# Computer Networks and the Internet

Instructor: C. Pu (Ph.D., Assistant Professor)

Lecture 04

*puc@marshall.edu*

# Overview of Delay

- view Internet as an infrastructure that provides services to distributed app. running on end systems

- ideally,
  - internet services can move as much data as needed between any two end systems
  - no any loss of data

- reality,
  - not gonna happen
  - networks necessarily constrain throughput, introduce delays, and cause packets loss
    - throughput: the amount of data per second that can be transferred

# Overview of Delay

- How a packet is transmitted in the network?
  - starts in a host (a source)
  - passes through a series of routers
  - ends in another host (a destination)

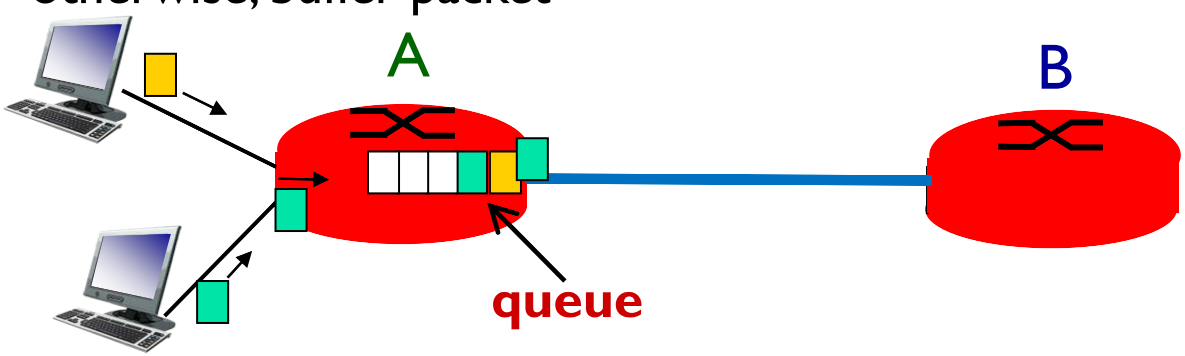
- As a packet travels from one node to the subsequent node, the packet suffers from several types of delays at *each* node
  - **Nodal Processing Delay**
  - **Queuing Delay**
  - **Transmission Delay**
  - **Propagation Delay**

  } **Total Nodal Delay**

- The performance of Internet application affected by network delay
  - Search, Web browsing, e-mail, instant messaging, etc.

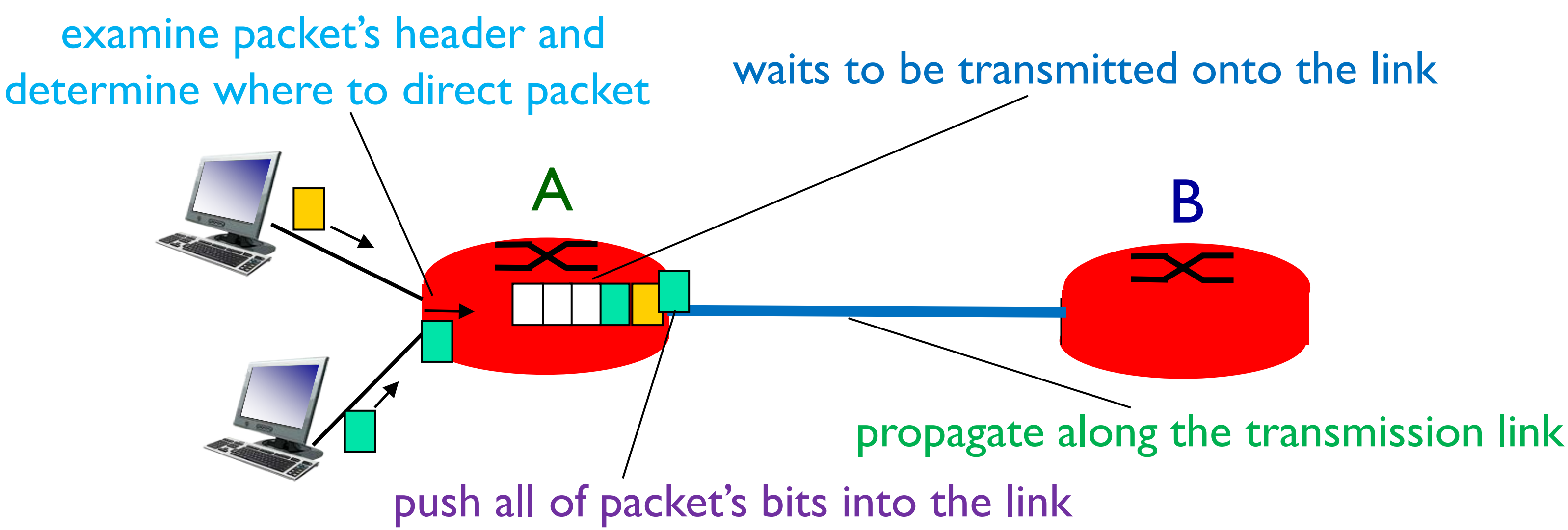
# Types of Delay

- A packet is sent from upstream node through router A to router B (characterize the nodal delay at router A)
  - when packet arrives at router A from upstream node
    - examine packet's header to determine the appropriate outbound link
    - direct the packet to this link
  - a packet can be transmitted on the link only
    - if no other packet currently being transmitted
    - if no other packet preceding it in the queue
    - otherwise, buffer packet

