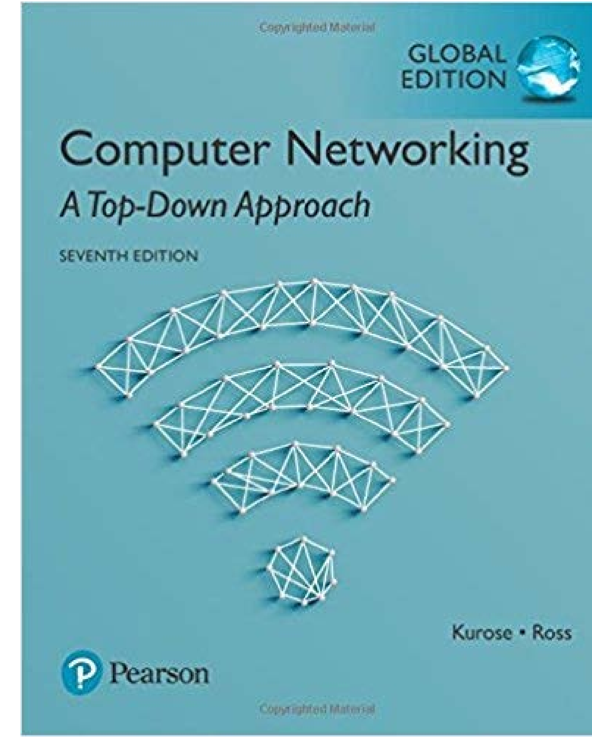
Chapter 1
Introduction – part 2

School of Computing
Gachon Univ.
Joohyung Lee

Most of slides from J.F Kurose and K.W. Ross. And, some slides from Prof. Joon Yoo



Computer Networking: A Top Down Approach

7^{th} edition
Jim Kurose, Keith Ross
Pearson, 2017

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

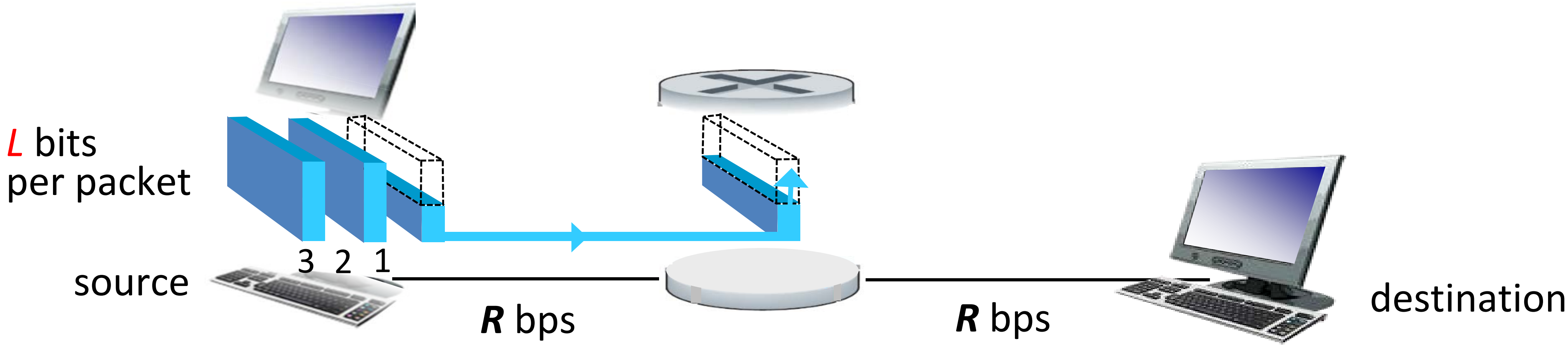
1.3 network core

- packet switching, circuit switching

- 1.4 delay, loss, throughput in networks

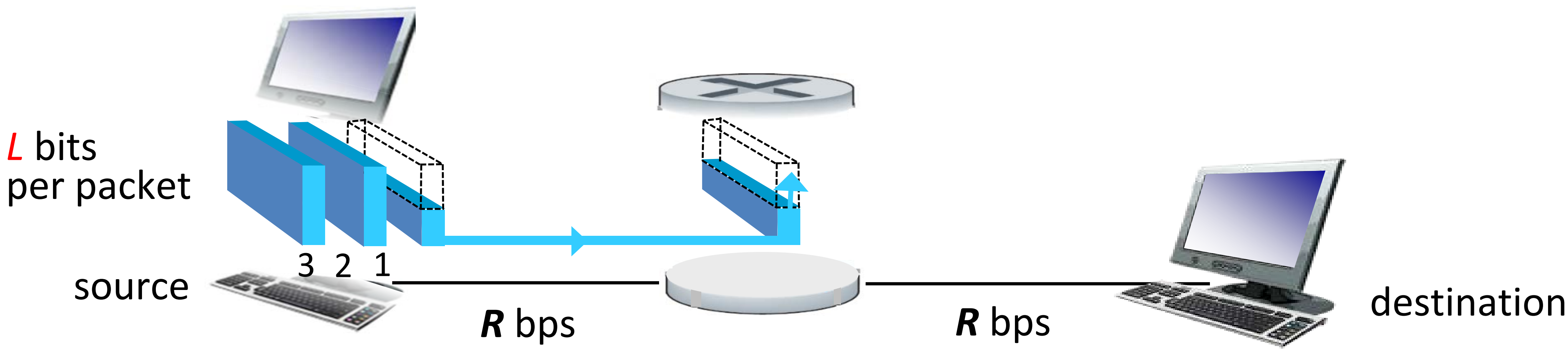
1.5 protocol layers, service models, network structure

Packet-switching: store-and-forward



- takes *L/R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
- one-hop **transmission delay** = L/R
- end-to-end **delay** = (L/R) x #hops

Packet-switching: store-and-forward



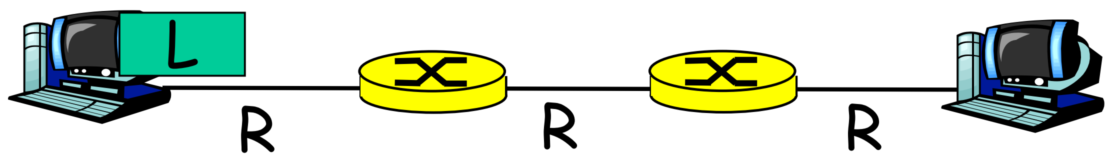
one-hop numerical example:

- L = 7.5 Mbits
- R = 1.5 Mbps
- Assuming no propagation delay

❖ one-hop transmission delay = 5 sec

❖ end-end delay = 10 sec

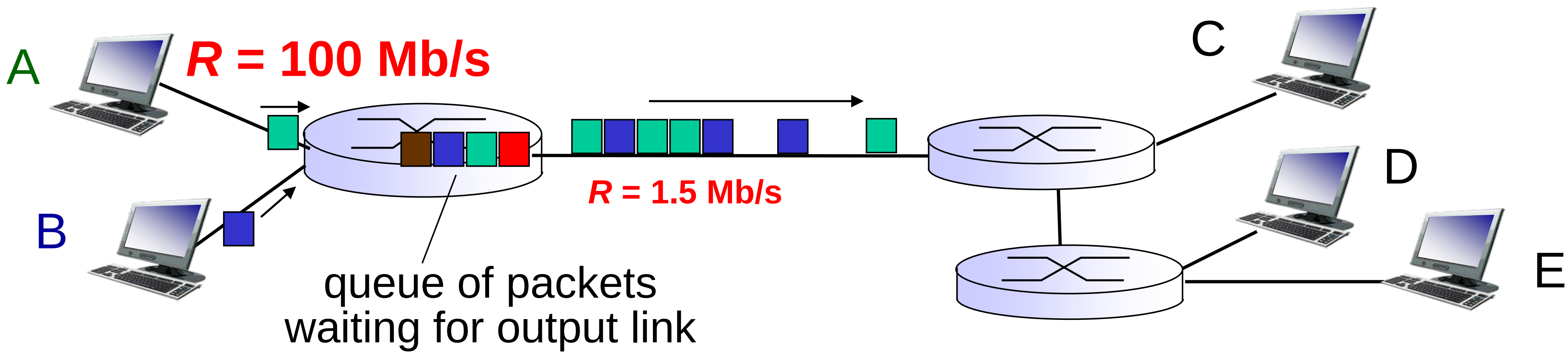
3 links example



- end-to-end delay = 3L/R

Example:
- L = 7.5 Mbits
- R = 1.5 Mbps
- delay = 15 sec

Packet Switching: queueing delay, loss



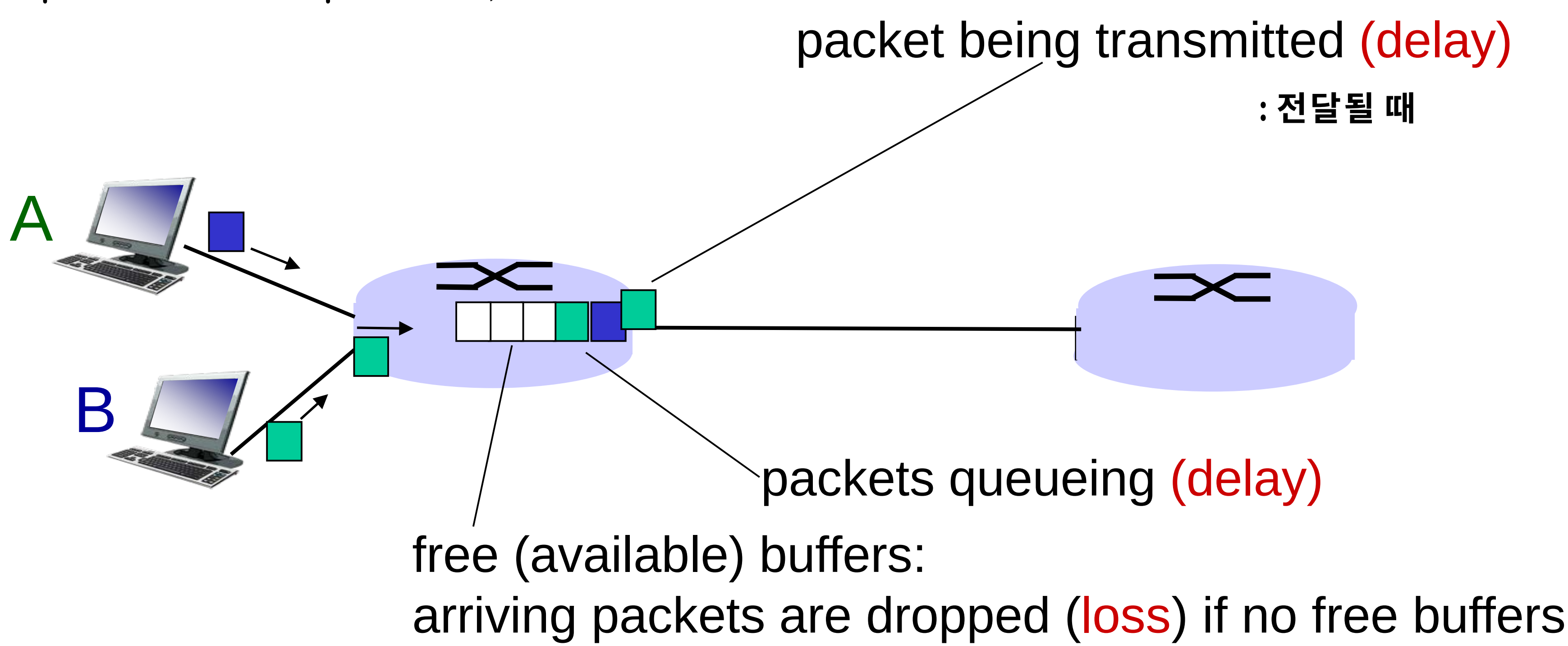
queuing and loss:

- If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  - packets will queue, wait to be transmitted on link – queueing delay
  - packets can be dropped if memory (buffer) fills up – loss

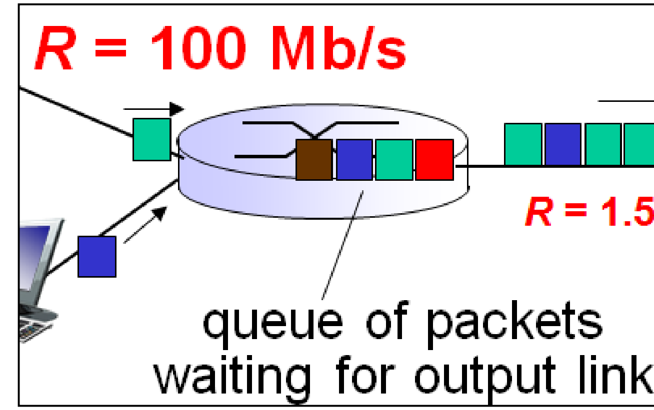
How do loss and delay occur?

packets *queue* in router buffers

- packet arrival rate to link (temporarily) exceeds output link capacity
- packets queue, wait for turn

