

컴퓨터 네트워크

- 컴퓨터 네트워크와 인터넷 -

순천향대학교 사물인터넷학과

목표 및 개요

■ 목표

- 네트워크에 대한 “느낌”, “전체 그림”, “용어” 습득
 - ✓ 이후 강의에서 깊게 다룰 예정
- 접근방법
 - ✓ “인터넷”을 예제로 컴퓨터 네트워크의 기본 지식 습득

■ 개요

- 인터넷이란 무엇인가?
- 프로토콜이란 무엇인가?
- 네트워크 엣지
 - ✓ 호스트, 액세스 네트워크, 물리 매체
- 네트워크 코어
 - ✓ 패킷 교환, 회선 교환, 인터넷의 구조
- 성능 지표: 손실 loss, 지연 delay, 처리율 throughput
- 프로토콜 계층, 서비스 모델
- 보안, 역사

컴퓨터 네트워크와 인터넷

1.1 인터넷이란 무엇인가?

1.2 네트워크 엣지

- 종단 시스템, 액세스 네트워크, 링크

1.3 네트워크 코어

- 패킷 교환, 회선 교환, 인터넷의 구조

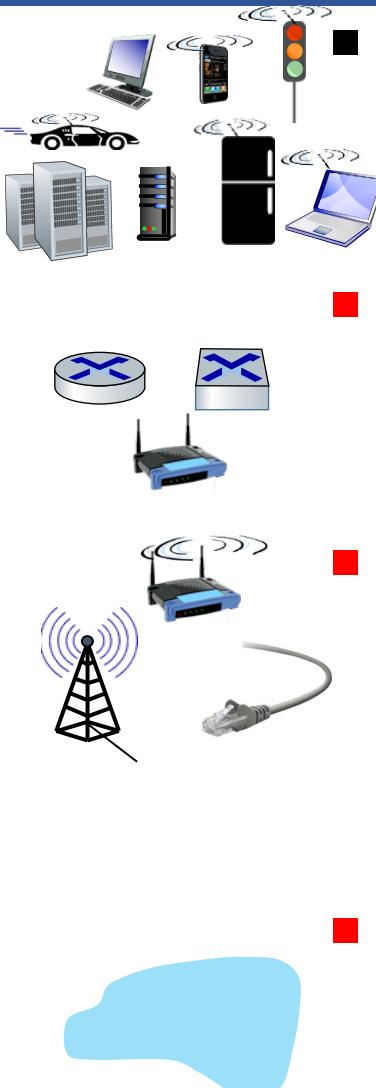
1.4 패킷 교환 네트워크에서의 지연, 손실과 처리율

1.5 프로토콜 계층, 서비스 모델

1.6 공격받는 네트워크: 보안

1.7 인터넷의 역사

인터넷이란 무엇인가? 구성요소 관점 (nuts and bolts view)



■ 많은 수의 연결된 장치들

- 호스트 host = 종단 시스템 end system
- 네트워크 응용 프로그램을 실행

■ 패킷 스위치 packet switch

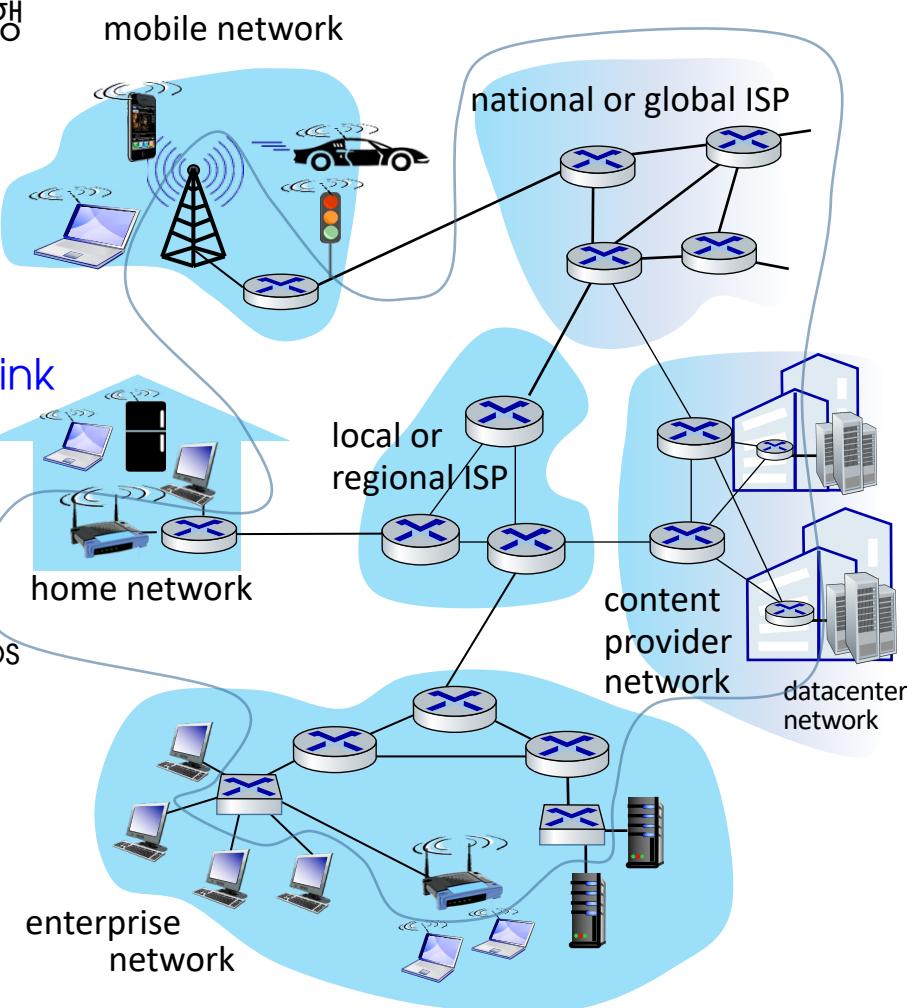
- 패킷 포워딩(전송)
- 라우터 router 와 스위치 switch

■ 통신 링크 communication link

- 구리선, 광섬유, 무선, 위성
- 전송률 transmission rate
 - ✓ throughput 또는 bandwidth
 - ✓ 예) 100Mbps, 1Gbps, 10Gbps

■ 네트워크 networks

- 호스트, 라우터, 링크의 집합
- 기관에 의해 관리됨



다양한 인터넷 연결 장치



IP picture frame
<http://www.ceiva.com/>



Web-enabled toaster +
weather forecaster



Tweet-a-watt:
monitor energy use



Slingbox: watch,
control cable TV remotely



sensorized,
bed
mattress



Nest Thermostat



Smart Refrigerator



Internet phones



bikes



cars



scooters 5



Fitbit



Amazon Echo



AR devices

인터넷이란 무엇인가? 구성요소 관점 (nuts and bolts view)

■ 인터넷 Internet: 네트워크들의 네트워크

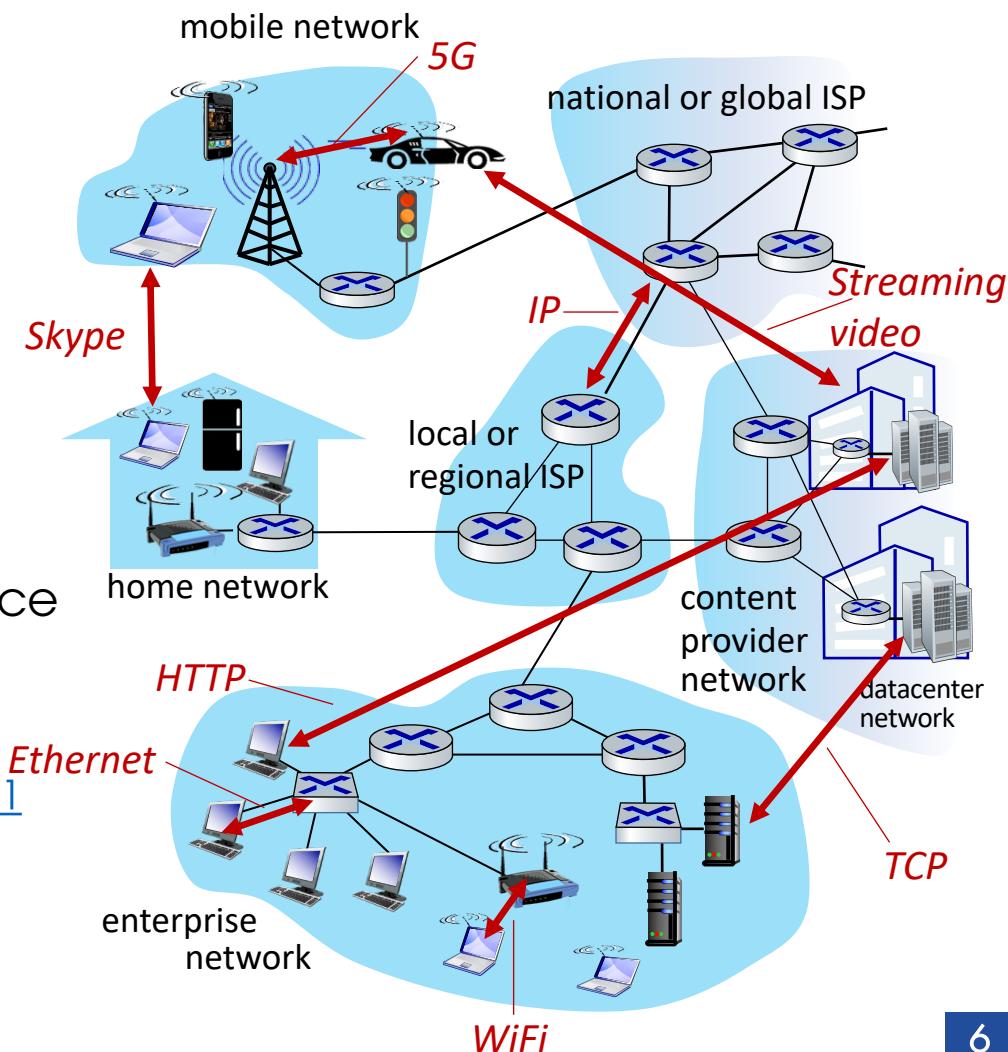
- 상호 연결된 ISP Internet Service Provider, 인터넷 서비스 제공자

■ 프로토콜 protocol

- 메시지 송수신 제어
- 예) HTTP (Web), 스트리밍 비디오, Skype, TCP, IP, WiFi, 4G/5G, 이더넷

■ 인터넷 표준 Internet standard

- IETF: Internet Engineering Task Force
- RFC: Request for comments
 - ✓ 예) TCP (RFC 793), IPv4 (RFC 791)
 - ✓ IPv4: <https://tools.ietf.org/html/rfc791>



인터넷이란 무엇인가? 서비스 관점 (service view)

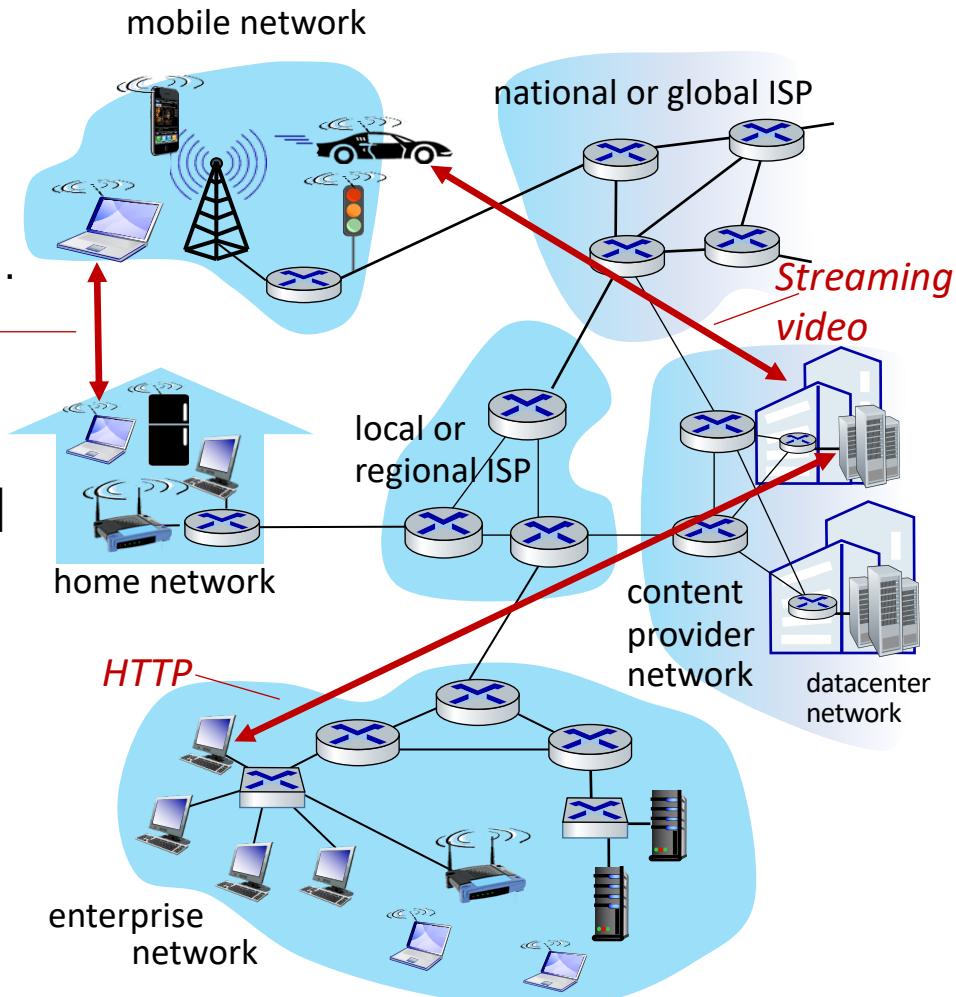
■ 인프라스트럭처 Infrastructure

- 아래와 같은 응용 프로그램에 네트워크 서비스 제공
- 웹, 스트리밍 비디오, 화상회의, 이메일, 게임, 전자상거래, SNS, ...

Skype

■ 프로그래밍 인터페이스

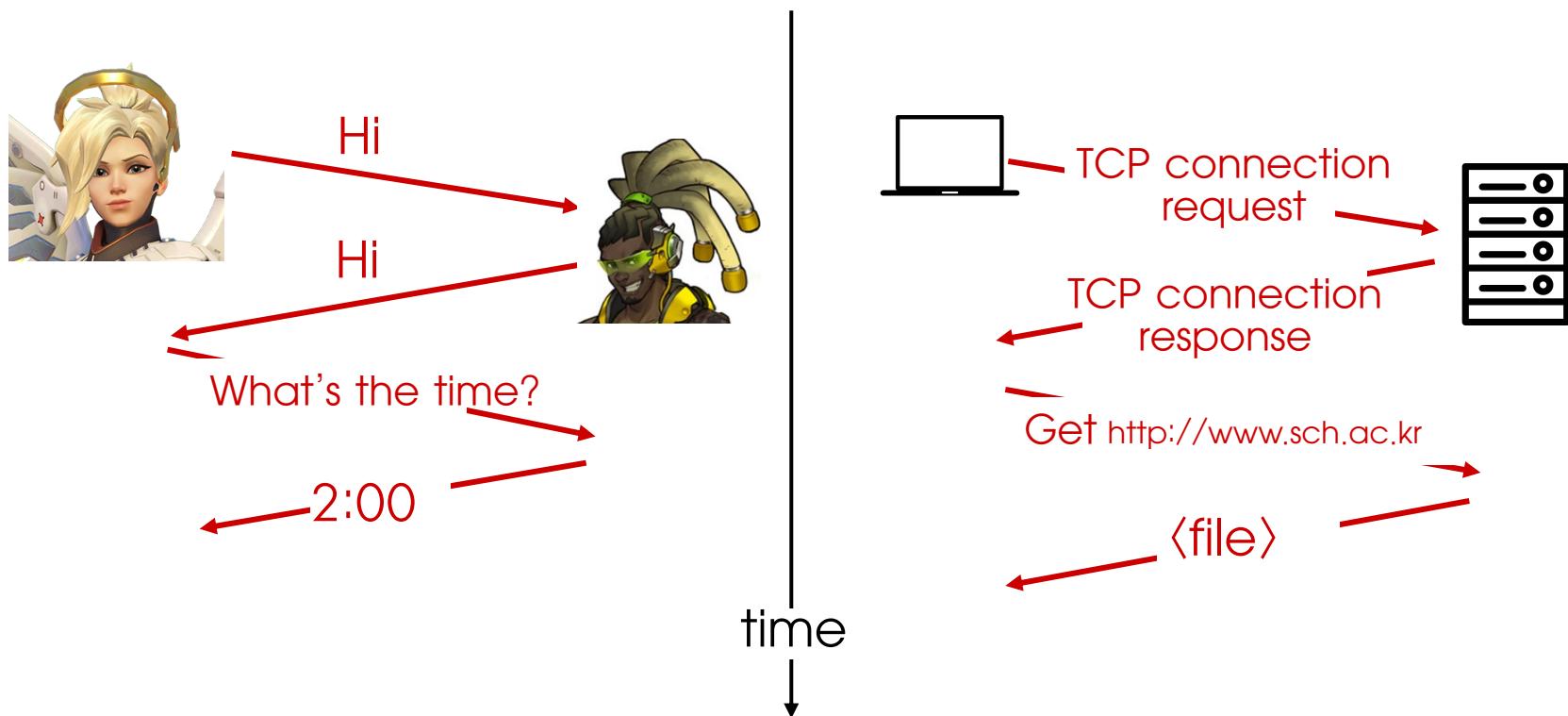
- 송수신 응용 프로그램을 인터넷에 연결시켜주는 인터페이스
- 우편 서비스와 유사한 서비스 옵션을 제공



프로토콜이란 무엇인가?

프로토콜 Protocol 은 송수신 메시지의 형식, 순서와 메시지 송수신 시 동작을 정의한다.

■ “사람 간 프로토콜” vs “네트워크 프로토콜”



컴퓨터 네트워크와 인터넷

1.1 인터넷이란 무엇인가?