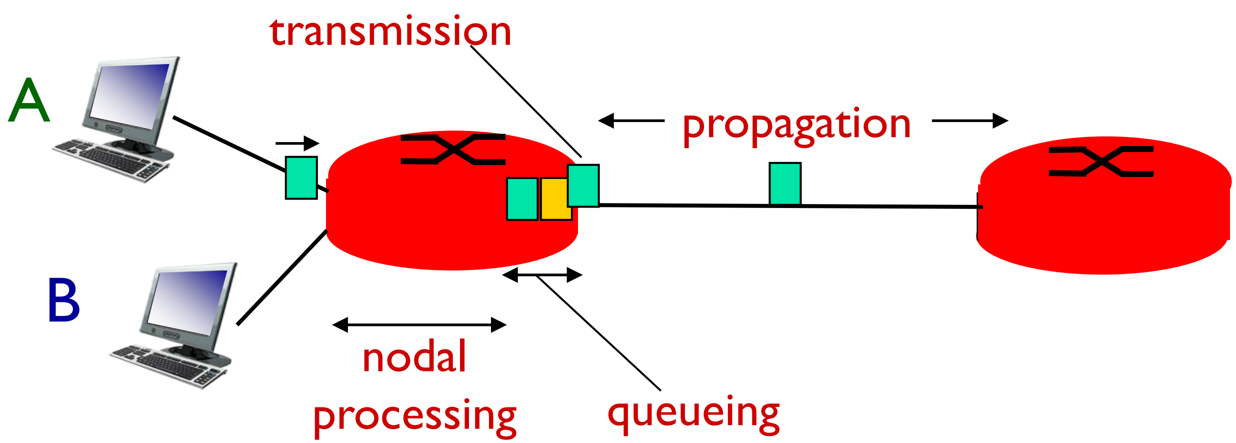


# Types of Delay

- A packet is sent from upstream node through router A to router B
  - **Processing Delay** — microseconds or less
  - **Queuing Delay** — microseconds to milliseconds
  - **Transmission Delay** — microseconds to milliseconds
  - **Propagation Delay** — milliseconds



# Four Sources of Packet Delay




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

$d_{proc}$: nodal processing

- determine output link
- check bit errors
- typically < micro sec

$d_{trans}$ and $d_{prop}$ *very* different

$d_{queue}$: queueing delay

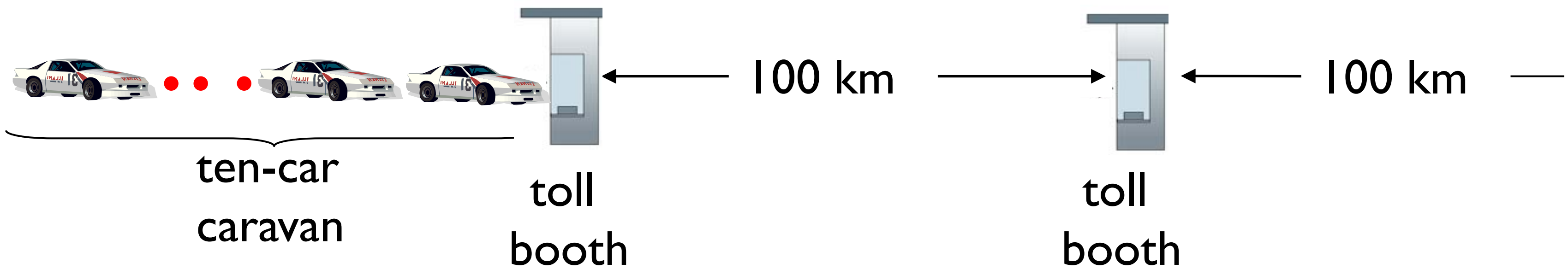
- time waiting at output link for transmission
- depends on congestion level of router

# Transmission vs. Propagation Delay

- **Transmission Delay**
  - the amount of time required for the router to push (this is, transmit) all of the packet's bits into the link
  - a function of packet's length (*L*) and transmission rate of the link (*R*)
    - L / R
  - has nothing to do with distance between two routers
- **Propagation Delay**
  - once a bit is pushed into the link, it needs to propagate
  - the amount of time required to propagate a bit from one router to another router
  - a function of distance between two routers (*d*) and propagation speed (*s*)
    - propagation speed = or little < the speed of light
    - d / s
  - has nothing to do with packet's length or trans. rate of the link

the time from when the caravan is stored in front of a tollbooth until the caravan is stored in front of the next tollbooth is the sum of transmission delay and propagation delay

