Computer Networks

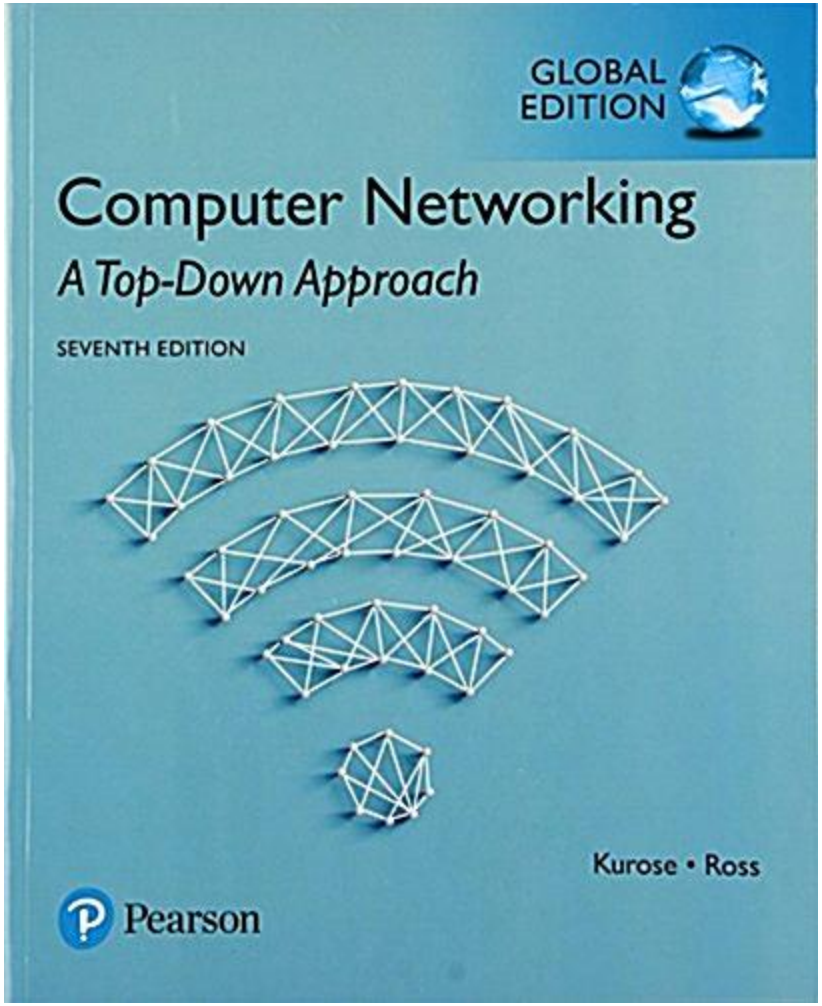
2024년 1학기

Myung Kyun Kim

(mkkim@ulsan.ac.kr)

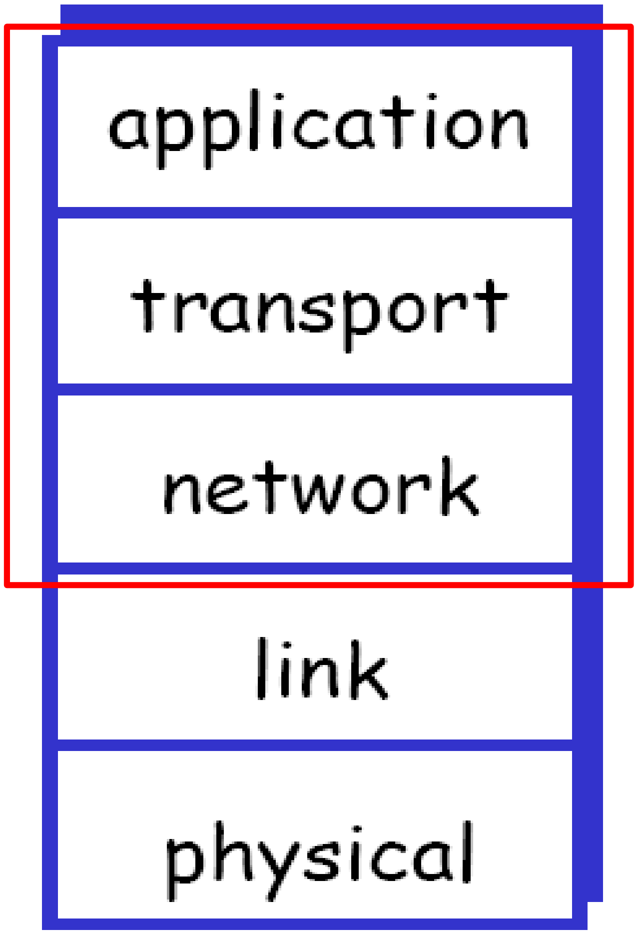
Contents

- Introduction to Data Communication
- Application Layer Technologies
- Transport Layer Technologies
- Multimedia Networking Technologies



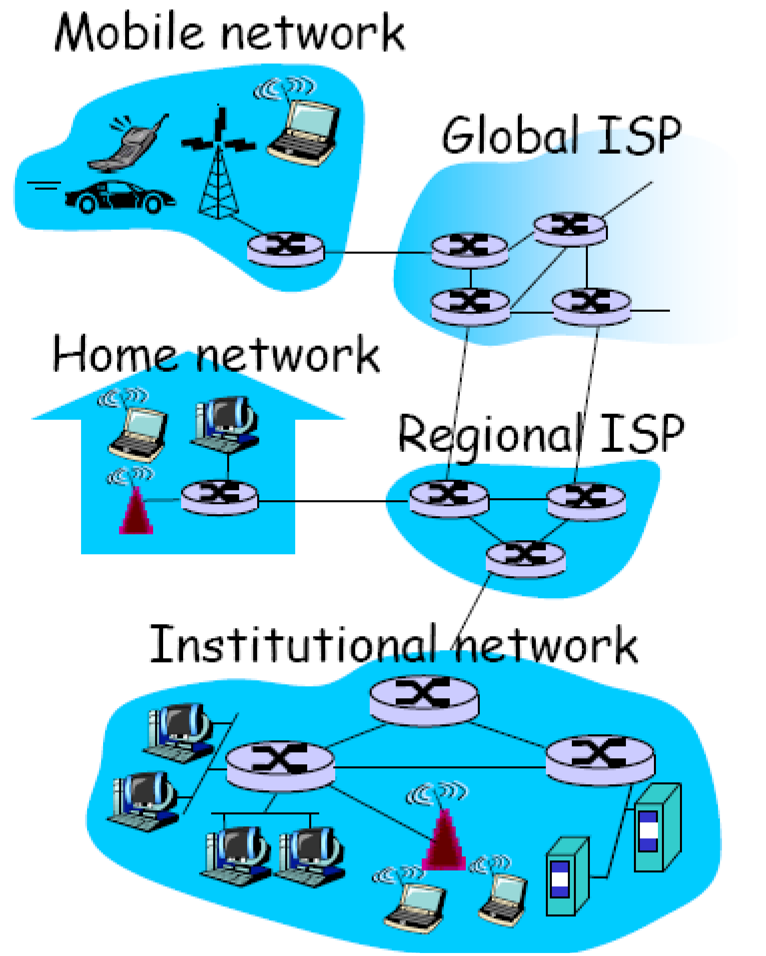
Contents

- Internet Protocol Stack



What's the Internet?

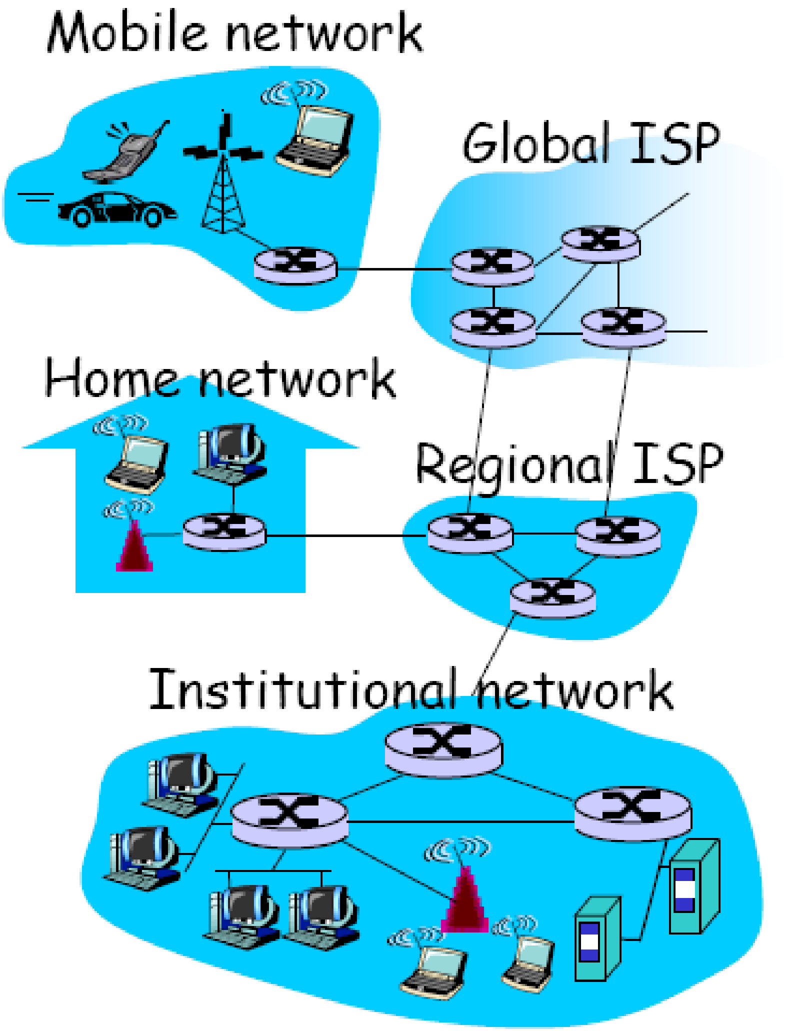
- Millions of connected computing devices:
  - *end systems*: hosts, things (Internet of Things)
  - running *network appls*
- Communication links
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth*
- *routers:* forward packets (chunks of data)



