

# Introduction: 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

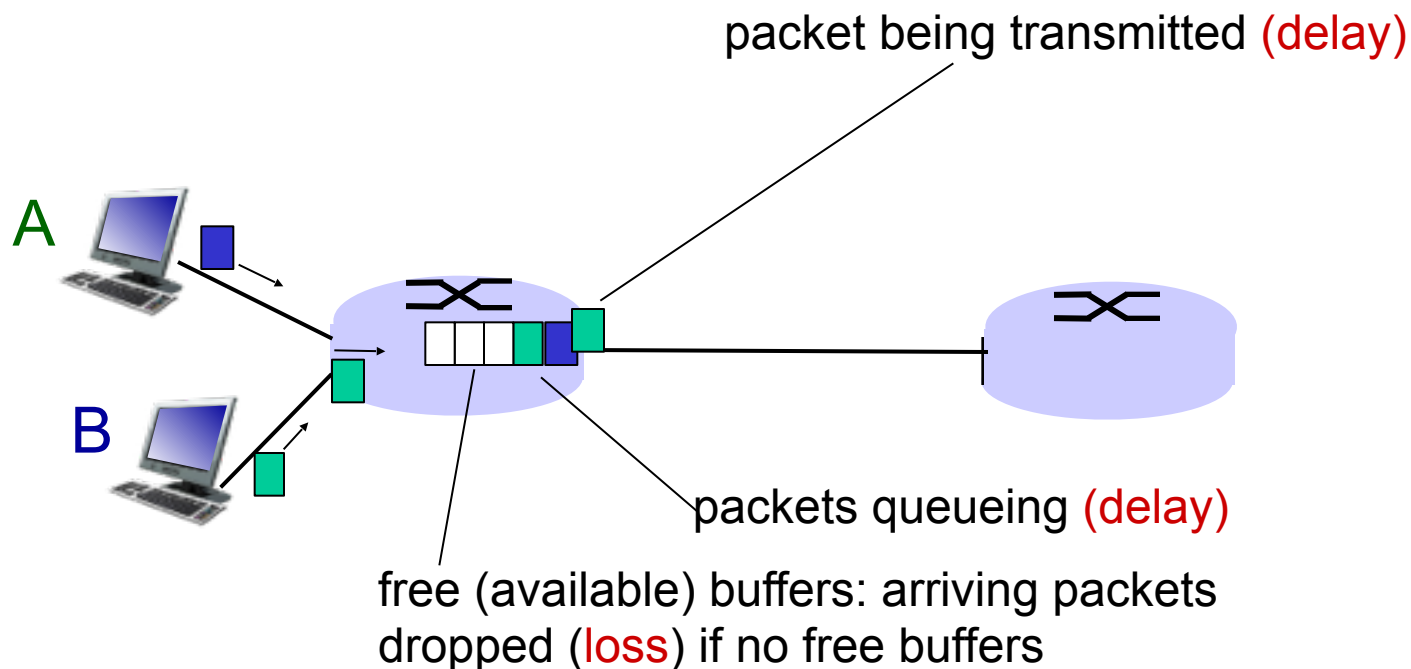
1.4 delay, loss, throughput in networks

1.5 protocol layers, service model

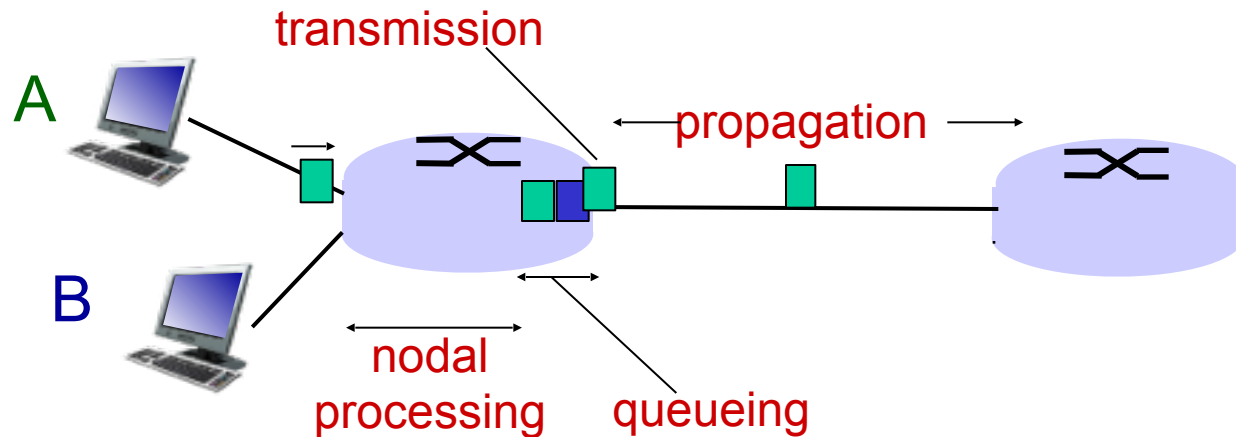
# How do loss and delay occur?

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