# Caravan Analogy

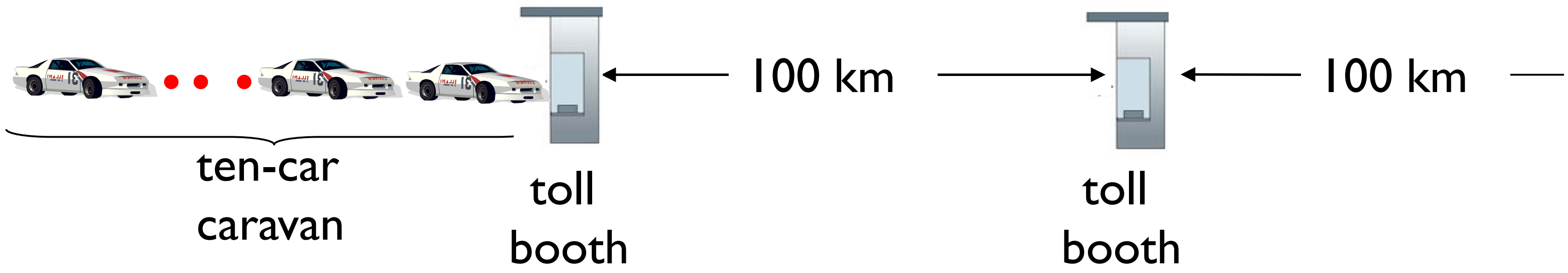



- tollbooth ~ router; highway segment ~ link
- cars "propagate" at 100 km/hr
- car ~ bit; caravan ~ packet
- tollbooth takes 12 sec to serve car (bit transmission time)
- the car arrives at a tollbooth, it waits for others
- *Q:* How long until caravan is lined up at 2nd toll booth?

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

the first bits in a packet can arrive at a router while many of the remaining bits in the packet are still waiting to be transmitted by the preceding router

# Caravan Analogy (cont.)



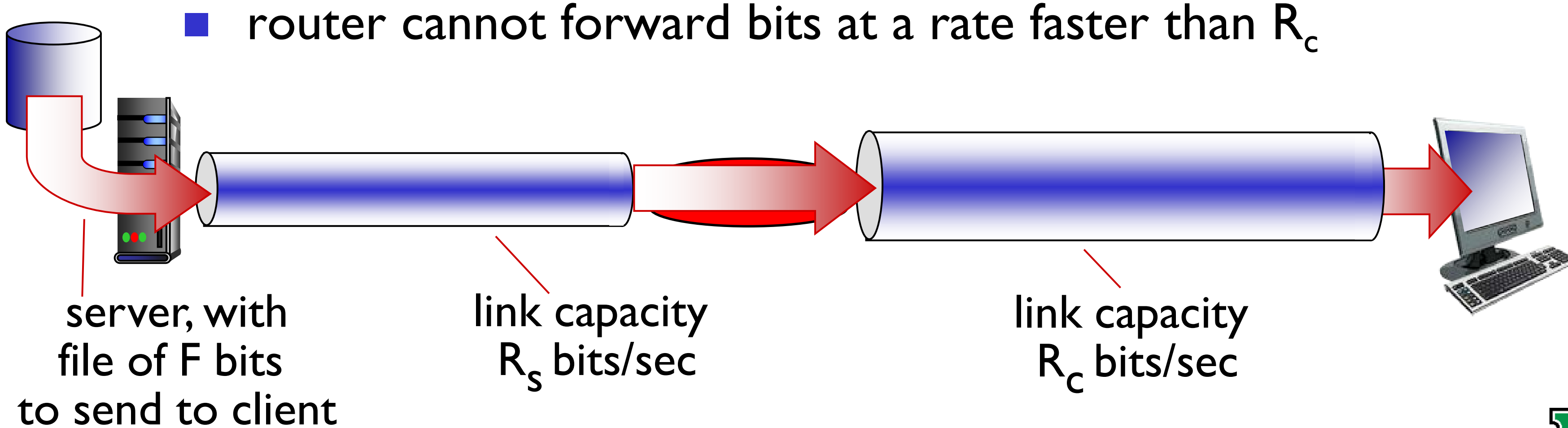

- suppose cars now "propagate" at 1000 km/hr
- suppose tollbooth now takes one min to service a car
- *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?
- *A: Yes!* after 7 min, 1st car arrives at second booth; three cars still at 1st booth.