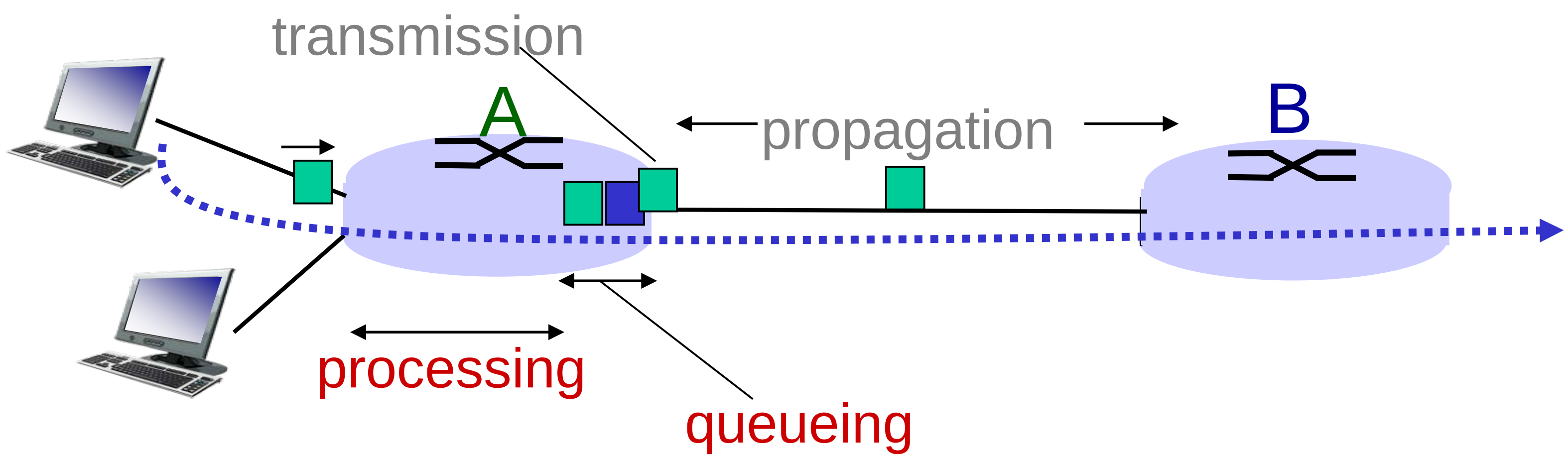


loss and delay



- Queueing delay
  - 10 packets arrive at router at the same time
    - *Currently the router buffer is empty*
  - First packet has no queueing delay
  - 10th packet has to wait for 9 other packets to be transmitted – **queueing delay** occurs
- Loss
  - If the router buffer is full
  - Next arriving packet will be **dropped** – **packet loss** occurs
- Queueing delay and loss will increase as traffic intensity (i.e., congestion) increases

Four sources of packet delay



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

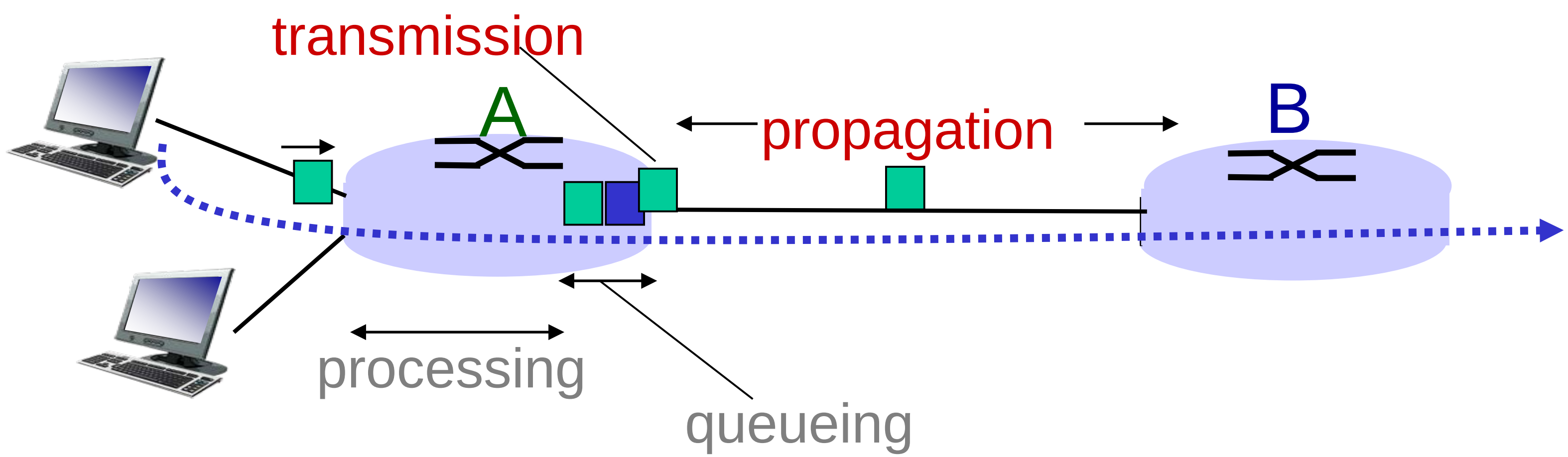
1. d_{proc}: processing delay

- examines packet header
- determine where to direct packet (=output link), e.g., link to router B.
- typically usec

2. d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on earlier arriving packets that are queued and waiting
- typically usec ~ msec

Four sources of packet delay



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

3. d_{trans}: transmission delay:

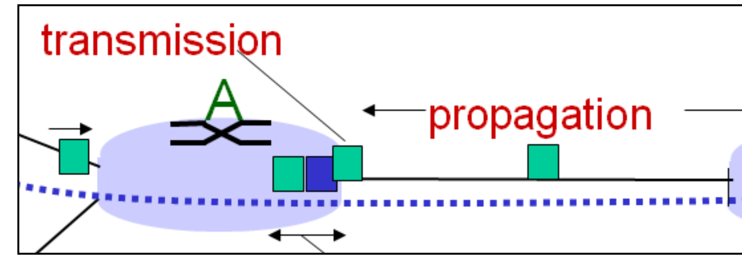
- time to transmit all the packet's bits into the link (e.g., link AB)
- $d_{trans} = L/R$
  - L: packet length (bits)
  - R: link *bandwidth (bps)*

4. d_{prop}: propagation delay:

- time to propagate from beginning of the link to next router (e.g., router B)
- $d_{prop} = d/s$
  - d: length of physical link
  - s: propagation speed in medium (~2×10^8 m/sec) – similar to speed of light

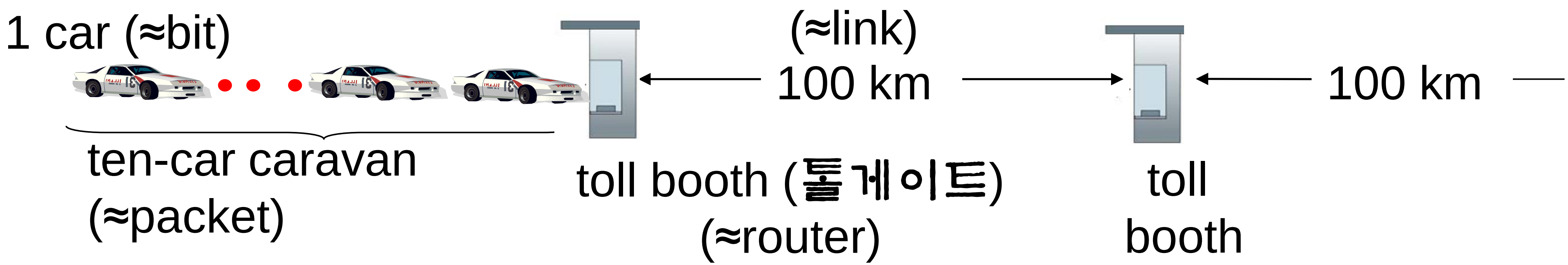
Transmission vs. Propagation Delay

- Transmission (전송) delay
  - Time for router to push out the packet
  - Function of packet's length (L bits) and transmission rate (R bps) of link
    - $d_{trans} = L/R$
  - Nothing to do with distance between two routers
- Propagation (전파) delay
  - Time a bit to propagate from one router to the next
  - Function of distance between the two routers
    - $d_{prop} = d/s$
  - Nothing to do with packet's length or the transmission rate of link



propagation delay는 data size랑 상관 x, link의 길이를 빛의속도를 나눈 값이 걸리는시간
거리랑 상관 많음.
transmission delay만 data size와 상관있음

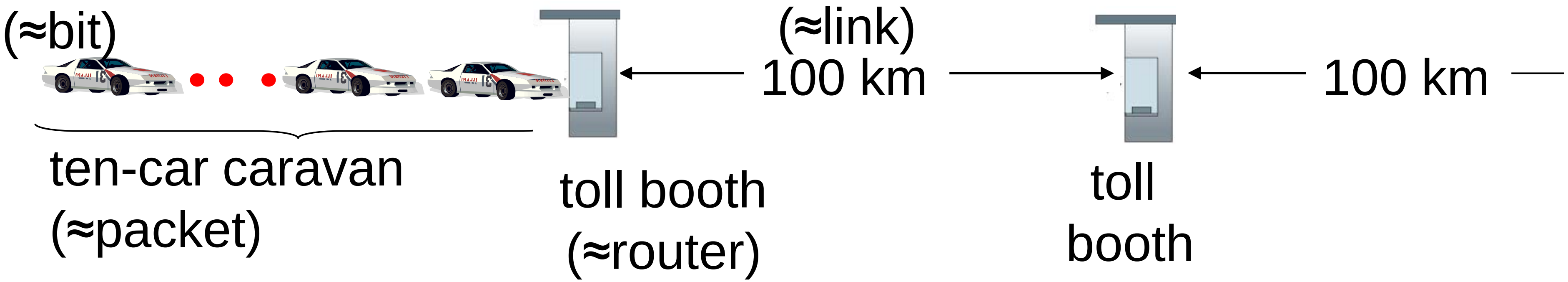
Caravan analogy



- cars “propagate” at 100 km/hr
- toll booth takes 12 sec to service one car (1 bit transmission time)
- car ≈ bit; caravan ≈ packet
- *Q:* How long until caravan is lined up before 2nd toll booth?

- **Transmission delay**: time to “push” entire caravan through toll booth onto highway = 12*10 = 120 sec (=2 minutes)
- **Propagation delay**: time for last car to propagate from 1st to 2nd toll booth: 100km/(100km/hr)= 1 hr (= 60mins)
- *A:* 62 minutes

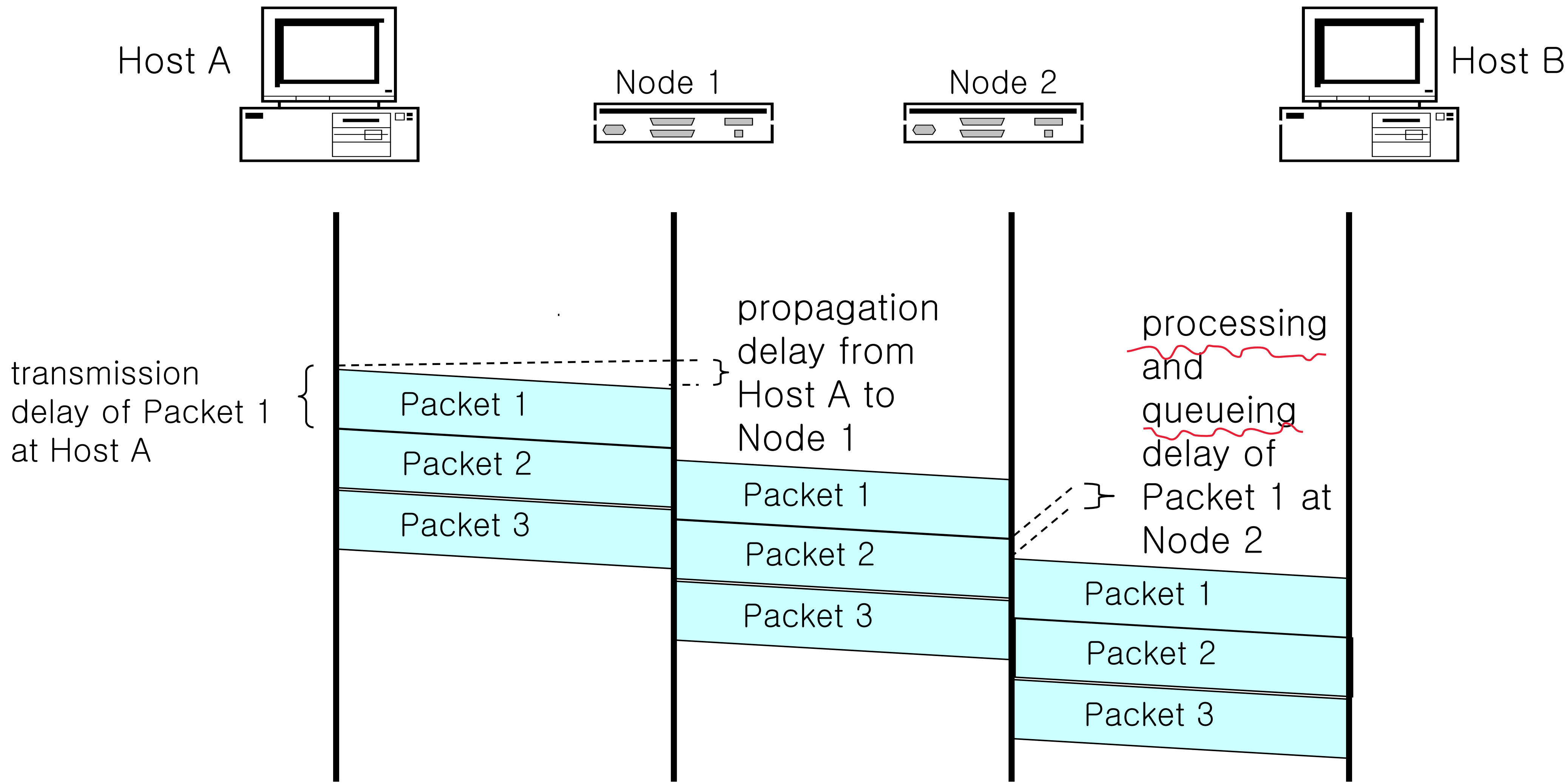
Caravan analogy (more)



- suppose cars now run (=propagate) at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- d_{trans} = 10mins, d_{prop} = 6mins

- *Q:* Will 1st car arrives to 2nd booth before all cars serviced at first booth?
  - *A: Yes!* after 7 min, 1st car arrives at second booth; three cars still at 1st booth.
- 1st bit of packet can arrive at 2nd router before packet is fully transmitted at 1st router!