ISP (Internet Service Provider)

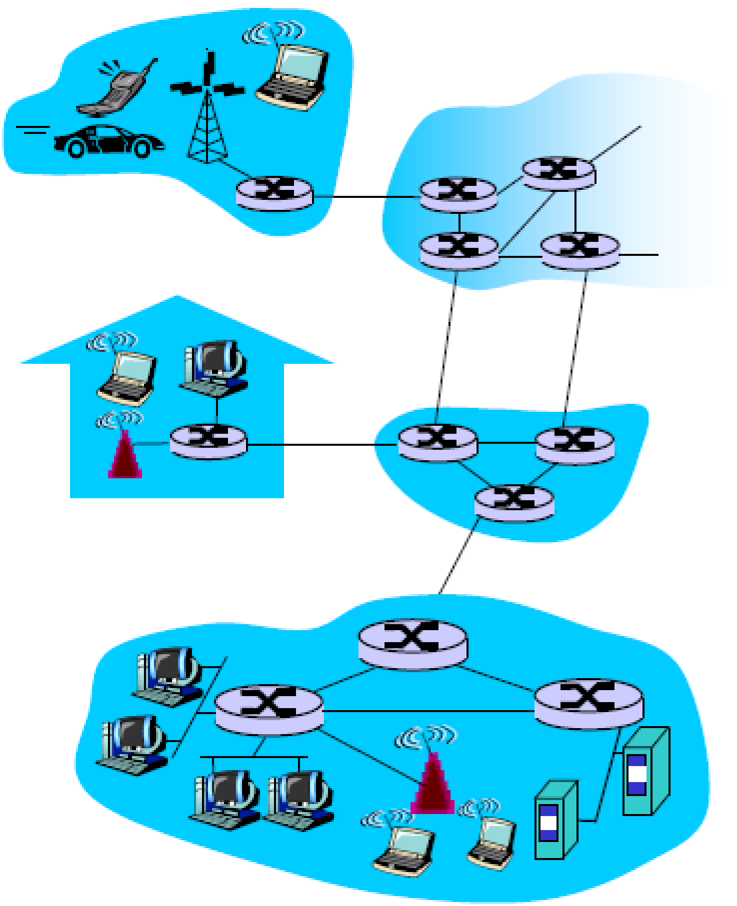
What’s the Internet?

- *Protocols*
  - control sending, receiving of msgs
  - TCP, IP, HTTP, Skype, Ethernet
- *Internet:* network of networks
  - loosely hierarchical
  - public Internet versus private intranet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



What's the Internet?

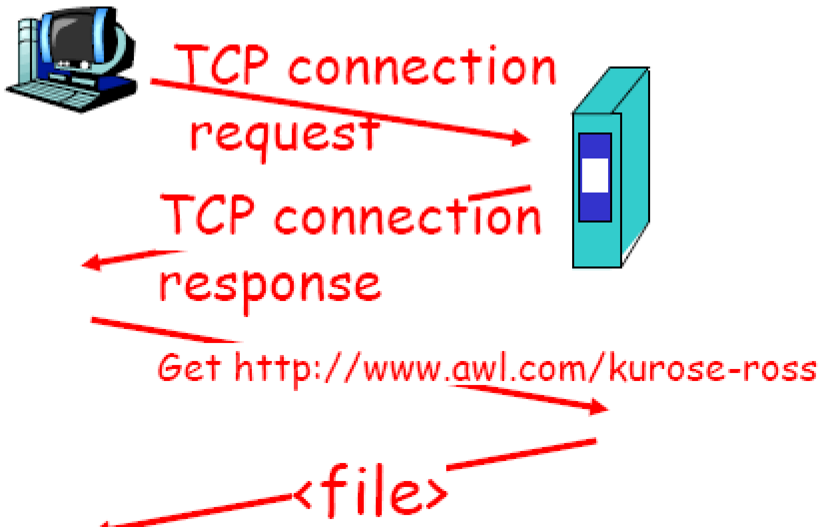
- Internet applications:
  - Web, SNS, VoIP, email, games, e-commerce, file sharing
- Internet provides comm. services to appls:
  - data delivery from source to destination
  - "best effort" (unreliable) data delivery



What’s a protocol?

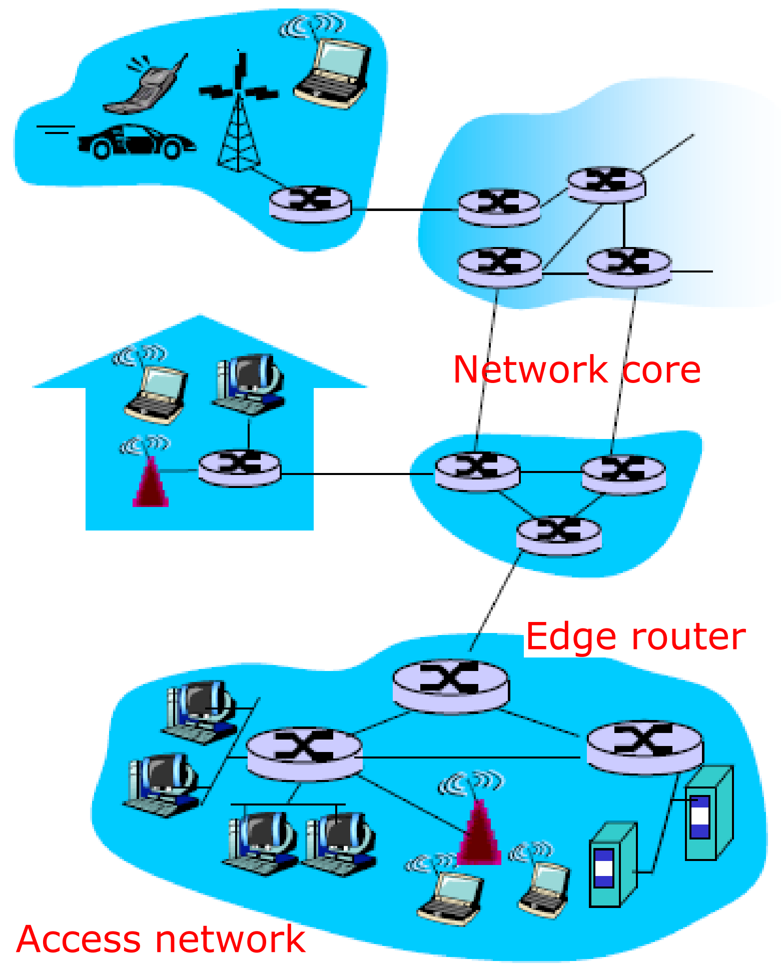
Protocols:

- Protocols define
  - message format,
  - order of messages sent and received among network entities, and
  - actions taken on msg transmission, receipt



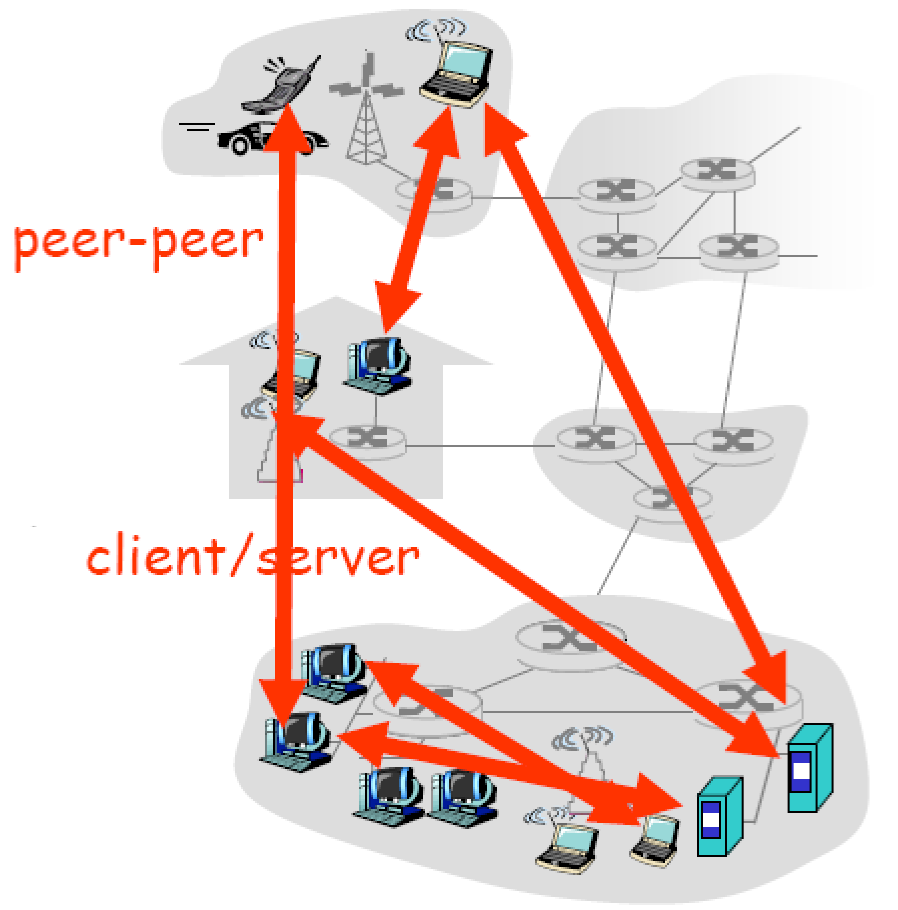
Network Edge

- Network edge:
  - applications and hosts
- Access networks:
  - physical media: wired, wireless communication links
- Network core:
  - interconnected routers
  - network of networks