1.2 네트워크 엣지

- 종단 시스템, 액세스 네트워크, 링크

1.3 네트워크 코어

- 패킷 교환, 회선 교환, 인터넷의 구조

1.4 패킷 교환 네트워크에서의 지연, 손실과 처리율

1.5 프로토콜 계층, 서비스 모델

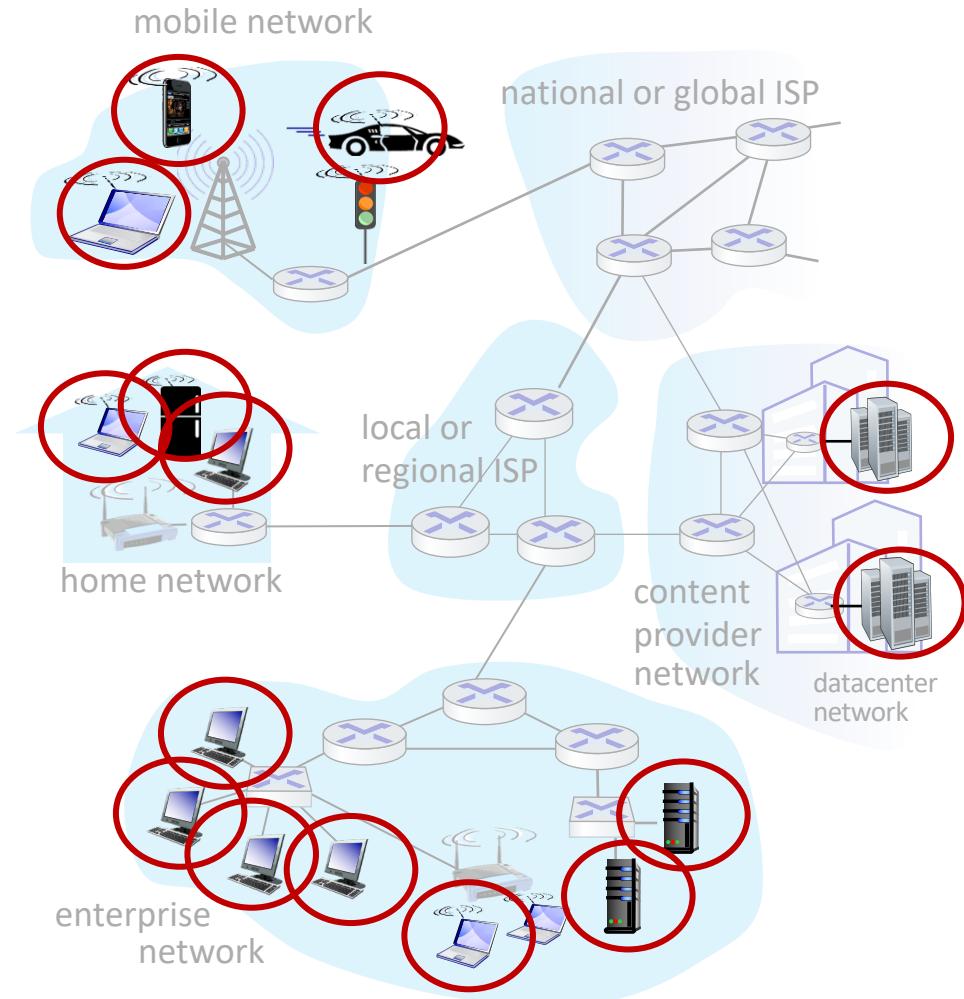
1.6 공격받는 네트워크: 보안

1.7 인터넷의 역사

네트워크 구조

■ 네트워크 엣지 | network edge

- 호스트: 클라이언트 client와 서버 server
- 서버는 종종 데이터 센터에 존재함



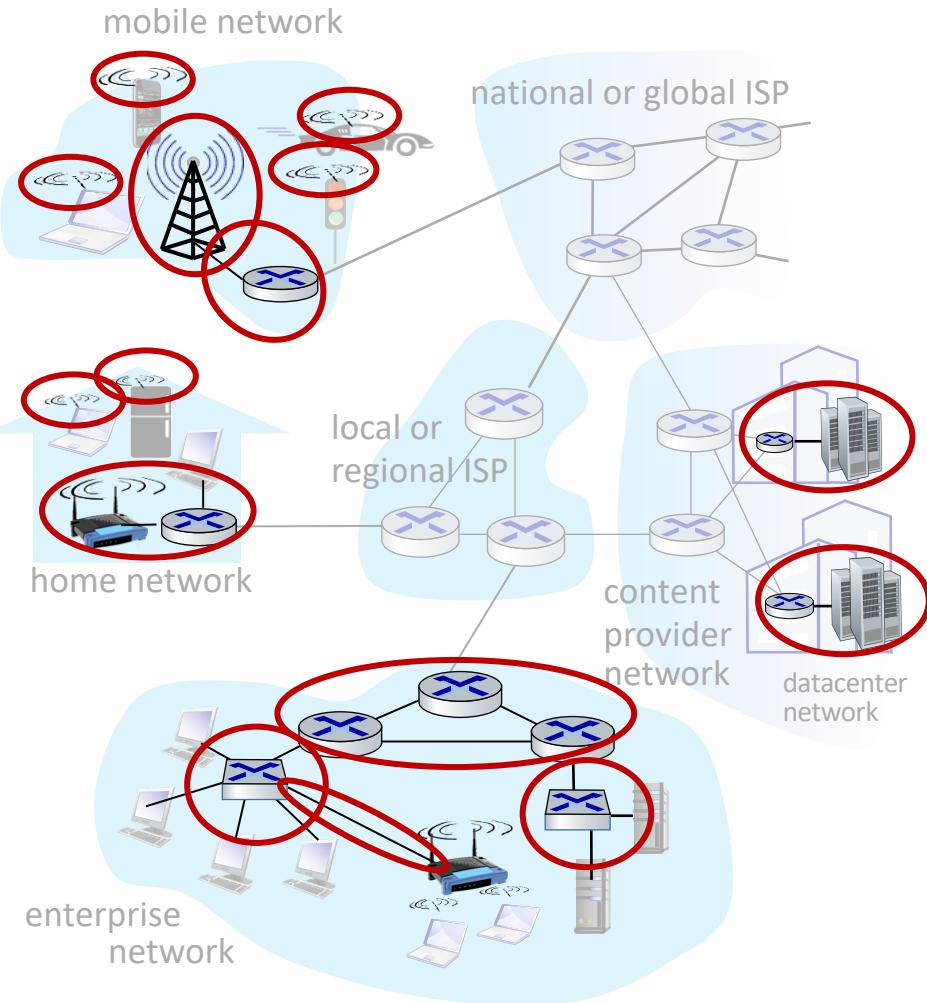
네트워크 구조

■ 네트워크 엣지 | network edge

- 호스트: 클라이언트 client와 서버 server
- 서버는 종종 데이터 센터에 존재함

■ 액세스 네트워크, 물리 매체

- 호스트를 네트워크 코어에 연결해주는 역할 수행
- 유선, 무선 통신 링크



네트워크 구조

■ 네트워크 엣지 network edge

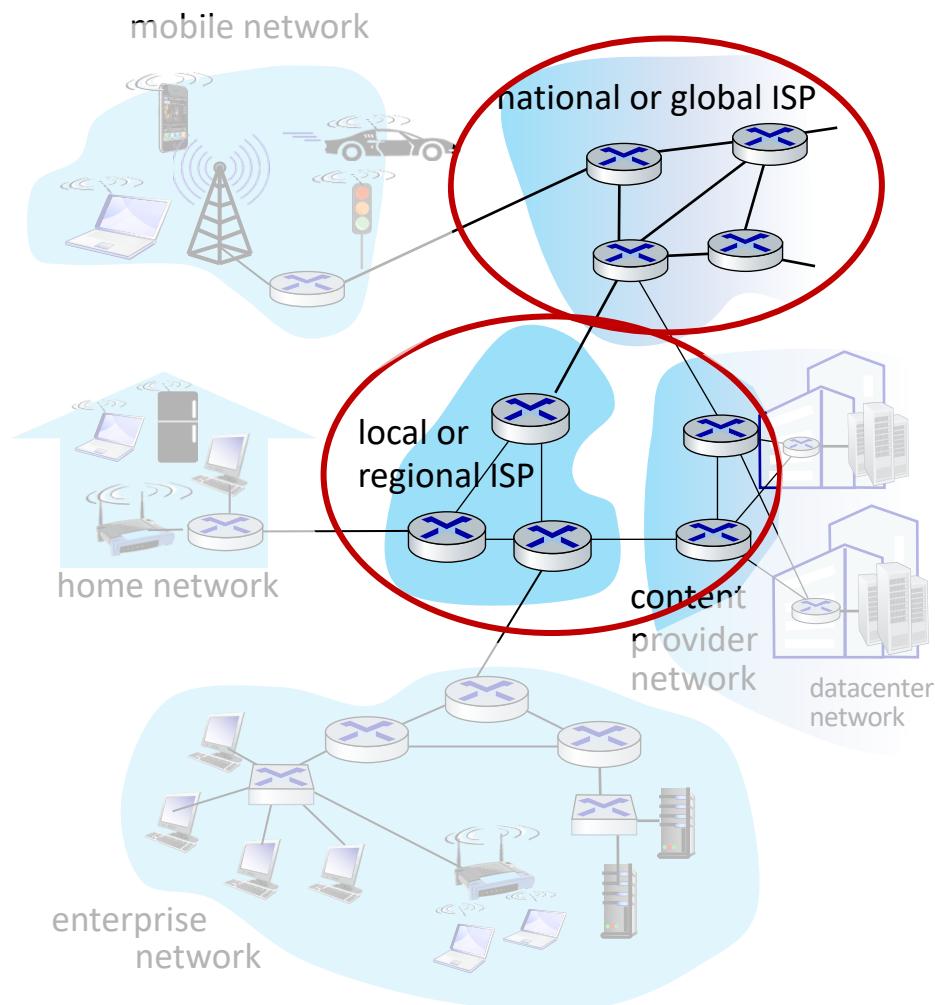
- 호스트: 클라이언트 client와 서버 server
- 서버는 종종 데이터 센터에 존재함

■ 액세스 네트워크, 물리 매체

- 호스트를 네트워크 코어에 연결해주는 역할 수행
- 유선, 무선 통신 링크

■ 네트워크 코어 network core

- 상호 연결된 라우터들
- 네트워크들의 네트워크



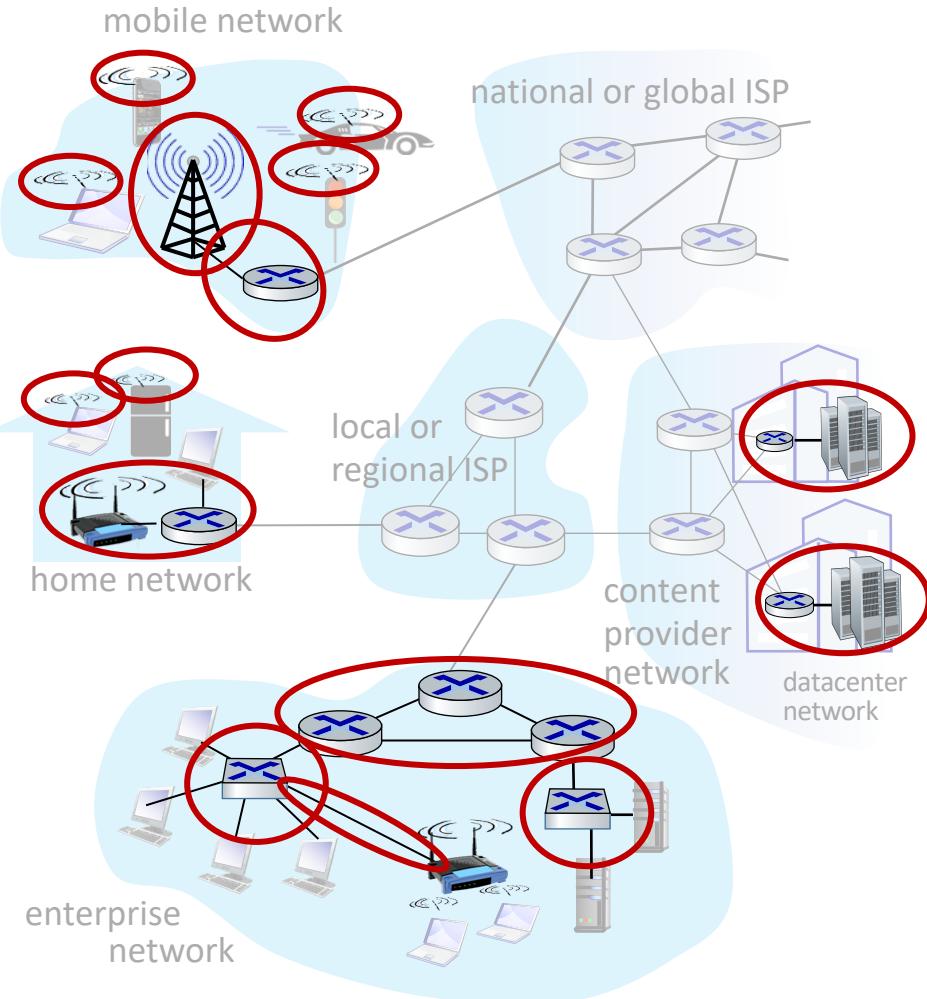
액세스 네트워크와 물리 매체

■ 호스트를 어떻게 엣지 라우터 edge router에 연결하는가?

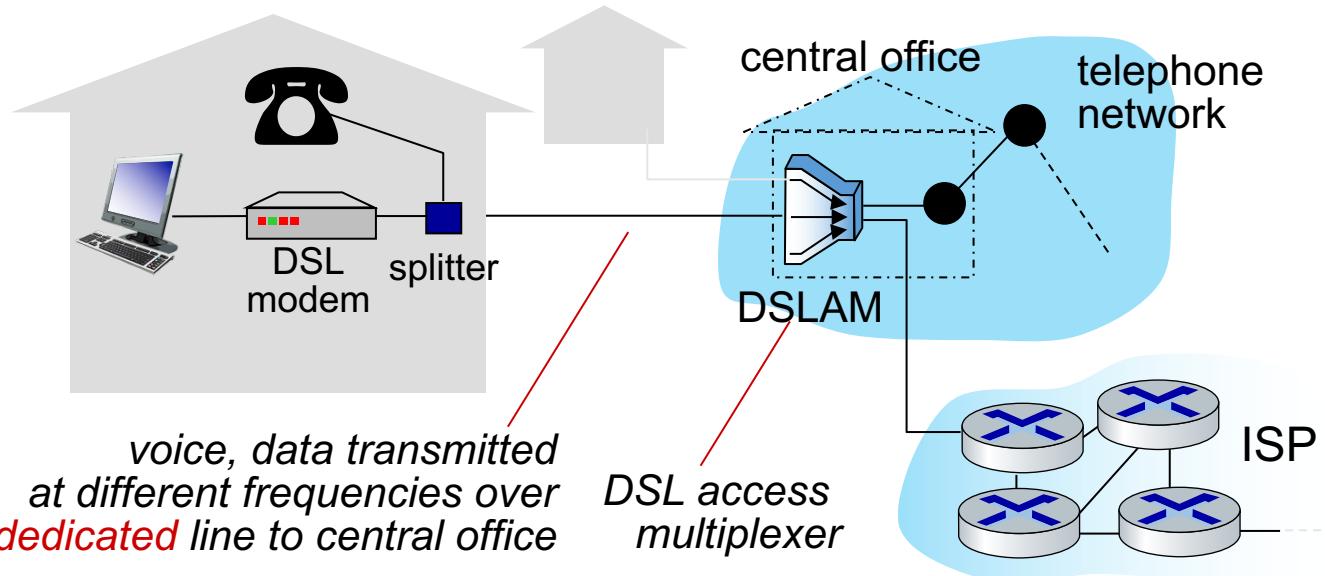
- 홈 액세스 네트워크
- 기관 ^{institutional} 액세스 네트워크
 - ✓ 학교, 회사 등
- 이동 ^{mobile} 액세스 네트워크
 - ✓ WiFi, 4G/5G

■ 중요사항

- 액세스 네트워크의 처리율 (bps, 초당 비트 수)
- 공유 ^{shared} 또는 전용 ^{dedicated?}

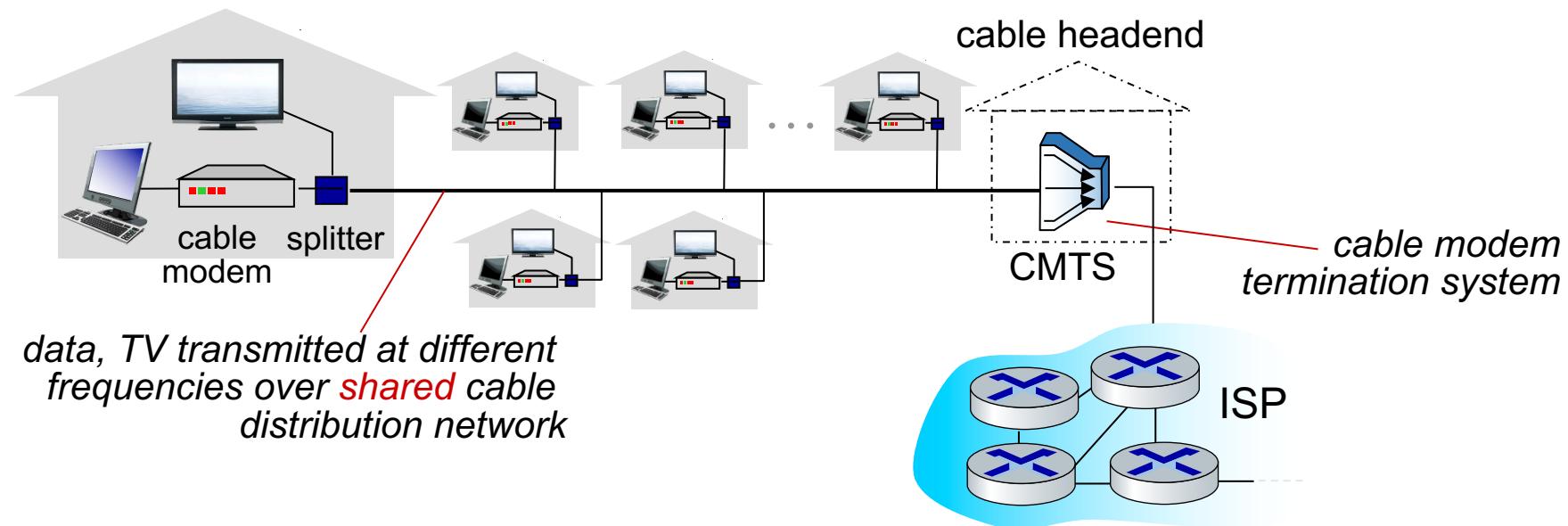


액세스 네트워크: digital subscriber line (DSL)



- 중앙국(CO^{central office})의 DSLAM까지는 **기존 전화선을 이용**
 - DSL 전화선으로 전송된 데이터는 인터넷으로 전달
 - DSL 전화선으로 전송된 음성은 전화망으로 전달
- 3.5~16 Mbps 전용 업스트림 전송 속도
- 24~52 Mbps 전용 다운스트림 전송 속도

액세스 네트워크: 케이블 네트워크 (cable network)



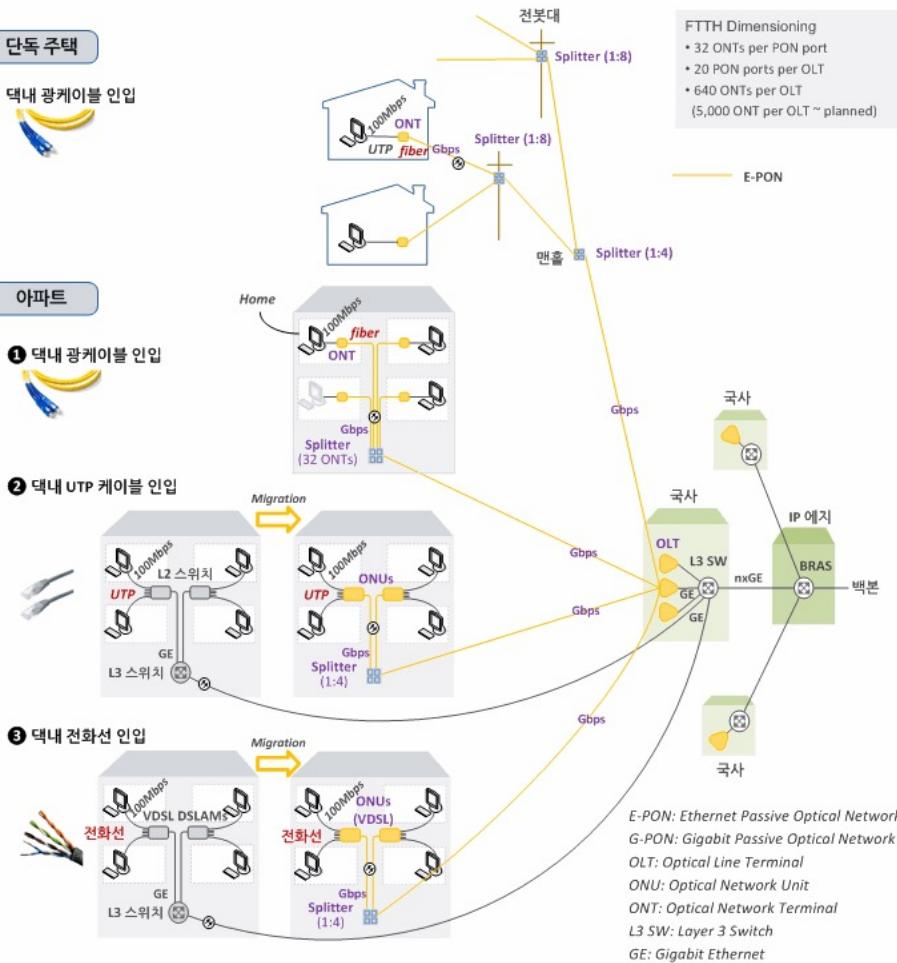
■ HFC^{hybrid fiber coax}

- 비대칭^{asymmetric}
 - ✓ 40Mbps~1.2Gbps 다운스트림, 30~100 Mbps 업스트림 전송 속도

■ 케이블과 광섬유 네트워크가 각 가정을 ISP 라우터에 연결

- 각 가정들이 케이블 헤드엔드까지 액세스 네트워크 **공유**
 - ✓ 케이블 네트워크와 달리 DSL은 중앙국까지 전용 회선 사용

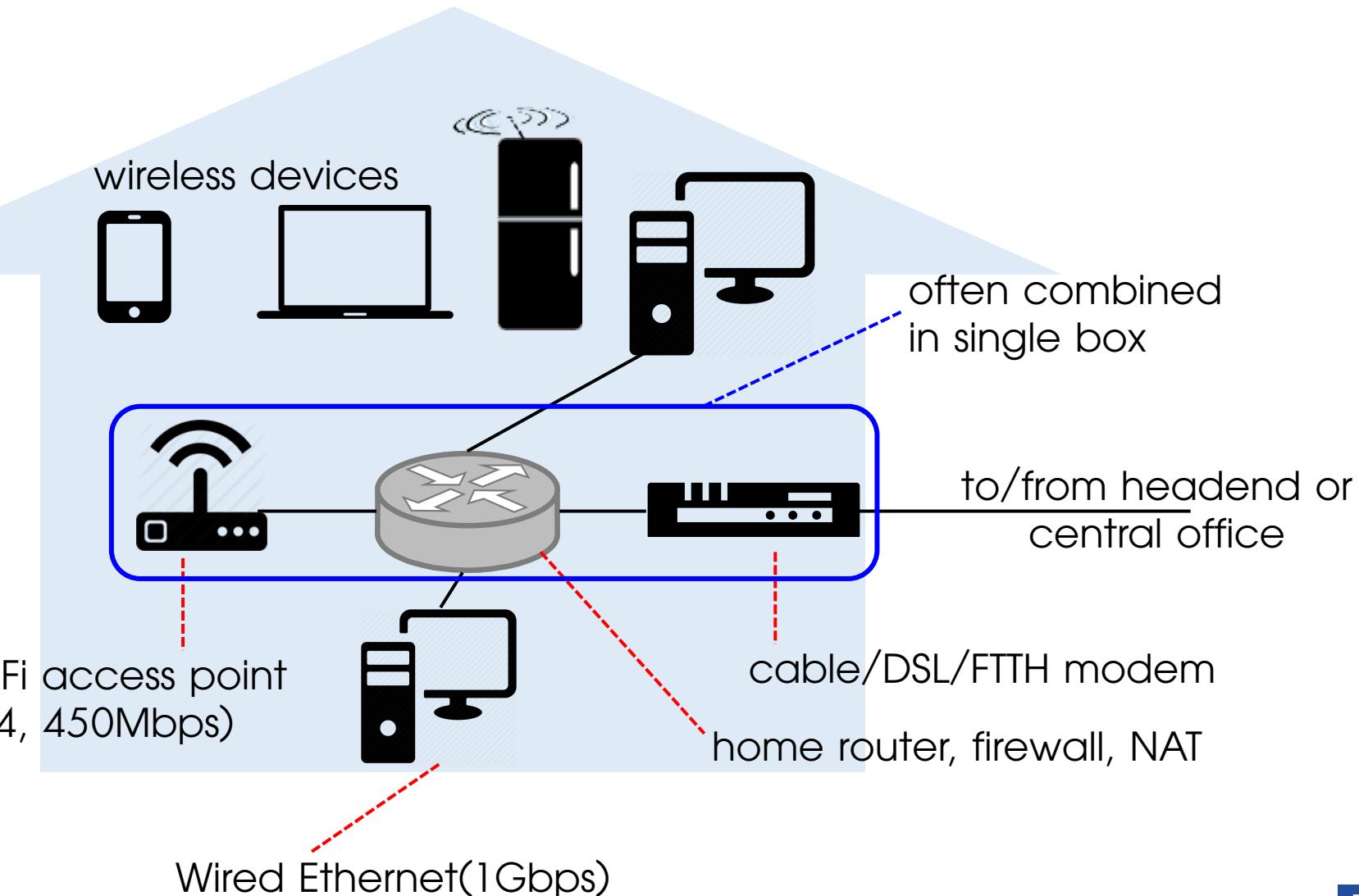
액세스 네트워크: FTTH (Fiber To The Home)