Timing Diagram of Packet Switching



Nodal delay

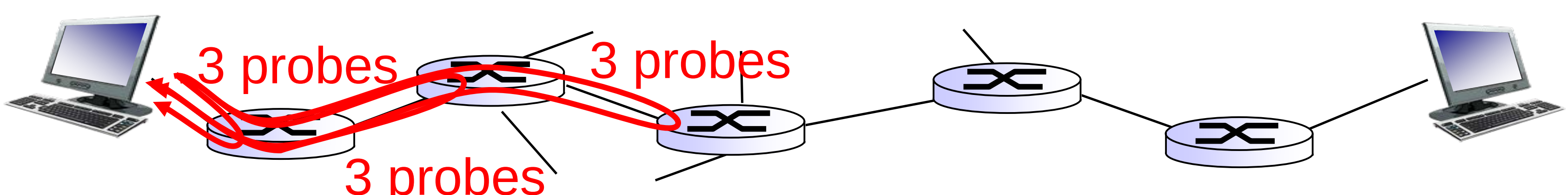
$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

- d_{proc} = processing delay
  - typically a few microsecs or less
- d_{queue} = <u>queuing delay</u>
  - depends on congestion of network →

- d_{trans} = transmission delay = L/R
  - negligible for 10Mbps and higher
  - but significant for low-speed links (e.g., dial-up modem, 2G)
- d_{prop} = propagation delay
  - a few microsecs (e.g., same campus) to hundreds of msecs (e.g., satellite link)

"Real" Internet delays and routes

- what do "real" Internet delay & loss look like?
- `traceroute` program: provides delay measurement from source to router along end-end Internet path towards destination. For all *i*:
  - sends three packets that will reach router *i* on path towards destination
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.



"Real" Internet delays, routes

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 * * *
19 fantasia.eurecom.fr (193.55.113.142) 132 ms 128 ms 136 ms

trans-oceanic link

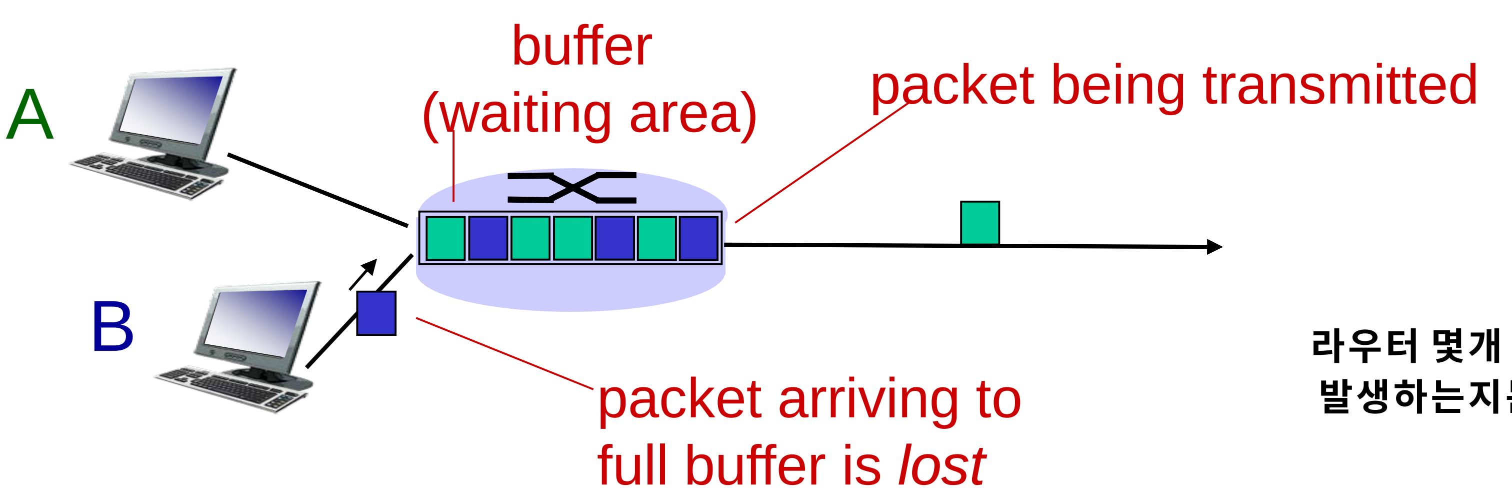
갑자기 느려진 걸 보니 바다를 건넜구나. 만약 바다를 건넌게 아니면 ~이 발생했다

* means no response (probe lost, router not replying)

* Do some traceroutes from exotic countries at www.traceroute.org

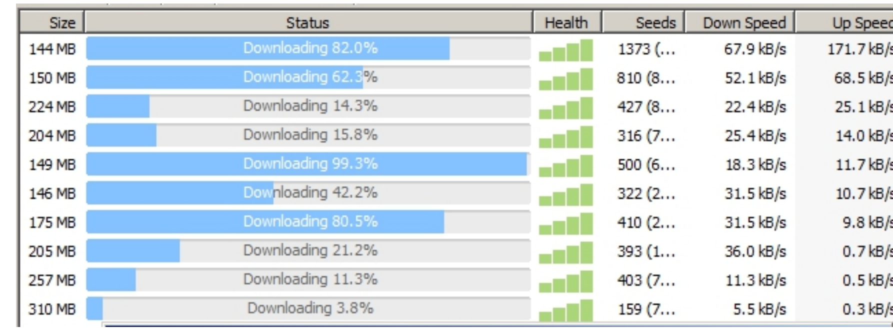
Packet loss

- queue (i.e., buffer) preceding link has finite capacity
  - e.g., 4M-bytes
- packet arriving to full queue dropped (i.e., lost)
- lost packet may be retransmitted
  - by previous node, by source end system, or not at all



라우터 몇개 거치는지 어느 라우터에서 packet loss가 발생하는지는 모르고 받지 못한 사실만 알 수 있음.

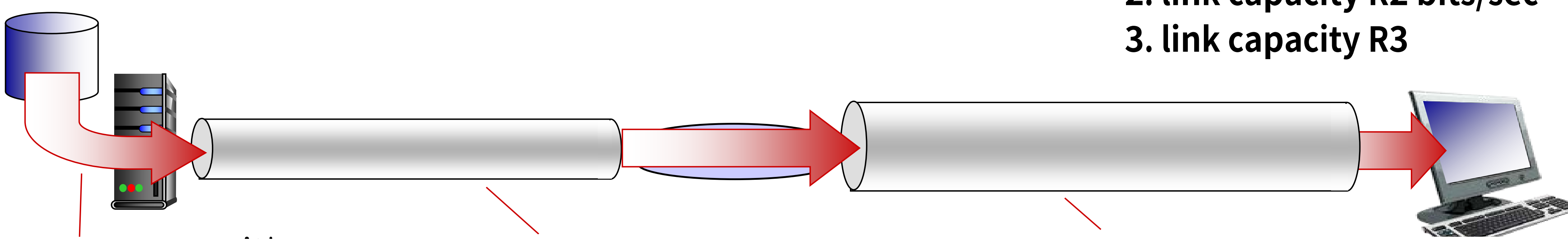
Throughput 다운로드 속도



Size	Status	Health	Seeds	Down Speed	Up Speed
144 MB	Downloading 82.0%		1373 (...	67.9 kB/s	171.7 kB/s
150 MB	Downloading 62.3%		810 (8...	52.1 kB/s	68.5 kB/s
224 MB	Downloading 14.3%		427 (8...	22.4 kB/s	25.1 kB/s
204 MB	Downloading 15.8%		316 (7...	25.4 kB/s	14.0 kB/s
149 MB	Downloading 99.3%		500 (6...	18.3 kB/s	11.7 kB/s
146 MB	Downloading 42.2%		322 (2...	31.5 kB/s	10.7 kB/s
175 MB	Downloading 80.5%		410 (2...	31.5 kB/s	9.8 kB/s
205 MB	Downloading 21.2%		393 (1...	36.0 kB/s	0.7 kB/s
257 MB	Downloading 11.3%		403 (7...	11.3 kB/s	0.5 kB/s
310 MB	Downloading 3.8%		159 (7...	5.5 kB/s	0.3 kB/s

- *throughput:* rate (bits/time) at which bits transferred between sender/receiver 즉각적인
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

1. server with file of F bits to send to client
2. link capacity R2 bits/sec
3. link capacity R3



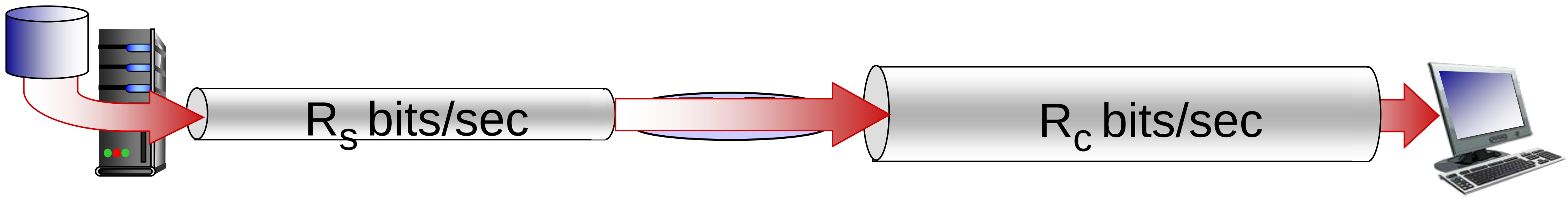
server sends bits (fluid) into pipe

pipe that can carry fluid at rate R_s bits/sec)

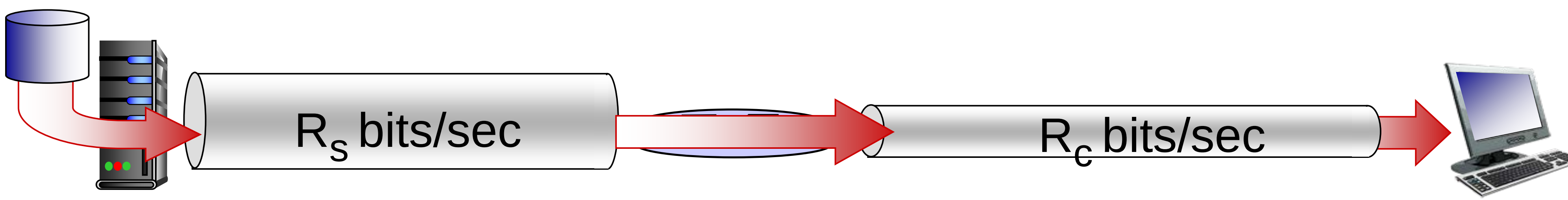
pipe that can carry fluid at rate R_c bits/sec)

Throughput (more)

- $R_s < R_c$ What is average end-end throughput?