# Throughput

- *end-to-end throughput*
- consider transferring a large file from host A to Host B
  - the transfer might be a large video clip
- ***the instantaneous throughput*** at any instant of time is the rate (in bits/sec) at which host B is receiving the file
- if the file consists of *F* bits and the transfer takes *T* seconds for host B to receive all *F* bits, then the ***average throughput*** of the file transfer is *F*/*T* bit/sec

# Throughput

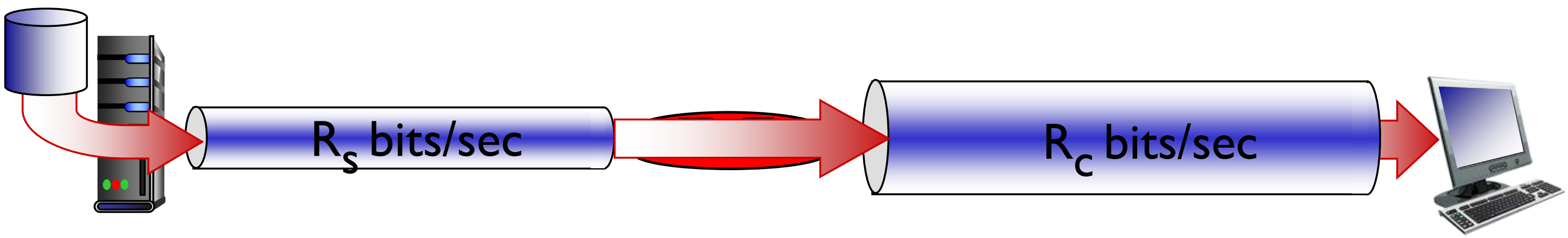
- two end systems:
  - a server and a client
  - connected by two communication links and a router
- we ask, in ideal scenario, what is the server-to-client throughput?
  - server cannot pump bits through its link at a rate faster than $R_s$
  - router cannot forward bits at a rate faster than $R_c$



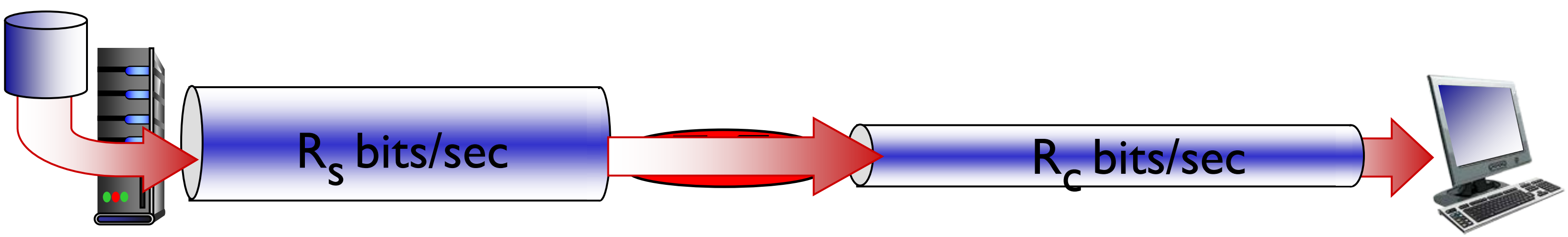
e.g., download an MP3 file of 32 million bits. transmission rate of server = 2 Mbps; transmission rate of your link = 1Mbps. The time needed to transfer the file is then 32 seconds.