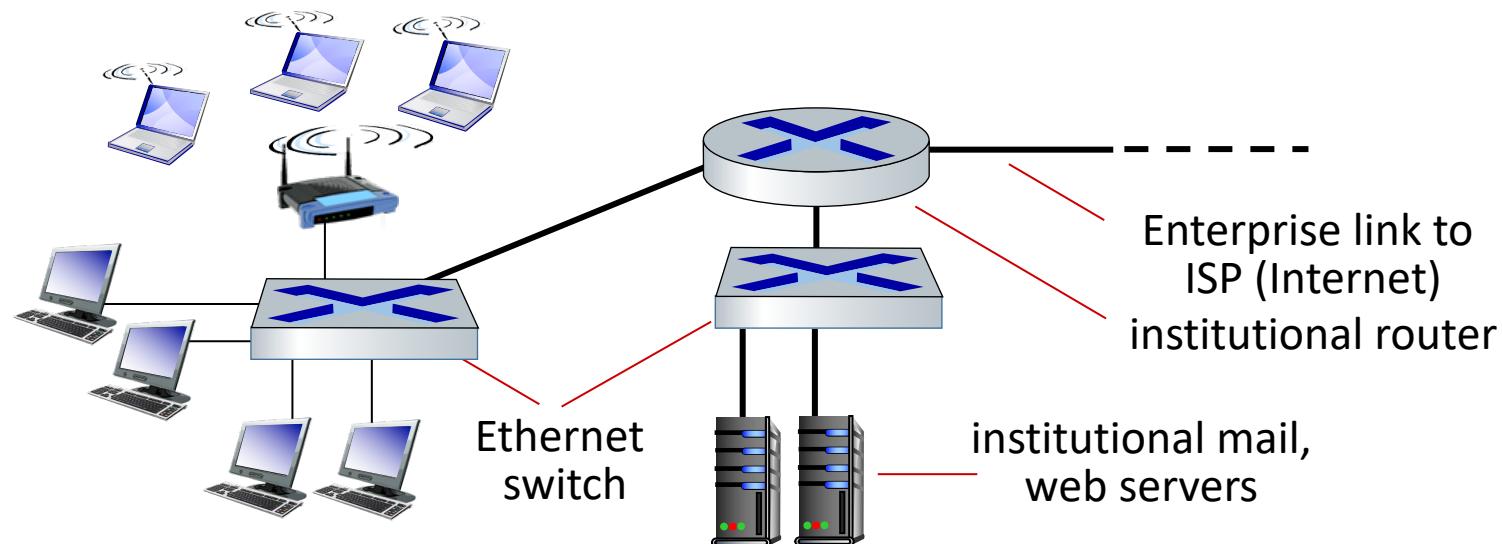
FTTH

- CO로부터 가정까지 직접 광섬유 경로를 제공
 - Gbps 범위의 인터넷 접속 속도 제공
 - CO의 OLT^{Optical Line Terminator}와 각 가정의 ONT^{Optical Network Terminator}를 광섬유로 연결하고, 일반적으로 홈 라우터와 ONT는 UTP로 연결됨

액세스 네트워크: Home Networks



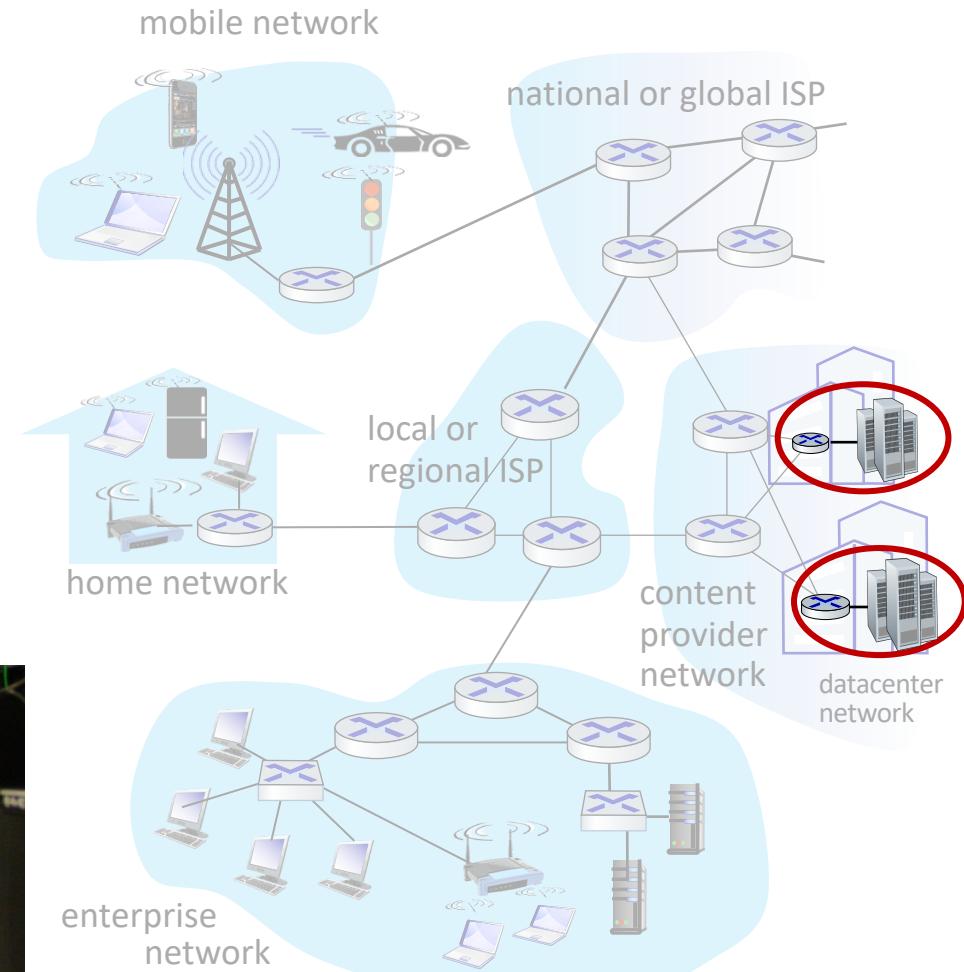
액세스 네트워크: Enterprise Networks



- 회사, 대학 등
- 유선, 무선 링크가 섞여 있음
 - 이더넷Ethernet: 유선 접속, 100Mbps, 1Gbps, 10Gbps 전송 속도
 - WiFi: 무선 접속, 11, 54, 600Mbps, 11Gbps(WiFi6)
- 호스트들은 이더넷 스위치, WiFi AP에 연결

액세스 네트워크: Data Center Networks

- 고대역폭 링크(10~100Gbps)를 통해 수십만개의 서버와 인터넷을 연결함

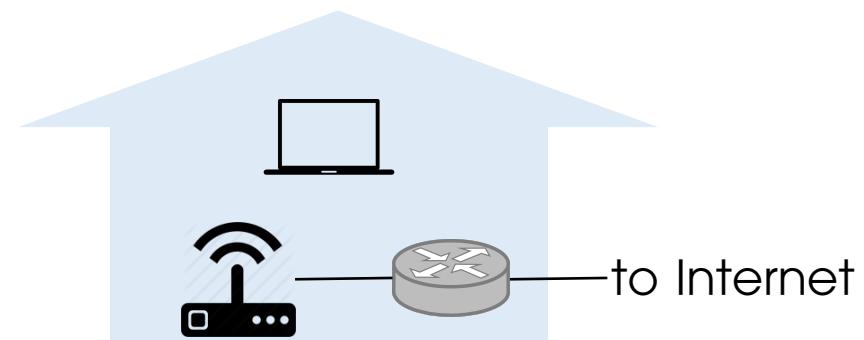


Courtesy: Massachusetts Green High Performance Computing Center (mghpcc.org)

무선 액세스 네트워크 Wireless Access Network

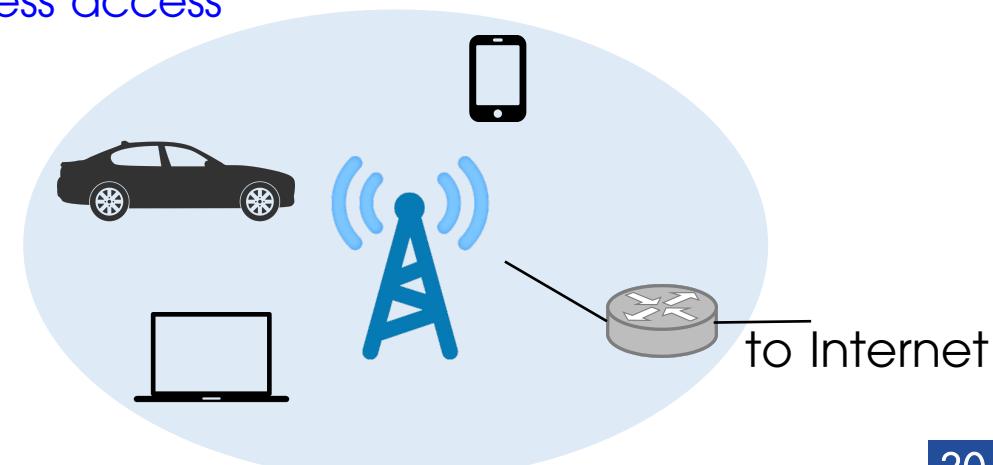
■ 무선 액세스 네트워크 wireless access network

- 공유 shared
- 호스트를 기지국 base station, BS 또는 AP access point를 통해 라우터에 연결시켜 줌



■ 무선 랜 wireless LAN

- 건물 내 (수십 미터)
- 802.11 a/b/g/n/ac/ad/ax
 - ✓ 11, 54, 600Mbps, 수 Gbps 전송률



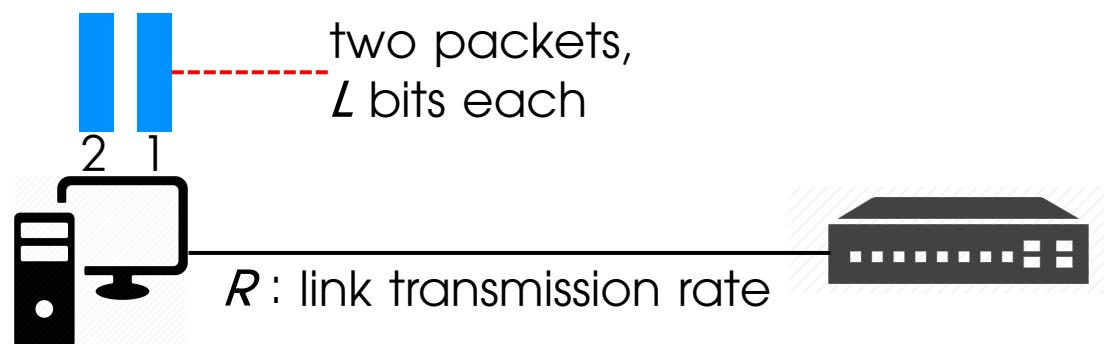
■ 광역 무선 액세스 wide-area wireless access

- 셀룰러 사업자에 의해 제공됨
 - ✓ 커버리지: 수십 km
- 수십 Mbps~최대 20Gbps
- 4G, 5G

호스트의 패킷 전송

■ 호스트의 송신 동작

- 애플리케이션의 메시지를 받음
- 메시지를 길이 L 비트의 작은 덩어리인 **패킷**으로 분할
- 패킷을 링크 전송률 R 로 전송
 - ✓ 링크 전송률은 링크 용량 capacity 또는 링크 대역폭 bandwidth라고 함



$$\text{packet transmission delay} = \frac{\text{time needed to transmit } L\text{-bit packet into link}}{R \text{ (bits/sec)}}$$

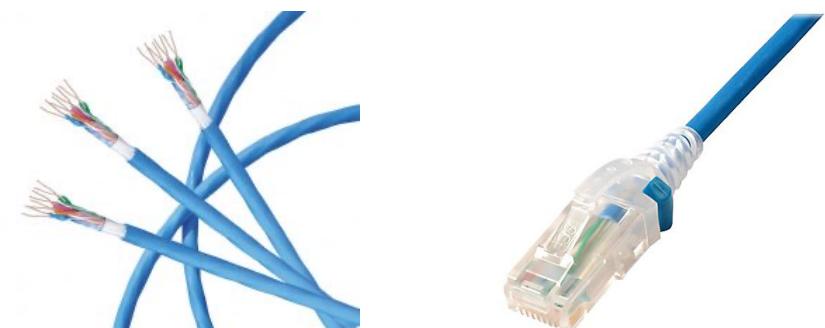
링크: 물리 매체

■ 물리 매체 physical media

- 송신/수신자 간 비트를 전송
- 물리적 링크 physical link
 - ✓ 송신자와 수신자 사이를 연결해 주는 것
 - ✓ 물리 매체 상에 전자파나 광신호를 전파하여 전송
- 유도 매체 guided media
 - ✓ 신호가 견고한 매체를 통해 전송됨
 - ✓ 예) UTP, 광섬유, 동축케이블
- 비유도 매체 unguided media
 - ✓ 무선 통신의 경우처럼, 야외 공간으로 파형을 자유롭게 전파
 - ✓ 예) 무선

■ UTP unshielded twisted pair

- 2개의 절연 구리선이 꼬여 있음
- Category 5: 100Mbps 이더넷
- Category 5e: 1Gbps 이더넷
- Category 6: 10Gbps 이더넷



링크: 물리 매체

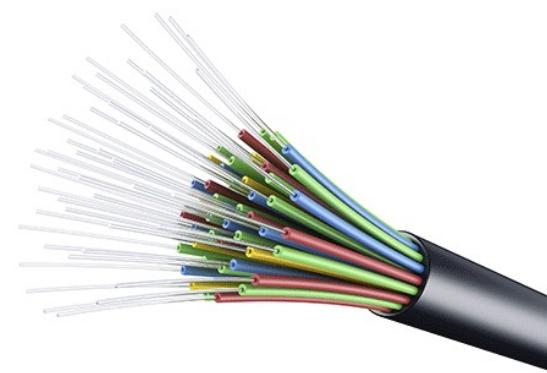
■ 동축 케이블 coaxial cable

- 동심원 형태의 두 개의 구리선
- 양방향 bidirectional
- 광대역 broadband
 - ✓ 케이블 상에서 다수의 주파수 채널
 - ✓ 채널 당 100Mbps



■ 광섬유 케이블 fiber optic cable

- 빛의 파동(각 파동 당 1bit)를 전달하는 유리 섬유
- 고속으로 동작
 - ✓ 고속 1대1 전송 (예: 10~100Gbps)
- 낮은 에러율
 - ✓ 리피터가 많이 필요 없음
 - ✓ 전자기 잡음에 강함



링크: 물리 매체

■ 무선

- 신호가 전자기파로 전달됨
- 물리적인 선이 없음
- 양방향
- 전송 시 환경적인 요인에 영향을 받음
 - ✓ 반사
 - ✓ 물체에 의한 방해
 - ✓ 간섭

■ 무선 링크의 종류

- 무선 랜 (WiFi)
- 블루투스
- 광역 (예: cellular(4G/5G))
- 위성
 - ✓ 270ms의 긴 전파 지연

컴퓨터 네트워크와 인터넷

1.1 인터넷이란 무엇인가?

1.2 네트워크 엣지

- 종단 시스템, 액세스 네트워크, 링크

1.3 네트워크 코어

- 패킷 교환, 회선 교환, 인터넷의 구조

1.4 패킷 교환 네트워크에서의 지연, 손실과 처리율

1.5 프로토콜 계층, 서비스 모델

1.6 공격받는 네트워크: 보안

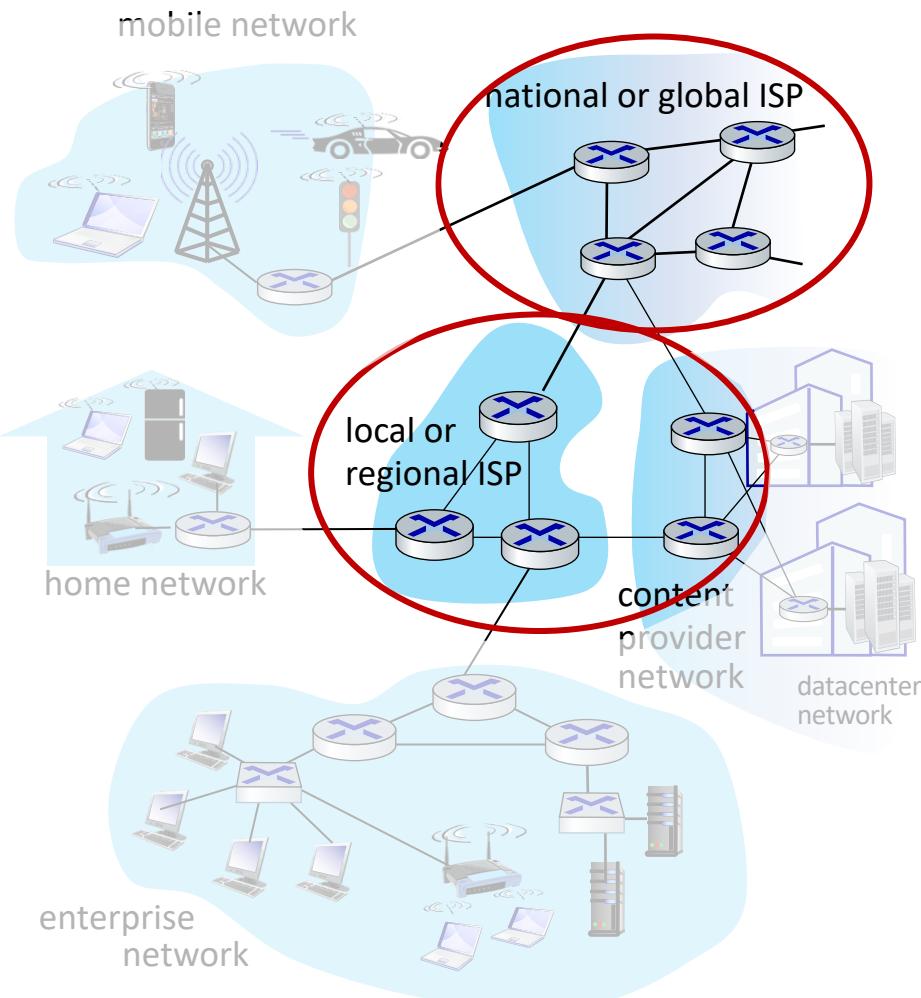
1.7 인터넷의 역사

네트워크 코어 Network Core

- 상호 연결된 라우터들로 이루어진 메시지^{mesh} 망

■ 패킷 교환 packet switching

- 호스트는 애플리케이션 계층의 메시지를 패킷^{packet}이라 알려진 작은 덩어리로 분할
- 패킷들은 송신 측과 수신 측의 경로 상의 링크와 라우터를 거쳐 전송
 - ✓ 패킷들은 한 라우터에서 다음 라우터로 전달^{포워딩, forwarding}
- 각 패킷은 링크의 최대 전송 속도로 전송됨



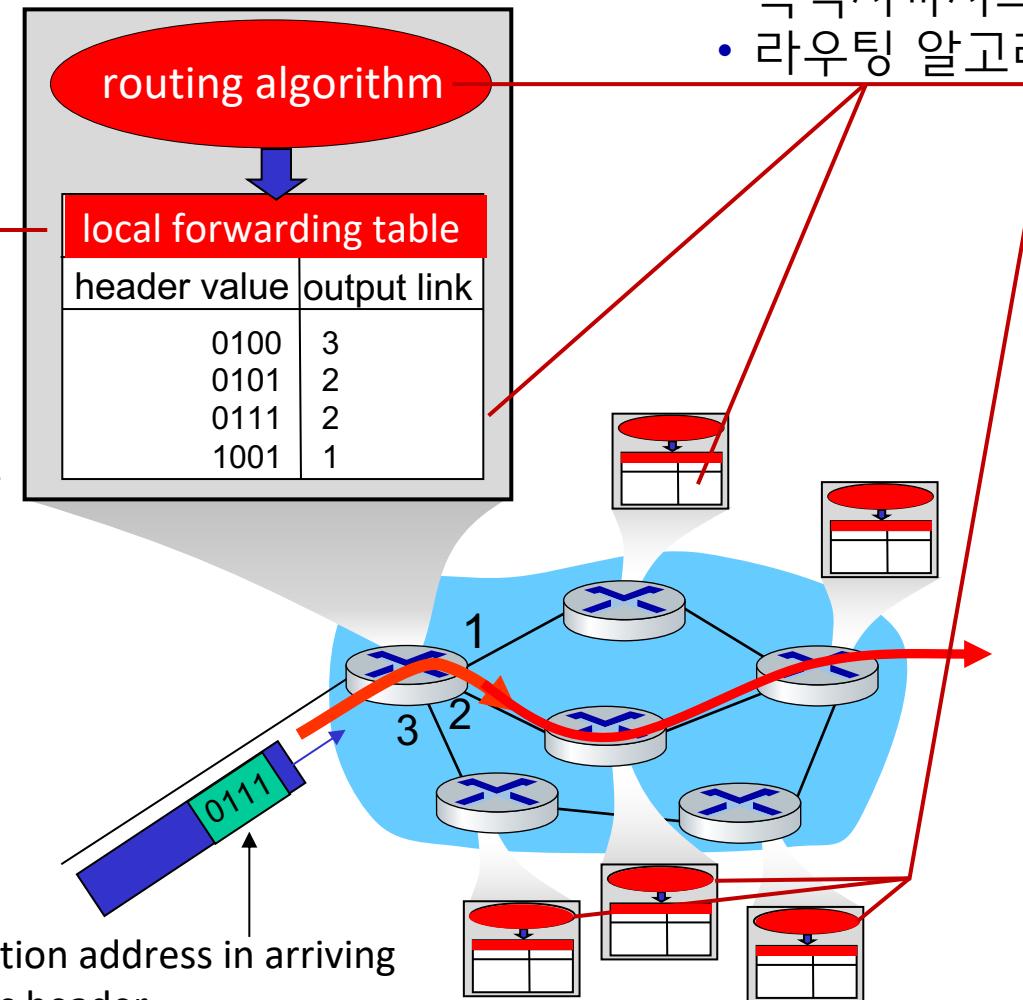
네트워크 코어의 2가지 주요 기능

Routing:

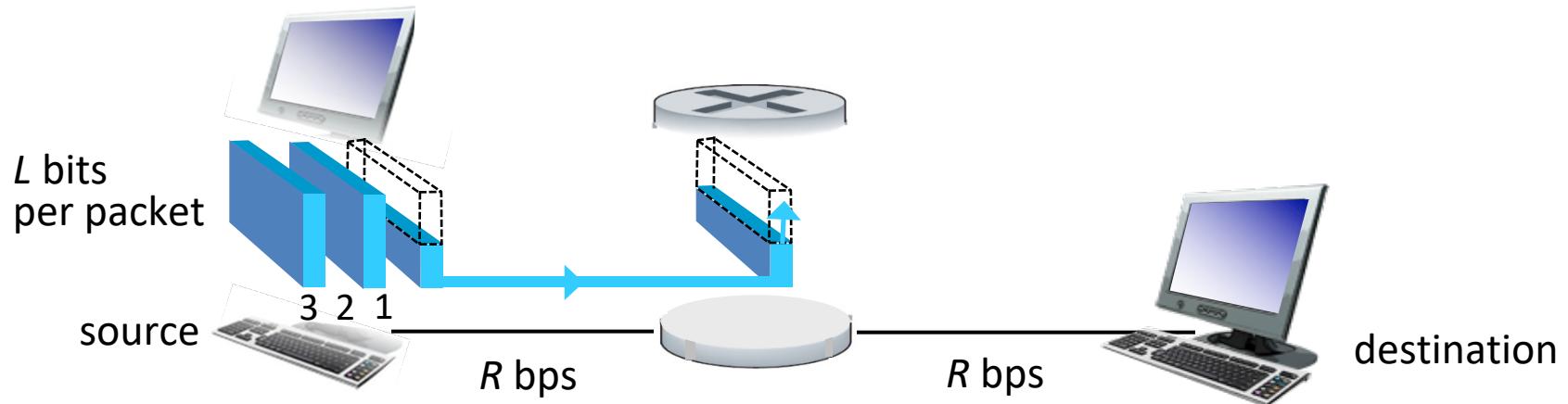
- global action: 발신지에서 목적지까지의 경로를 결정
- 라우팅 알고리즘

Forwarding:

- local action:
라우터의 입력
포트에서 적절한
출력 포트로 패킷을
이동시키는 것

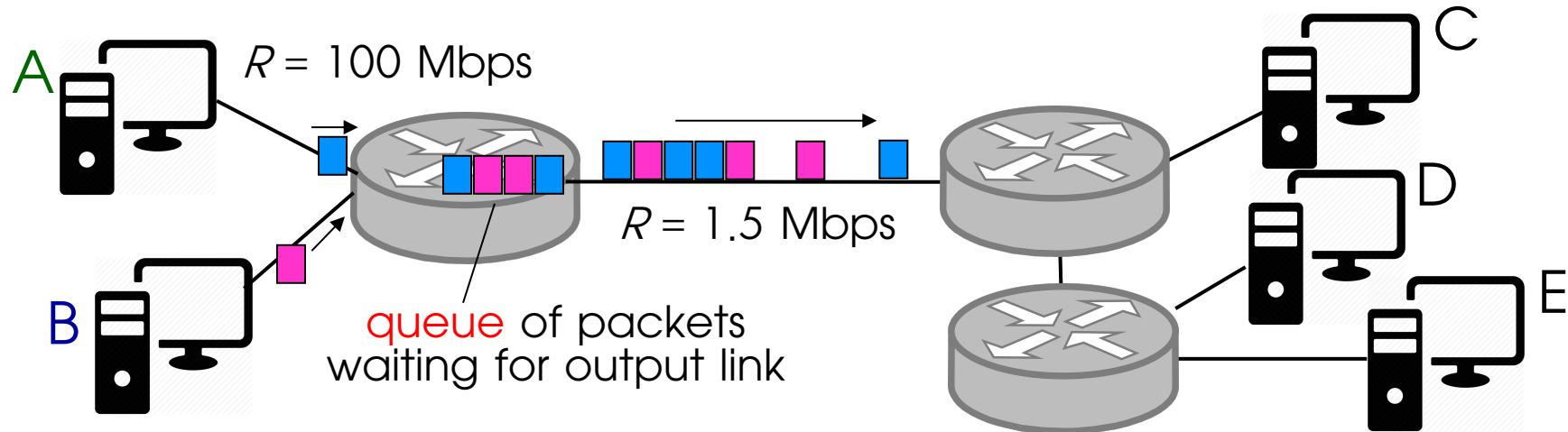


패킷 교환: 저장 후 전달 (store-and-forward)



- L 비트의 패킷을 R bps의 링크로 전송하는 시간: L/R
- 저장 후 전달 *store and forward*
 - 다음 링크로 전송되기 전에 전체 패킷이 라우터에 수신 되어야 함
- 위 그림에서의 지연 = $2L / R$ (전파 지연 무시)
 - $L = 10 \text{ Kbits}$, $R = 100 \text{ Mbps}$
 - 1-hop(홉) 전송 지연 = $L / R = 0.1 \text{ msec}$
 - 종단 간 *end-to-end* 전송 시간 = $2L / R = 0.2 \text{ msec}$

패킷 교환: 큐잉 Queueing, 패킷 손실 Loss



■ 큐잉 Queueing과 패킷 손실 Loss

- 일정시간 동안 라우터의 입력 링크에 패킷이 도착하는 속도가 링크 전송률을 초과하면,
- 패킷은 큐 queue에 저장되고, 출력 링크 상에서 전송 되기를 기다림
- 큐의 메모리가 꽉 차면, 패킷은 손실됨

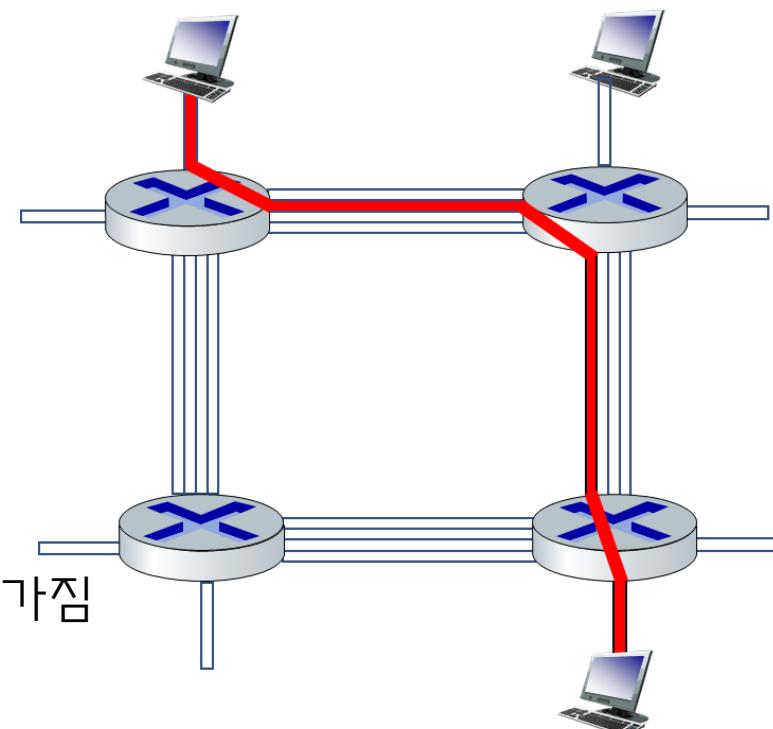


회선 교환 Circuit Switching

- 네트워크 코어 구성 방식은 회선 교환과 패킷 교환 두 가지 방식이 있음

■ 회선 교환 Circuit switching

- 경로 상에 필요한 자원(버퍼, 링크 전송률)을 통신 세션 동안 예약
- 자원이 공유되지 않음
- 송신자와 수신자 간의 경로에 있는 스위치들이 연결 상태를 유지
- 송신자와 수신자 간의 전송률 보장
- 기존의 전화망에 사용



■ 한 링크는 n개의 회선을 연결

- 오른쪽 그림에서 각 링크는 4개의 회선을 가짐
- 네트워크 자원을 분할(다중화)
- 통신이 없는 상황에서, 회선이 낭비됨

회선 교환: FDM과 TDM

■ FDM Frequency Division Multiplexing

- 설정된 연결은 링크의 주파수 스펙트럼을 공유
- 각 연결에 고정된 주파수 대역이 할당

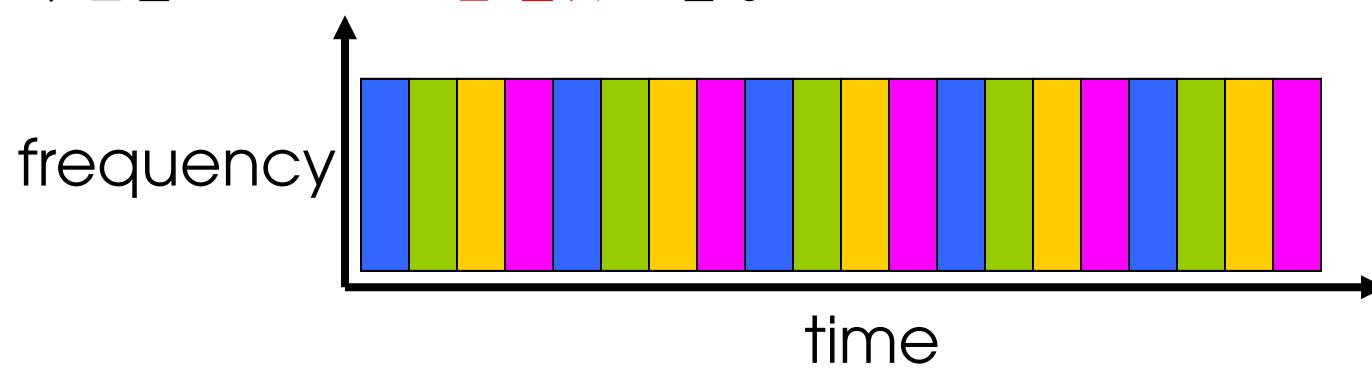
예제

4 users



■ TDM Time Division Multiplexing

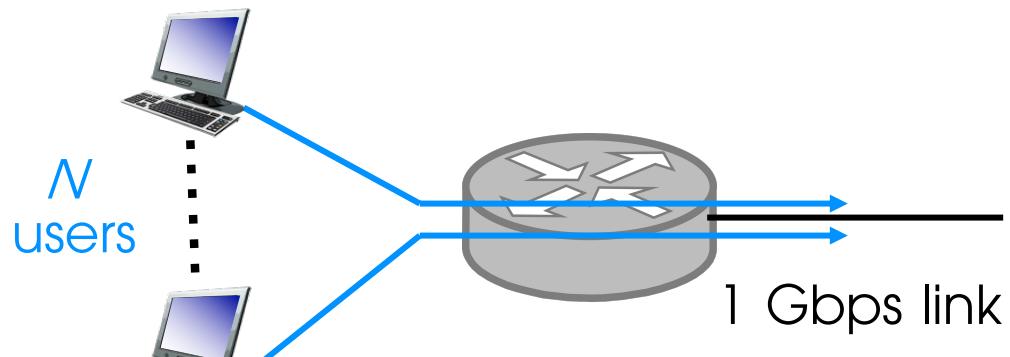
- 시간을 일정 주기의 프레임으로 구분하고, 각 프레임은 고정된 수의 시간 슬롯으로 분할
- 각 연결에 하나의 시간 슬롯이 할당



패킷 교환 vs 회선 교환

■ 예제

- 1Gbps 링크
- 각 사용자는
 - ✓ active일 때 100Mbps 사용
 - ✓ 전체 시간의 10%만 active함
- 회선 교환
 - ✓ 100Mbps가 각 사용자에게 예약
 - ✓ $1\text{ Gbps}/100\text{Mbps} = 10\text{명}$ 의 사용자
- 패킷 교환
 - ✓ 35명 사용자의 경우, 동시에 11명 이상이 active일 확률은 0.0004이하임
 - ✓ Q1. 0.0004가 나온 이유?
 - ✓ Q2. 사용자가 35명보다 많은 경우에는 어떻게 되는가?



패킷 교환은 더 많은 사용자(위의 예제의 경우 3배 이상)가 네트워크를 대등한 자연 성능을 가지면서 사용하게 해줌

패킷 교환 vs 회선 교환

Q. 그럼 패킷 교환이 항상 나은가?

- bursty data에 유리
 - 자원 공유
 - 연결 설정이 없으므로, 간단함
- 과도한 혼잡 발생 가능
 - 패킷 전송 지연 및 손실 발생
 - 신뢰적 데이터 전송 및 혼잡제어 프로토콜 필요

Q. 패킷 교환이 회선 교환처럼 동작할 수 있는가?

- 오디오/비디오 애플리케이션을 위한 bandwidth 보장 필요
- chapter 7의 주제!!!

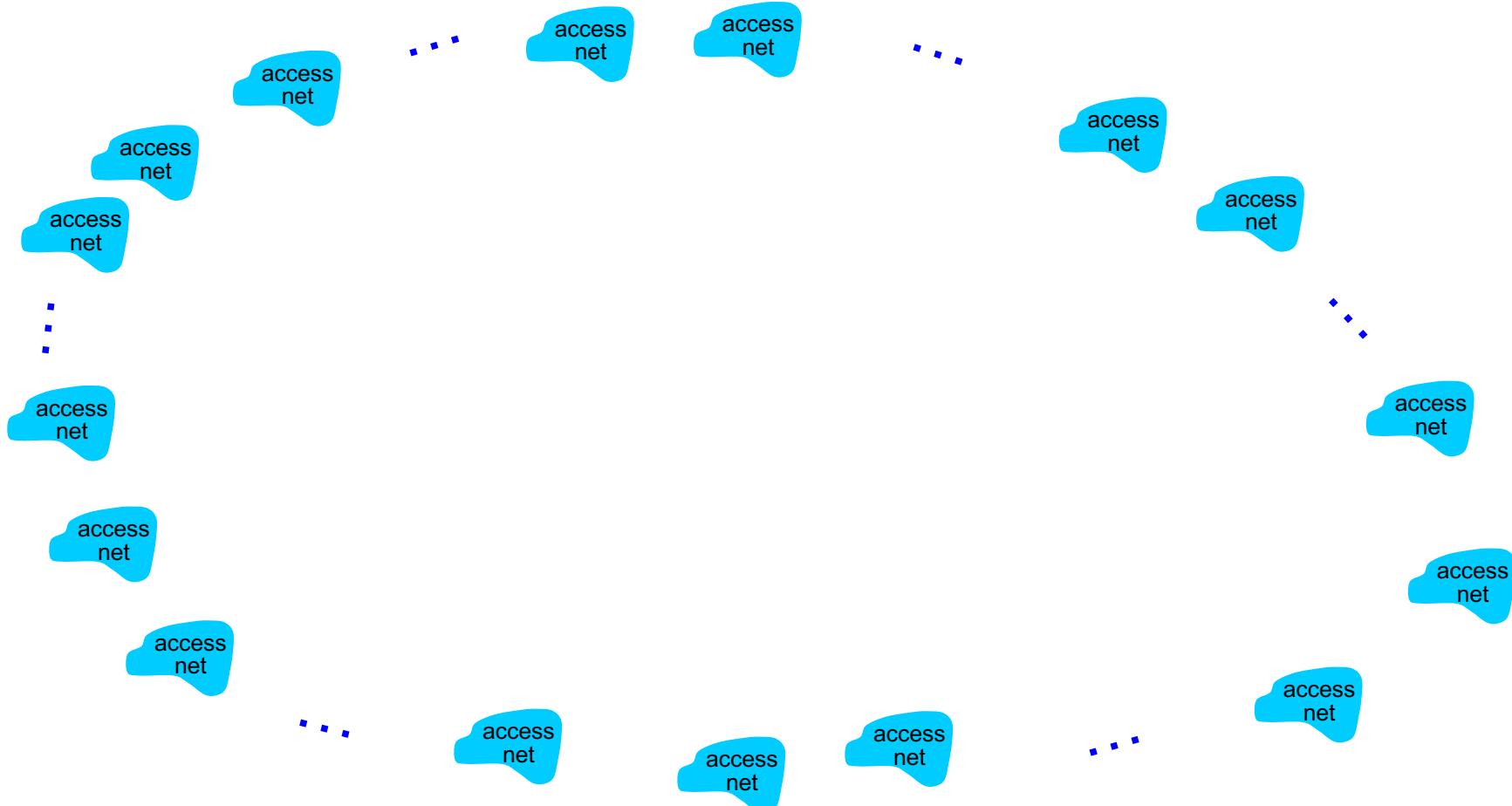
인터넷의 구조: 네트워크들의 네트워크

- 종단 시스템은 **액세스 ISP** access Internet Service Provider를 통해 인터넷에 연결
 - 통신회사, 회사, 대학교
- 액세스 ISP들은 상호 연결되어야 함
 - 따라서, 어떤 2개의 호스트도 서로 패킷을 주고 받을 수 있음
 - 대신, 네트워크의 구조는 네트워크들의 네트워크 형태로 매우 복잡해짐
 - ✓ 성능보다 경제적, 국가 정책에 의해 진화
- 지금부터 단계적으로 현재 인터넷 구조를 설명

인터넷의 구조: 네트워크들의 네트워크

■ Question

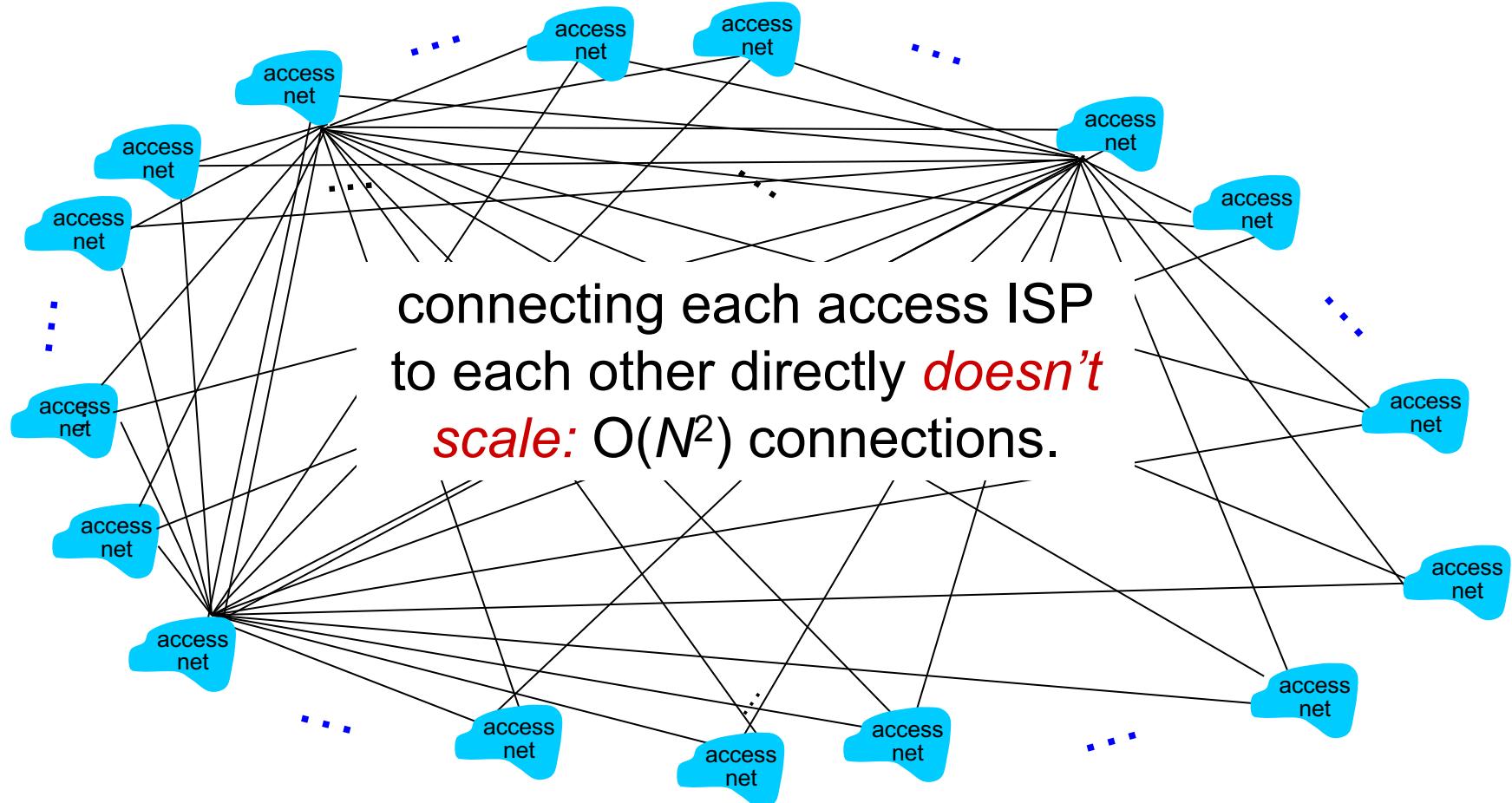
- 수백만 개의 액세스 ISP를 어떻게 연결하는가?



