

# Computer Networks Basic Protocols

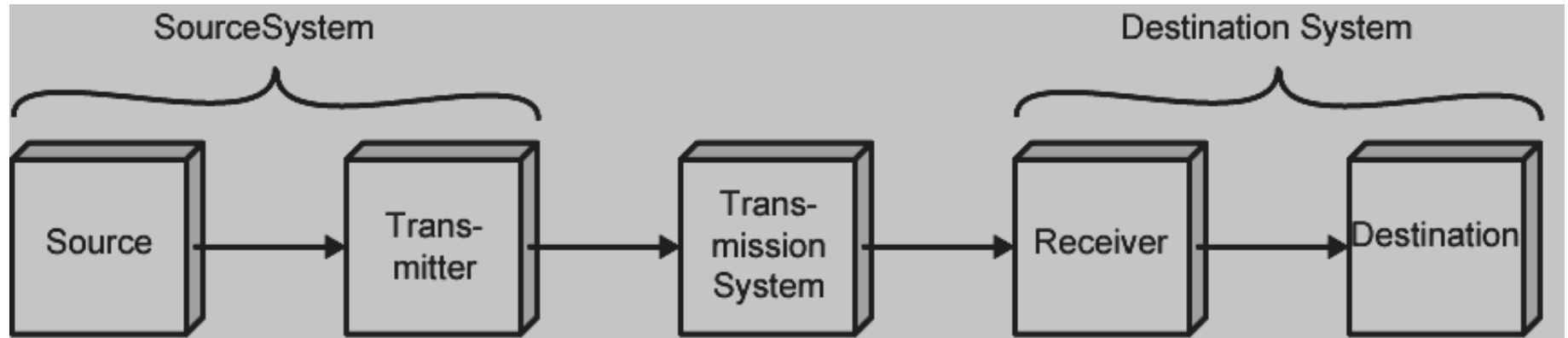
Assoc. Prof. Dr. Berk CANBERK

**13 September 2017**  
**-Introduction-**

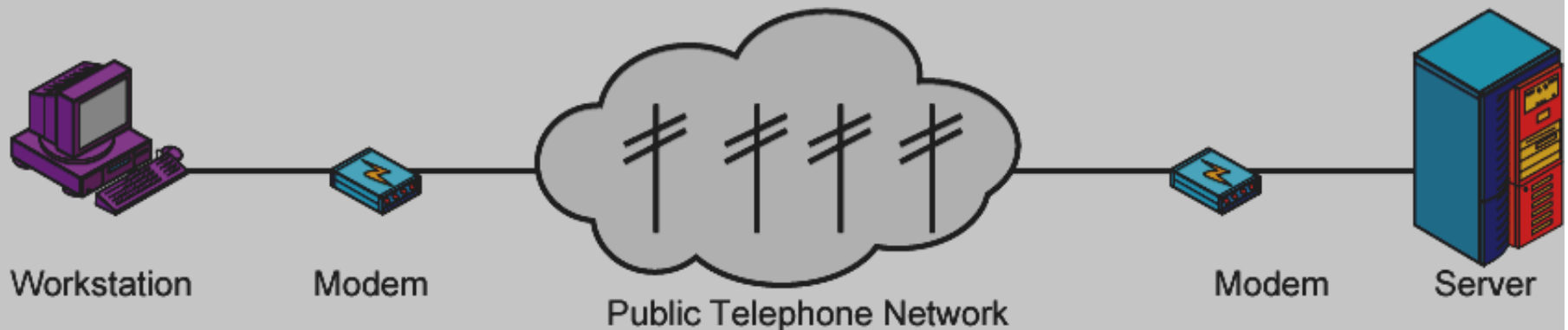
## References:

- Data and Computer Communications*, William Stallings, Pearson-Prentice Hall, 9<sup>th</sup> Edition, 2010.
- Computer Networking, A Top-Down Approach Featuring the Internet*, James F.Kurose, Keith W.Ross, Pearson-Addison Wesley, 6<sup>th</sup> Edition, 2012.