# Throughput (cont.)

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




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



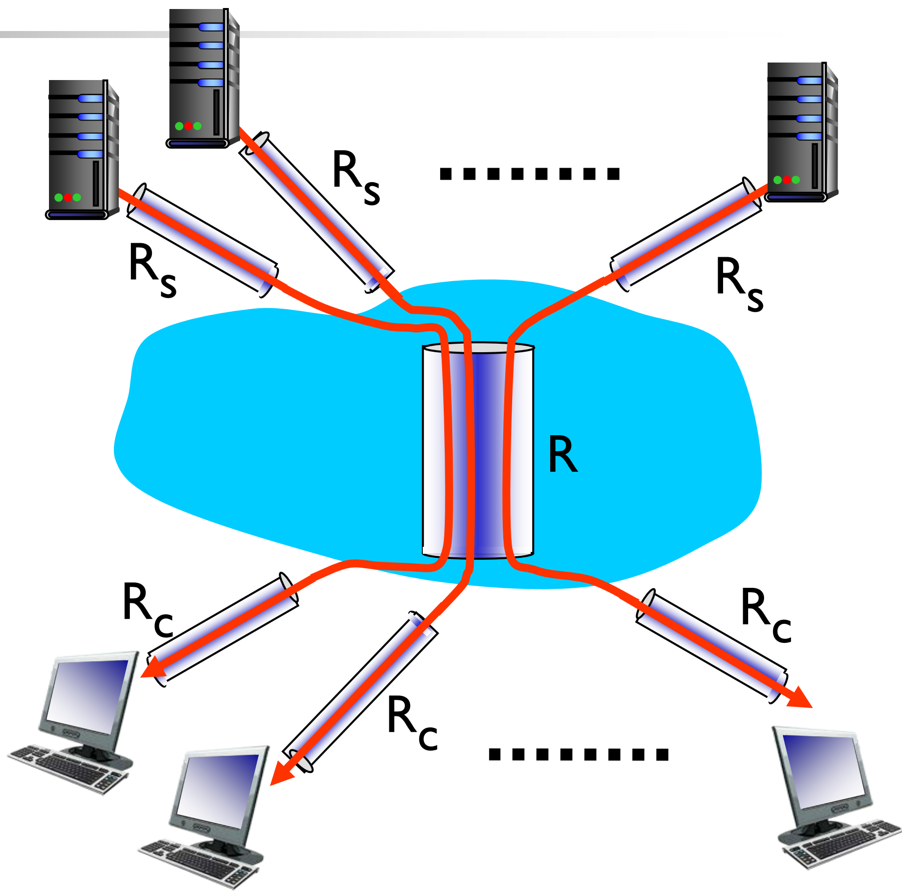

*For this simple two-link network, the throughput is min{$R_s$, $R_c$}*

*bottleneck link*

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

# Throughput: Internet Scenario

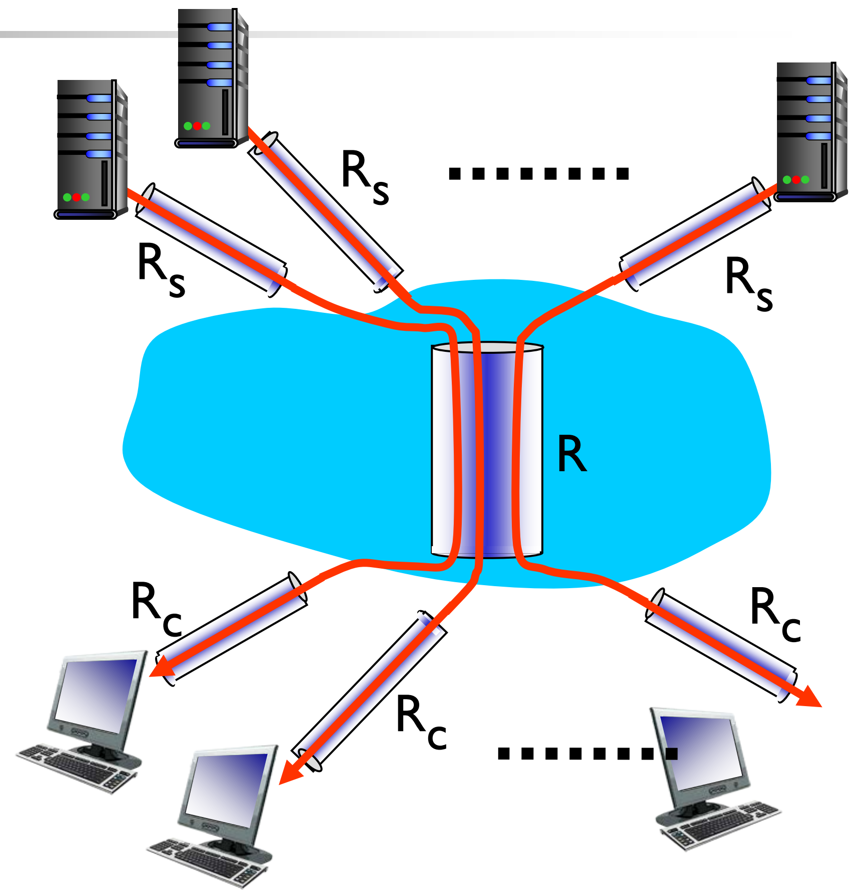
- 10 server and 10 clients
- 10 simultaneous downloads
  - 10 client-server pairs
- The link in the core that is traversed by all 10 downloads
- R: transmission rate of the link in the core
- $R_s$: transmission rate of server access link
- $R_c$: transmission rate of client access link
- What are the throughputs of downloads?




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

# Throughput: Internet Scenario

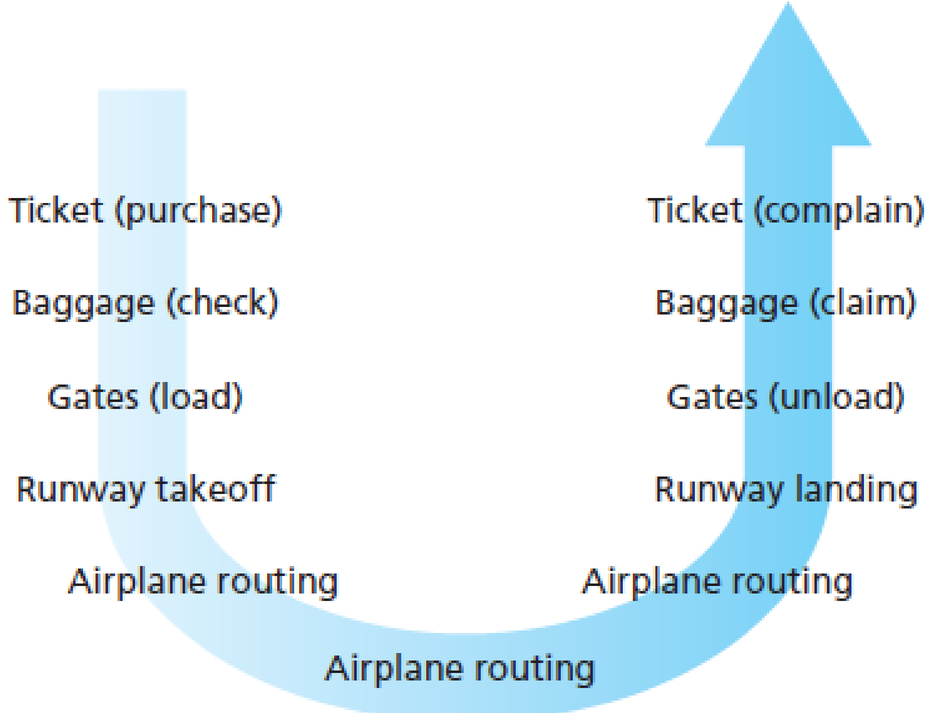
- if R is very large,
  - per-connection end-to-end throughput: min{$R_c$, $R_s$}
- if R is of the same order or $R_c$ and $R_s$
  - per-connection end-to-end throughput: min{$R_c$, $R_s$, R/10}
  - e.g., $R_s$ = 2 Mbps, $R_c$ = 1 Mbps, R = 5 Mbps
    - end-to-end throughput: 500 kbps
- in practice: $R_c$ or $R_s$ is often bottleneck