- $R_s > R_c$ What is average end-end throughput?



bottleneck link

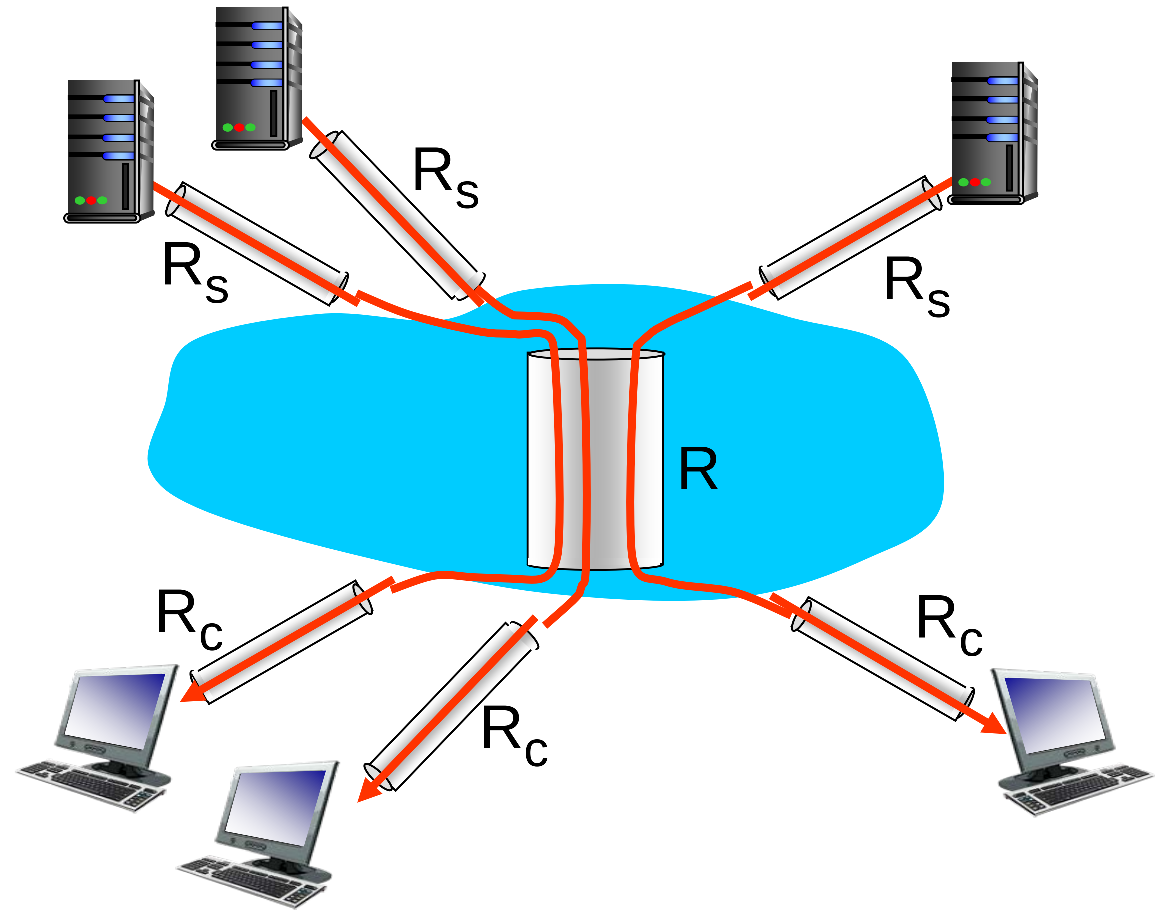
제한하다

link on end-end path that constrains end-end throughput

bottleneck link : 전체 링크 중 가장 느린 링크(가장 많이 영향을 줌)

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
  - R is often much larger
  - Even R/10, R/100, … is larger
- Throughput depends on the transmission rates of links on the data path



10 connections

Chapter 1: roadmap

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

Protocol "layers"

*Networks are **complex**,*
with many "pieces":

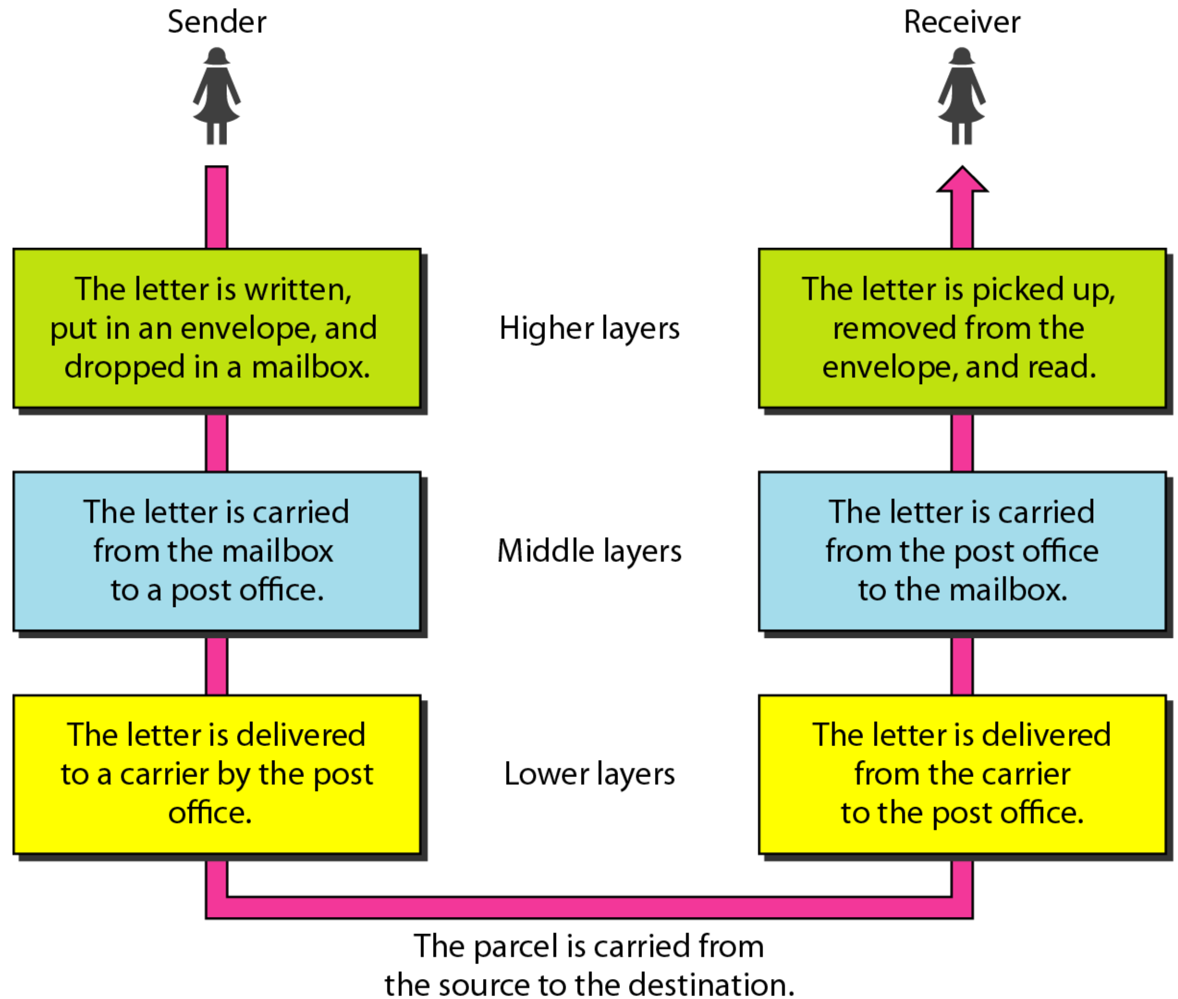
- end system/hosts
- routers
- links of various media
- applications/protocols
- hardware, software

Question:

is there any hope of *organizing* structure of network?

…. or at least our discussion of networks?

Layering Example: Postal mail



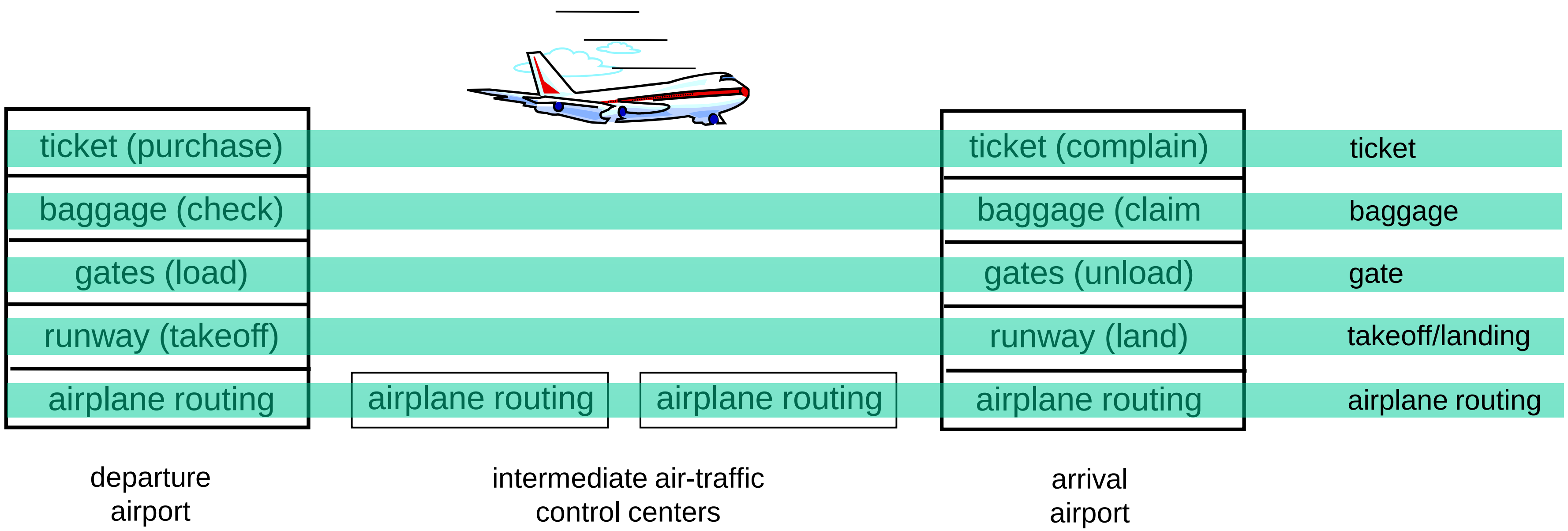
Organization of air travel

ticket (purchase)	ticket (complain)
baggage (check)	baggage (claim)
gates (load)	gates (unload)
runway takeoff	runway landing
airplane routing	airplane routing

airplane routing

- a series of steps

Layering of airline functionality



layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

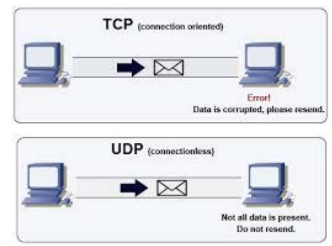
Why layering?

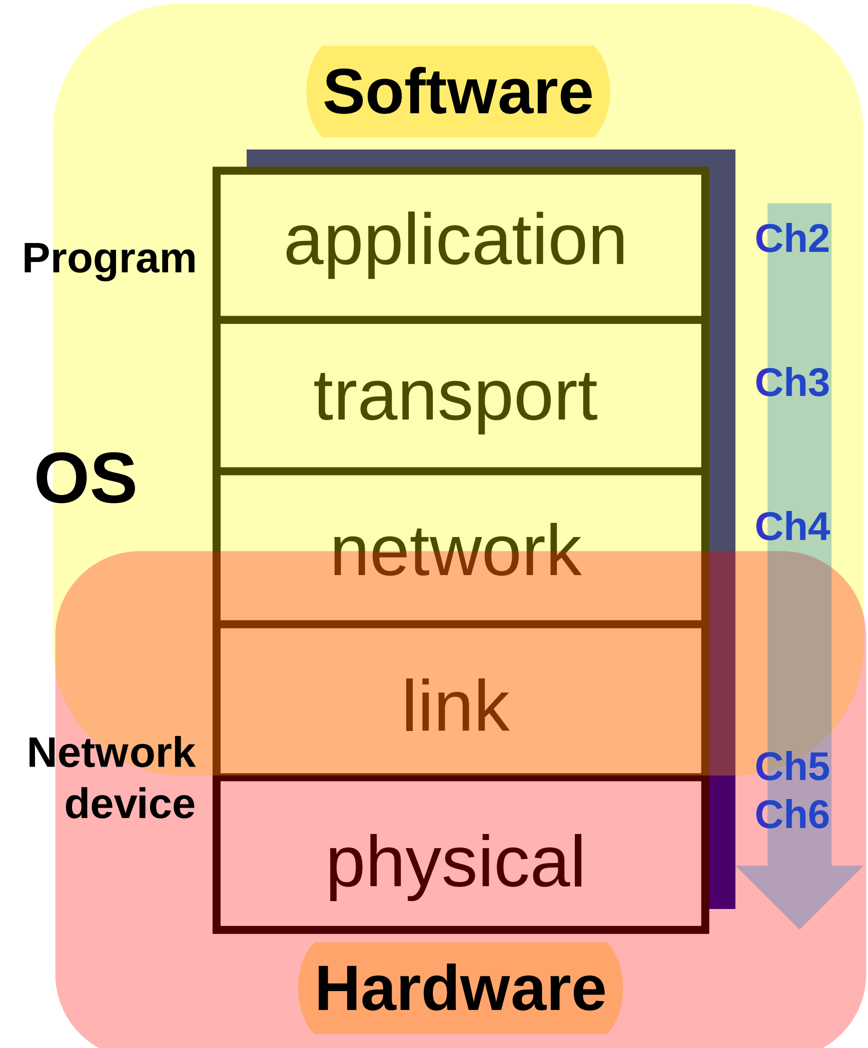
dealing with **complex systems:**

- simplification!
- modularization eases maintenance, updating of system
  - change of implementation of layer's service **transparent** to rest of system as long as
    - the layer provides same service to upper layer
    - the layer uses same service from lower layer
  - Example: change in gate procedure doesn't affect rest of system
    - e.g., embark at gate in FCFS order → in priority order (disabled first, then first class, …)
    - This does not change Baggage or Runway layers