# Communications Model

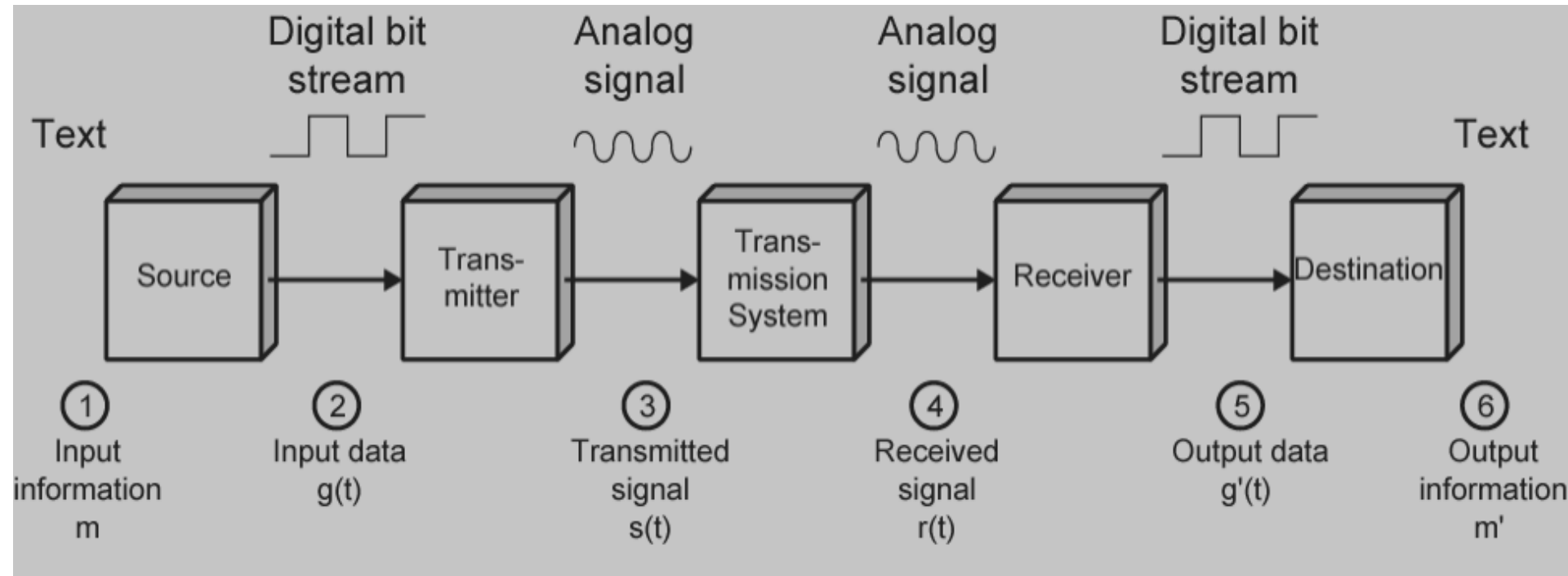


(a) General block diagram

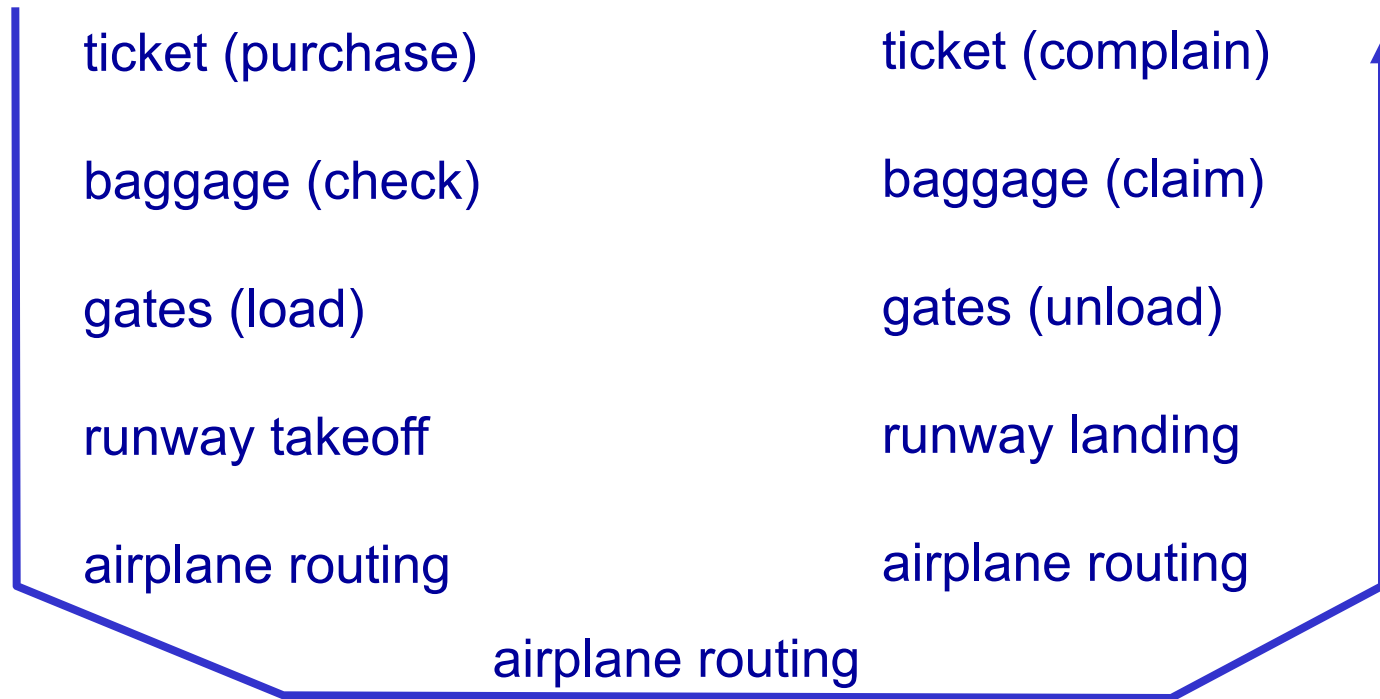


(b) Example

# Data Communications Model

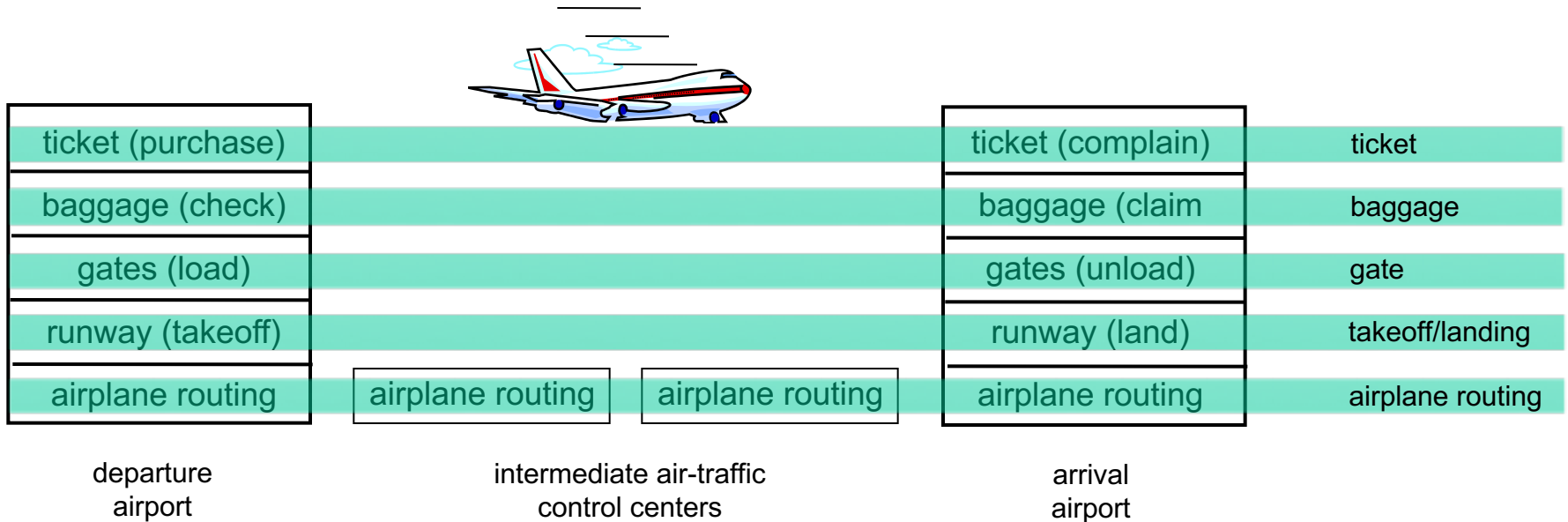


# Organization of air travel



❖ 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:

- ❖ explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- ❖ modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- ❖ layering considered harmful?