인터넷의 구조: 네트워크들의 네트워크

■ Option 1

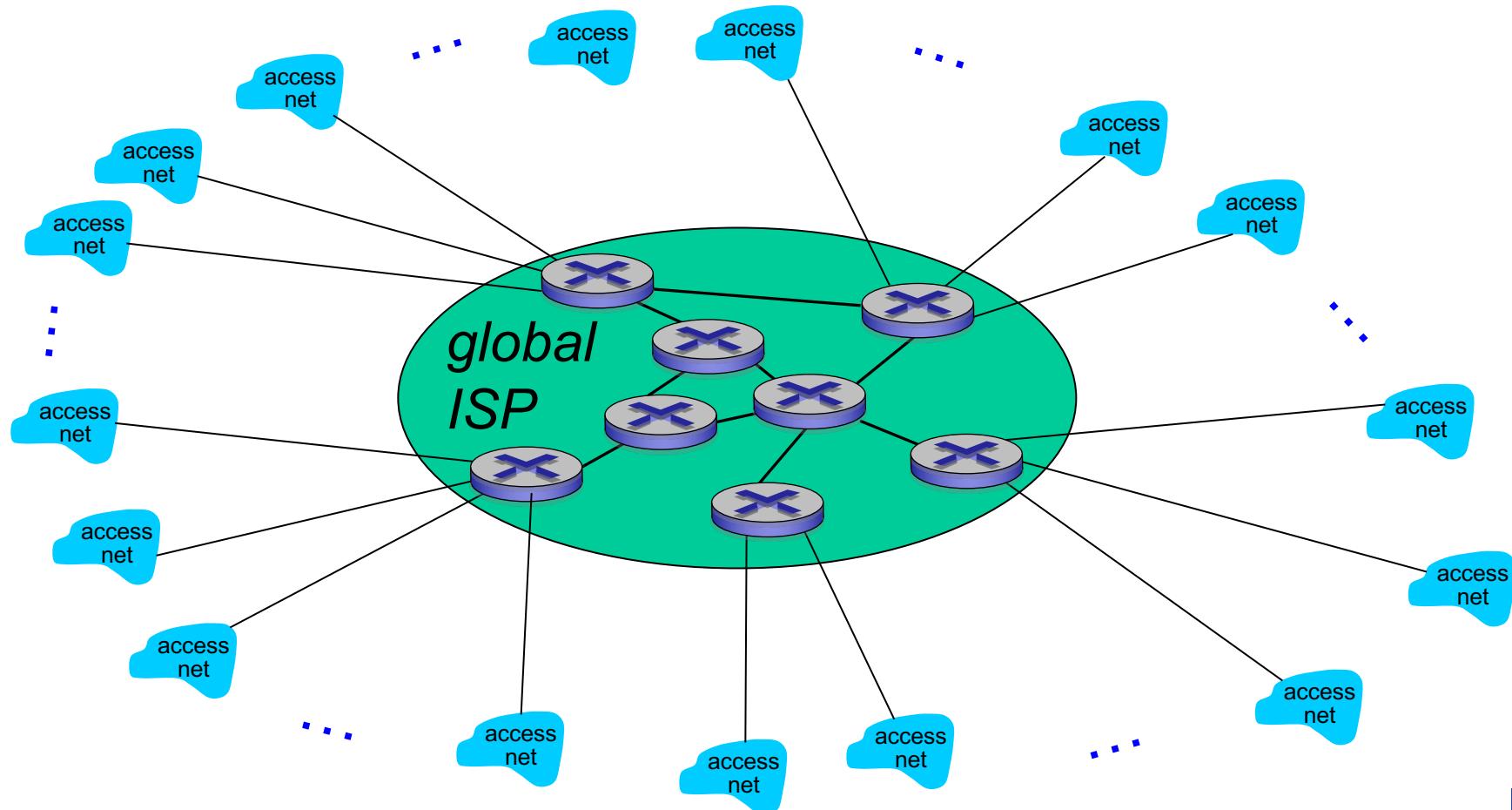
- 각 액세스 ISP를 모든 다른 액세스 ISP에 연결?



인터넷의 구조: 네트워크들의 네트워크

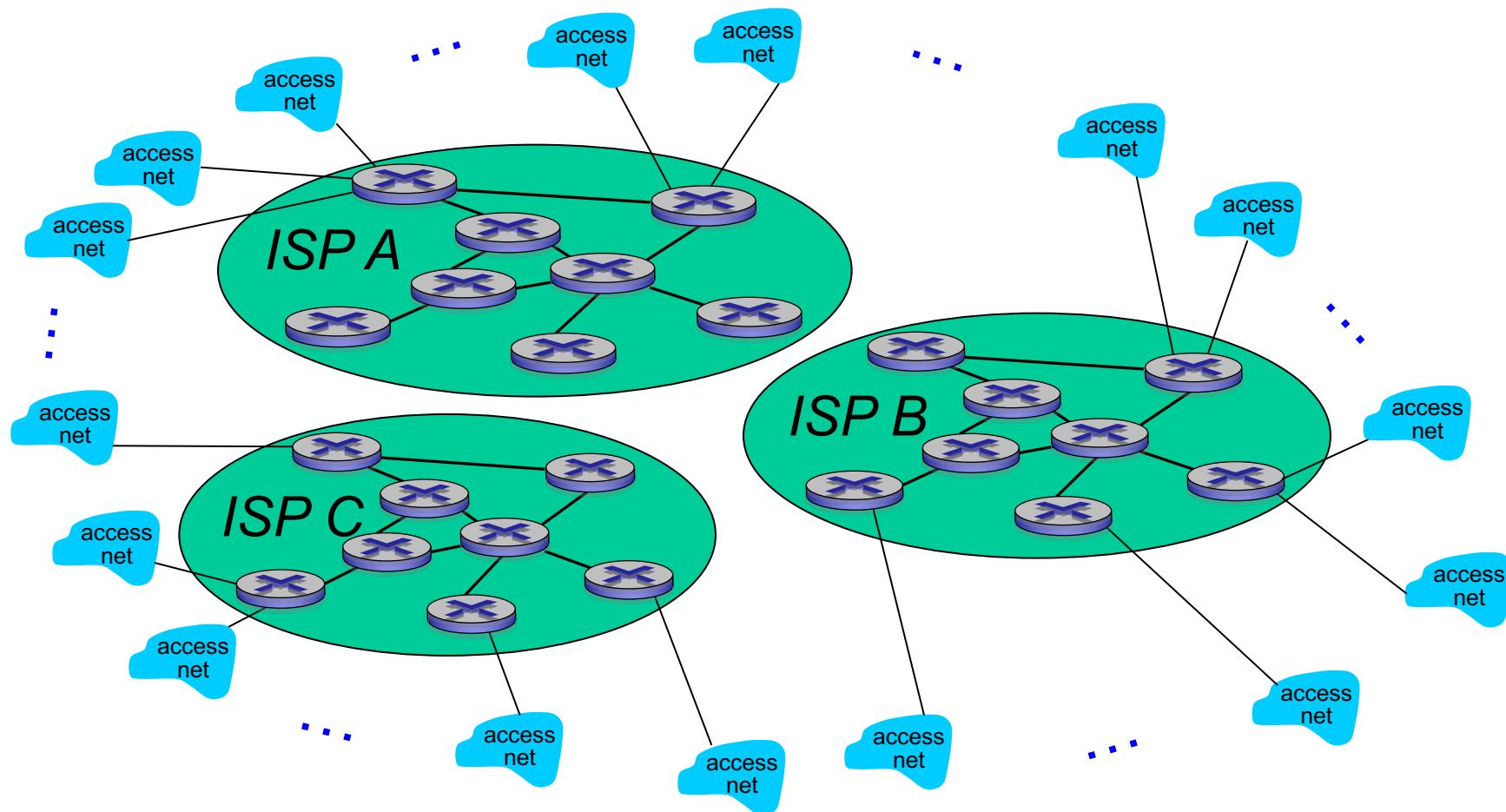
■ Option 2

- 각 액세스 ISP들을 하나의 글로벌 통과^{transit} ISP에 연결?
- 액세스 ISP와 글로벌 ISP 간 요금 합의 필요



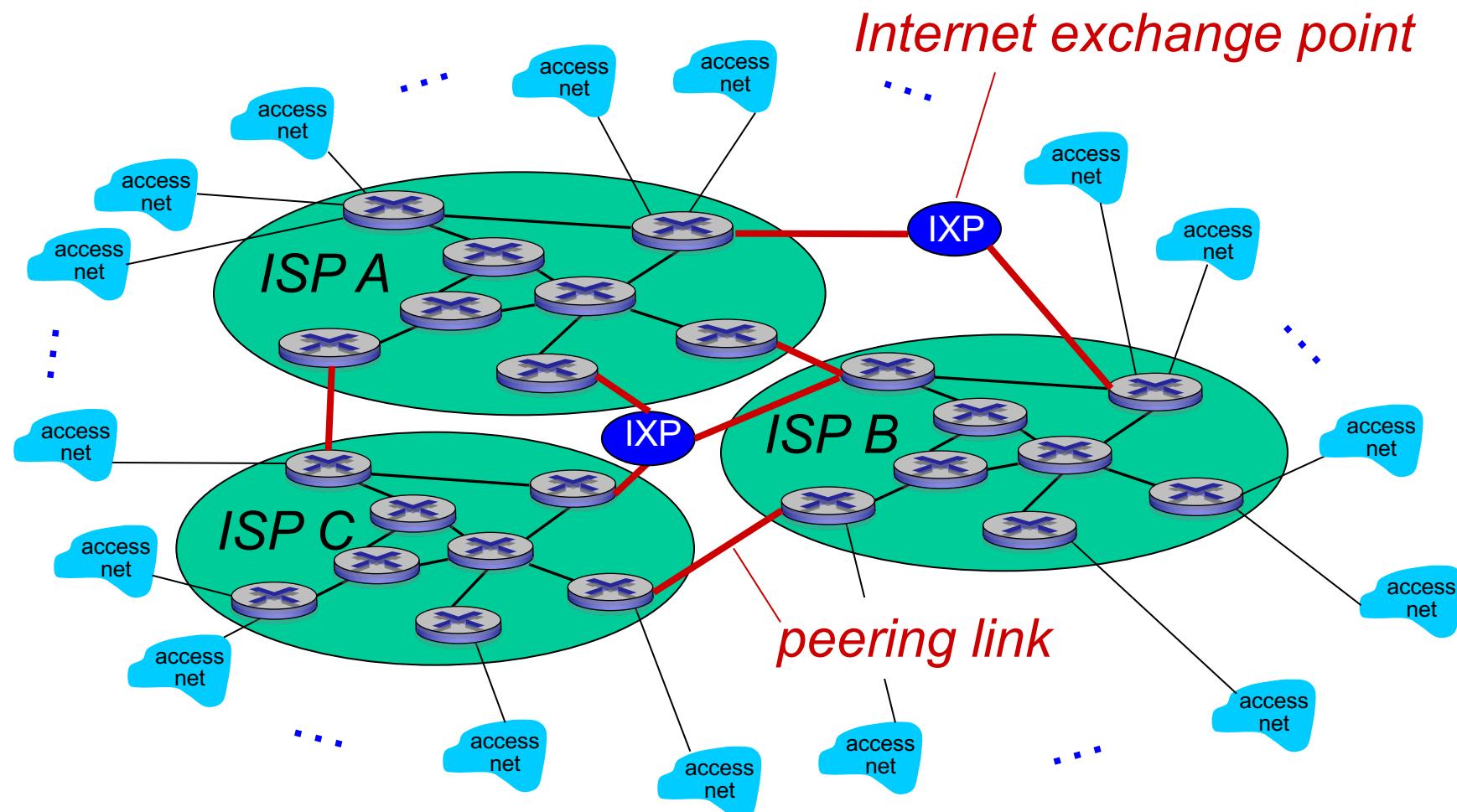
인터넷의 구조: 네트워크들의 네트워크

- 하나의 글로벌 ISP가 수익을 낸다면, 다른 새로운 글로벌 ISP도 생겨나서 경쟁하게 됨



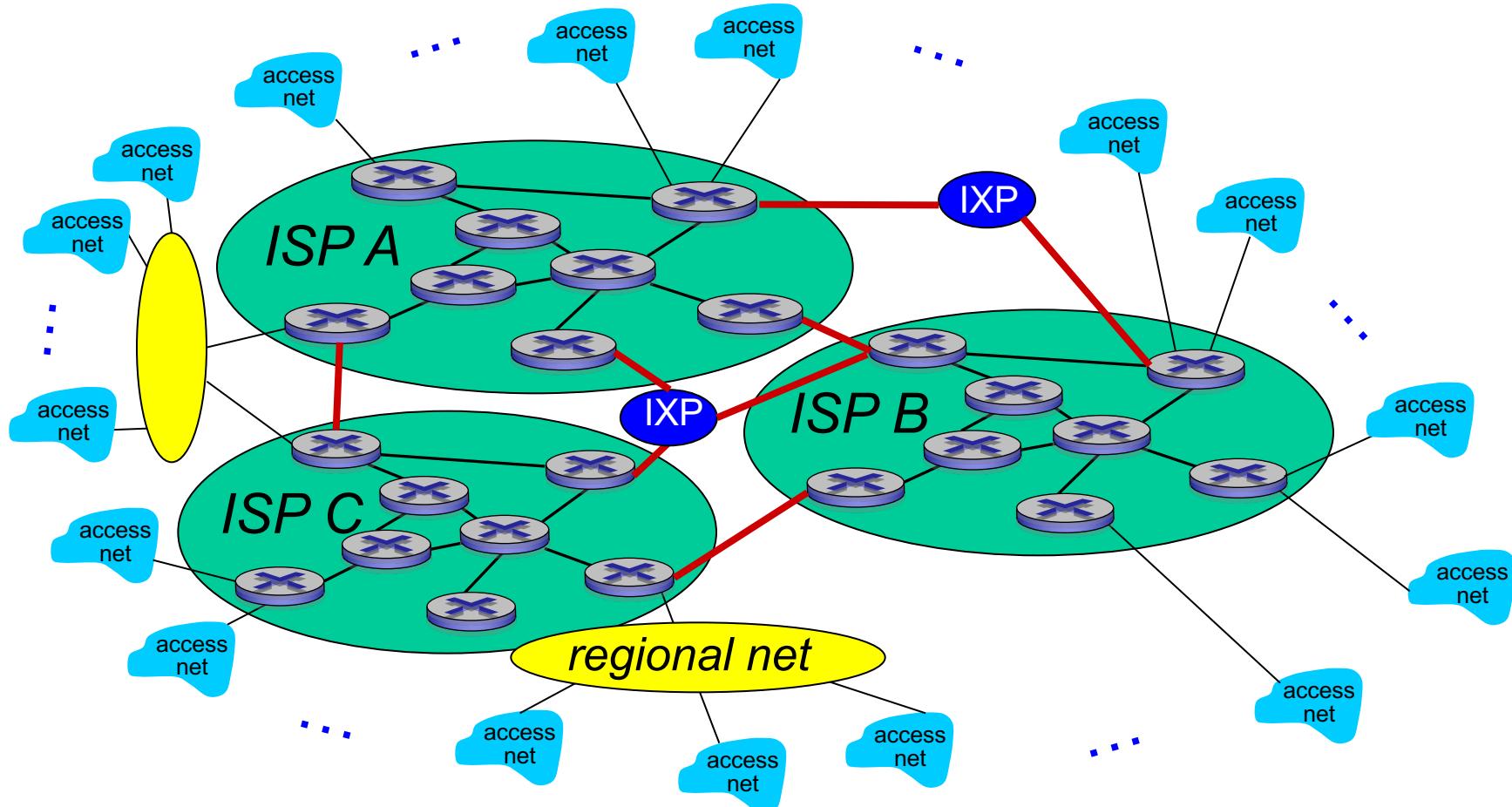
인터넷의 구조: 네트워크들의 네트워크

- 글로벌 ISP들 간에도 상호 연결되어야 함



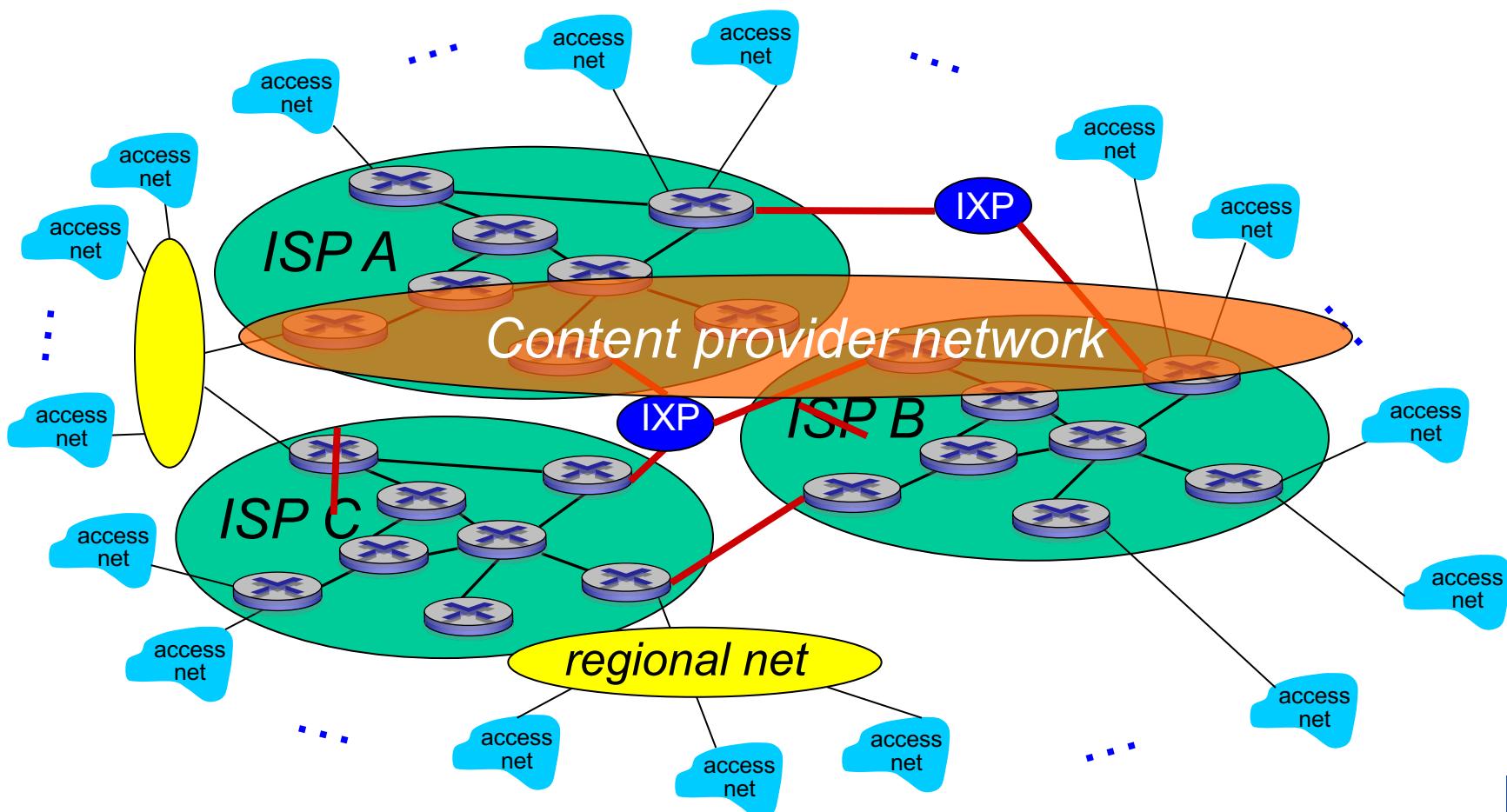
인터넷의 구조: 네트워크들의 네트워크

- 액세스 ISP를 연결하는 지역 ISP^{regional ISP}도 생겨남
 - 액세스 ISP와 글로벌 ISP를 직접 연결하기는 물리적으로 어려움

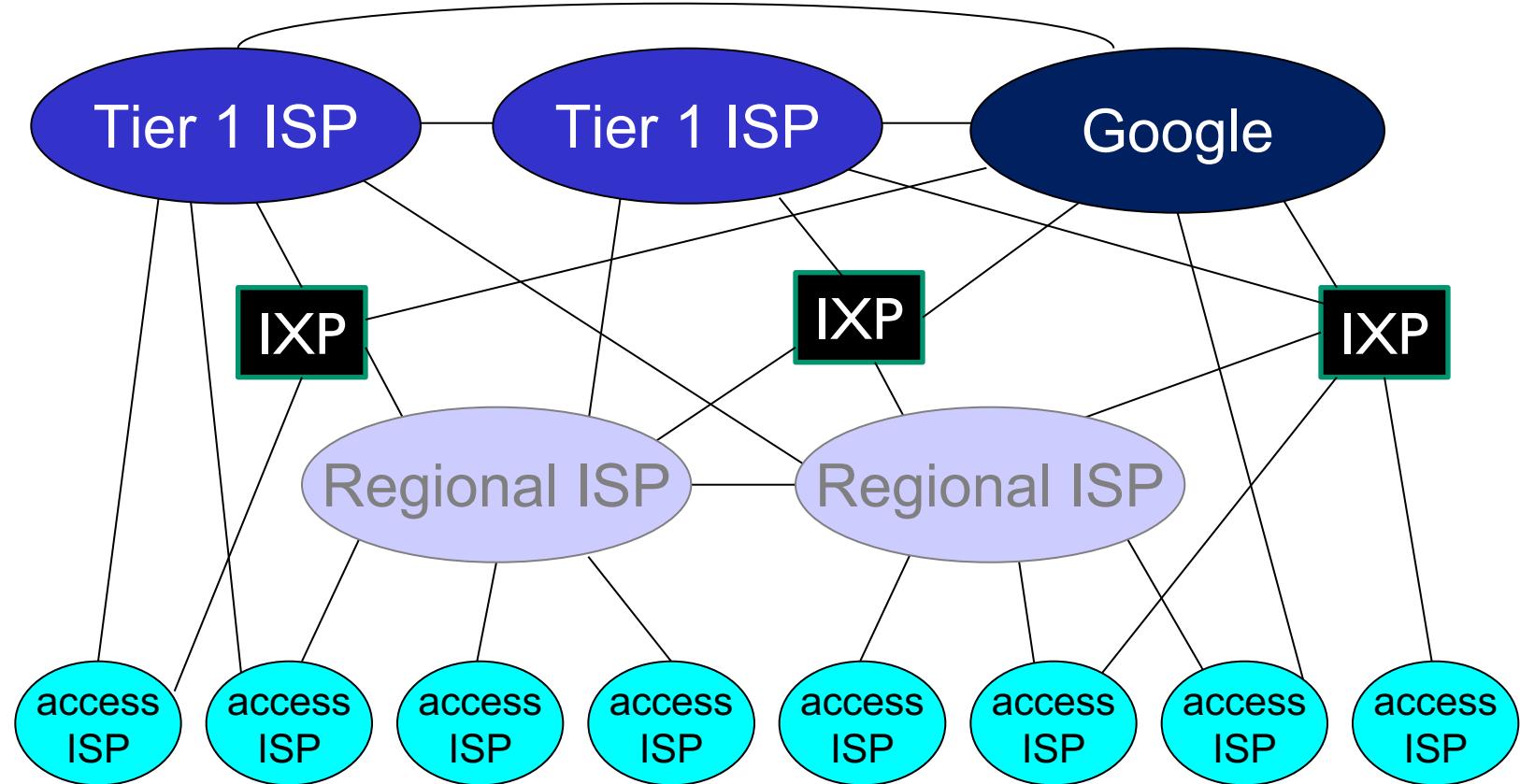


인터넷의 구조: 네트워크들의 네트워크

- 컨텐츠 제공자 네트워크 contents provider network는 서비스와 컨텐츠를 사용자에게 빠르게 제공하기 위해 자신의 네트워크 운영
 - 구글, 마이크로소프트, 페이스북 등



인터넷의 구조: 네트워크들의 네트워크



- 오늘날의 인터넷 - 네트워크들의 네트워크 - 은 매우 복잡하며, 소수의 1-계층^{tier-1} ISP들과 수십만 개의 하위 계층 ISP들로 구성
 - 1-계층 ISP(AT&T, NTT, Sprint)는 국가 간의 연결을 담당
 - 컨텐츠 제공자 네트워크는 자신의 데이터 센터와 인터넷을 연결하는 사설 네트워크를 구축, 1-계층, 지역 ISP를 우회하기도 함

컴퓨터 네트워크와 인터넷

1.1 인터넷이란 무엇인가?