Network Edge

- End systems (hosts):
  - run application programs at "edge of network"
  - e.g. Web, email, SNS
- Client-server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- Peer-peer model:
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent



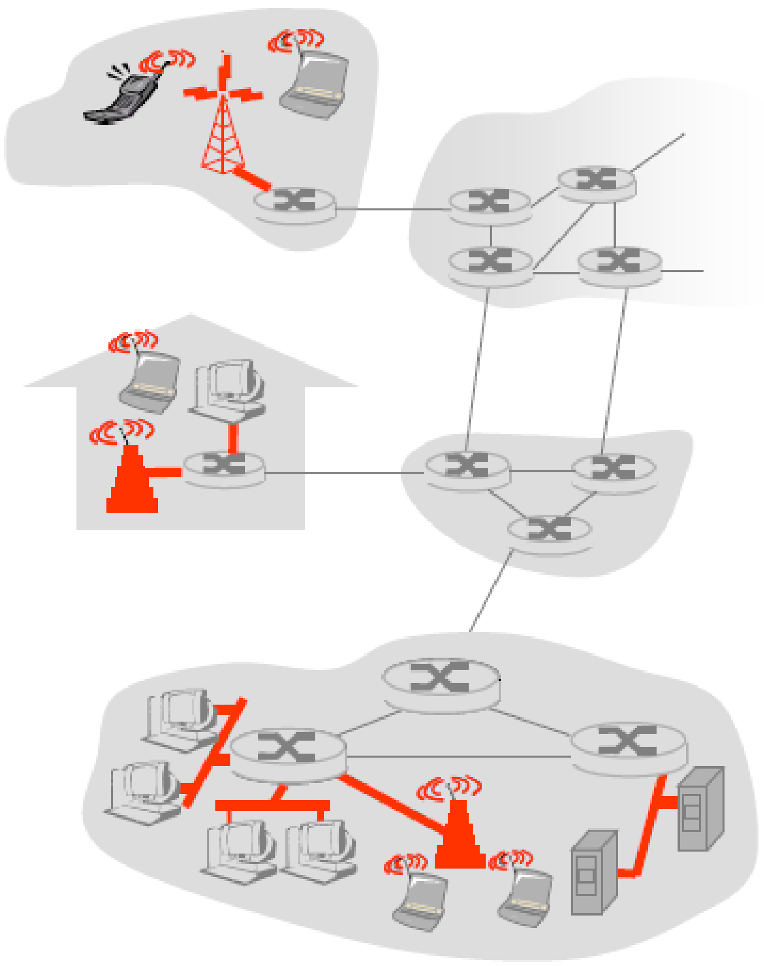
Network Edge

Q: How to connect end systems to edge router?

- residential access networks
- institutional access networks (school, company)
- mobile access networks

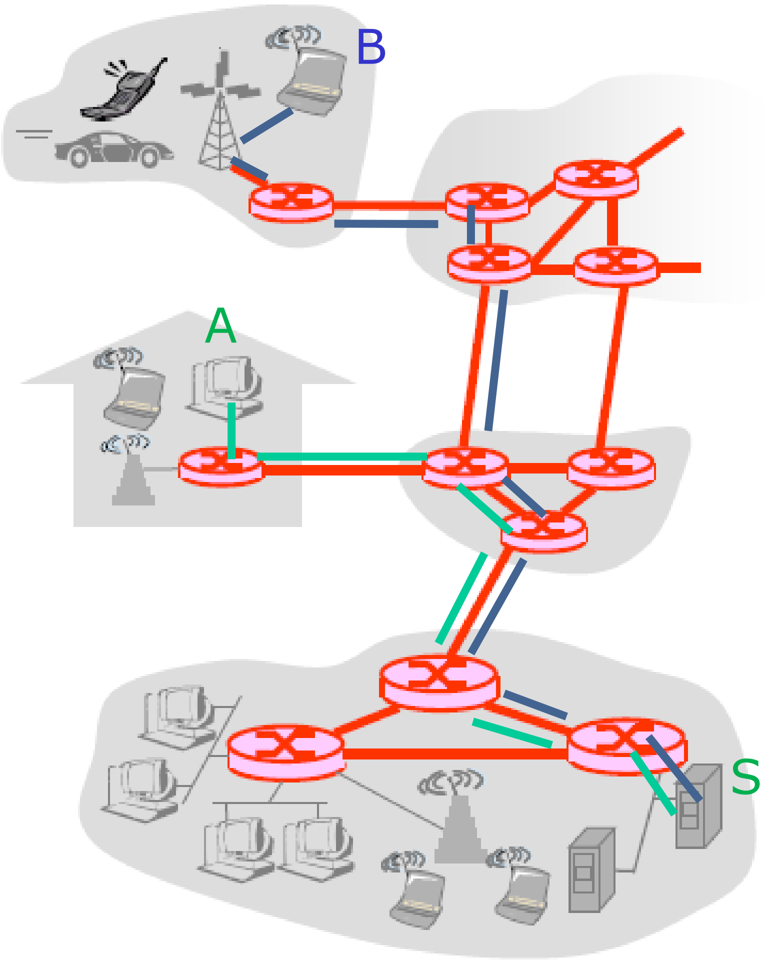
Keep in mind:

- bandwidth (bits per second) of access network?
- shared or dedicated?



Network Core

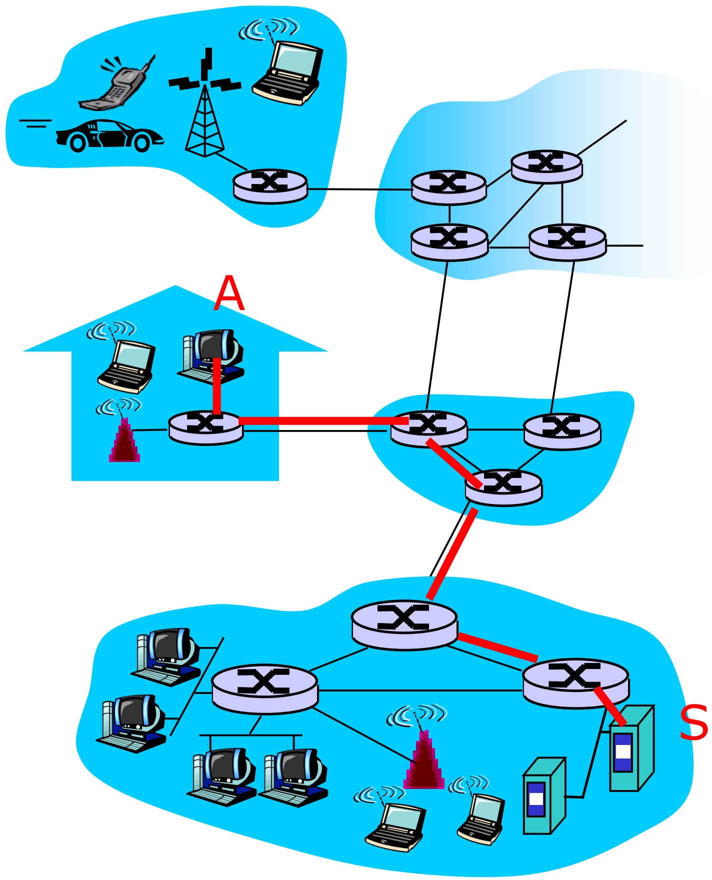
- Mesh of interconnected routers
- Switching methods: how is data transferred through net?
  - Circuit-switching: dedicated circuit per call: telephone network
  - Packet-switching: data sent thru network in discrete “chunks” (packets)



Network Core: Circuit Switching

End-to-end resources reserved for "call"

