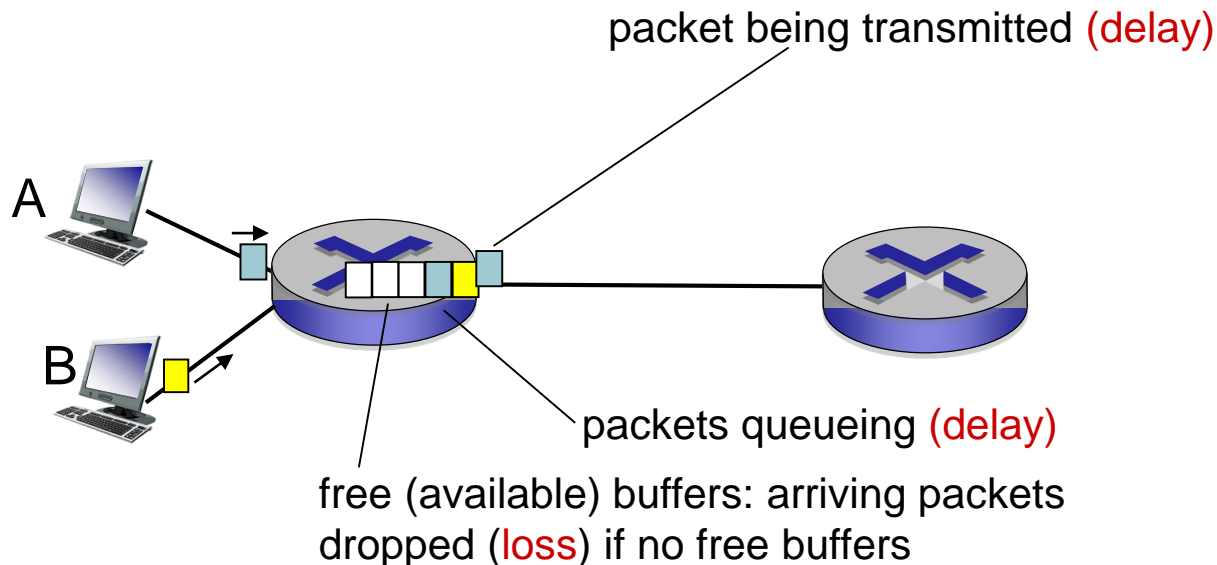
4. d_{prop} : propagation delay (network 사용자 수[data량]에 따라 달라짐) (전파 지연)

- packet의 마지막 bit가 link에 올라와서 다음 router에 도달 할 때까지의 시간 delay ($d_{\text{prop}} = d/s$)

d: length of physical link
s: signal speed

Queueing delay

- packets **queue** in router buffers
 - packet arrival rate이 일시적으로 output link capacity를 넘을 때,
→ packets은 queue에서 대기하고 차례가 오면 output link로 나감
 - queue가 다 채워진 후 도착한 packet → dropped (손실)



Queueing delay - traffic intensity

- R : link bandwidth [bps] → bit가 router(queue)에서 나와서 link에 실려 전송되는 속도 (bit / sec)
- L : packet length [bits / packet] → packet당 bit 수
- a : **average** packet arrival rate → 평균적으로 router에 도착하는 packet 수 [packet / sec]

- ❖ 초당 router(Queue)에 도착하는 평균 bit 수 구하기
(초당 유입되는 traffic(or data) 양)

$$a \text{ [packet / sec]} \times L \text{ [bit / packet]} = La \text{ [bit/sec]}$$

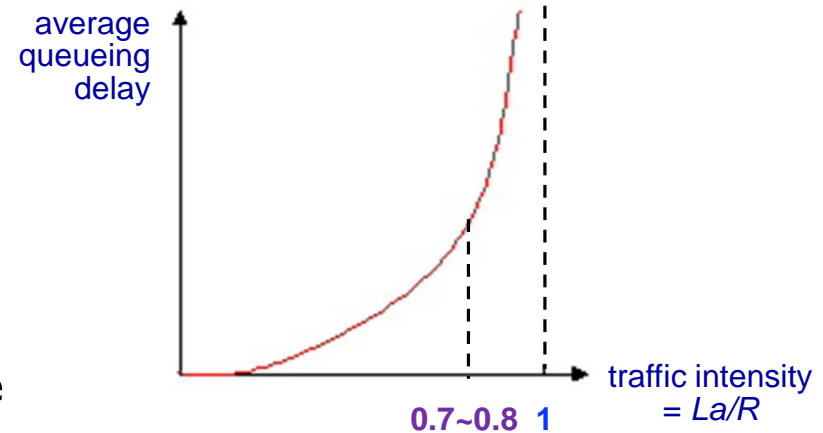
- router에서 bit가 link에 실리는 속도(나가는 속도)는 R [bit/sec] 인데, 평균적으로 router에 유입되는 bit는 1초당 La [bit] 이므로, La/R 을 traffic intensity(트래픽 강도)로 정의하여 사용한다.