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

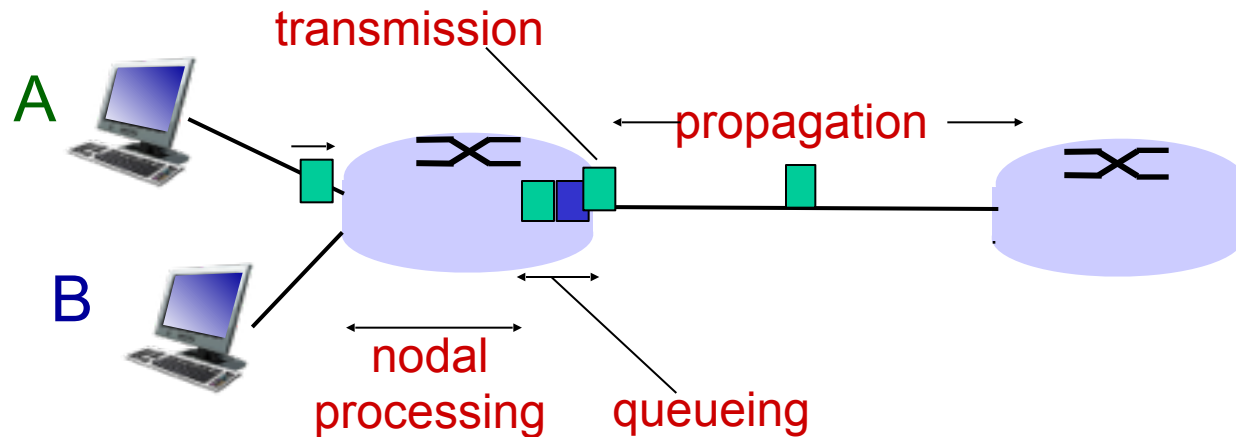
$d_{\text{proc}}$ : nodal processing

- 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:

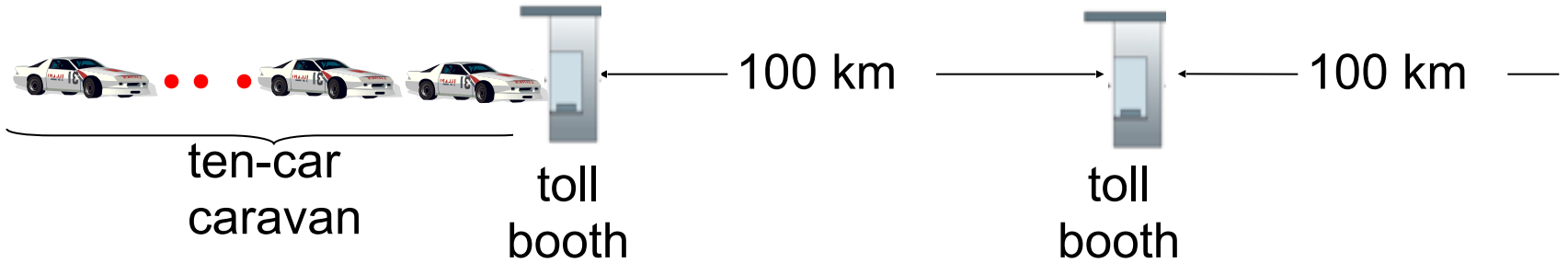
- $L$ : packet length (bits)
- $R$ : link bandwidth (bps)
- $d_{\text{trans}} = L/R$