# What's a protocol?

## *human protocols:*

- ❖ “what's the time?”
  - ❖ “I have a question”
  - ❖ introductions
- ... specific msgs sent
- ... specific actions taken  
when msgs received, or  
other events

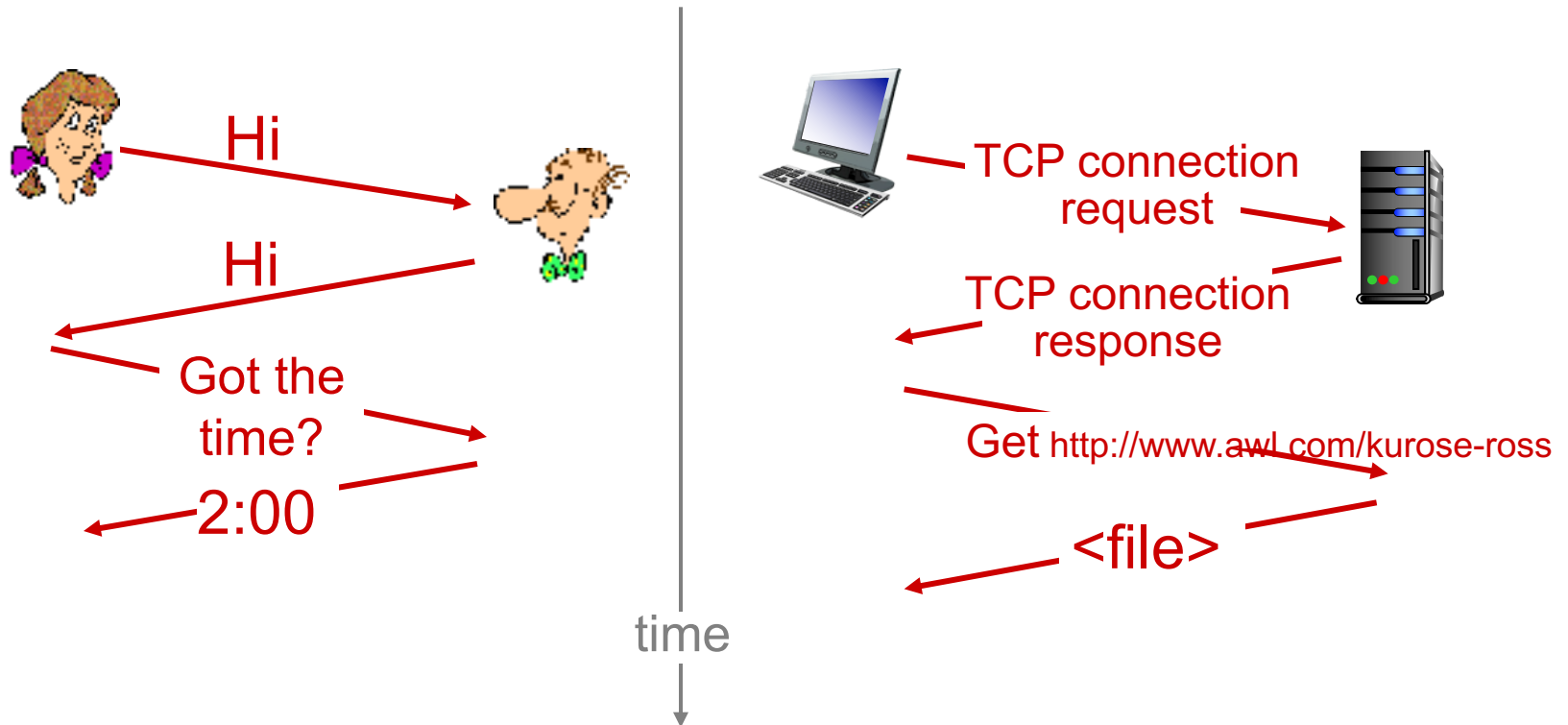
## *network protocols:*

- ❖ machines rather than humans
- ❖ all communication activity in Internet governed by protocols

*protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt*

# What's a protocol?

a human protocol and a computer network protocol:

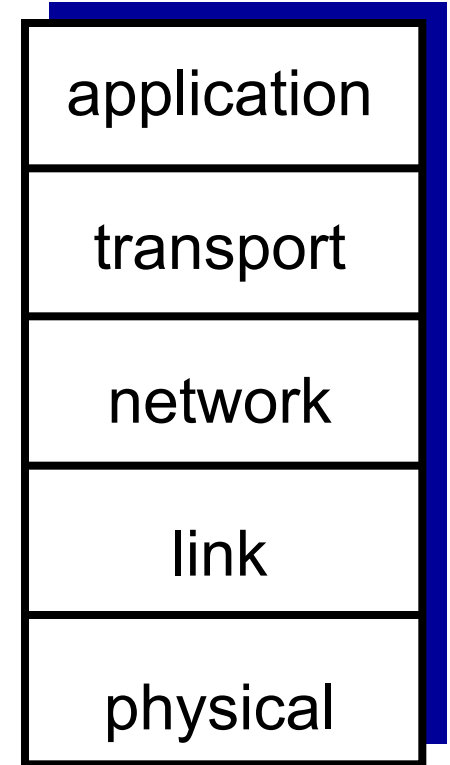


**Q:** other human protocols?



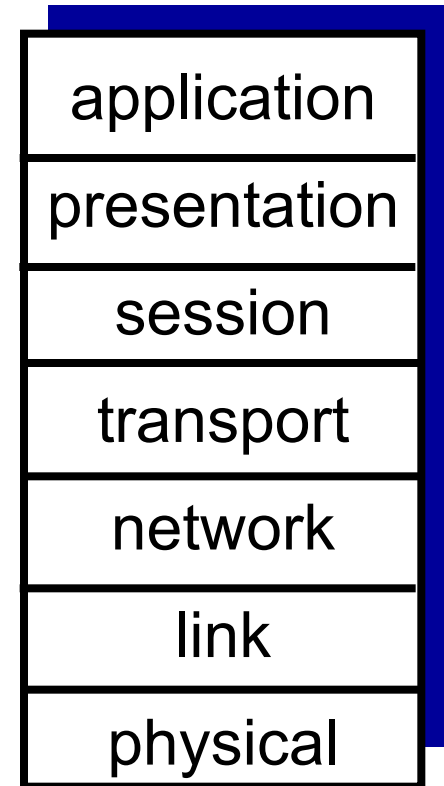
# Internet protocol stack

- ❖ *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
  - Ethernet, 802.111 (WiFi), PPP
- ❖ *physical*: bits “on the wire”

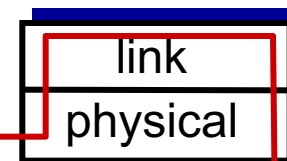
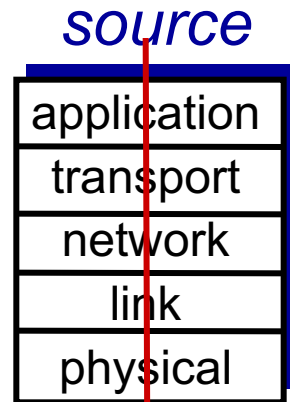
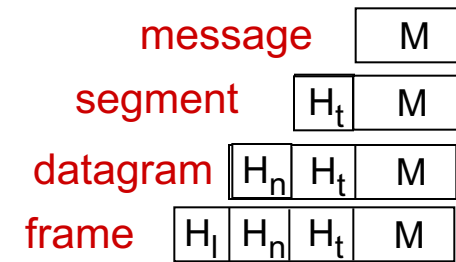


# ISO/OSI reference model

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

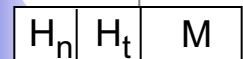
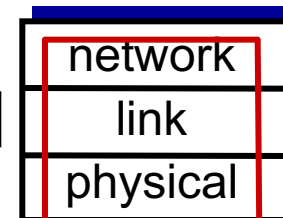
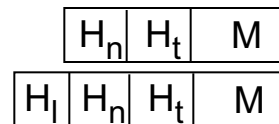
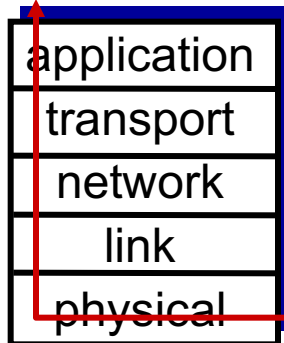
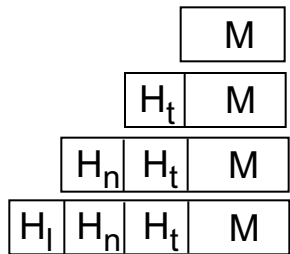