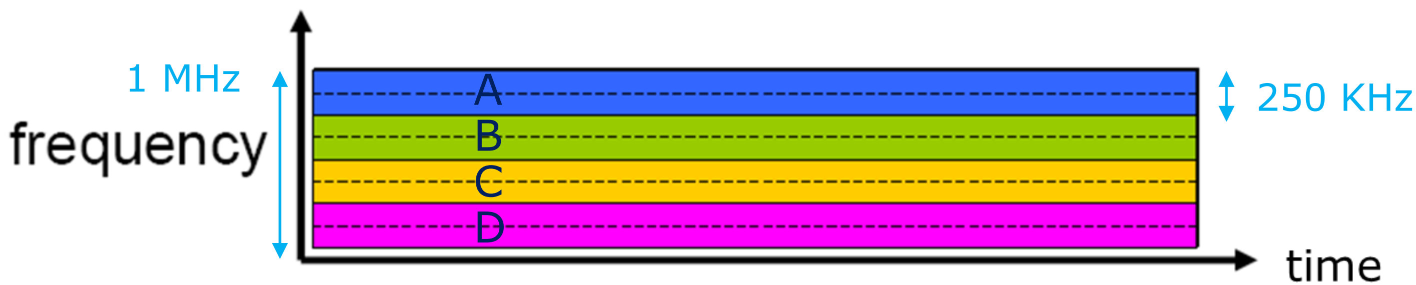
- Call setup required
  - Call setup – msg transmission – call release
- Dedicated resources:
  - Link bandwidth, switch capacity
  - no sharing
- Guaranteed performance: constant delay
- Resource waste if not used



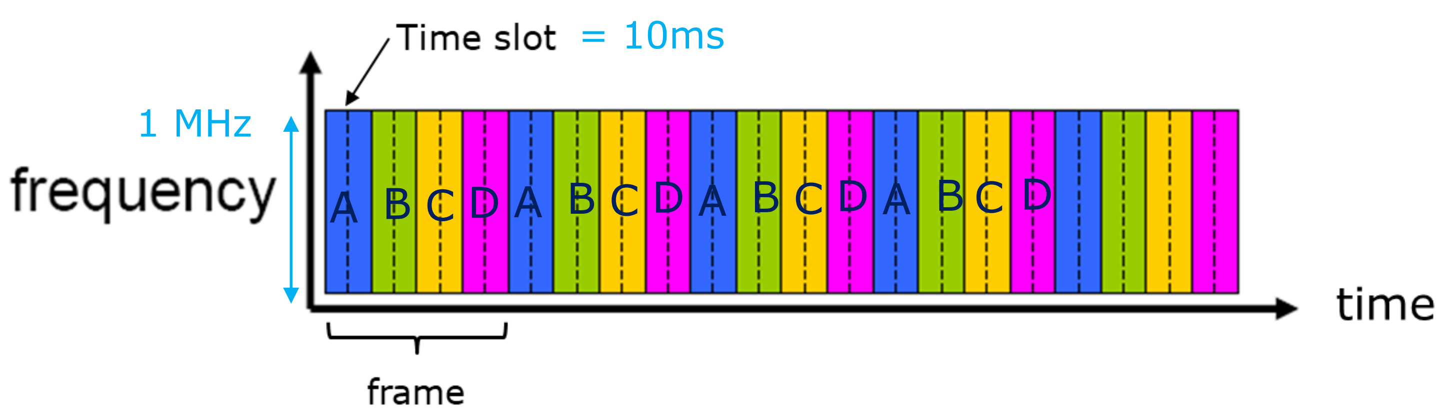
Network Core: Circuit Switching

Dividing link bandwidth into calls

- FDM (Frequency Division Multiplexing)



- TDM (Time Division Multiplexing)



Numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - All links are 1.536 Mbps
  - Each link uses TDM with 24 slots/frame, 1 frame/sec
  - 500 msec to establish end-to-end circuit

Numerical example

- Delay from A to B
  - Delay for connection set-up: 500 ms
  - Data transmission delay:
    - time-slot size: (1/24) sec
    - data to be transmitted in one time-slot (one frame):

      $= (1.536 * 10^6) * (1/24) = 64{,}000$ bits
    - Num. of frames to transmit total data : 640,000 / 64,000 = 10
    - Data transmission delay = 10 (frames) * 1 (sec/frame) = 10 sec
  - Total delay: 10.5 sec

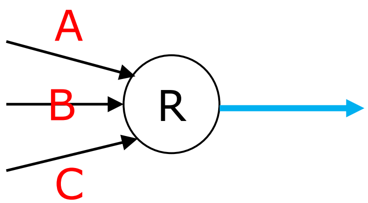
Numerical example

- How long does it take to send a file of 1 Mbits from host A to host B over a circuit-switched network?
  - All links are 1 Mbps
  - Each link uses TDM with 100 slots/frame, frame-length = 2 sec
  - 500 msec to establish end-to-end circuit

Network Core: Packet Switching

Each data stream divided into *packets*

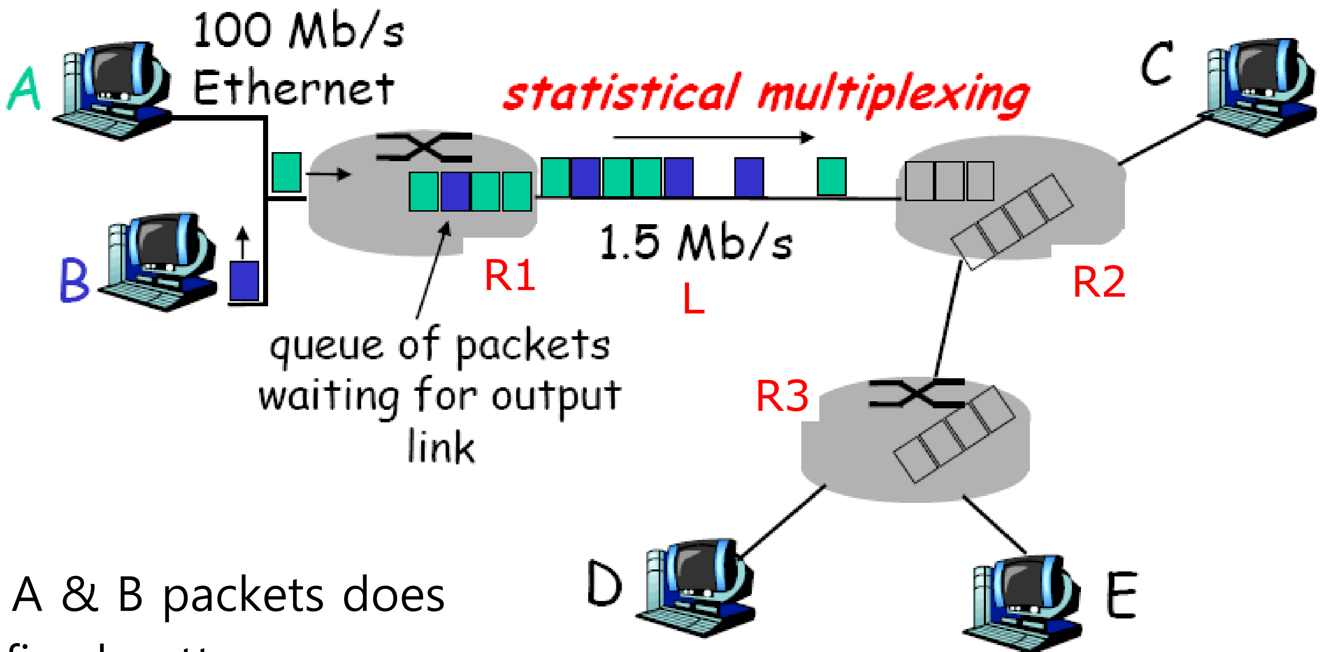
- User packets *share* network resources
- Resources assigned to packet *as needed*
- Each packet uses full link bandwidth



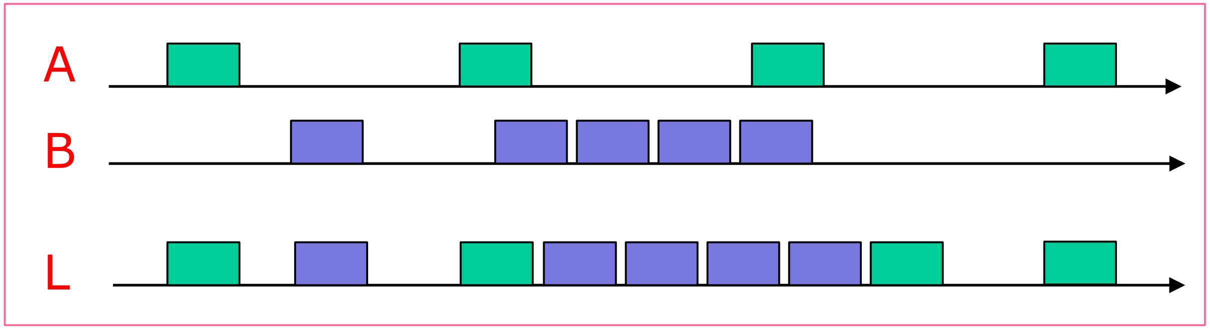
Resource contention:

- aggregate resource demand can exceed amount available -> needs packet queue (buffer)
- congestion: packets queued, wait for link use
- store and forward: packets move one hop at a time
  - Node receives complete packet before forwarding

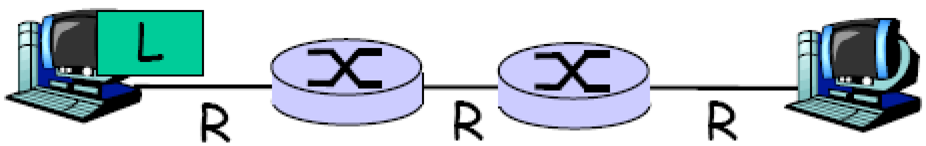
Packet Switching: Statistical Multiplexing



Sequence of A & B packets does
not have fixed pattern,
bandwidth shared on demand :
statistical multiplexing