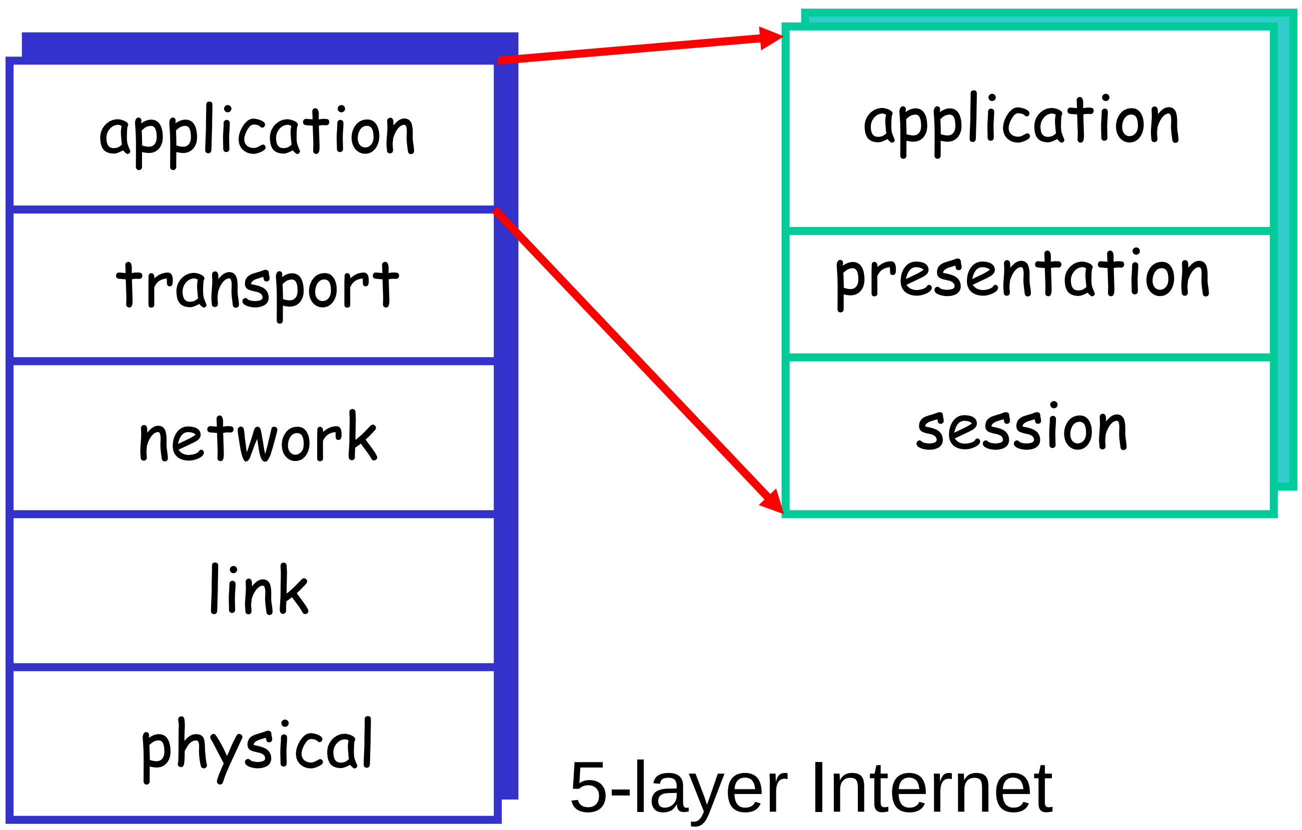
Internet protocol stack

- *application:* supporting various network applications
  - HTTP, SMTP, FTP, DNS
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi)
- *physical:* bits "on the wire"





ISO 7-layer reference model

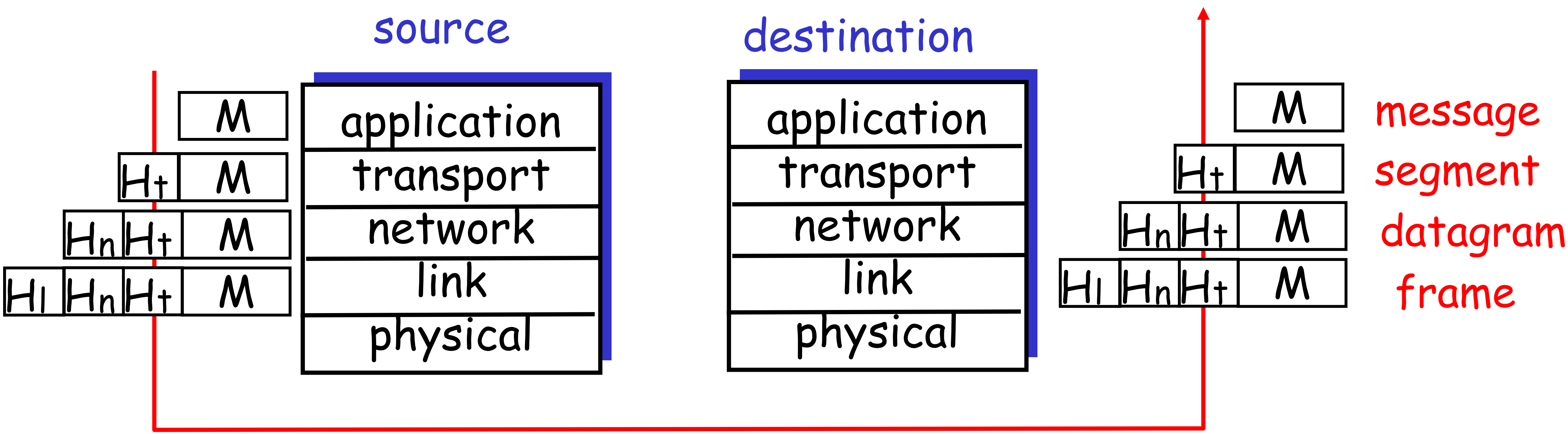


5-layer Internet
Protocol Stack

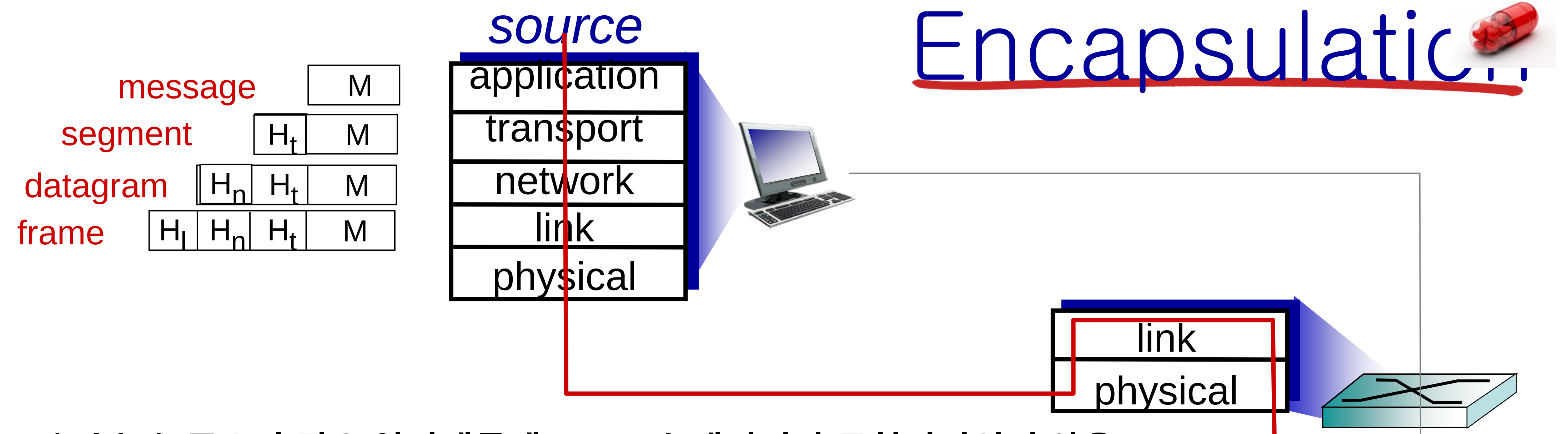
Protocol layering and data

Each layer takes data from above

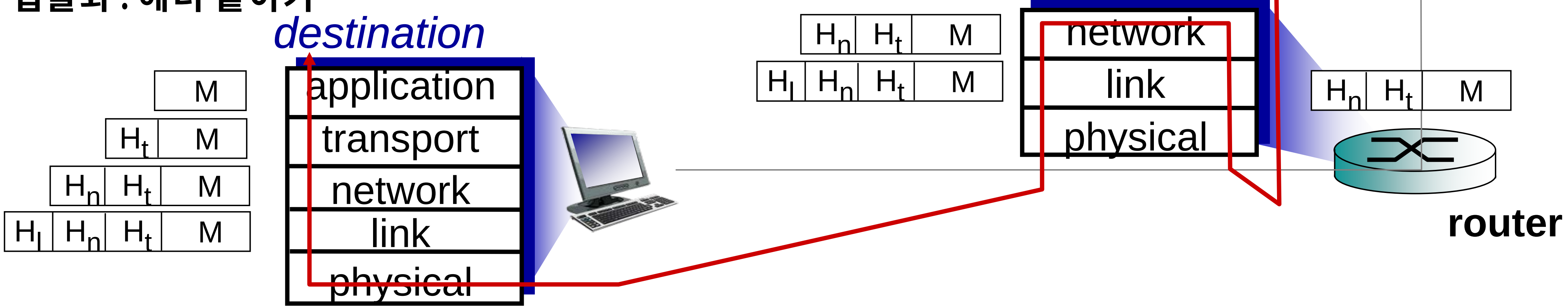
- adds header information to create new data unit
- passes new data unit to layer below



Encapsulation



switch는 ip주소가 필요 없기때문에 network 레이어가 구현되어있지 않음
router는 ip주소가 필요하기 때문에(어느 ip 주소/router로 보내야 더 빨리 도달할지)
network 헤더를 봐야돼서 network 레이어가 구현됨
캡슐화 : 헤더 붙이기



Internet structure: network of networks (Ch. 1.3)

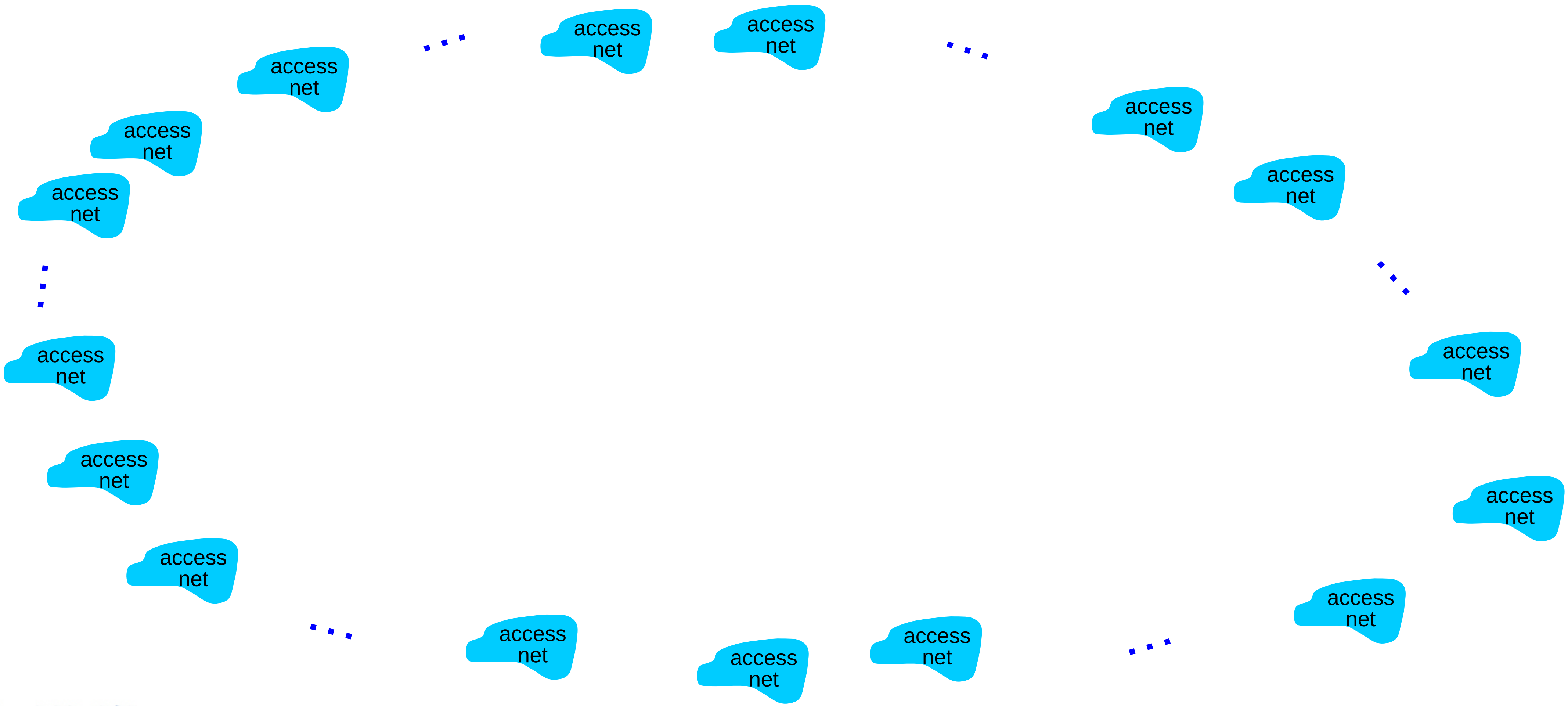
- End systems connect to Internet via access ISPs (Internet Service Providers)
  - DSL, cable, FTTH, Wi-Fi, cellular, …
  - Telco (e.g., AT&T, Sprint, KT, SKT), Residential, company and university ISPs
- Access ISPs in turn must be interconnected.
  - Network of Networks
  - So that any two hosts can send packets to each other
- Resulting network of networks is very complex
  - Evolution was driven by economics and national policies
- Let's take a stepwise approach to describe current Internet structure

access ISOs = access internet service providers :
skt, kt, lg u+ ...