# Encapsulation



switch

*destination*



router



# A closer look at network structure:

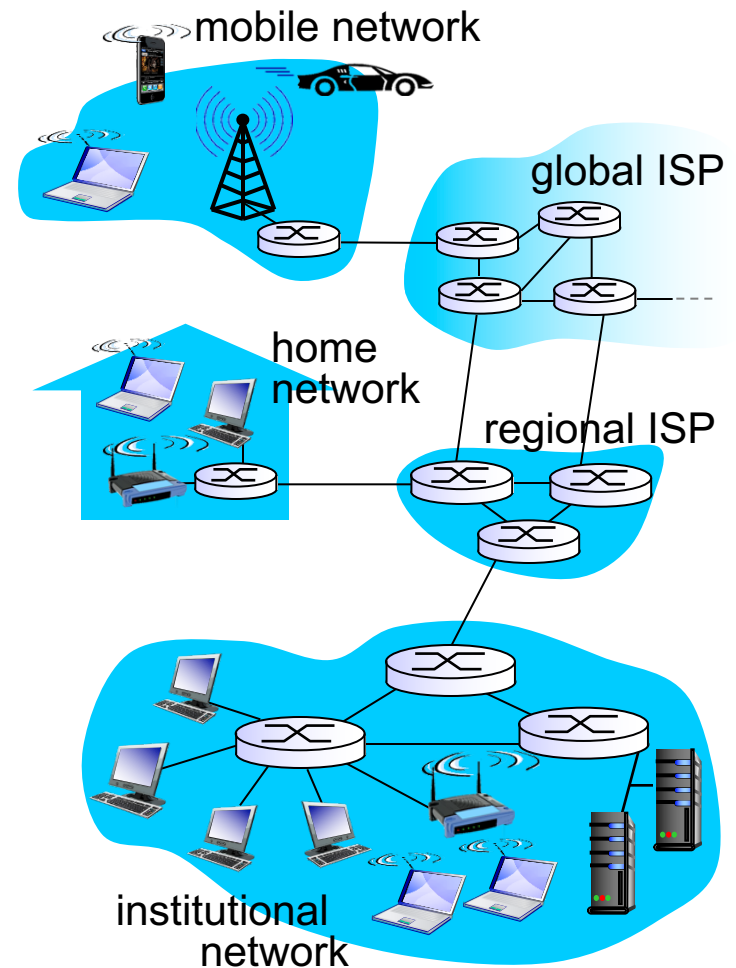
## ❖ *network edge:*

- hosts: clients and servers
- servers often in data centers

## ❖ *access networks, physical media:* wired, wireless communication links

## ❖ *network core:*

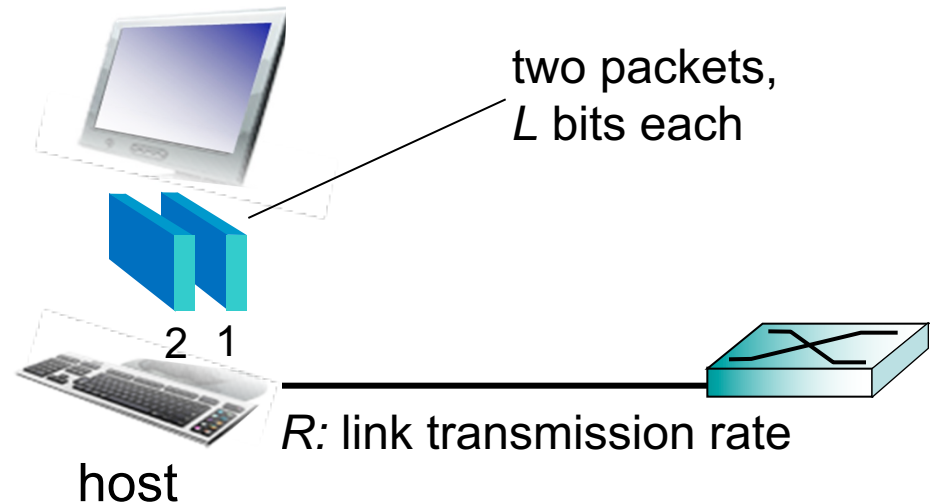
- interconnected routers
- network of networks



# Host: sends *packets* of data

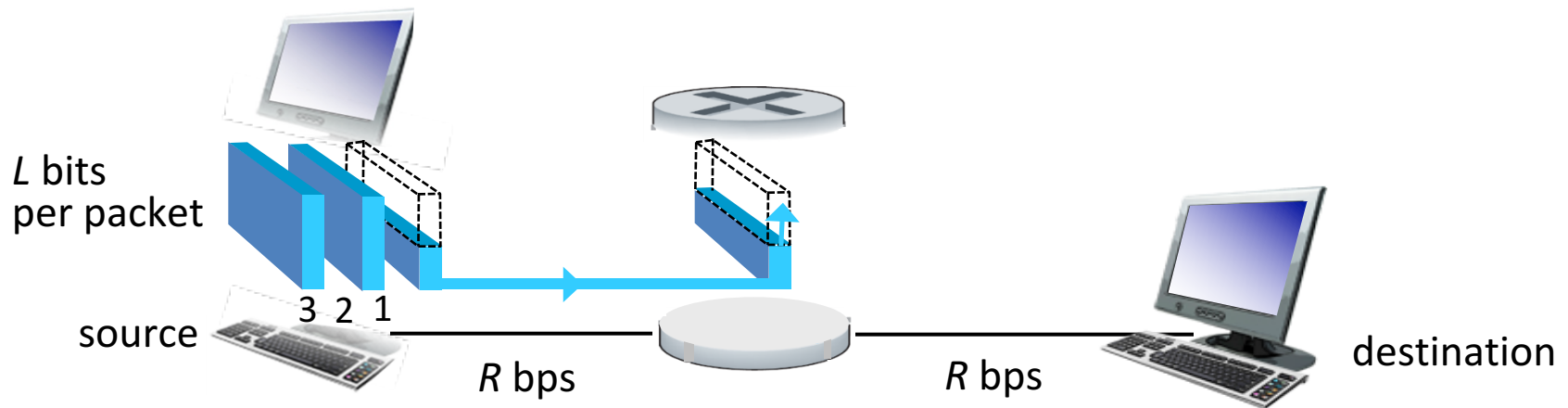
host sending function:

- ❖ takes application message
- ❖ breaks into smaller chunks, known as *packets*, of length  $L$  bits
- ❖ transmits packet into access network at *transmission rate  $R$* 
  - If this is max link transmission rate, then link *capacity*



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

# Packet-switching: store-and-forward



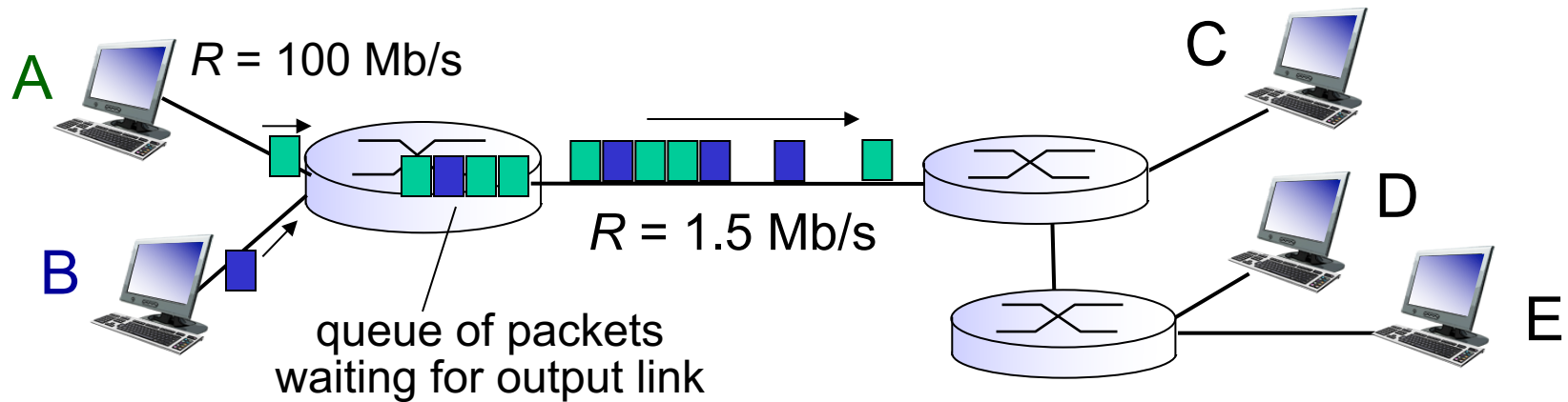
- ❖ takes  $L/R$  seconds to transmit (push out)  $L$ -bit packet into link at  $R$  bps
- ❖ *store and forward*: entire packet must arrive at router before it can be transmitted on next link
- ❖ end-end delay =  $2L/R$  (assuming zero propagation delay)

*one-hop numerical example:*

- $L = 7.5$  Mbits
- $R = 1.5$  Mbps
- one-hop transmission delay = 5 sec

} more on delay shortly ...