Packet-switching: store-and-forward



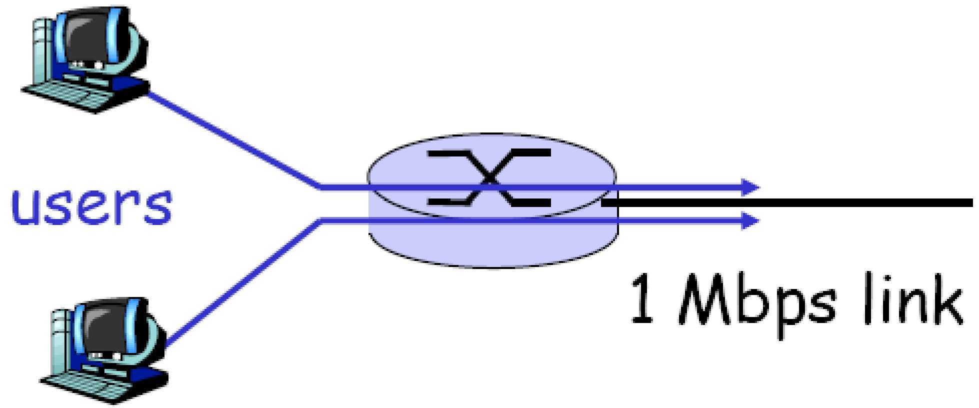
- L/R seconds to transmit packet of L bits on a link at R bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
- End-to-end delay = 3*L/R (assuming zero queuing delay and propagation delay)

Example:

- L = 7.5 Mbits
- R = 1.5 Mbps
- Transmission delay per link = L/R = 5 sec

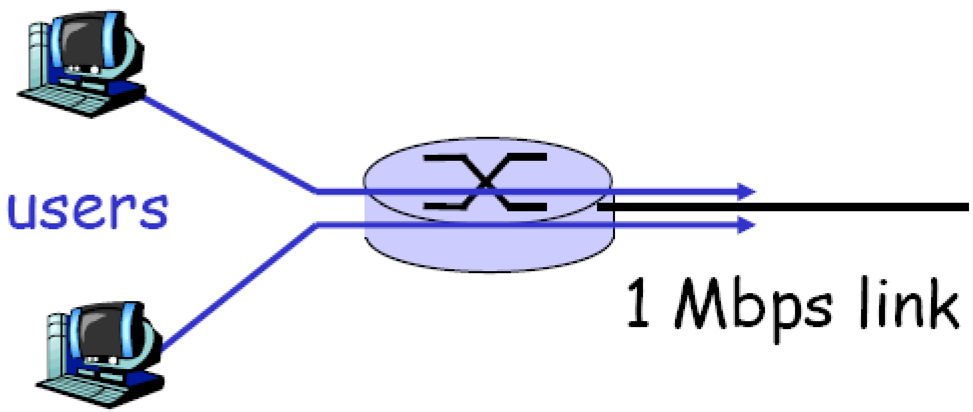
Packet switching versus circuit switching

- 1 Mb/s link
- Each user:
  - 100 kb/s when "active"
  - active 10% of time
- *Circuit-switching:*
  - 10 users



Packet switching versus circuit switching

- 1 Mb/s link
- Each user:
  - 100 kb/s when "active"
  - active 10% of time



- *Packet switching:*
  - no limits on the number of users: with 35 users, probability > 10 active at same time is less than .0004 (how to get?)

$$ {}_{35}C_{11}(0.1)^{11}*(0.9)^{24} + {}_{35}C_{12}(0.1)^{12}*(0.9)^{23} + \dots $$
$$ \dots $$
$$ {}_{35}C_{35}(0.1)^{35}*(0.9)^{0} < 0.0004 $$

Packet switching versus circuit switching

Packet switching

- great for bursty data
  - resource sharing → efficient use of network resources
  - simpler, no call setup
- congestion: packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still an unsolved problem

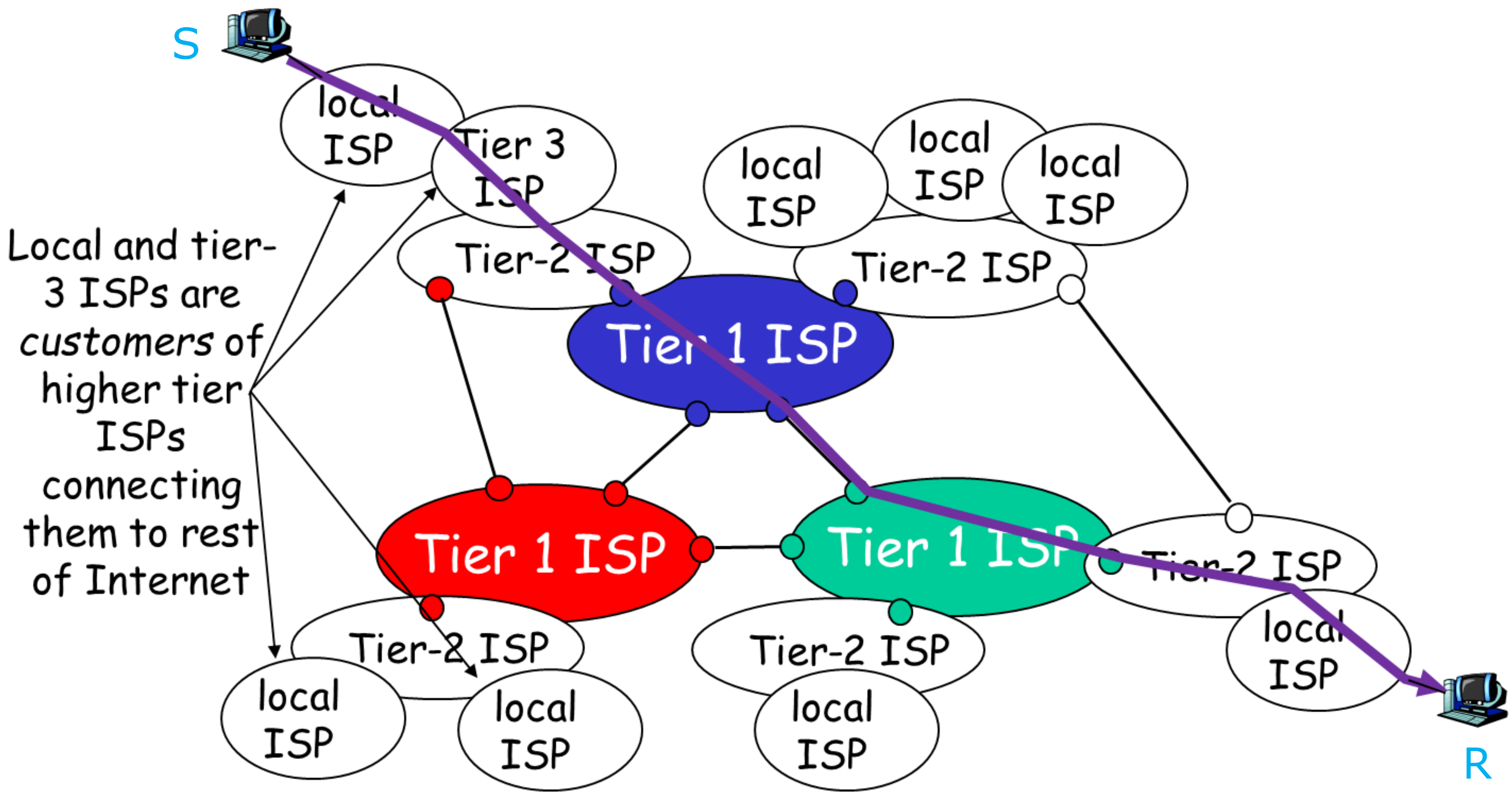
Network Core: Internet Structure

Internet

- network of networks
- roughly hierarchical
- Multi-levels of ISPs
  - Tier1 ISP – tier2 ISP – tier3 ISP (local ISP)
  - Lower tier ISPs rely on higher tier ISPs for Internet connectivity to larger area

Network Core: Internet Structure

Internet: Hierarchical networks

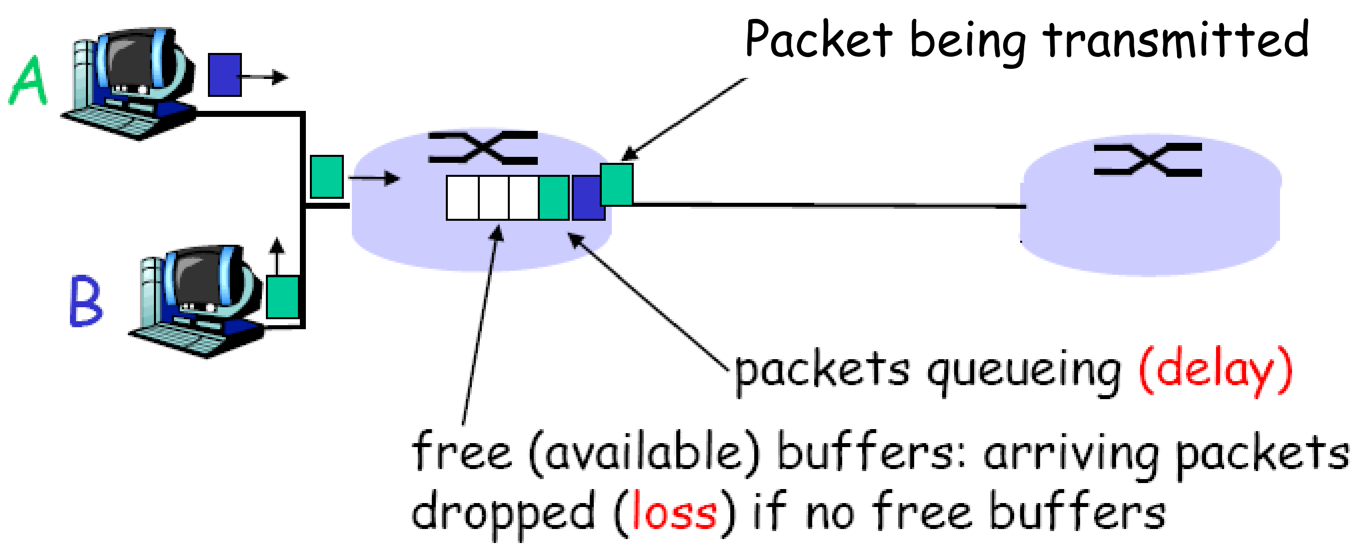


Network Performance: Loss, Delay, Throughput

Loss and delay in Internet:

- packet *queue* in router buffers: store-and-forward
- packet arrival rate to link exceeds output link capacity

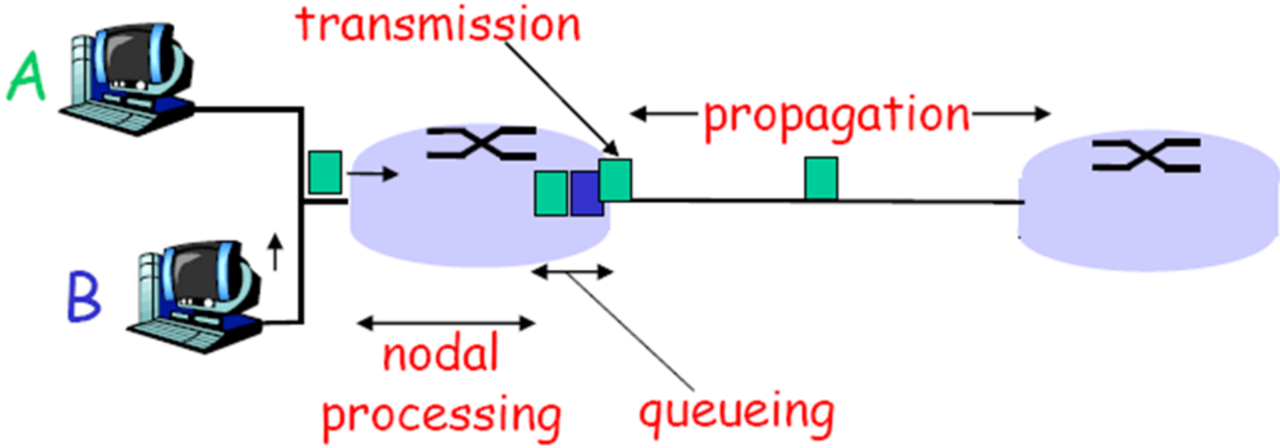
  → packets queue, wait for turn



Network Performance

Four sources of packet delay (1-hop delay):

- nodal processing:
  - check bit errors
  - determine output link
- queuing
  - time waiting at output link queue for transmission
  - depends on congestion level
- Transmission delay
  - time to send bits into link = L/R
- Propagation delay
  - d = length of physical link
  - s = propagation speed in medium (~2×10^8 m/sec)
  - propagation delay = d/s



Network Performance

Nodal delay (1-hop delay):

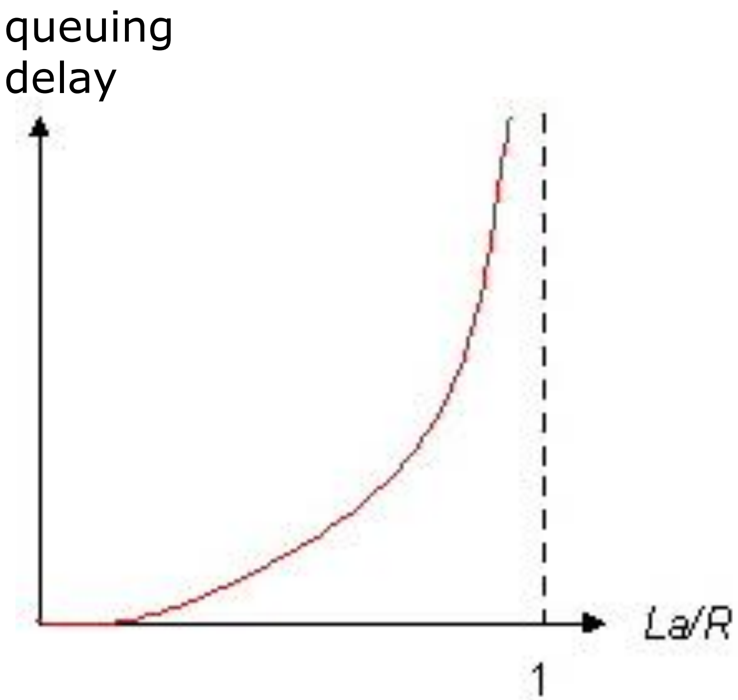
$$\boxed{d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}}$$

- d_{proc} = processing delay
  - typically a few microsecs or less
- d_{queue} = queuing delay
  - depends on congestion level
- d_{trans} = transmission delay
  - = L/R, significant for low-speed links
- d_{prop} = propagation delay
  - depends on length of the link; a few microsecs to 100s msecs

Queuing Delay

Traffic intensity: L*a/R

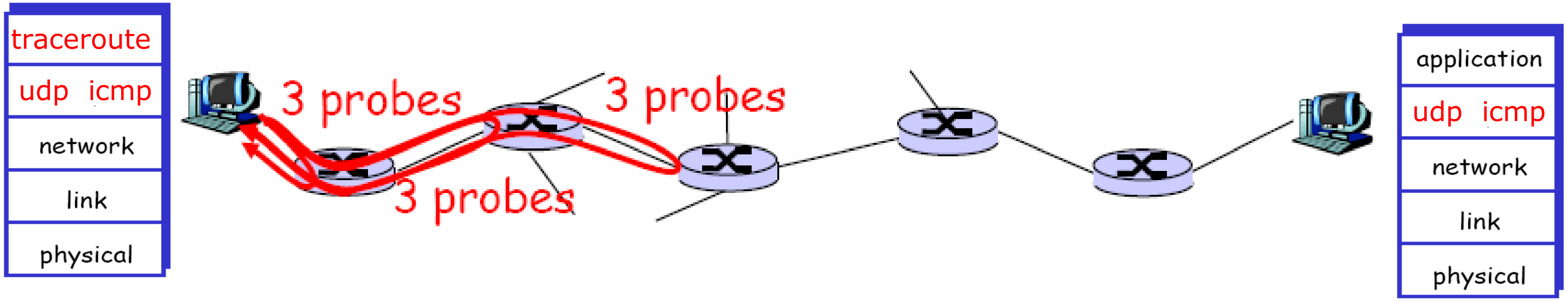
- R=link bandwidth (bps)
- L=packet length (bits)
- a=average packet arrival rate



- La/R ~ 0: average queuing delay small
- La/R -> 1: delays become large
- La/R > 1: more "work" arriving than can be serviced, average delay infinite!

Delay on Internet

- Experiment using traceroute
- **Traceroute program:** provides delay measurement from source to router along end-end Internet path towards destination: for all *i:*
  - sends packets that will reach router *i* on path towards dest.
  - router *i* sends response packet to sender
  - sender times interval between transmission and reply