1.2 네트워크 엣지

- 종단 시스템, 액세스 네트워크, 링크

1.3 네트워크 코어

- 패킷 교환, 회선 교환, 인터넷의 구조

1.4 패킷 교환 네트워크에서의 지연, 손실과 처리율

1.5 프로토콜 계층, 서비스 모델

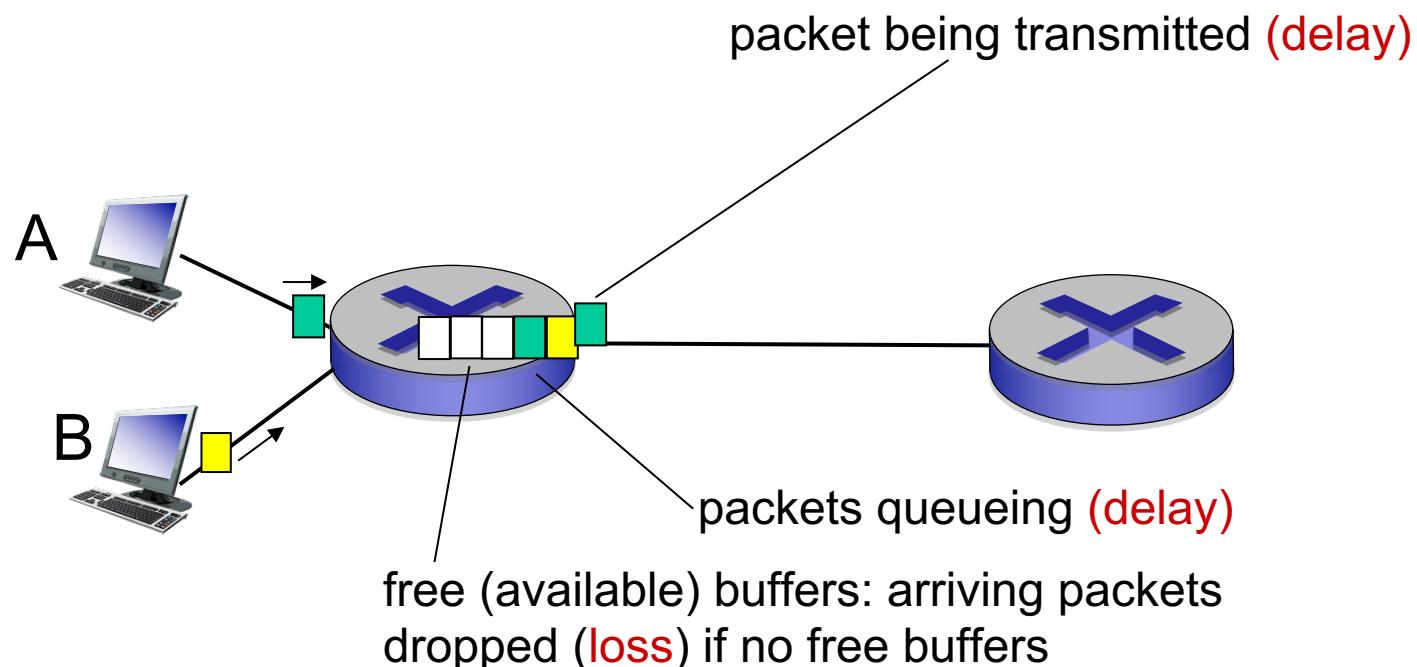
1.6 공격받는 네트워크: 보안

1.7 인터넷의 역사

손실과 지연은 왜 발생하는가?

■ 패킷은 라우터의 버퍼(큐)에서 대기

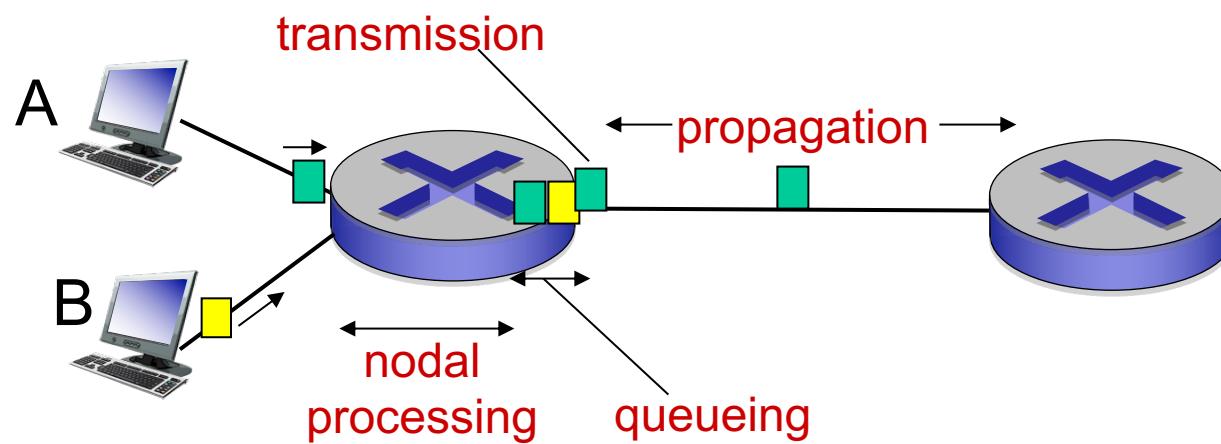
- 대기 중인 패킷이 없으면, 바로 처리
 - ✓ 전송 지연 발생
- 패킷 도착 속도가 출력 링크 용량을 초과하면, 패킷은 queue에 저장되어, 차례를 기다림
 - ✓ 큐잉 지연, 손실 발생



패킷 지연 유형

■ 패킷 지연 유형

- 노드 처리 지연 (nodal processing delay)
- 큐잉 지연 (queueing delay)
- 전송 지연 (transmission delay)
- 전파 지연 (propagation delay)



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

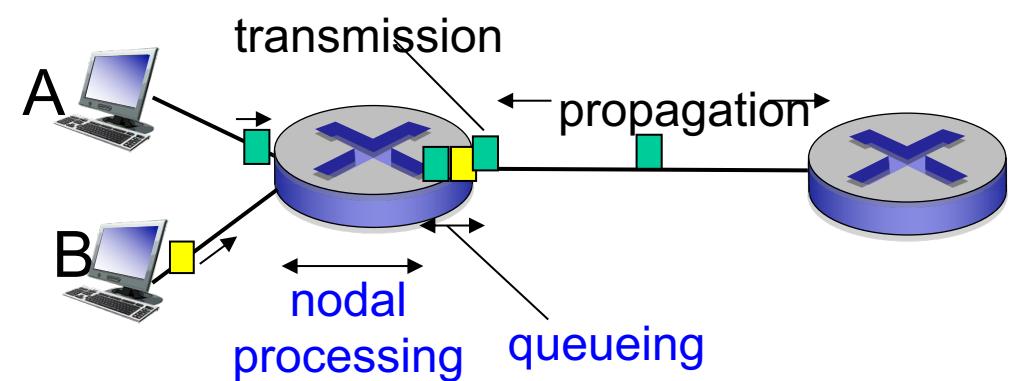
노드 처리 지연과 큐잉 지연

■ 노드 처리 지연

- 라우터에서의 패킷을 처리하는데 걸리는 시간
- 패킷의 헤더를 조사하고 어느 출력 링크로 보낼지 결정하는 시간
- 패킷의 비트 수준 오류를 조사하는데 필요한 시간
- 일반적으로 < ms

■ 큐잉 지연

- 패킷이 큐에서 출력 링크로 전송되기를 기다리는 시간
- 라우터의 혼잡 수준 *congestion level*에 좌우됨
 - ✓ 이미 큐에 저장된 패킷들의 수에 의해 결정됨



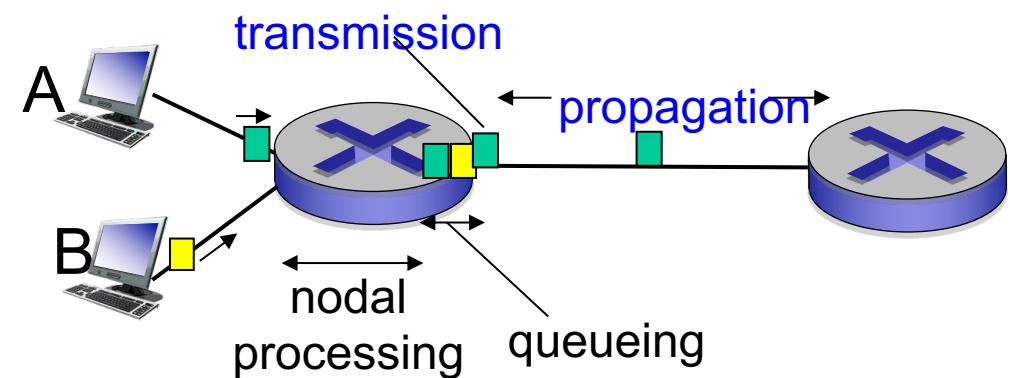
전송 지연과 전파 지연

■ 전송 지연

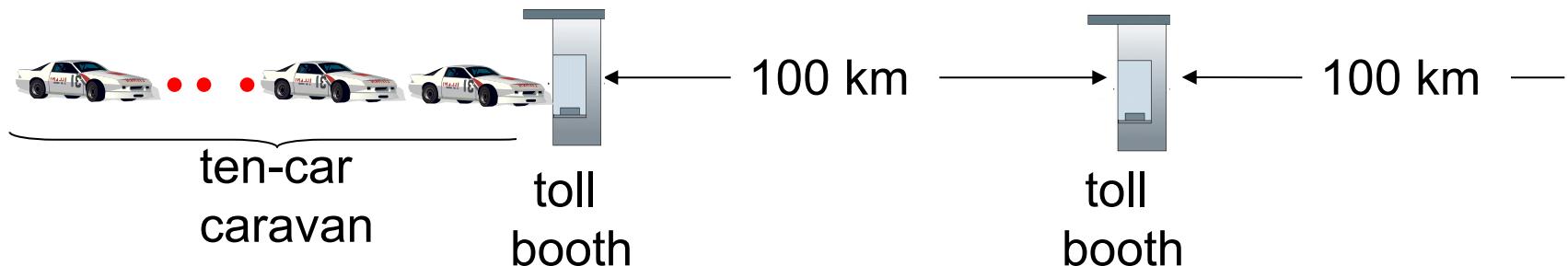
- 패킷의 모든 비트들을 링크로 밀어내는(전송)데 필요한 시간
- 저장 후 전달 지연
- 전송 지연 = L/R
 - ✓ R : 링크 대역폭 (bps)
 - ✓ L : 패킷 길이 | packet length (bits)

■ 전파 지연

- 출력 링크에서 다음 라우터까지 전파되는데 필요한 시간
- 전파 속도는 링크의 물리 매체(광섬유, UTP 등)에 좌우됨
- 전파 지연 = d/s
 - ✓ d : 두 라우터 간의 거리
 - ✓ s : 매체의 전파 속도 ($\sim 2 \times 10^8$ m/sec)



전송 지연과 전파 지연 비교: 자동차 대열 비유 (1)



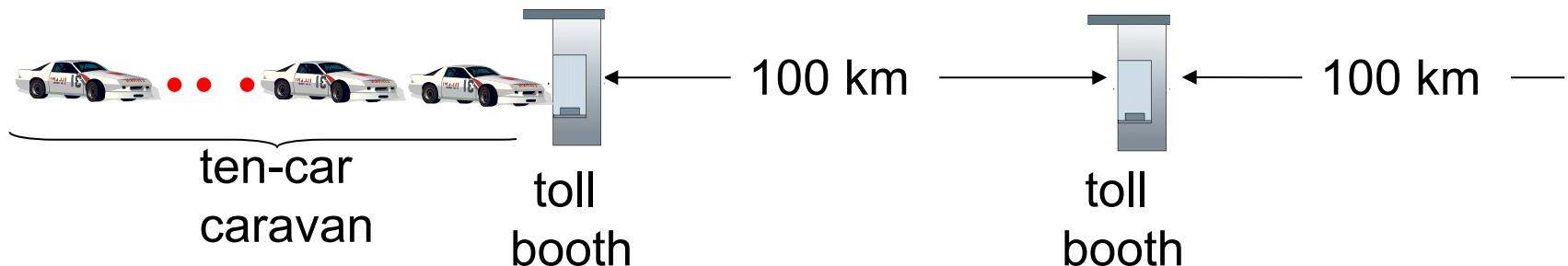
■ 함께 여행하는 10대의 자동차 대열을 가정

- 10대의 자동차 대열 (패킷), 1대의 자동차 (비트)
- 100km마다 요금 계산소 (라우터)
- 각 요금 계산소는 한대의 자동차 서비스 (전송)에 12초 걸림
- 자동차는 시속 100km로 주행 (전파)

Q. 자동차 대열이 2번째 요금 계산소에 도착할 때까지 걸리는 시간?

- 전체 자동차를 고속도로로 내보내는데 걸리는 시간
✓ $12\text{초} \times 10\text{대} = 120\text{초} = 2\text{분}$ (전송 지연)
- 한 자동차가 한 요금 계산소에서 다음 요금 계산소로 이동하는 시간
✓ $100\text{km} / 100\text{km/h} = 1\text{시간}$ (전파 지연)
- 자동차 대열이 모두 2번째 요금 계산소에 도착하는데 걸리는 시간
✓ 전송 지연 + 전파 지연 = 62분

전송 지연과 전파 지연 비교: 자동차 대열 비유 (2)



- 각 요금 계산소는 한대의 자동차 서비스 (전송)에 1분 걸림
- 자동차는 시속 1000km로 주행 (전파)

Q. 모든 차가 1번째 요금 계산소에서 서비스 받기 전에 다른 차들이 2번째 요금 계산소에 도착할 수 있는가?

- 한 자동차가 한 요금 계산소에서 다음 요금 계산소로 이동하는 시간
✓ $100\text{km} / 1000\text{km/h} = 6\text{분}$ (전파 지연)
- 7분 후에는 1번째 자동차가 2번째 요금 계산소에 도착
✓ 3대의 자동차는 여전히 1번째 요금 계산소에서 대기 중
- 모든 패킷이 첫 번째 라우터에서 전송되기 전에 패킷의 첫 번째 비트가 2번째 라우터에 도착할 수 있음

전체 노드 지연 nodal delay

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

- d_{proc} : 처리 지연
 - 일반적으로 수 마이크로 초 정도
- d_{queue} : 큐잉 지연
 - 혼잡도에 좌우
- d_{trans} : 전송 지연
 - L/R, 저속의 링크에서는 상당한 지연
- d_{prop} : 전파 지연
 - 수 마이크로 초에서 수백 밀리 초(정지 위성의 경우)

큐잉 지연

■ 트래픽 강도 traffic intensity

- La / R

- ✓ R: 출력 링크 속도 (bps)
- ✓ L: 패킷 길이 (bits)
- ✓ a: 평균 패킷 도착 속도 (packets/sec)
- ✓ La: 평균 비트 도착 속도 (bps)

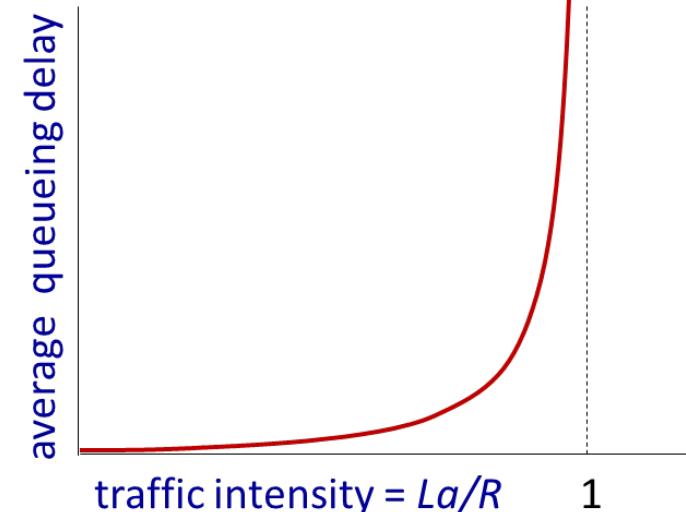
- 큐잉 지연의 정도를 측정하는 데 매우 중요

- ✓ 여기서는, 트래픽 강도와 큐가 매우 커서 무한대 비트를 저장한다고 가정

- $La/R \sim 0$: 평균 큐잉 지연 작음

- $La/R \rightarrow 1$: 평균 큐잉 지연 아주 커짐

- $La/R > 1$: 서비스 용량을 초과하여 평균 지연 무한대



$La/R \sim 0$

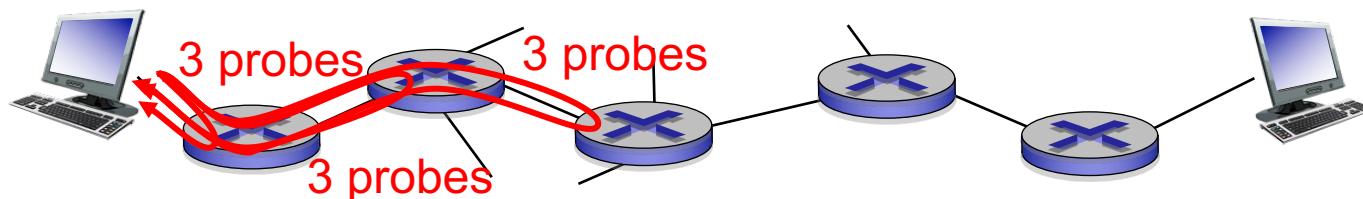


$La/R \rightarrow 1$

실제 인터넷 상에서의 지연과 경로

■ traceroute 진단 프로그램

- 지정된 목적지 경로에 따라 출발지에서 라우터까지의 지연을 측정
- 경로 상의 모든 라우터 i에 대해 다음을 수행
 - ✓ 경로 상의 라우터 i에 대해 3개의 패킷을 송신
 - ✓ 라우터 i는 송신자에게 패킷을 리턴
 - ✓ 송신자는 패킷 송신과 응답 사이 시간을 측정



실제 인터넷 상에서의 지연과 경로

traceroute: gaia.cs.umass.edu to www.eurecom.fr

3 delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu						
1	cs-gw (128.119.240.254)	1 ms	1 ms	2 ms		
2	border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)	1 ms	1 ms	2 ms		
3	cht-vbns.gw.umass.edu (128.119.3.130)	6 ms	5 ms	5 ms		
4	jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)	16 ms	11 ms	13 ms		
5	jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)	21 ms	18 ms	18 ms		
6	abilene-vbns.abilene.ucaid.edu (198.32.11.9)	22 ms	18 ms	22 ms		
7	nycm-wash.abilene.ucaid.edu (198.32.8.46)	22 ms	22 ms	22 ms		
8	62.40.103.253 (62.40.103.253)	104 ms	109 ms	106 ms		
9	de2-1.de1.de.geant.net (62.40.96.129)	109 ms	102 ms	104 ms		
10	de.fr1.fr.geant.net (62.40.96.50)	113 ms	121 ms	114 ms		
11	renater-gw.fr1.fr.geant.net (62.40.103.54)	112 ms	114 ms	112 ms		
12	nio-n2.cssi.renater.fr (193.51.206.13)	111 ms	114 ms	116 ms		
13	nice.cssi.renater.fr (195.220.98.102)	123 ms	125 ms	124 ms		
14	r3t2-nice.cssi.renater.fr (195.220.98.110)	126 ms	126 ms	124 ms		
15	eurecom-valbonne.r3t2.ft.net (193.48.50.54)	135 ms	128 ms	133 ms		
16	194.214.211.25 (194.214.211.25)	126 ms	128 ms	126 ms		
17	***					
18	***				* means no response (probe lost, router not replying)	
19	fantasia.eurecom.fr (193.55.113.142)	132 ms	128 ms	136 ms		