# Packet Switching: queueing delay, loss



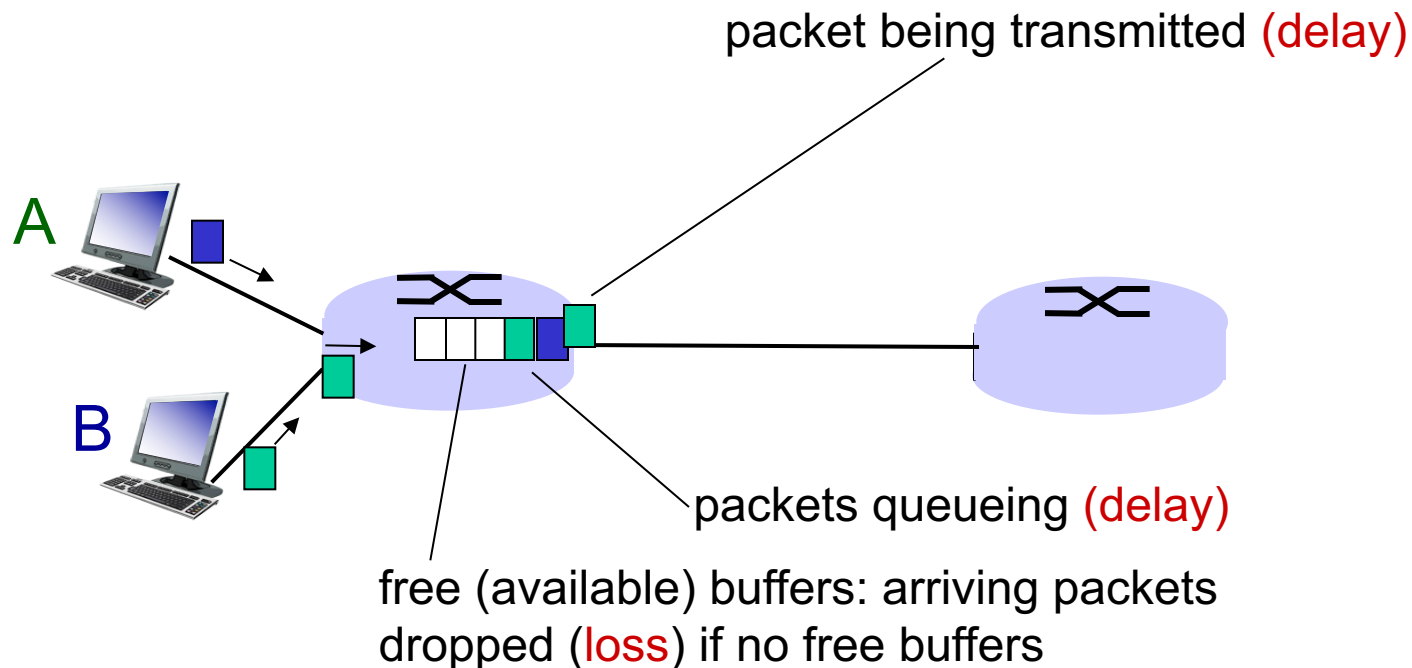
## queuing and loss:

- ❖ 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
  - packets can be dropped (lost) if memory (buffer) fills up