Delay on Internet

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

Three delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

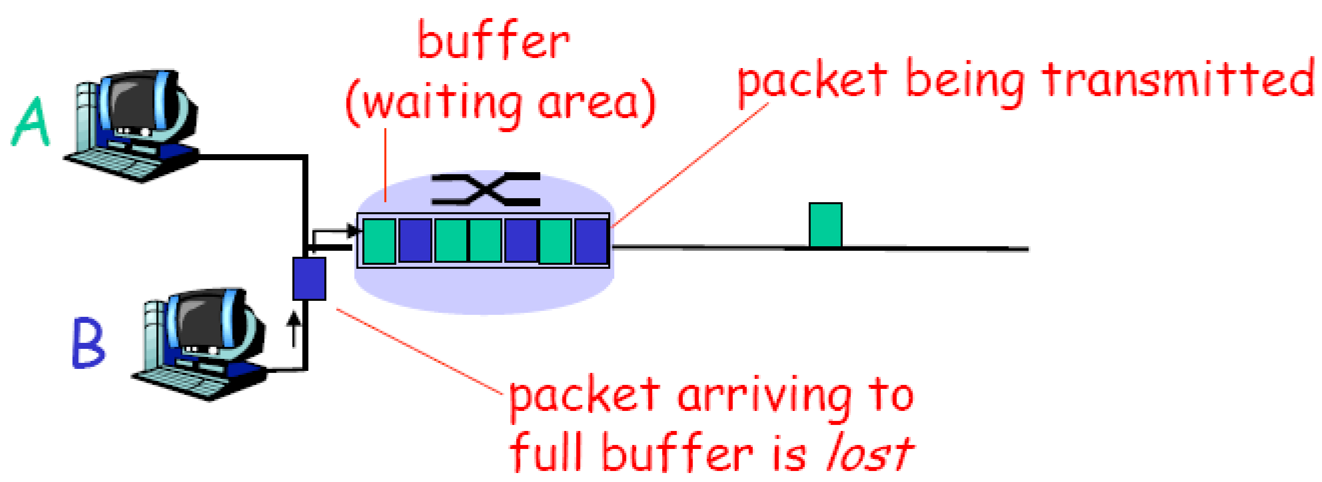
```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 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

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

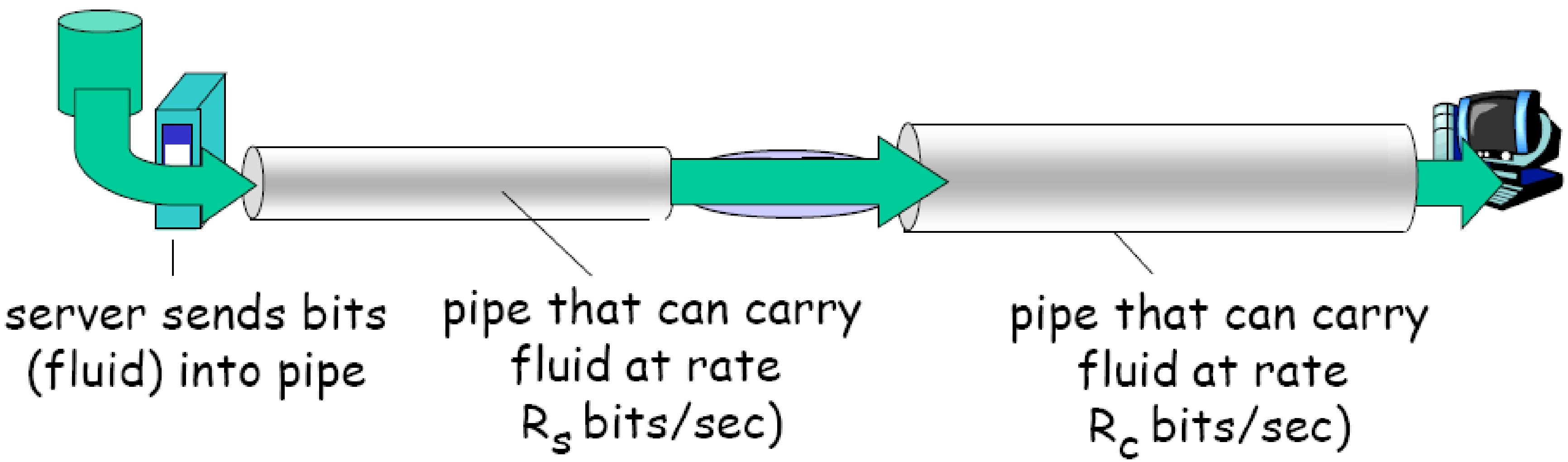
Packet Loss

- Packet buffer has finite capacity
- packet arriving to full queue dropped (lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



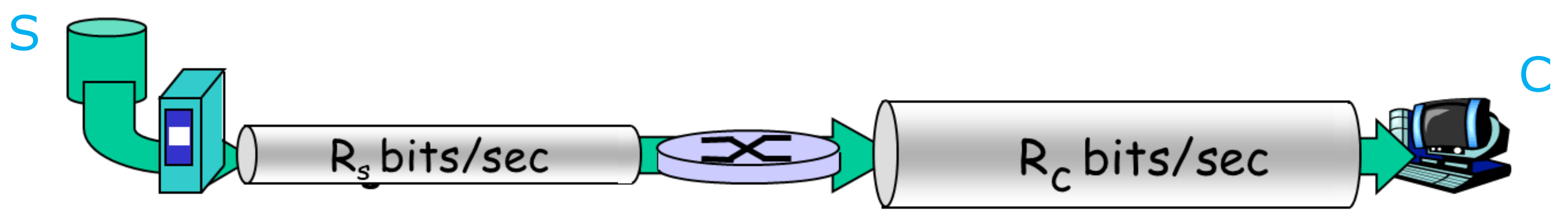
Throughput

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

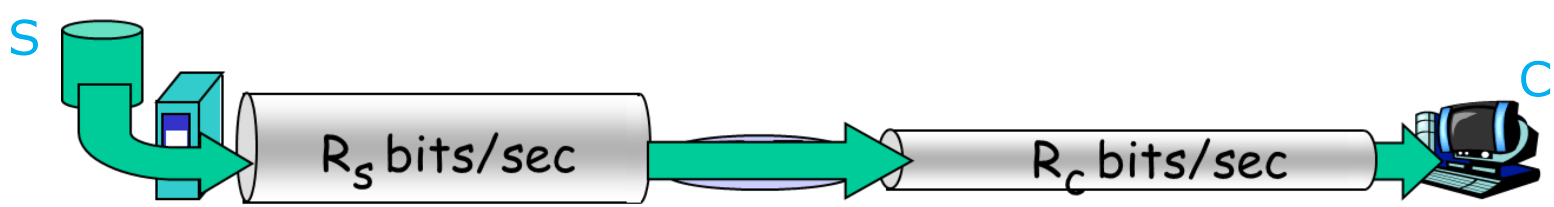


Throughput (more)

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



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



bottleneck link

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

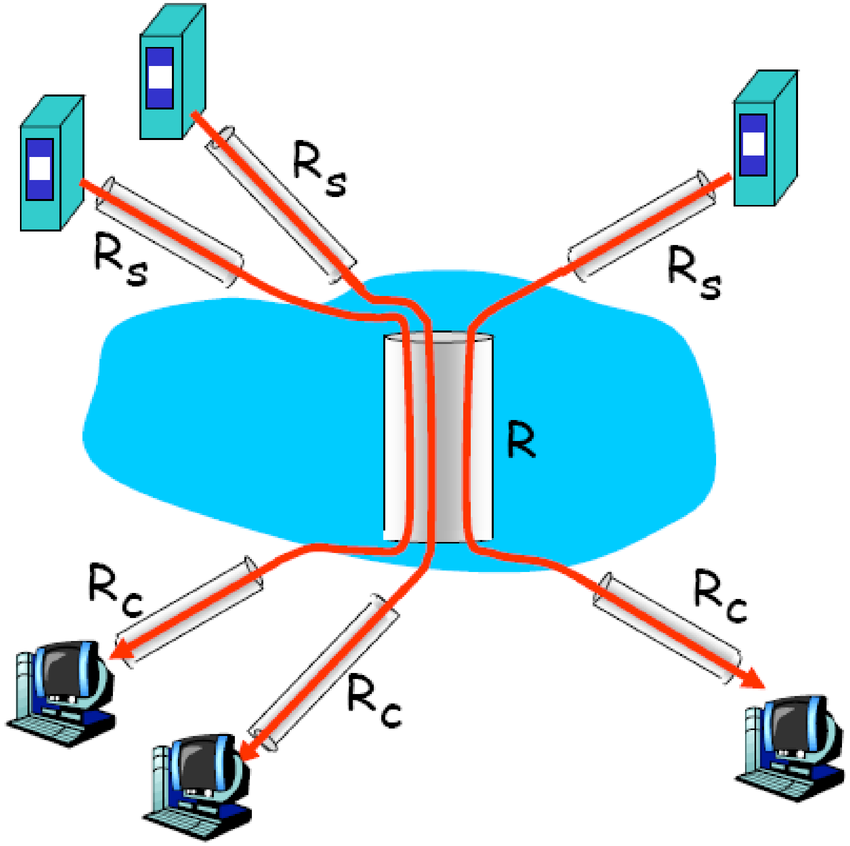
Throughput (more)

Internet scenario

- per-connection end-end throughput:

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

- in practice: R_c or R_s is often bottleneck



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

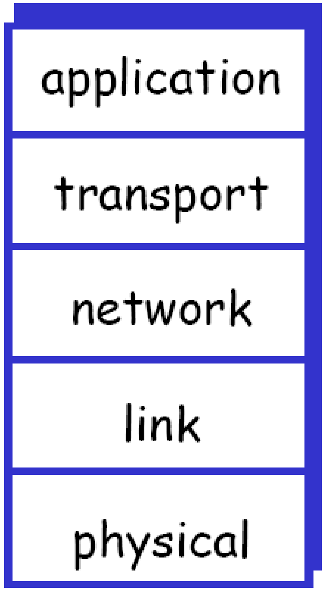
Protocol Layers

Networks are complex!

- many "pieces":
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

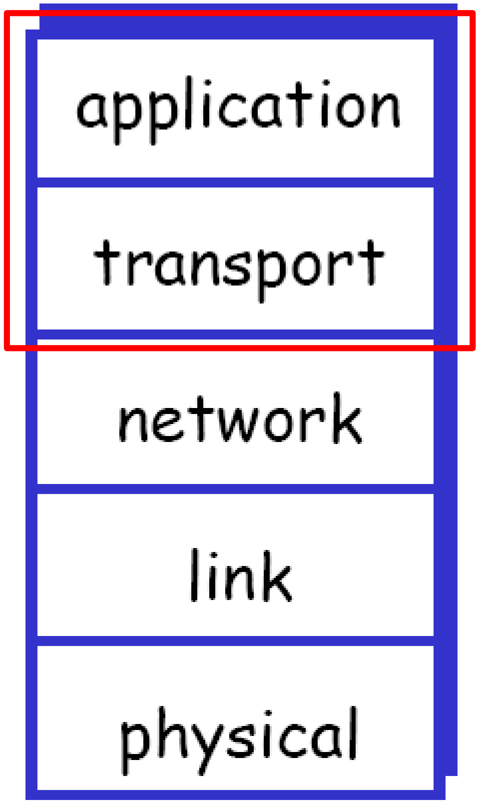