$$\text{traffic intensity} = \frac{La}{R}$$

Queueing delay - traffic intensity

$$\text{traffic intensity} = \frac{La}{R}$$

- $La/R \sim 0$: avg. queueing delay - small
- $La/R \rightarrow 1$: avg. queueing delay - large
- $La/R \sim 1$: avg. queueing delay - infinite



$La/R \sim 0$

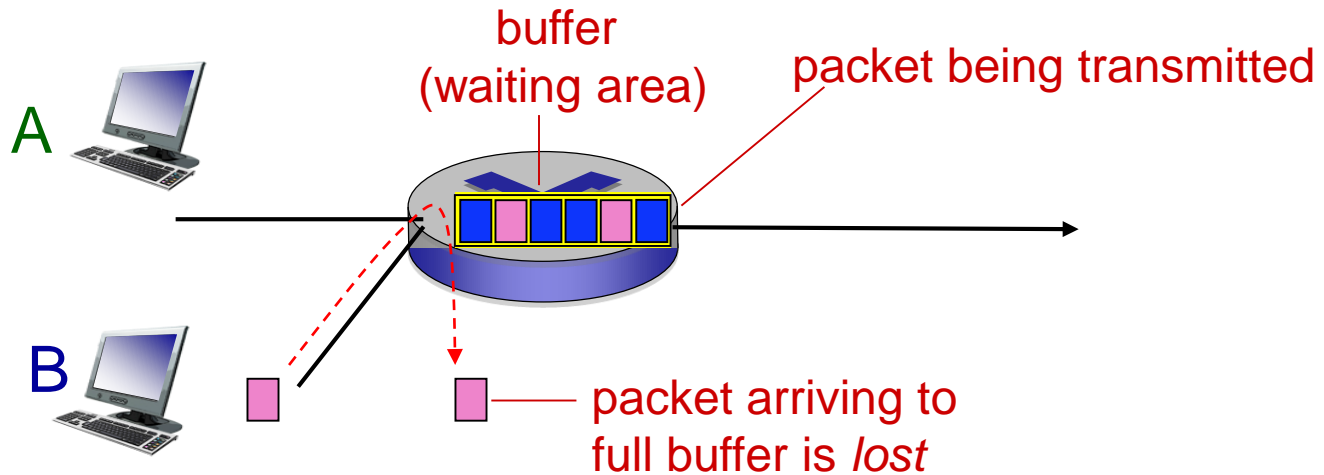


$La/R \rightarrow 1$

* La 는(traffic이 아주 많을 때와 없을 때의) 평균값이므로 La/R 값이 커질 수록 queueing delay가 급격하게 커진다.

Packet loss

- Packet loss는 대부분 router에서 발생
 - queue (aka buffer) 의 capacity보다 많은 data들이 동시에 도착할 때 queue에 저장 못하고 넘치는 data들은 dropped (aka lost)
 - lost packet은 host에서 다시 전송 → 사용자는 delay를 느낌
네트워크는 resource 낭비

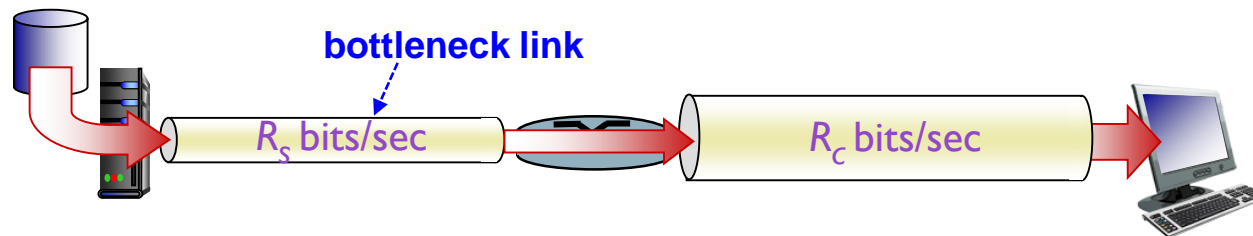


Throughput

■ *throughput*:

- sender/receiver 사이의 bits의 전달률 [rate(bits/time unit)]
- 단위 시간(unit time) 동안 전달할 수 있는 traffic의 총량 (data의 이동 총량)
- *instantaneous*: 어떤 순간(peak)의 throughput
순간 throughput
- *average*: 전체 시간의 평균의 throughput
평균 throughput

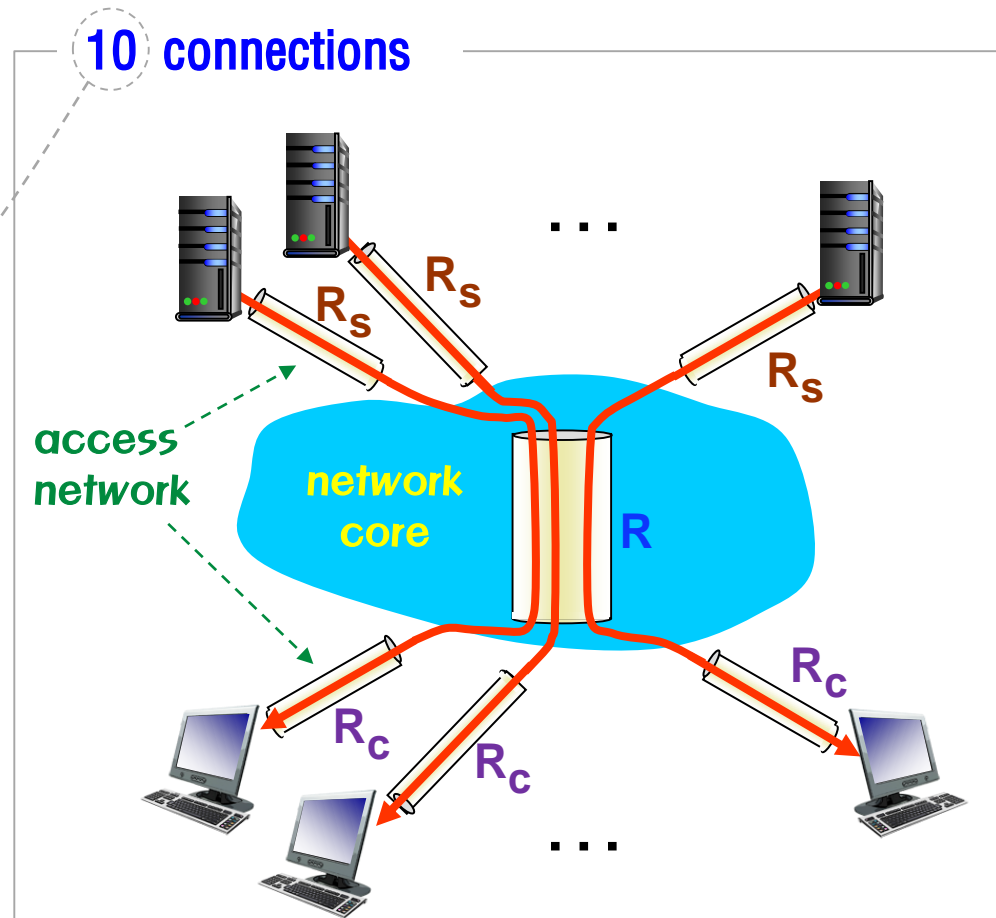
throughput $R_s < R_c$



- **bottleneck link**: 전체 link에서 전송용량이 가장 낮은 link
 - bottleneck line가 throughput에 악영향을 미침

Throughput: Internet scenario

- per-connection end-end throughput:
 $\min(R_c, R_s, R/10)$
- in practice:
 R_c or R_s is often bottleneck



실제 network에서 core 부분은
bottleneck link가 되지 않도록 설계됨



감사합니다.