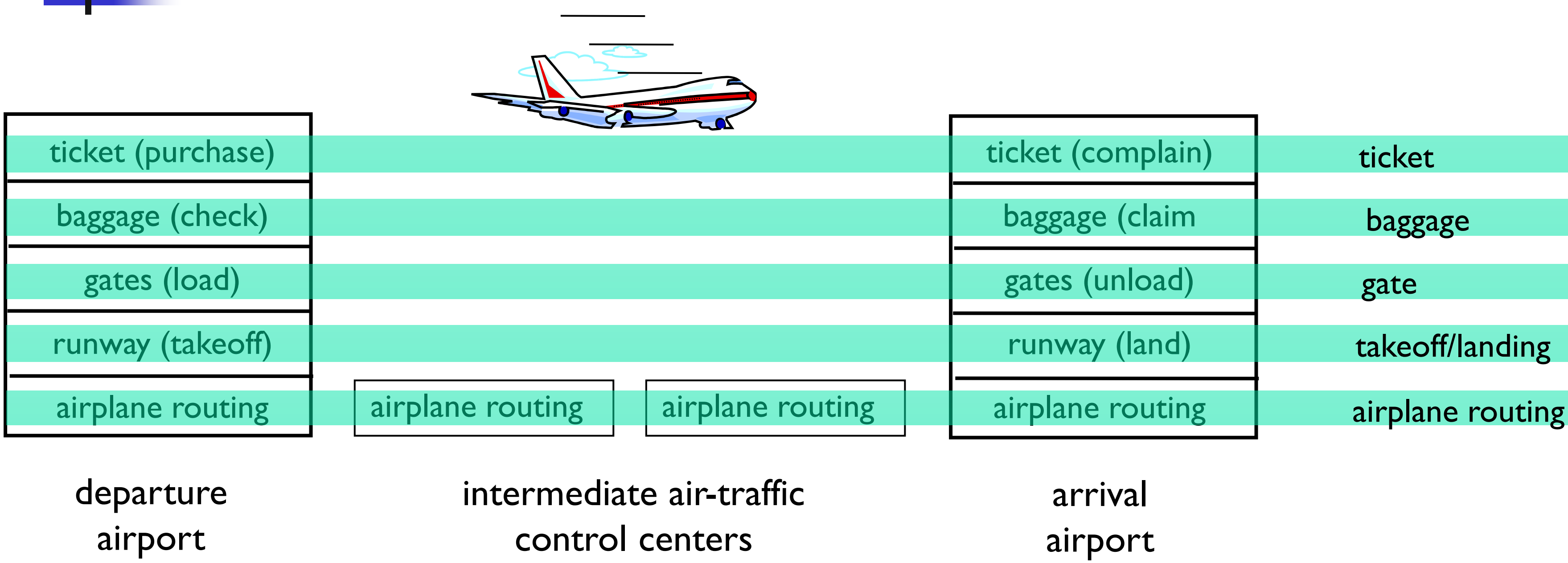

10 connections (fairly) share backbone bottleneck link R bits/sec

# Airline System

- How to describe the airline system?
  - One way to describe this system might be to describe the series of actions you take when you fly on an airline.