실제 인터넷 상에서의 지연과 경로

■ 해보기

```
C:\> tracert gaia.cs.umass.edu
```

최대 30홉 이상의

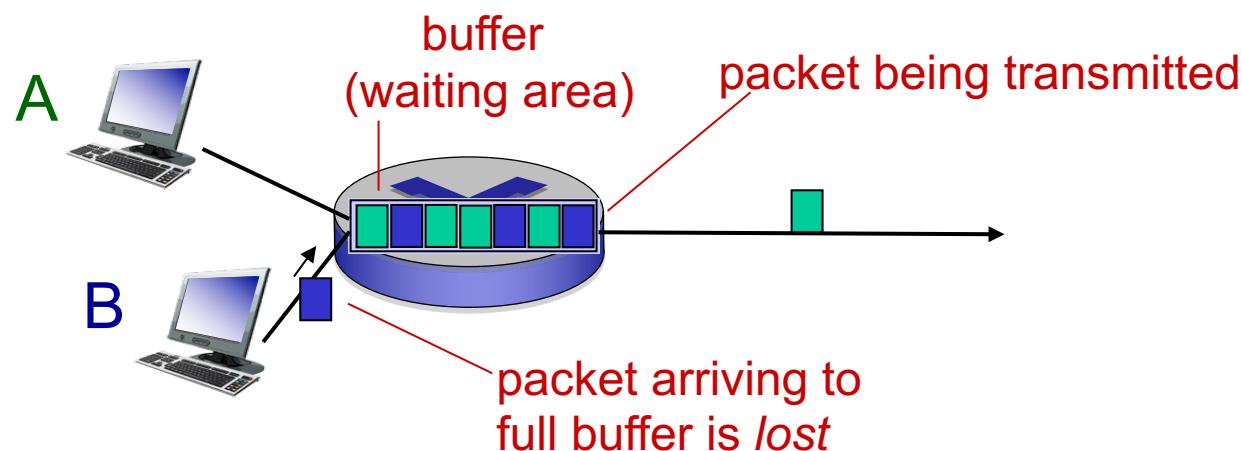
gaia.cs.umass.edu [128.119.245.12](으)로 가는 경로 추적:

1	2 ms	3 ms	2 ms	114.71.220.1
2	*	1 ms	<1 ms	10.10.180.1
3	8 ms	4 ms	5 ms	220.69.192.1
4	8 ms	10 ms	11 ms	175.204.250.5
5	9 ms	10 ms	12 ms	112.188.160.41
6	5 ms	6 ms	8 ms	112.188.165.85
7	10 ms	14 ms	10 ms	112.188.133.237
8	*	*	*	요청 시간이 만료되었습니다.
9	*	10 ms	28 ms	112.174.49.62
10	16 ms	14 ms	16 ms	112.174.84.38
11	128 ms	129 ms	130 ms	112.174.88.174
12	156 ms	156 ms	149 ms	sl-mpe51-sea-.sprintlink.net [144.224.113.125]
13	147 ms	148 ms	148 ms	sea-brdr-02.inet.qwest.net [63.146.27.73]
14	199 ms	195 ms	195 ms	cmb-edge-03.inet.qwest.net [67.14.30.158]
15	229 ms	232 ms	228 ms	core2-rt-et-4-3-0.gw.umass.edu [192.80.83.105]
16	221 ms	219 ms	214 ms	core2-rt-et-4-3-0.gw.umass.edu [192.80.83.105]
17	222 ms	*	223 ms	cics-rt-xe-0-0-0.gw.umass.edu [128.119.3.32]
18	272 ms	272 ms	271 ms	nscs1bbs1.cs.umass.edu [128.119.240.253]
19	228 ms	*	258 ms	nscs1bbs1.cs.umass.edu [128.119.240.253]
20	214 ms	209 ms	209 ms	gaia.cs.umass.edu [128.119.245.12]

추적을 완료했습니다.

패킷 손실 Packet loss

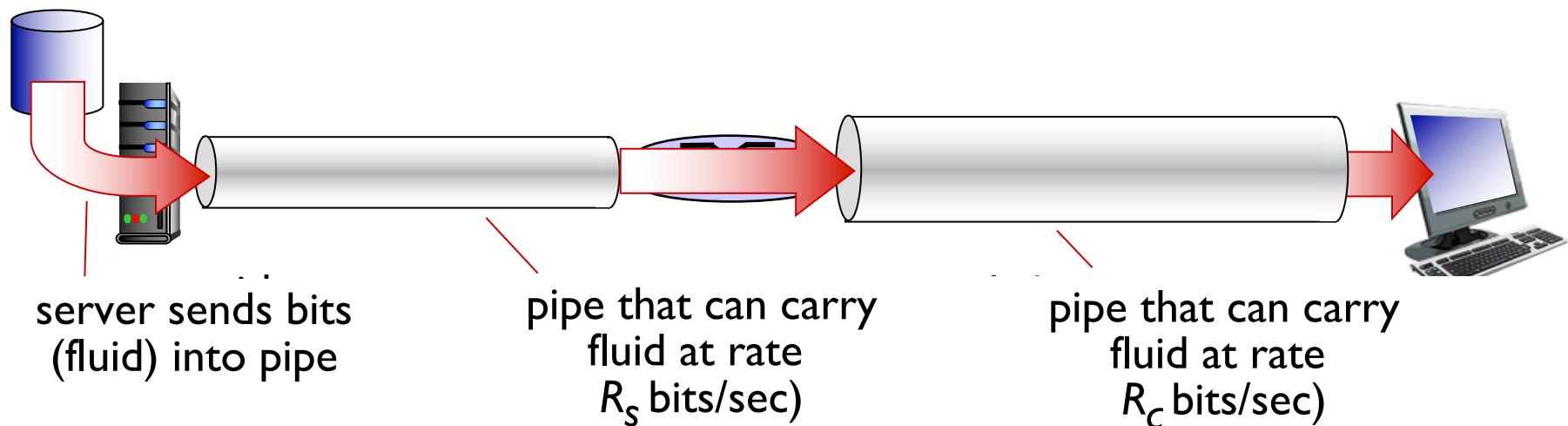
- 앞에서 큐잉 자연 분석 시 큐(버퍼)가 무한대 패킷을 저장한다고 가정하였으나...
- 실제, 라우터의 큐(버퍼)는 유한한^{finite} 용량을 가짐
 - 큐가 가득차서 도착한 패킷을 저장할 수 없으면, 패킷을 버리게 되어(drop)
패킷을 잃어버림(lost)
- 잃어버린 패킷은 이전 노드나 발신지 노드에 의해 재전송될 수 있음



처리율 Throughput

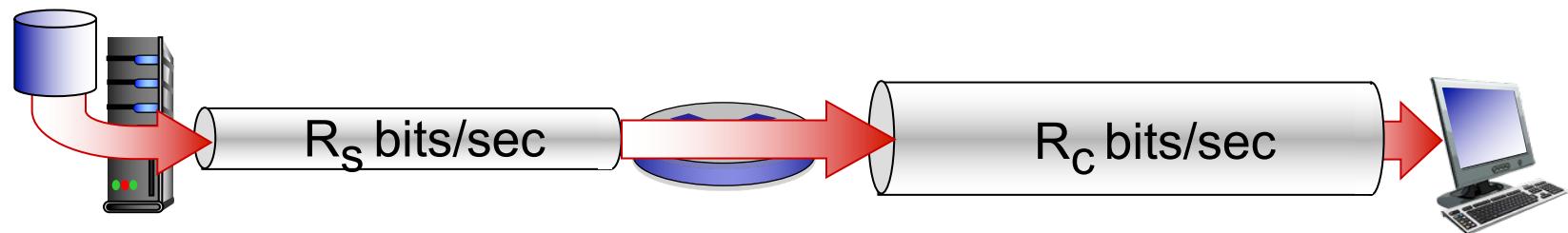
■ 처리율

- 종단(송신자와 수신자) 간 비트가 전송되는 속도 (bps, 비트/초)
- 순간적 Instantaneous 처리율
 - ✓ 주어진 순간의 전송 속도
 - ✓ 예) 파일 수신 시 특정 시점에서 파일을 수신하는 속도
- 평균 Average 처리율
 - ✓ 일정 시간 동안의 전송 속도
 - ✓ 예) 크기가 F인 파일을 모두 수신하는데 T초가 걸리면, 평균처리율은 F/T

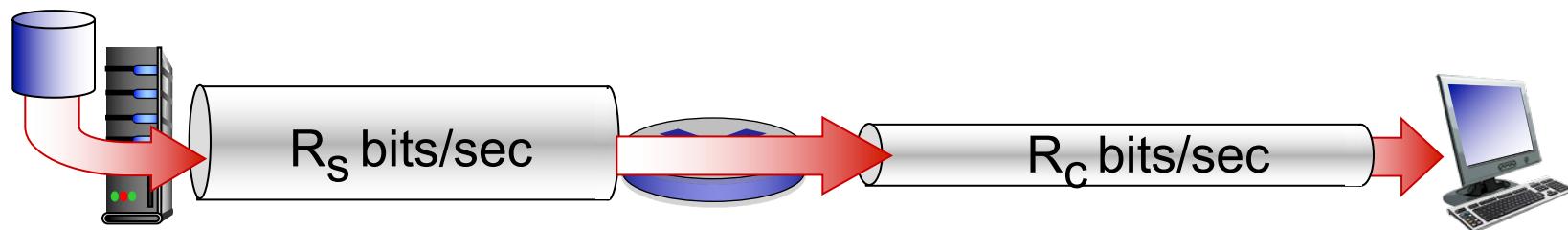


처리율 Throughput

- $R_s < R_c$: 종단 간 처리율은?



- $R_s > R_c$: 종단 간 처리율은?



- 병목 링크 bottleneck link의 처리율이 종단 간 처리율
 - 병목 링크는 종단 간 처리율을 제약하는 링크
 - 처리율은 $\min(R_s, R_c)$

처리율: 인터넷 시나리오

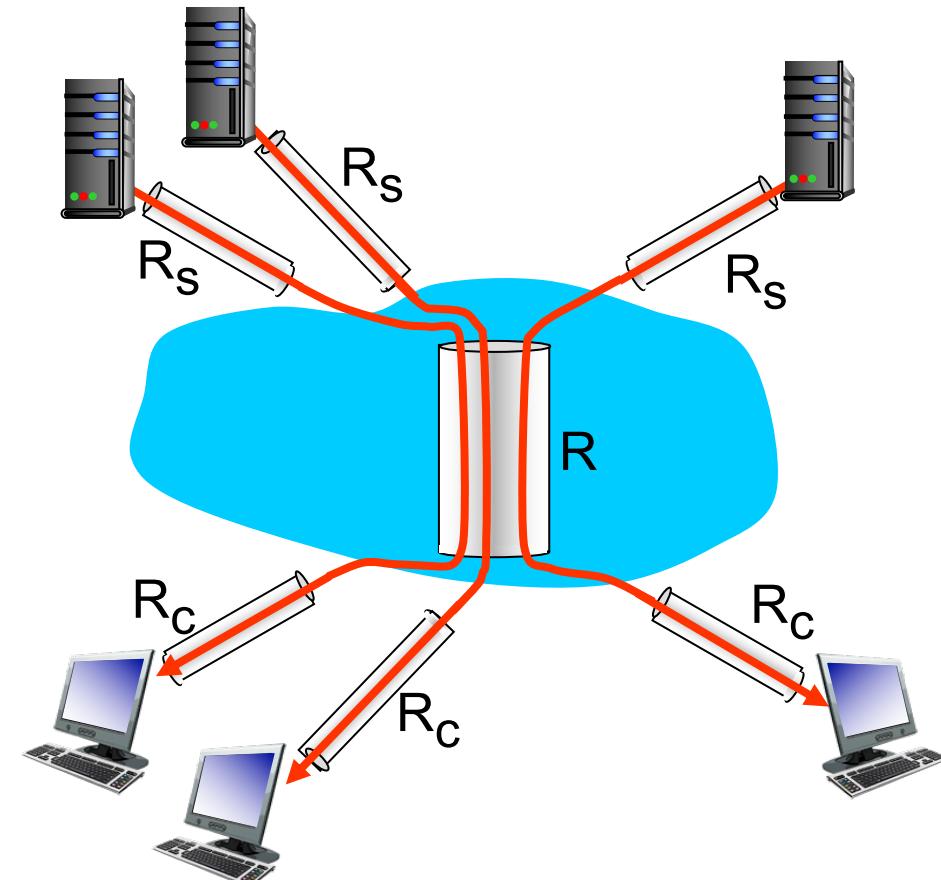
- 10개의 클라이언트가 10개의 서버로부터 다운로드하는 예

- 공통 링크(backbone)는 10개의 연결에 공평하게 전송속도를 나눈다고 가정

- 종단 간 처리율

- $\min(R_c, R_s, R/10)$

- 실제 인터넷에서는 R_c 또는 R_s 가 병목 링크임



10 connections (fairly) share
backbone bottleneck link R bits/sec

컴퓨터 네트워크와 인터넷

1.1 인터넷이란 무엇인가?

1.2 네트워크 엣지

- 종단 시스템, 액세스 네트워크, 링크

1.3 네트워크 코어

- 패킷 교환, 회선 교환, 인터넷의 구조

1.4 패킷 교환 네트워크에서의 지연, 손실과 처리율

1.5 프로토콜 계층, 서비스 모델

1.6 공격받는 네트워크: 보안

1.7 인터넷의 역사

프로토콜 계층 Protocol Layers

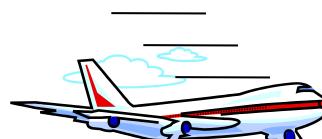
- 많은 요소로 구성된 인터넷은 **매우 복잡한 시스템**
- 많은 인터넷 구성 요소
 - 호스트, 라우터, 다양한 물리 매체로 구성된 링크
 - 다양한 애플리케이션
 - 프로토콜
 - 하드웨어, 소프트웨어

Question:

복잡한 네트워크 구조를 효율적으로 조직화 할 수 있을까?
할 수 있다면 어떻게?

비행기 여행 과정 조직화

- 비행기 여행: 일련의 여러 과정 수행

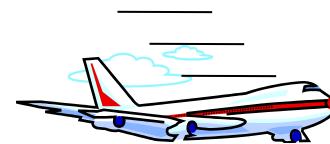


ticket (purchase)
baggage (check)
gates (load)
runway takeoff
airplane routing

ticket (complain)
baggage (claim)
gates (unload)
runway landing
airplane routing

airplane routing

항공 시스템 기능의 계층화



ticket (purchase)	<i>ticketing service</i>	ticket (complain)
baggage (check)	<i>baggage service</i>	baggage (claim)
gates (load)	<i>gate service</i>	gates (unload)
runway takeoff	<i>runway service</i>	runway landing
airplane routing	<i>routing service</i>	airplane routing

■ 계층 Layers: 각 계층은 서비스를 구현

- 자신의 계층 내부의 동작을 서비스로 구현
- 아래 계층에서 제공되는 서비스에 의존

계층화의 이유

- 계층화를 통해 **크고 복잡한 시스템**을 관리
- 명확한 구조화를 통해 복잡한 시스템 요소를 구분하고, **요소 간의 관계 설정**이 가능
 - 계층화된 참조 모델 layered reference model
 - 잘 정의된 특정 부분에 대한 논의가 가능
- 시스템의 **유지보수와 변경**이 용이
 - 한 계층의 서비스 구현이 변경되어도 시스템의 나머지 부분에 영향 없음
 - 예) 탑승구의 기능 변화가 비행 시스템 나머지에 영향을 미치지 않음
- Q. 계층화의 단점은?