ISP is an "administrative view" of the network

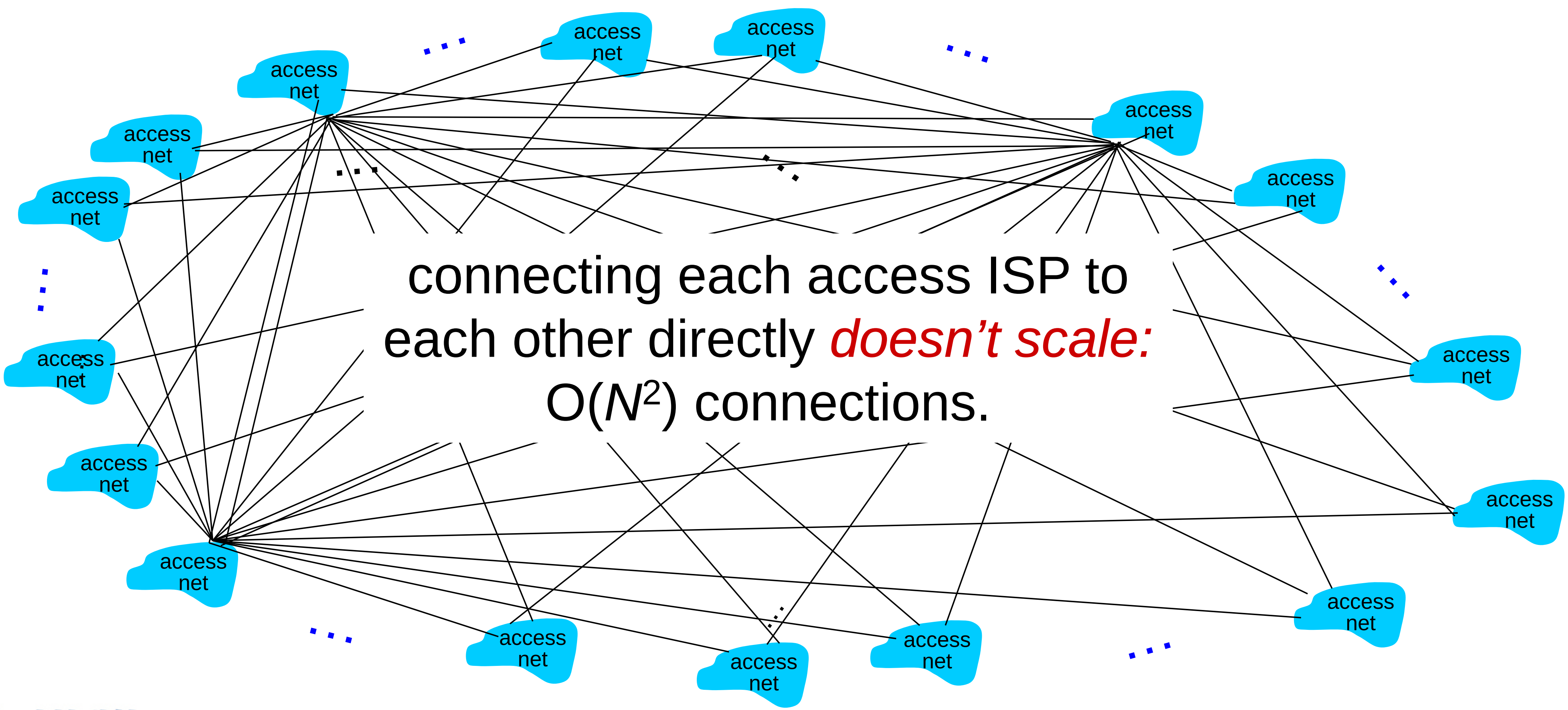
Internet structure: network of networks

Question: given *millions* of access ISPs, how to connect them together?



Internet structure: network of networks

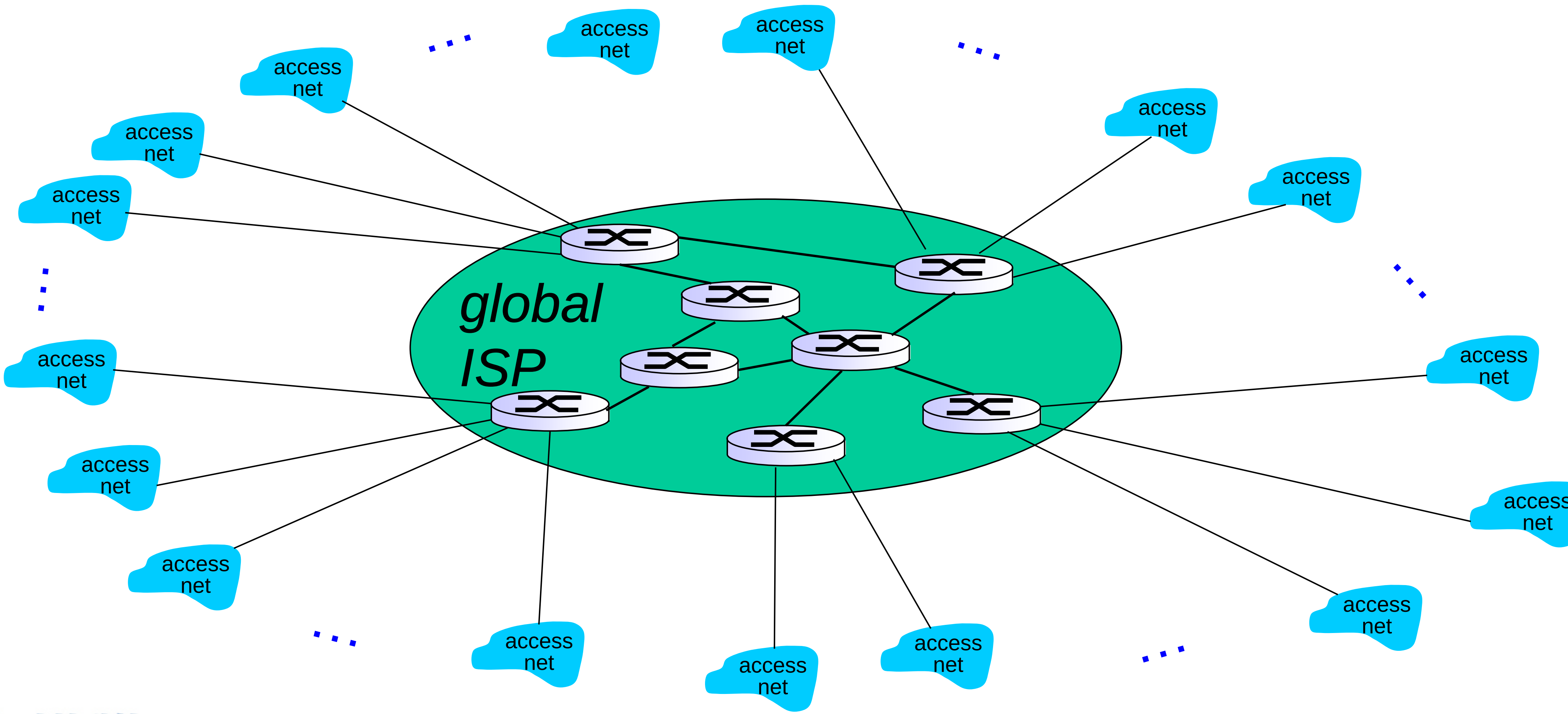
Option: *connect each access ISP to every other access ISP?*



connecting each access ISP to each other directly *doesn't scale:* O(N^2) connections.

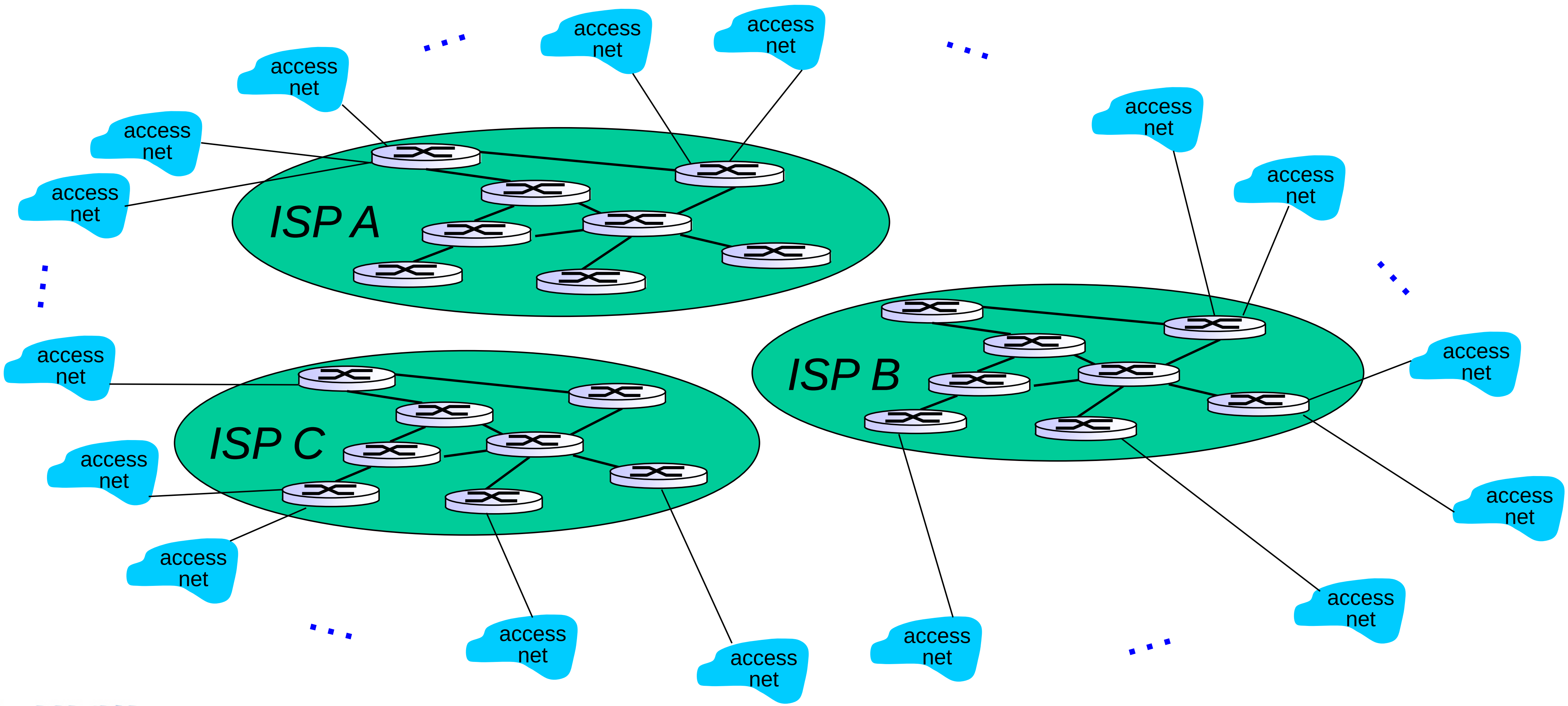
Internet structure: network of networks

Option: connect each access ISP to a global transit ISP?
***Customer** and **provider** ISPs have economic agreement.*