# How do loss and delay occur?

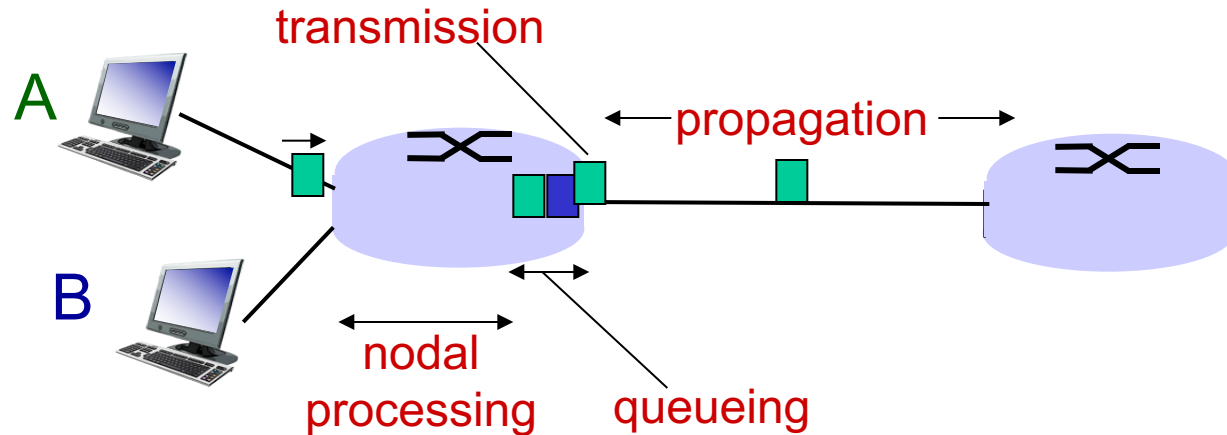
packets *queue* in router buffers

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





# Four sources of packet delay



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

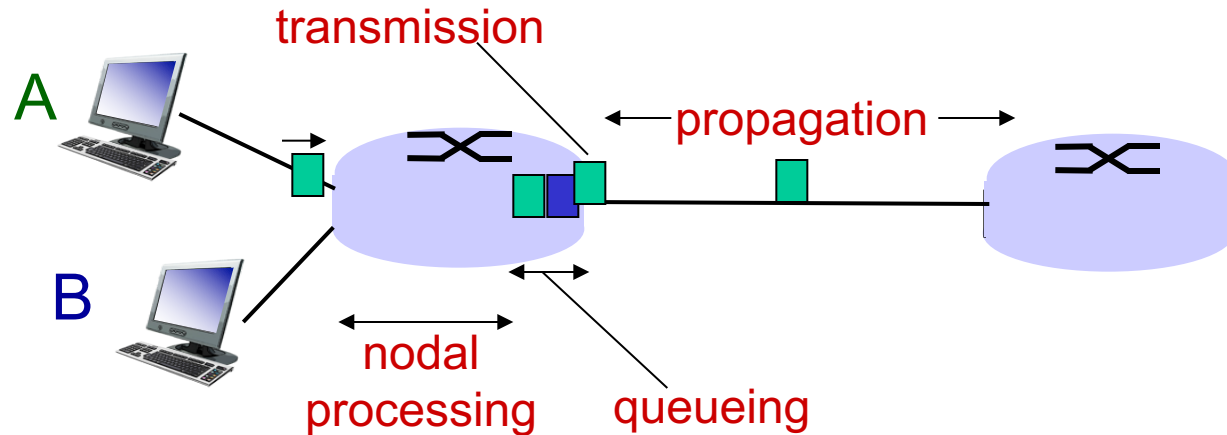
## $d_{\text{proc}}$ : nodal processing

- check bit errors
- determine output link
- typically < msec

## $d_{\text{queue}}$ : queueing delay

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

# Four sources of packet delay



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

$d_{\text{trans}}$ : transmission delay:

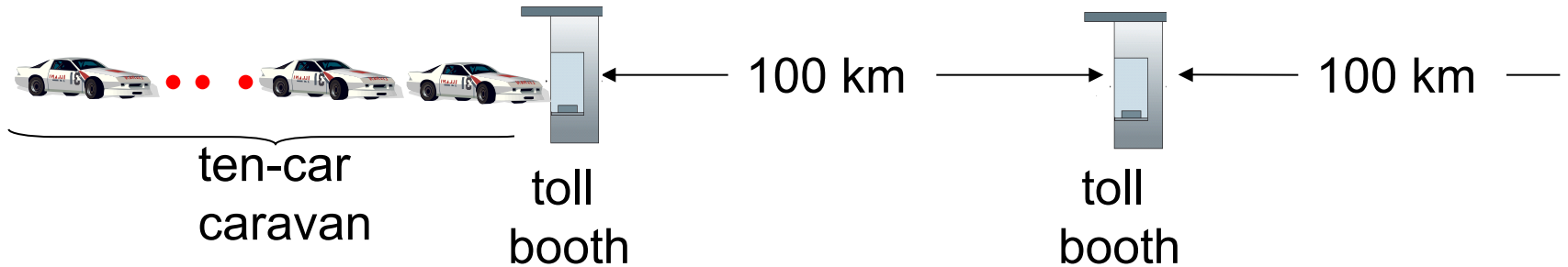
- $L$ : packet length (bits)
- $R$ : link bandwidth (bps)
- $d_{\text{trans}} = L/R$

$d_{\text{prop}}$ : propagation delay:

- $d$ : length of physical link
- $s$ : propagation speed in medium ( $\sim 2 \times 10^8$  m/sec)
- $d_{\text{prop}} = d/s$

$d_{\text{trans}}$  and  $d_{\text{prop}}$   
very different

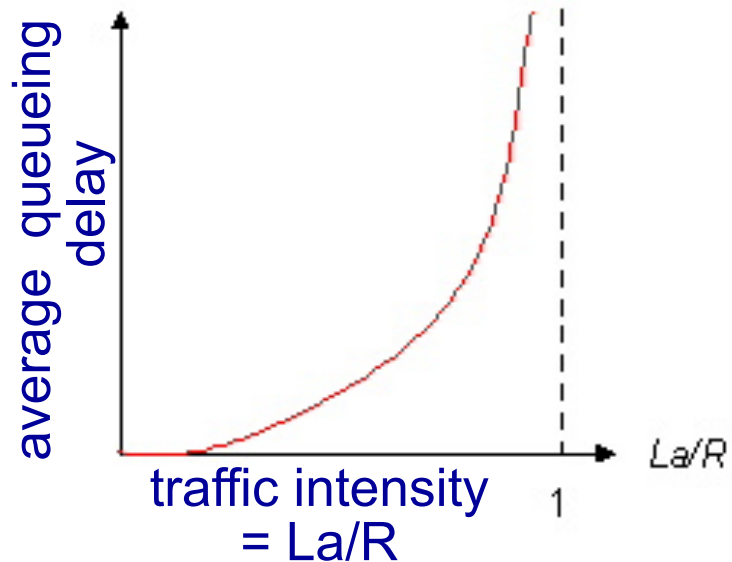
# Caravan analogy



- ❖ cars “propagate” at 100 km/hr
- ❖ toll booth takes 12 sec to service car (bit transmission time)
- ❖ car ~ bit; caravan ~ packet
- ❖ Q: How long until caravan is lined up before 2nd toll booth?
- time to “push” entire caravan through toll booth onto highway =  $12 \times 10 = 120$  sec
- time for last car to propagate from 1st to 2nd toll booth:  
 $100 \text{ km} / (100 \text{ km/hr}) = 1 \text{ hr}$
- A: 62 minutes

# Queueing delay

- ❖  $R$ : link bandwidth (bps)
- ❖  $L$ : packet length (bits)
- ❖  $a$ : average packet arrival rate: average occupancy of a server or resource during a specified period of **time**
- ❖  $La/R \sim 0$ : avg. queueing delay small
- ❖  $La/R \rightarrow 1$ : avg. queueing delay large
- ❖  $La/R > 1$ : more “work” arriving than can be serviced, average delay infinite!