Why layering?

- modularization eases maintenance, updating of system
- Implementation change of layer's service transparent to rest of system
- sometimes cross-layer design used



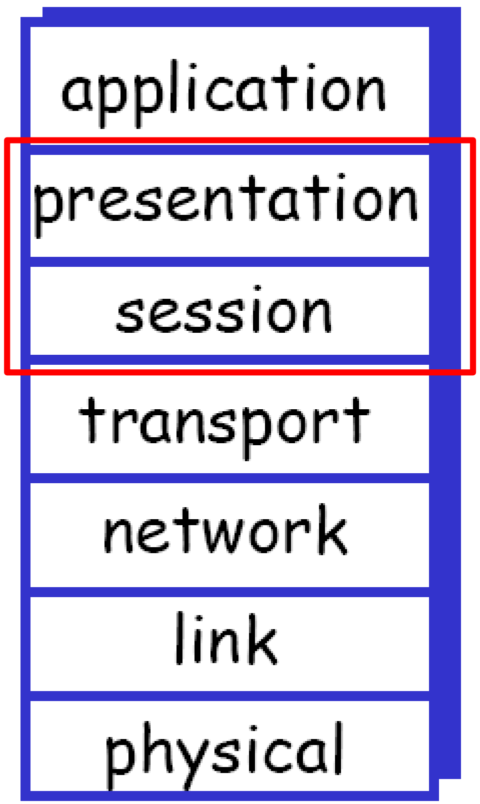
Internet Protocol Stack

- application: supporting network applications
  - FTP, SMTP, HTTP
- 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
  - PPP, Ethernet, WLAN
- physical: bits “on the wire”



ISO/OSI Reference Model

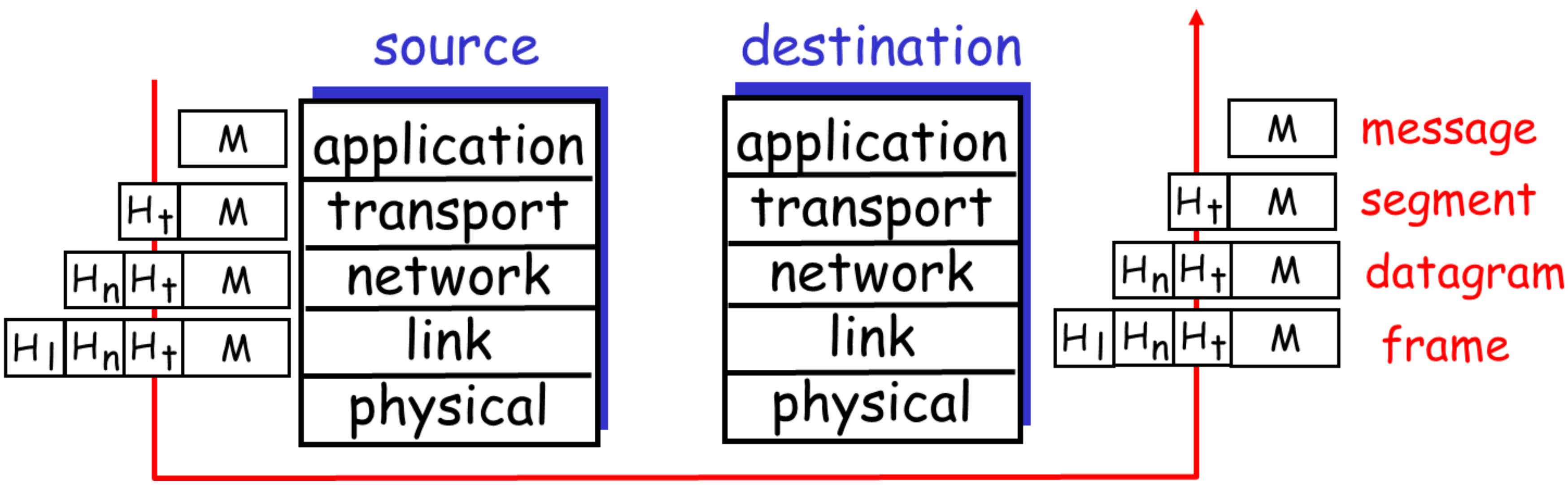
- presentation:
  - allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- session:
  - synchronization, checkpointing, recovery of data exchange
- These services, *if needed,* are implemented in application



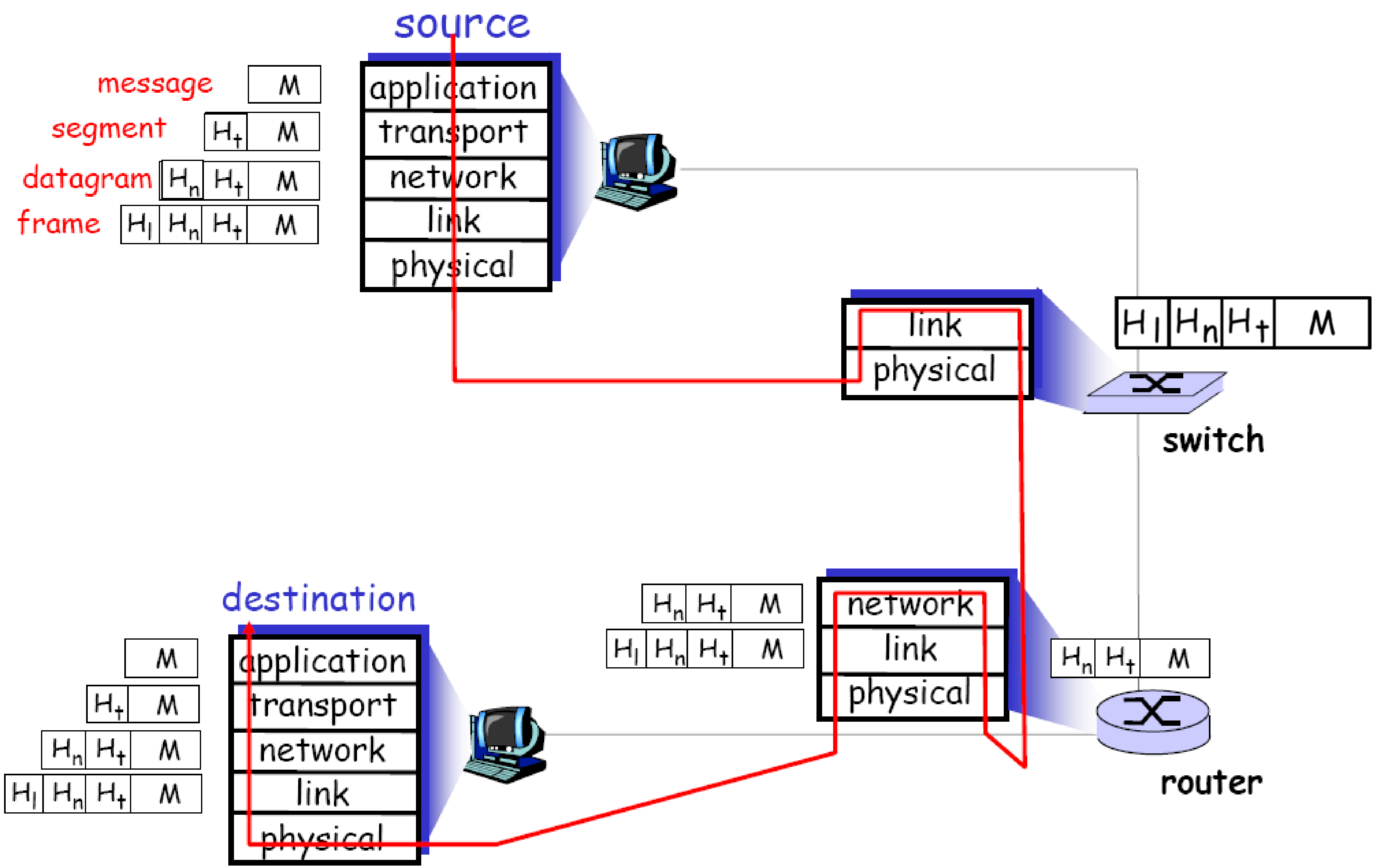
Internet Protocol Stack

Encapsulation/decapsulation

- Each layer takes data from above
- adds header information to create new data unit
- passes new data unit to layer below



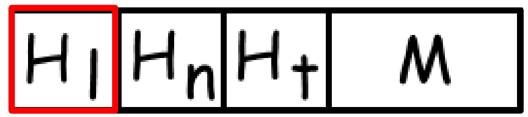
Internet Protocol Stack



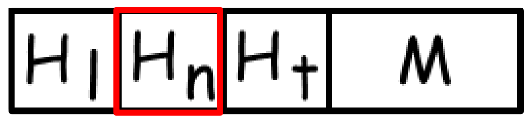
Internet Protocol Stack

Packet forwarding

- Switch: L2 device



  - frame forwarding based on MAC address in L2 layer
- Router: L3 device



  - datagram forwarding based on IP address in L3 layer
- L4 switch or Application gateway: L4 or L7 device
  - packet forwarding based on L4 or higher layer information