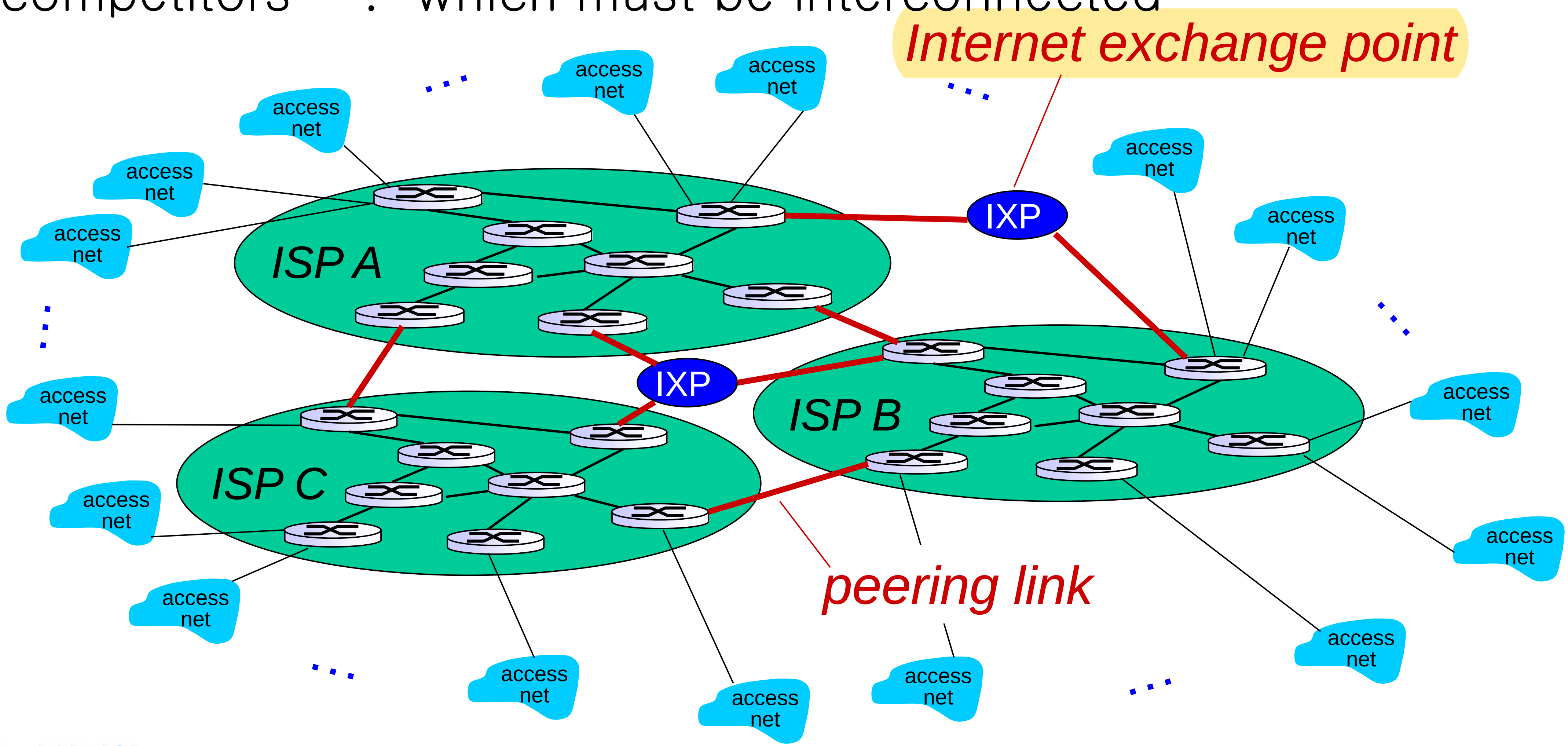
Internet structure: network of networks

But if one global ISP is viable business, there will be competitors ….



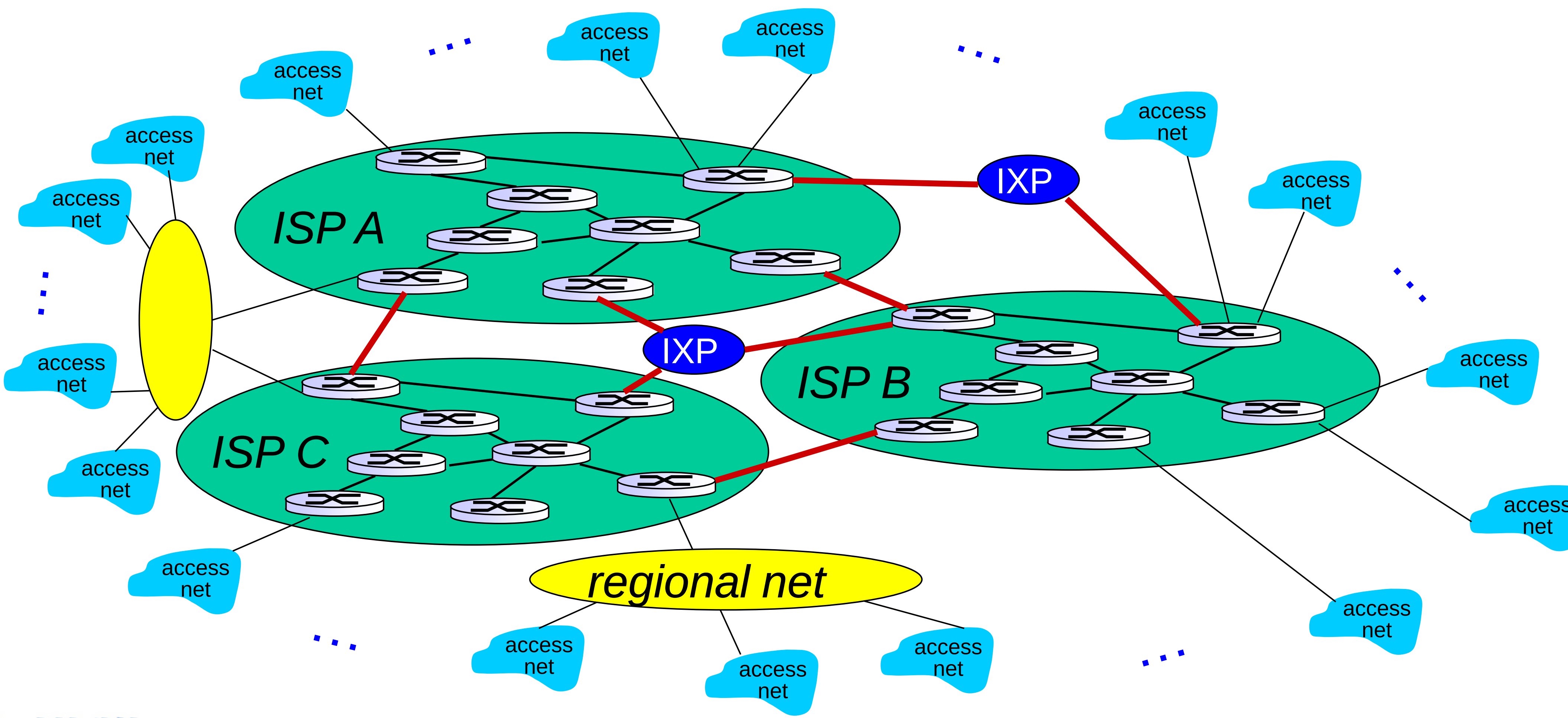
Internet structure: network of networks

But if one global ISP is viable business, there will be competitors …. which must be interconnected