인터넷 프로토콜 스택 Internet Protocol Stack

■ 애플리케이션(응용) 계층 Application layer

- 네트워크 애플리케이션을 지원
- HTTP, SMTP, IMAP

■ 트랜스포트(전송) 계층 Transport layer

- 프로세스 간 데이터 전송
- TCP, UDP

■ 네트워크 계층 Network layer

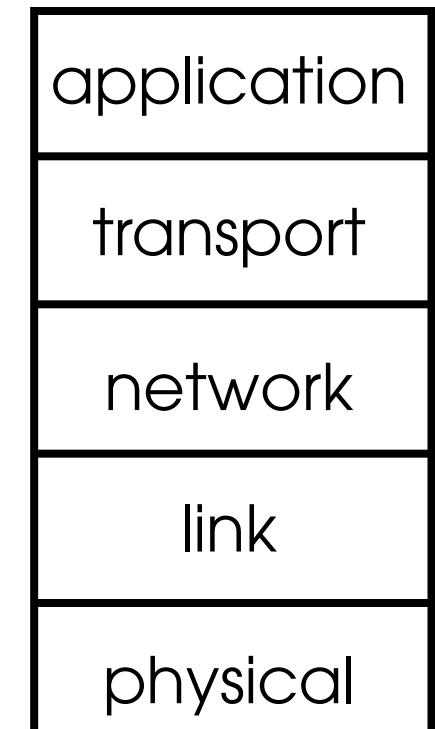
- 발신지에서 수신지까지 데이터그램을 라우팅
- IP, 라우팅 프로토콜

■ 링크 계층 link layer

- 경로 상의 인접한 노드 간 데이터 전송
- Ethernet, WiFi

■ 물리 계층 Physical layer

- 물리 매체를 통해 비트를 전송



OSI 참조 모델

■ 프리젠테이션 계층 Presentation layer

- 교환되는 데이터의 의미를 해석
- 데이터의 압축, 암호화, 호스트 내부 데이터 표시 규칙

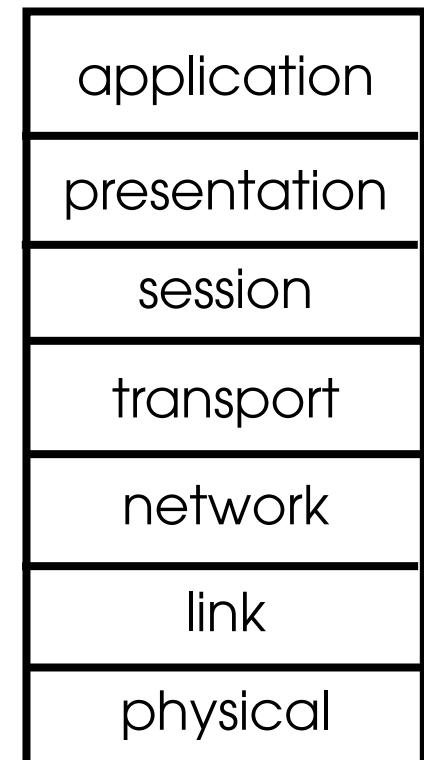
■ 세션 계층 Session layer

- 데이터 교환의 경계와 동기화

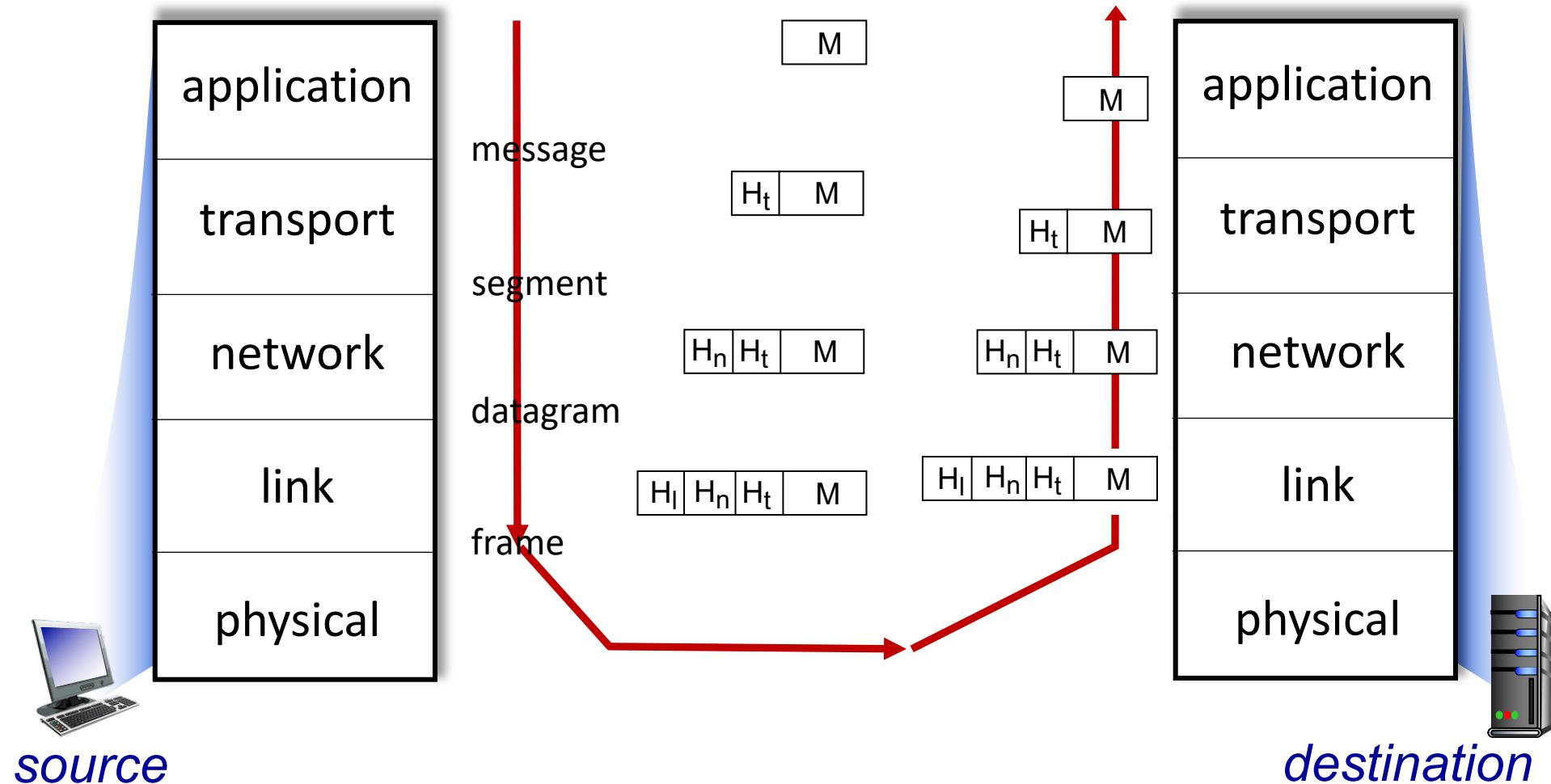
■ 인터넷 계층에는 위 두 계층이 없음

Q. 인터넷에서 이러한 서비스가 필요할 경우에는?

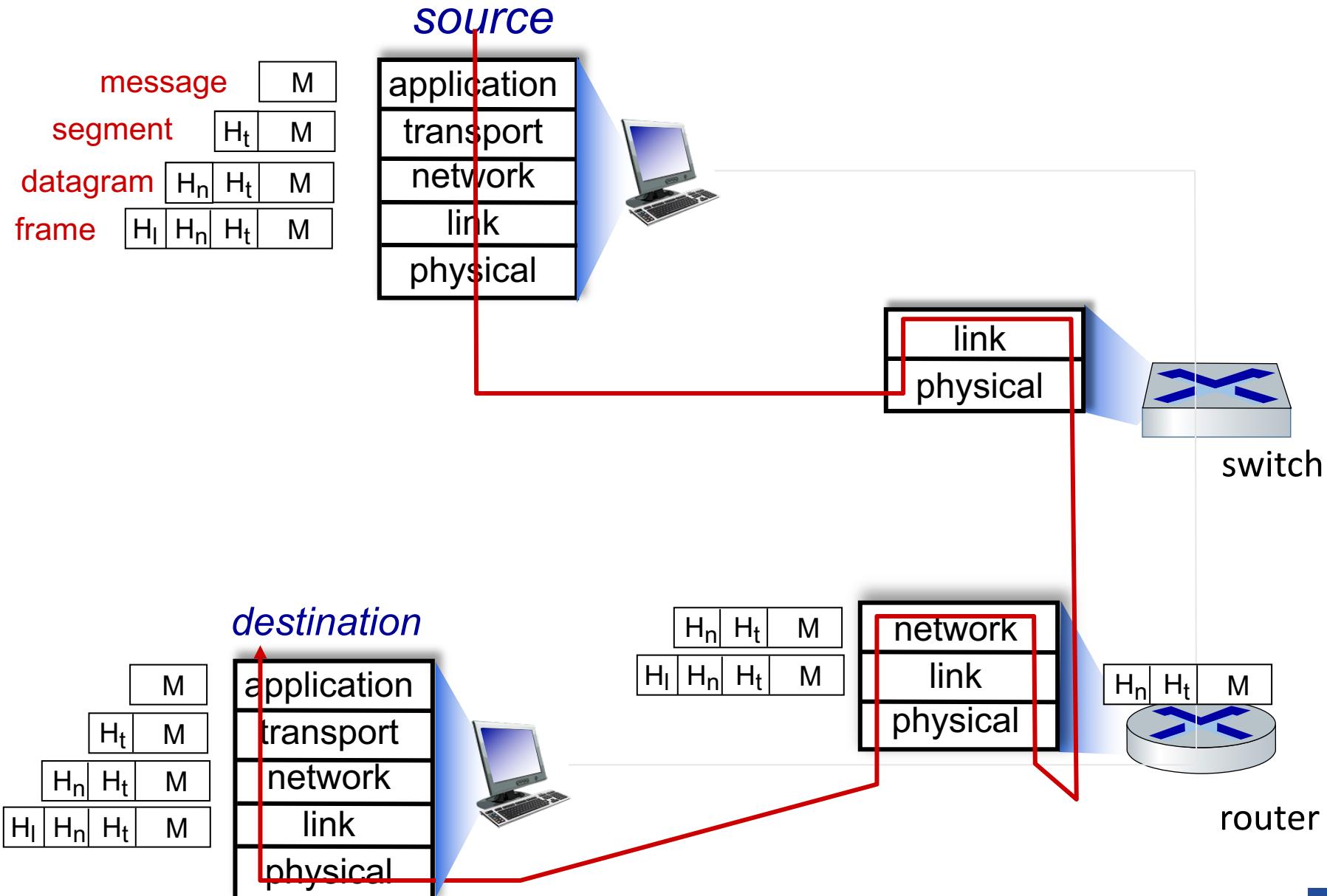
A. 필요하면, 애플리케이션 계층에서 구현해야 함



계층화 Layering / 캡슐화 Encapsulation



캡슐화: 종단 간 관점



컴퓨터 네트워크와 인터넷

1.1 인터넷이란 무엇인가?

1.2 네트워크 엣지

- 종단 시스템, 액세스 네트워크, 링크

1.3 네트워크 코어

- 패킷 교환, 회선 교환, 인터넷의 구조

1.4 패킷 교환 네트워크에서의 지연, 손실과 처리율

1.5 프로토콜 계층, 서비스 모델

1.6 공격받는 네트워크: 보안

1.7 인터넷의 역사

네트워크 보안 Network Security

■ 네트워크 보안 분야가 다루는 주제

- 나쁜 친구들(bad guys)이 어떻게 컴퓨터 네트워크를 공격(attack)할 수 있는가?
- 어떻게 이러한 공격으로부터 네트워크를 방어(defend)할 수 있는가?
- 어떻게 이러한 공격에 영향을 받지 않는 구조를 설계하는가?

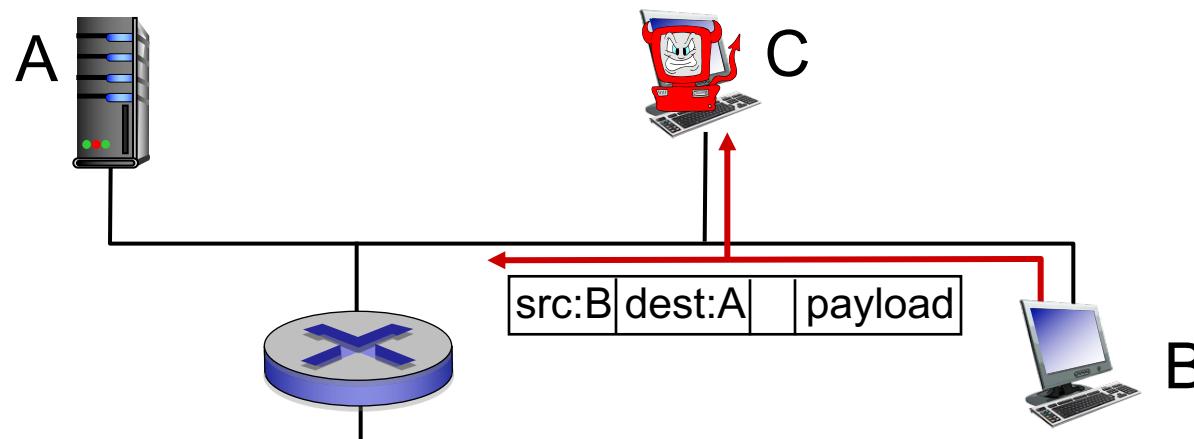
■ 초기 인터넷은 보안을 염두에 두고 설계되지 않았음

- 초기에는 서로 상호 신뢰할 수 있는 그룹들이 투명한 네트워크에 연결되었다고 가정
- 이후 인터넷 프로토콜 설계자들이 보안을 고려하여 초기 가정의 취약점을 보안
- 모든 계층에서 보안을 고려해야 함!!!

패킷 탐지 Packet Sniffing

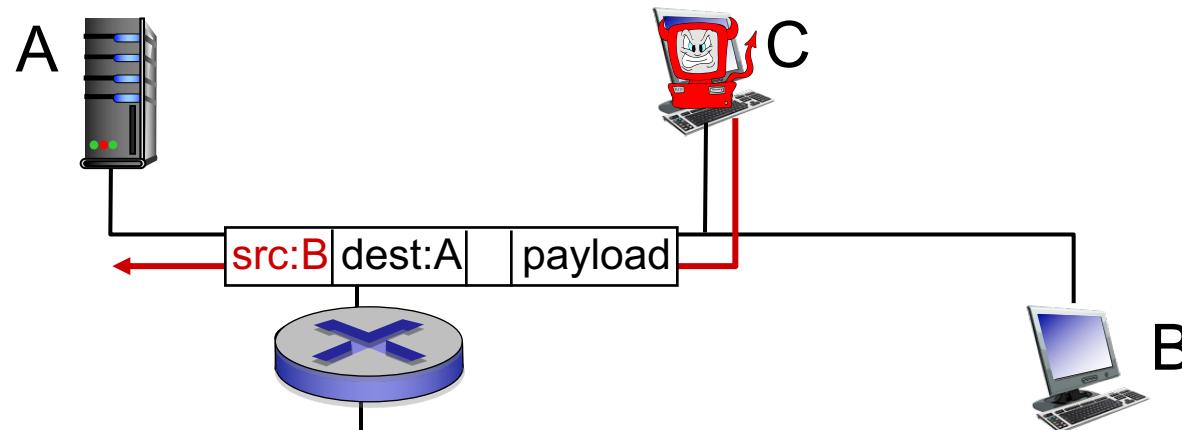
■ 나쁜 친구들은 패킷을 탐지할 수 있음

- 방송 매체 broadcast media 상에서 송수신되는 패킷의 사본을 탐지하여 읽고 기록
 - ✓ 이더넷, WiFi 환경 등
- 패킷 스니퍼 packet sniffer
 - ✓ 지나가는 모든 패킷 사본을 기록하는 수동적인 수신자
 - ✓ 본 수업에서 사용하는 Wireshark 소프트웨어도 패킷 스니퍼



위장 masquerading

- 나쁜 친구들은 신뢰하는 사람인 것처럼 “**위장**” 할 수 있음
- IP 스퓌핑 spoofing
 - 거짓의 출발지 수소를 가진 패킷을 전송
 - 한 사용자가 다른 사용자인 것처럼 위장



서버와 네트워크 기반구조 공격

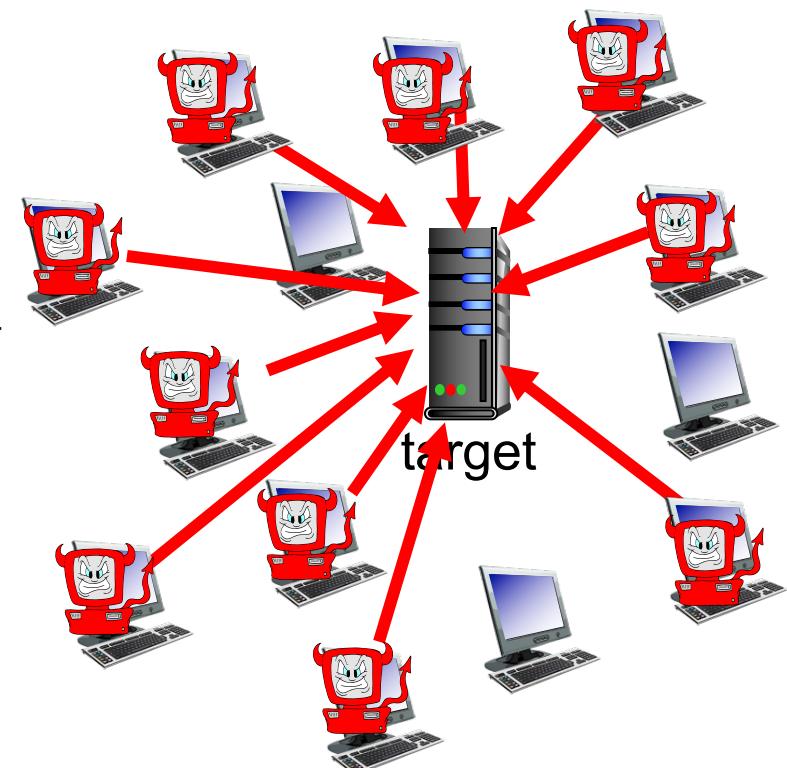
■ DoS^{Denial of Service} 공격

- 공격자가 가짜 트래픽을 발생하여 자원(서버, 대역폭 등)을 과점하여 정상적인 사용자가 사용할 수 없게 함

■ DDoS^{분산 DoS, Distributed DoS}

- 수천 개의 호스트로 구성된 봇넷 이용
- 공격자는 다수의 소스를 제어하고 각 소스는 목표 대상에게 트래픽을 전송

1. 공격대상(target) 선택
2. 공격에 사용할 호스트 침투
3. 감염된 호스트에서 패킷 전송



호스트에 맬웨어 malware 침투

■ 나쁜 친구들이 인터넷을 통해 호스트에 맬웨어 malware를 침투시킴

- 바이러스 virus

- ✓ 사용자의 장치에 영향을 주기 위해서는 사용자의 상호작용이 필요한 맬웨어
- ✓ 예) email 첨부 파일

- 웜 worm

- ✓ 사용자의 직접적인 상호작용 없이 장치에 침투하는 맬웨어
- ✓ 공격자가 취약한 네트워크 애플리케이션을 수행하여 맬웨어 송신
- ✓ 취약한 네트워크 애플리케이션을 수행하는 다른 호스트들을 찾고자 인터넷을 스캔 scan하여 자기복제

■ 스파이웨어 맬웨어 spyware malware

- 키보드 입력, 방문한 웹사이트, 업로드 정보 등을 기록하여 수집 사이트에 전달

■ 감염된 호스트는 봇넷 botnet에 등록되어 스팸 spam, DDoS Distributed Denial-of-Service 공격에 사용됨

- 봇넷: 감염된 장치들로 구성된 네트워크

컴퓨터 네트워크와 인터넷

1.1 인터넷이란 무엇인가?

1.2 네트워크 엣지

- 종단 시스템, 액세스 네트워크, 링크

1.3 네트워크 코어

- 패킷 교환, 회선 교환, 인터넷의 구조

1.4 패킷 교환 네트워크에서의 지연, 손실과 처리율

1.5 프로토콜 계층, 서비스 모델

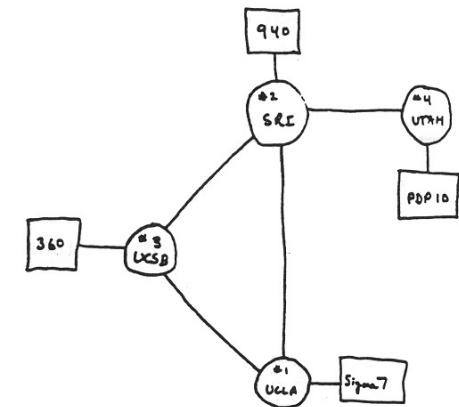
1.6 공격받는 네트워크: 보안

1.7 인터넷의 역사

인터넷의 역사

■ 초기 패킷 교환 개발 (1961-1972)

- 1961: Kleinrock - 큐잉 이론을 사용하여 패킷 교환의 효율성을 보여줌
- 1964: Baran - 군사 네트워크에 패킷 교환 적용을 조사
- 1967: ARPA(Advanced Research Projects Agency)에서 ARPAnet 계획
- 1969: 첫 번째 ARPAnet 노드 동작
 - ✓ 4개의 노드로 구성
 - ✓ 첫 번째 패킷 교환 컴퓨터 네트워크, 인터넷의 직계 원조
- 1972:
 - ✓ ARPAnet 공개 시연
 - ✓ NCP(Network Control Protocol): 첫 번째 호스트 간 프로토콜
 - ✓ 최초 전자메일 프로그램
 - ✓ ARPAnet은 15개 노드로 확장

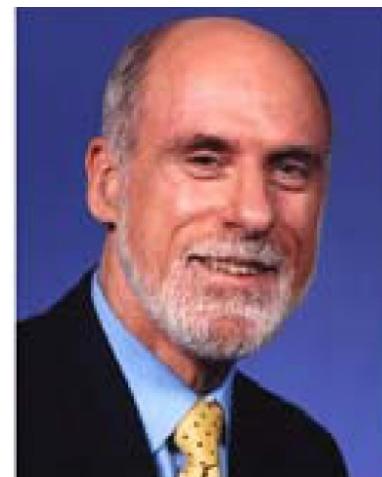


THE ARPANET

인터넷의 역사

■ 독점 네트워크와 인터네트워킹 (1972-1980)

- 1970: 하와이에서 ALOHAnet 위성 네트워크
- 1974: Cerf and Kahn - 상호 연결 네트워크인 인터네트워킹internetworking 네트워크 구조
 - ✓ 이 구조의 원리는 TCP 프로토콜로 구체화
- 1976: Xerox PARC에서 이더넷 프로토콜 개발
- 1970년대 후반: 독점 네트워크proprietary network
 - ✓ DECnet, SNA, XNA
- 1970년대 후반: 고정된 길이의 패킷 교환
 - ✓ ATM의 전신
- 1979: ARPAnet이 200개 노드로 확장



빈트 세프



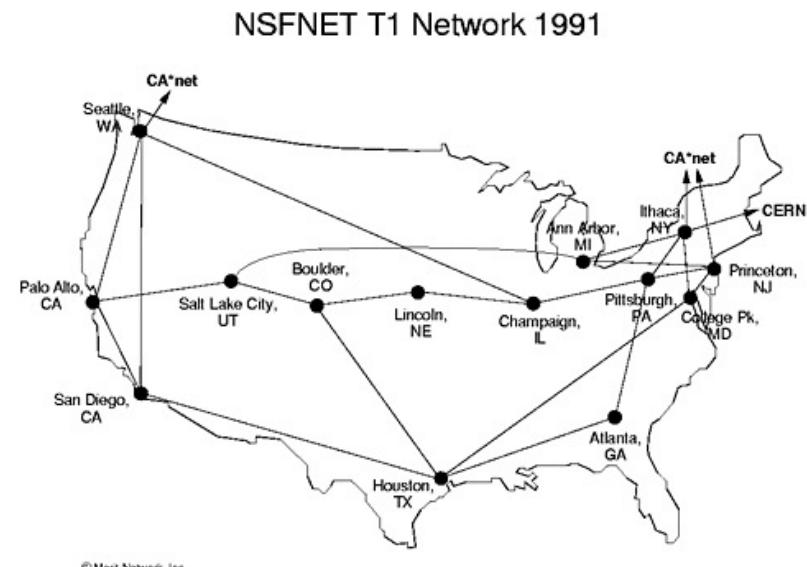
로버트 칸

인터넷의 역사

■ 새로운 프로토콜, 네트워크의 확산 (1980-1990)

- 1983: ARPAnet의 새로운 호스트 표준 프로토콜 **TCP/IP**가 공식 적용
- 1982: SMTP 전자메일 프로토콜 정의
- 1983: 이름-IP 주소 변환 **DNS** 정의
- 1985: FTP 프로토콜 정의
- 1988: TCP 혼잡 제어 **congestion control**

- 새로운 네트워크들의 등장
 - ✓ Cernet, BITnet, NSFNET, Minitel
- 10만개의 호스트들이 네트워크들의 동맹으로 연결



인터넷의 역사

■ 인터넷 상업화, 웹, 새로운 애플리케이션 (1990, 2000년)

- 1990년대 초반: ARPAnet 종료
- 1991: NSFNET 상업화 제한 해제
 - ✓ 1995년 NSFNET 종료
- 1990년대 초반: 웹(WWW, World Wide Web) 출현
 - ✓ 하이퍼텍스트 [hypertext](#)
 - ✓ HTML, HTTP: Berners-Lee
 - ✓ 1994: Mosaic, Netscape
 - ✓ 1990년대 후반: 웹의 상업화
- 1990년대 후반 - 2000년대:
 - ✓ 더 많은 컬러 애플리케이션: 인스턴트 메시징, P2P 파일 공유
 - ✓ 네트워크 보안 전면 등장
 - ✓ 5천만개 호스트, 1억 이상의 사용자
 - ✓ 백본 [backbone](#) 링크가 Gbps 급에서 동작



인터넷의 역사

■ 2005 ~ 현재: new application, Internet is ‘everywhere’

- 인터넷에 연결된 ~180억개의 디바이스 (2017)
 - ✓ 스마트폰, 태블릿, 다양한 사물들
- 광대역 접속의 확산
- 고속 무선 접속을 활용한 유비쿼터스 접속: 4G/5G, WiFi
- 온라인 **소셜 네트워크** 출현
 - ✓ 페이스북: ~25억명의 사용자
- 서비스 제공자(구글, 페이스북, 마이크로소프트)가 자신들의 네트워크 구축
 - ✓ 인터넷을 우회하여 검색, 비디오 컨텐츠, 이메일 등 빠른 접속 제공
- **클라우드**^{cloud} 상에서 전자상거래, 대학, 기업 등이 자신의 서비스를 수행
 - ✓ 아마존 EC2, 구글 application engine, 마이크로소프트 Azure
- IoT, 빅데이터, 인공지능

요약

■ 컴퓨터 네트워크 전 분야를 소개

- 인터넷의 개요
- 프로토콜
- 네트워크 엣지
 - ✓ 액세스 네트워크
- 네트워크 코어
 - ✓ 패킷 교환 vs 회선 교환
 - ✓ 인터넷 구조
- 성능: 손실, 지연, 처리율
- 계층, 서비스 모델
- 보안
- 인터넷의 역사