$La/R \sim 0$

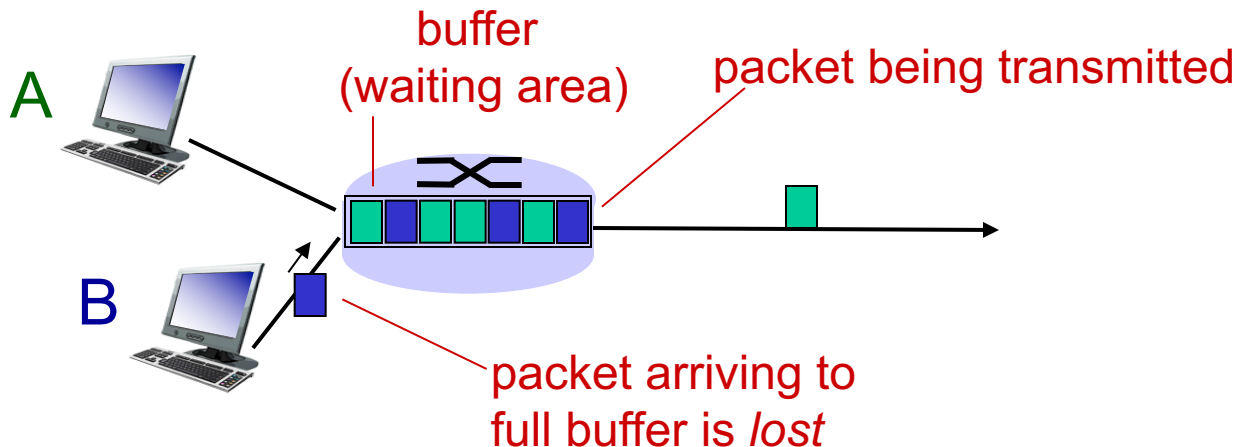


$La/R \rightarrow 1$

\* Check out the Java applet for an interactive animation on queueing and loss

# Packet loss

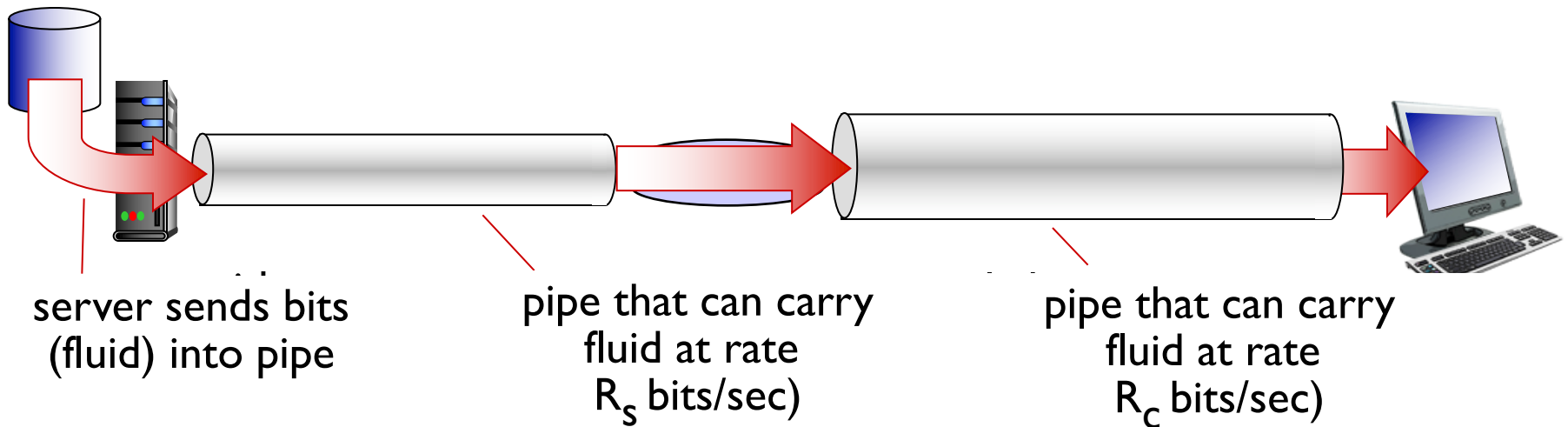
- ❖ queue (aka buffer) preceding link in buffer has finite capacity
- ❖ packet arriving to full queue dropped (aka lost)
- ❖ lost packet may be retransmitted by previous node, by source end system, or not at all



\* Check out the Java applet for an interactive animation on queuing and loss

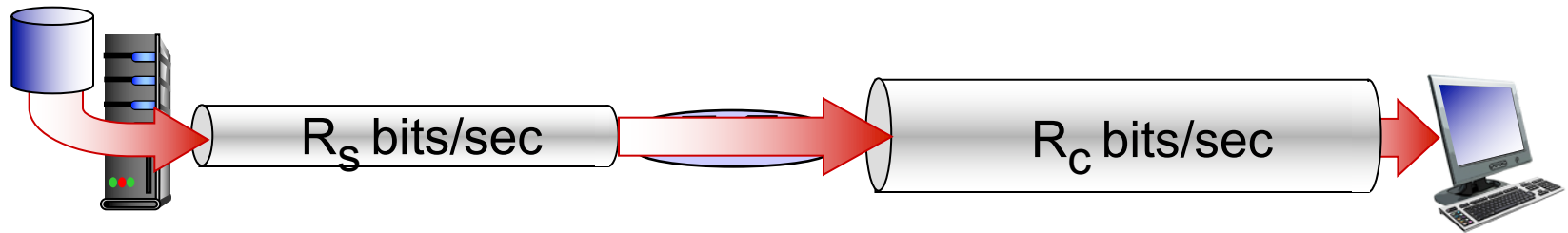
# Throughput

- ❖ *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

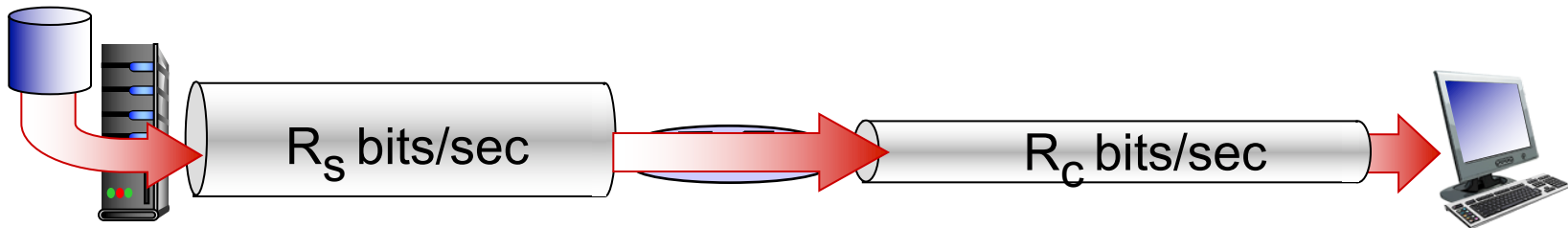


# Throughput (more)

❖  $R_s < R_c$  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