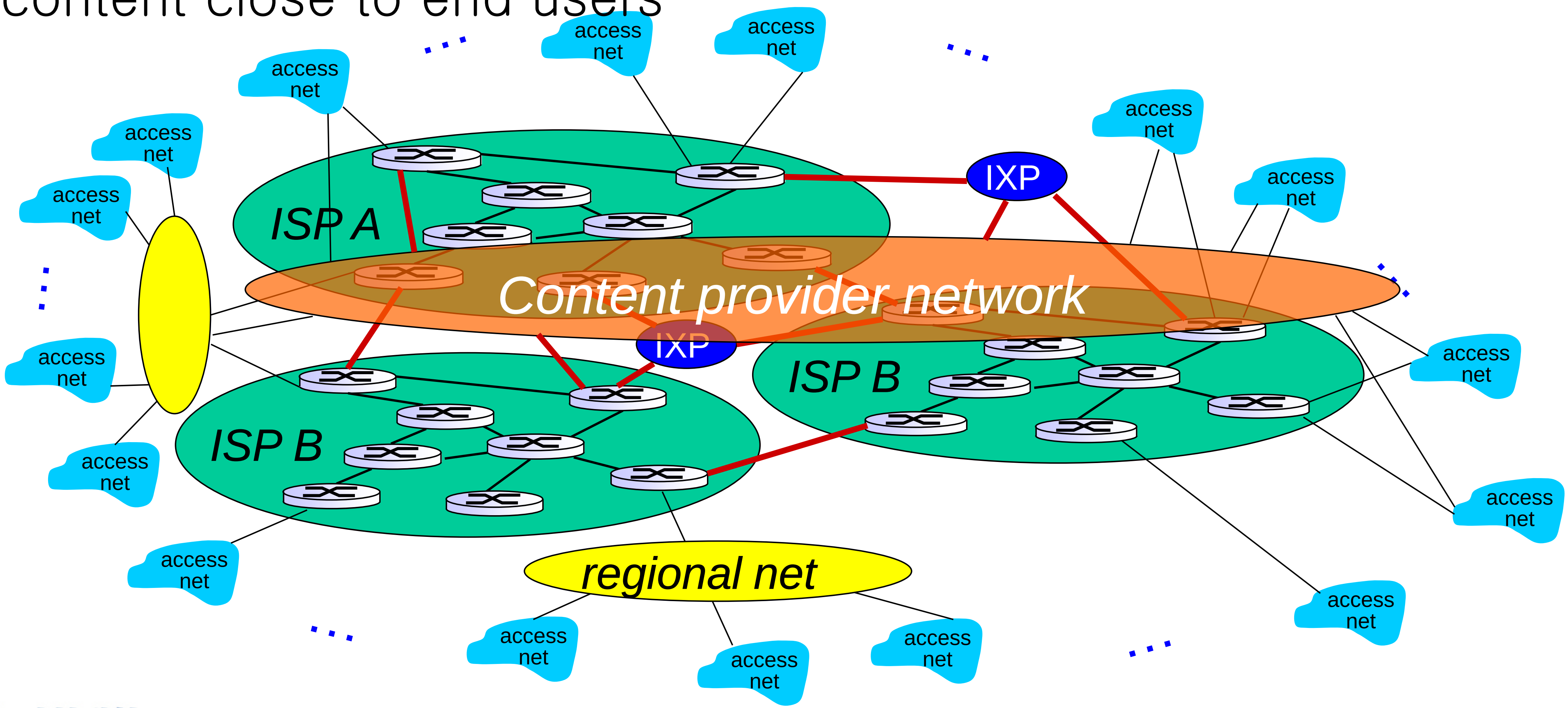
Internet structure: network of networks

… and regional networks may arise to connect access nets to ISPS



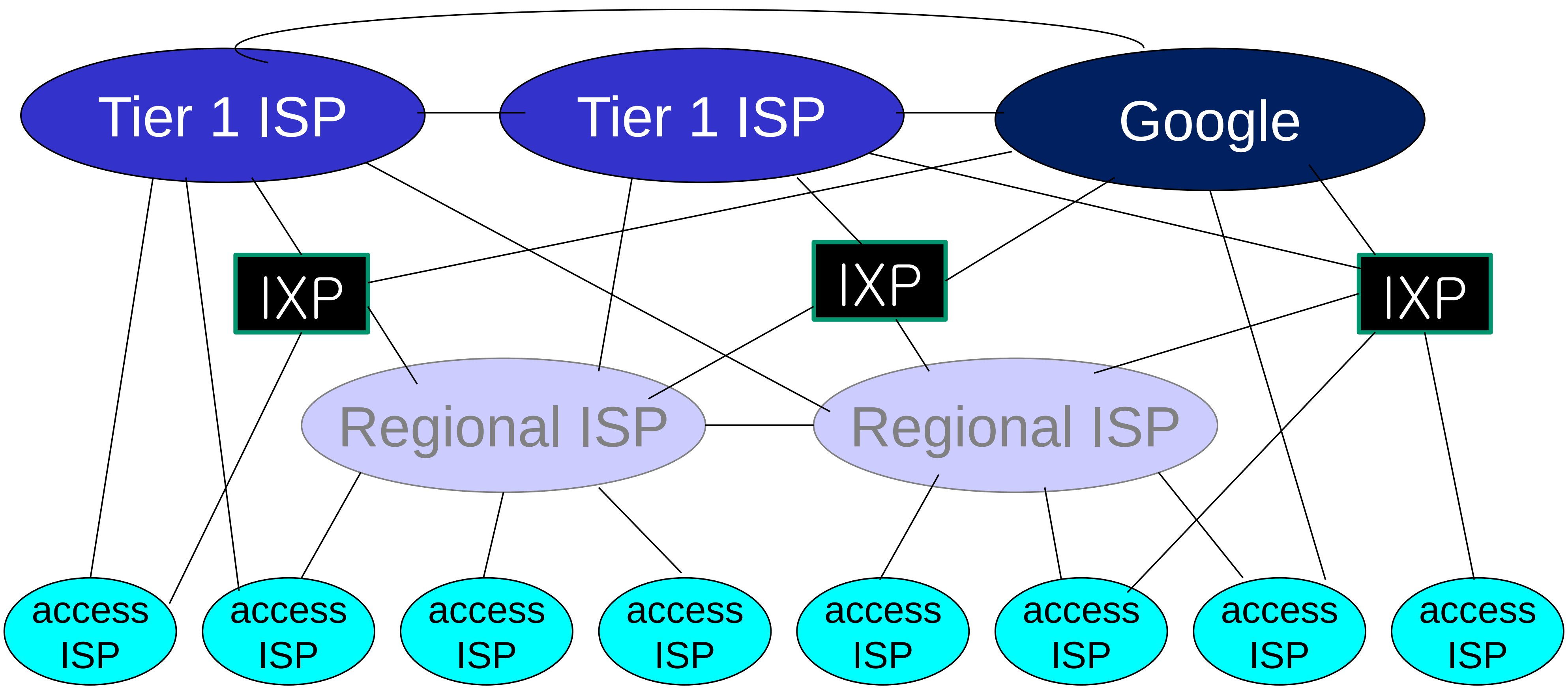
Internet structure: network of networks

… and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users



Gachon University 가천대학교

Internet structure: network of networks

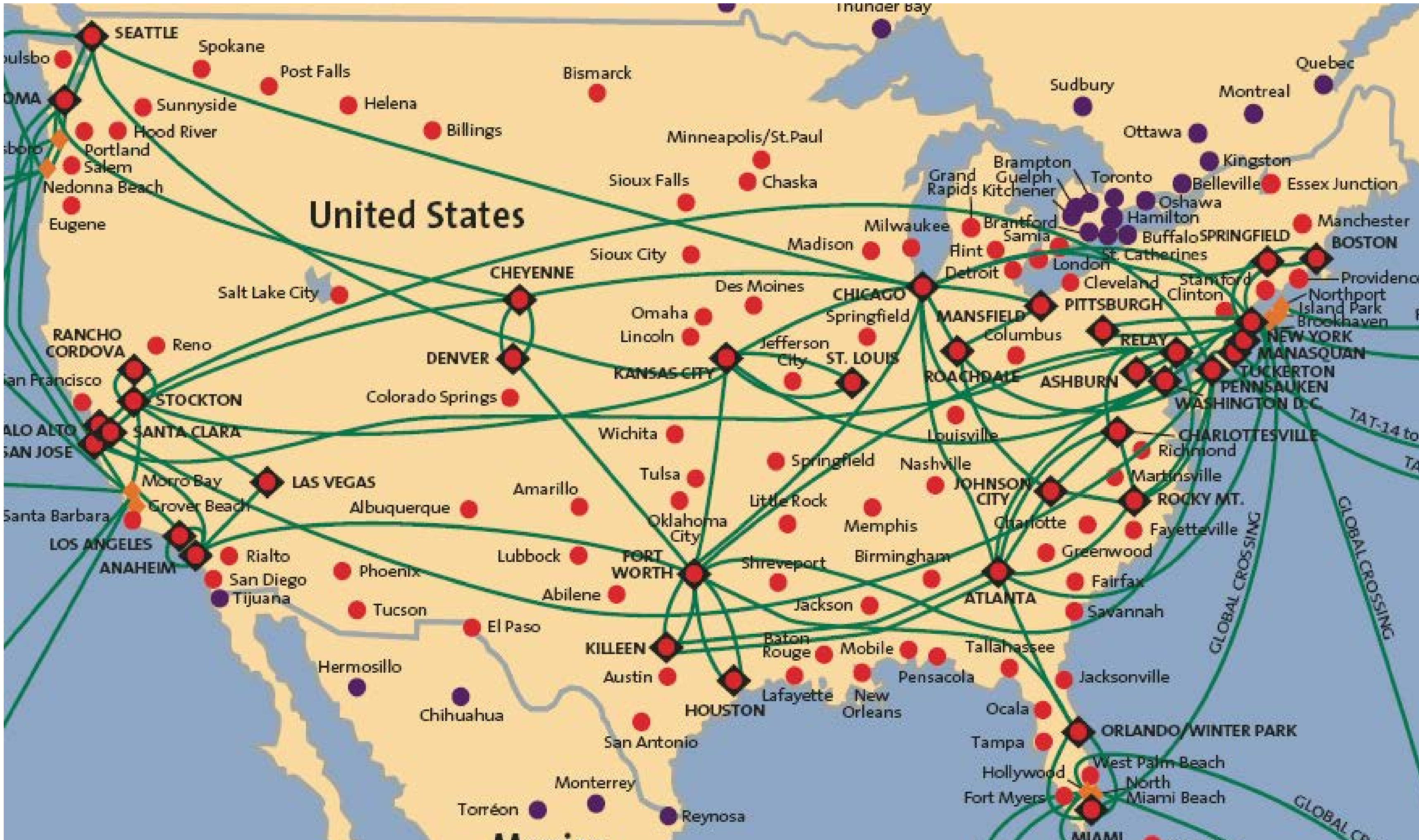


- at center: small # of well-connected large networks
  - "tier-1" commercial ISPs (e.g., Level 3 communications, Sprint, AT&T, NTT), national & international coverage
  - content provider network (e.g, Google): private network that connects it data centers to Internet, often bypassing tier-1, regional ISPs

가천대학교 Gachon University

youtube나 netflix는 최종적으로 연결은 되어하지만 자체(content provider network)

Tier-1 ISP: e.g., Sprint



Chapter 1: roadmap

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

Network security

- field of network security:
  - how bad guys can attack computer networks
  - how we can defend networks against attacks
  - how to design architectures that are immune to attacks
- Internet not originally designed with (much) security in mind
  - *original vision:* "a group of mutually trusting users attached to a transparent network" ☺
  - Internet protocol designers playing "catch-up"
  - security considerations in all layers!

Bad guys: put malware into hosts via Internet

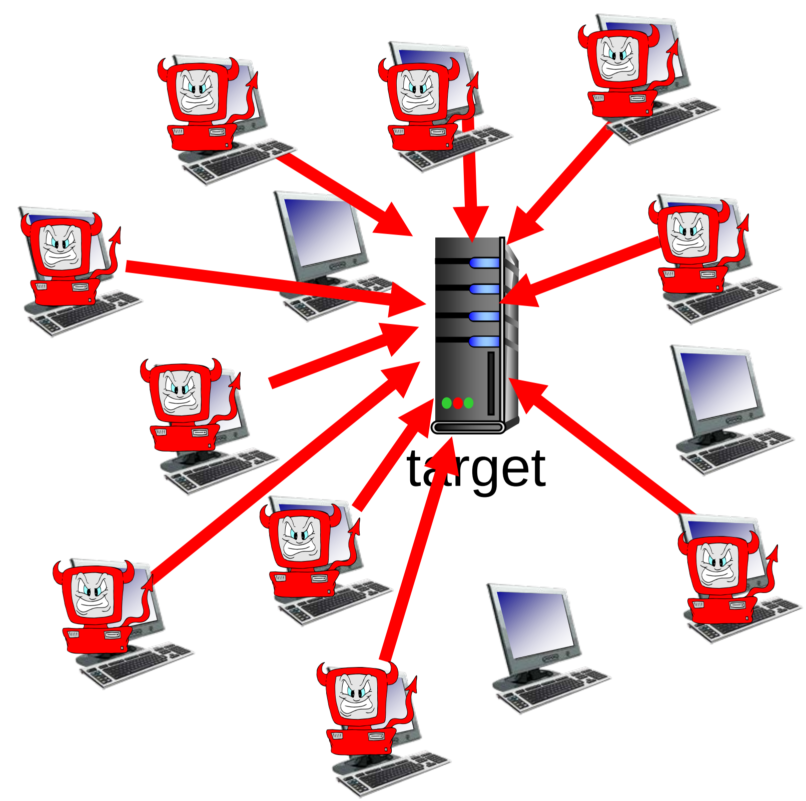
- malware can get in host from:
  - *virus:* self-replicating infection by receiving/executing object (e.g., e-mail attachment)
  - *worm:* self-replicating infection by passively receiving object that gets itself executed
- spyware malware can record keystrokes, web sites visited, upload info to collection site
- infected host can be enrolled in botnet, used for spam, DDoS attacks

Bad guys: attack server, network infrastructure

Denial of Service (DoS): attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic

가짜의

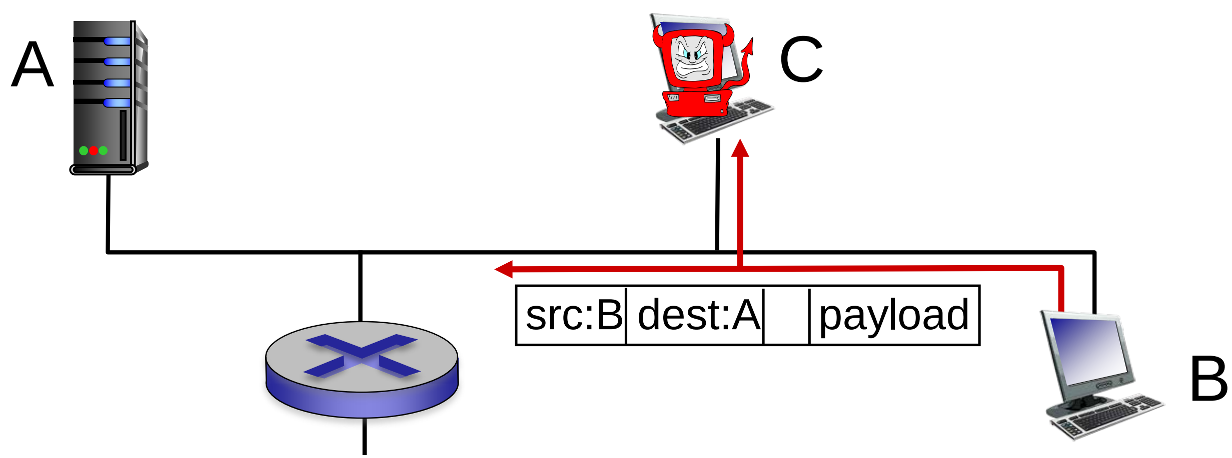
1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



Bad guys can sniff packets

packet "sniffing":

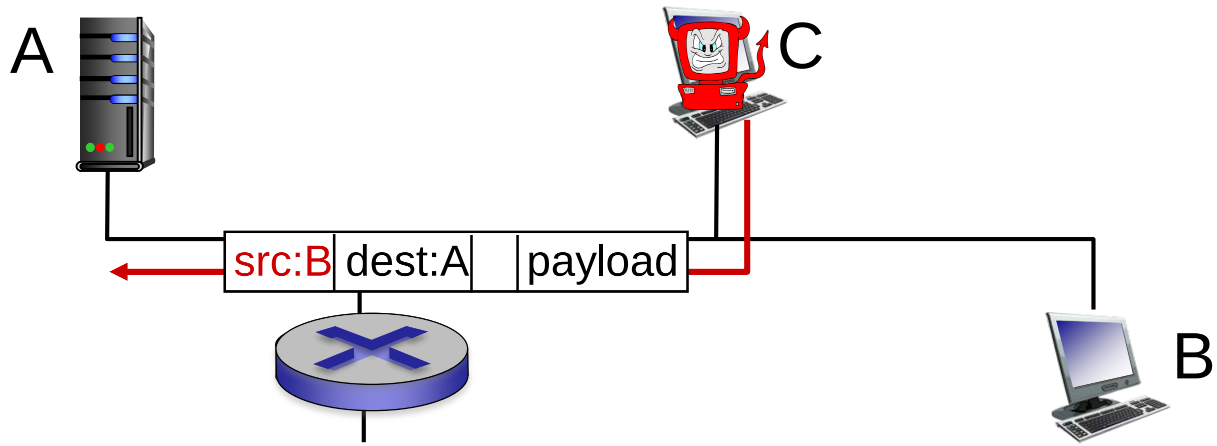
- broadcast media (shared Ethernet, wireless)
- 난잡한 promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



- wireshark software used for end-of-chapter labs is a (free) packet-sniffer

Bad guys can use fake addresses

IP spoofing: send packet with false source address



IP spoofing : C가 B인척

CH1: summary

covered a "ton" of material!

- Internet overview
- what's a protocol?
- network edge, core, access network
  - packet-switching versus circuit-switching
  - Internet structure
- performance: loss, delay, throughput
- layering, service models

you now have:

- context, overview, "feel" of networking
- more depth, detail *to follow!*