# 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?

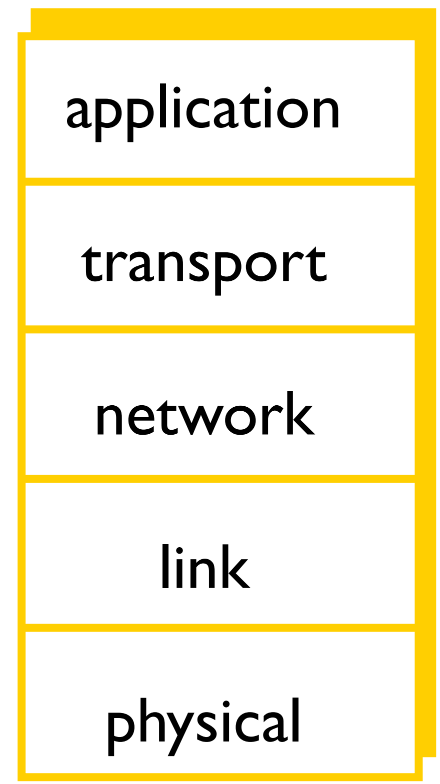
- dealing with complex systems
- explicit structure allows identification, relationship of complex system's pieces

- **modularization** eases maintenance, updating of system
  - change of implementation of layer's service is transparent to the rest of system

- layering – potential drawback?
  - one layer may duplicate low-layer functionality
  - functionality at one layer may need information in another layer
    - violate the goal of separation of layers

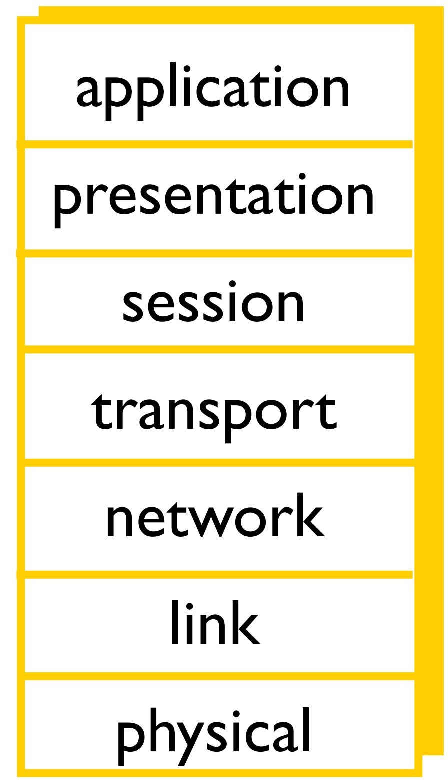
# Internet Protocol Stack

- 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
- Physical:
  - bits “on the wire”

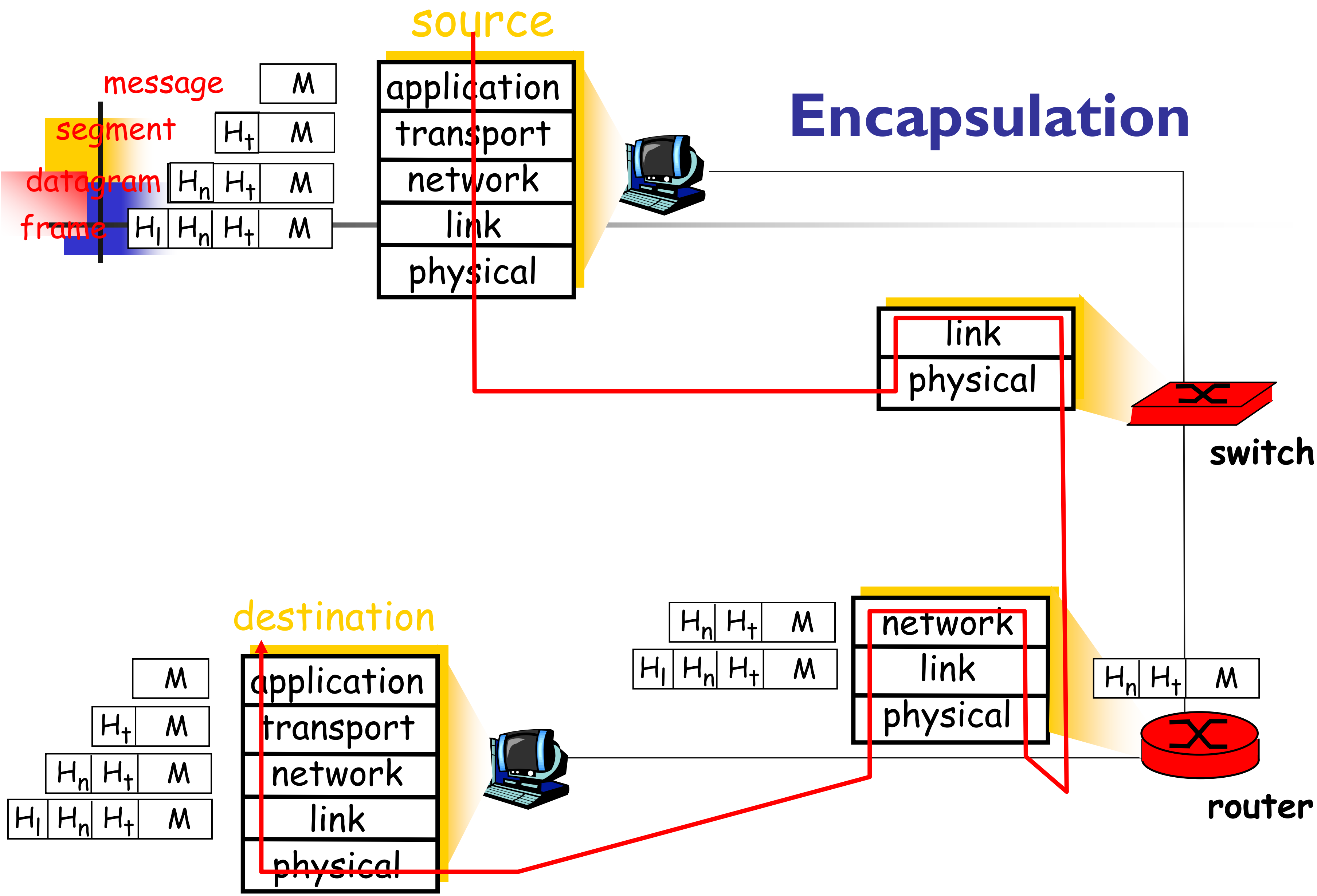


# OSI Reference Model

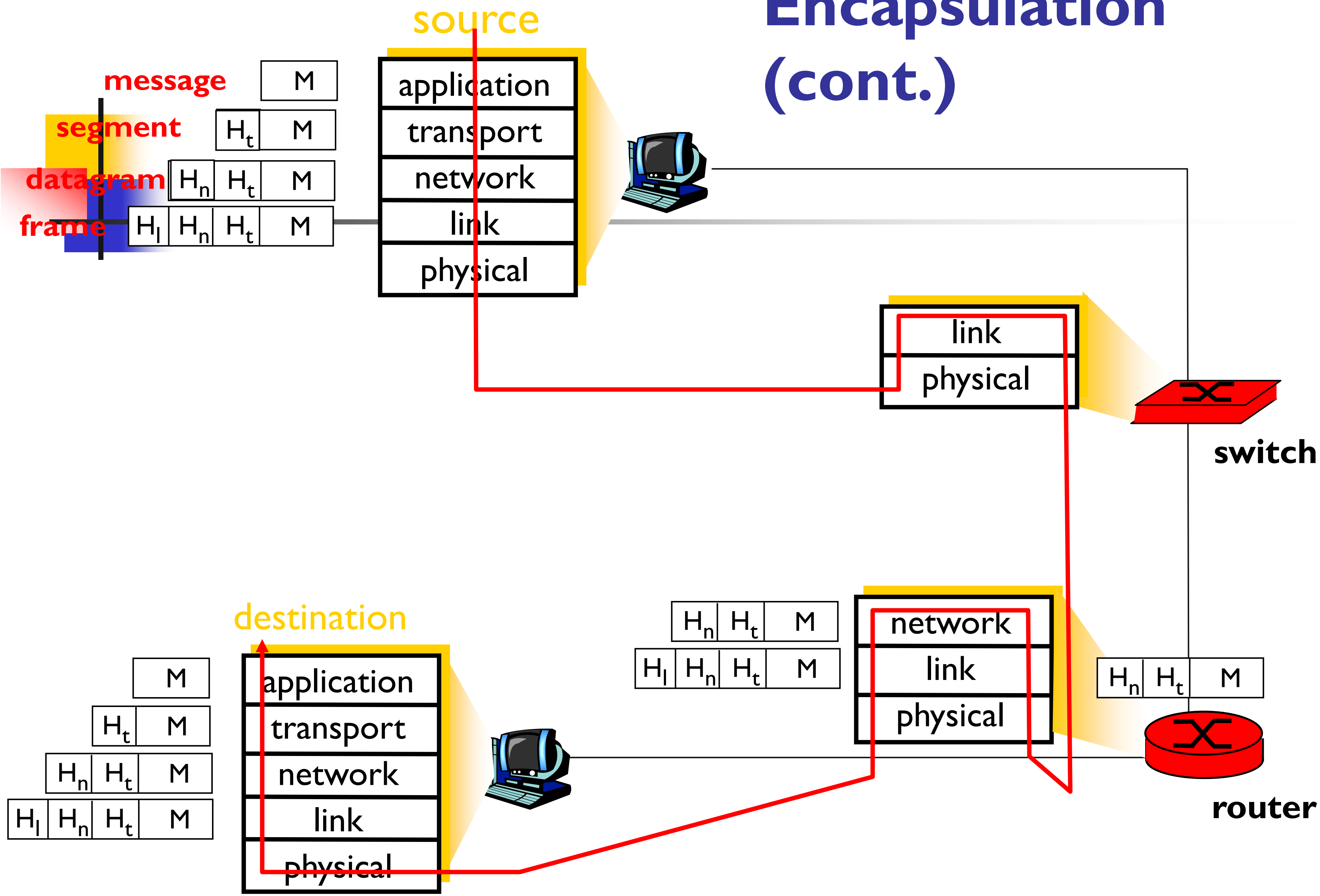
- presentation: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session:* synchronization, checkpointing, recovery of data exchange
- Internet stack "missing" these layers!
  - these services, *if needed,* must be implemented in application



# Encapsulation



# Encapsulation (cont.)



# Another Simple Reference Model