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

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

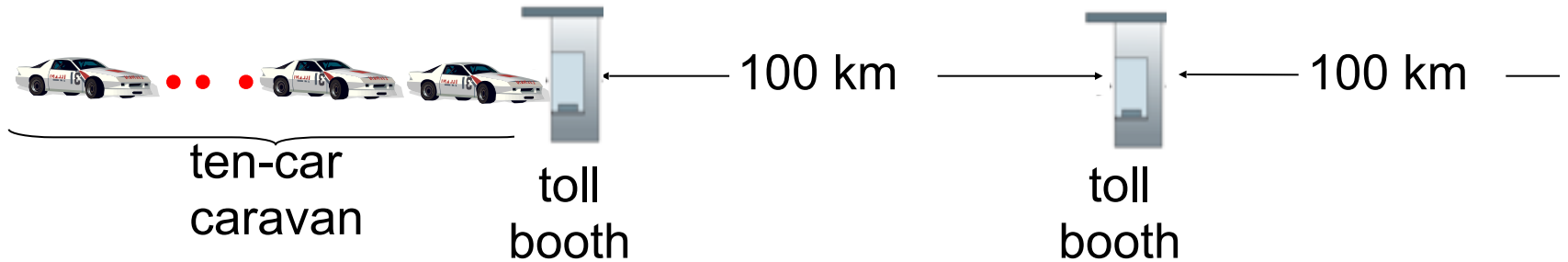
$d_{\text{trans}}$  and  $d_{\text{prop}}$   
very different

# 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:  $100\text{km} / (100\text{km/hr}) = 1$  hr
  - **A: 62 minutes**

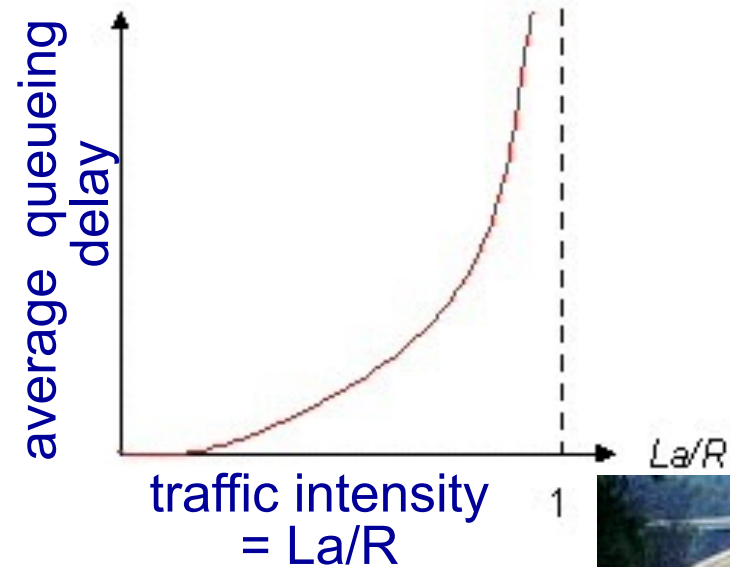
# Caravan analogy (more)



- ❖ suppose cars now “propagate” at 1000 km/hr
- ❖ and suppose toll booth now takes one min to service a car
- ❖ Q: Will cars arrive to 2<sup>nd</sup> booth before all cars serviced at first booth?
- ❖ Q: If yes, how many cars will be at 1<sup>st</sup> booth when the first car arrives at the 2<sup>nd</sup> booth
  - A: Yes! after 7 min, 1st car arrives at second booth; three cars still at 1st booth.

# Queueing delay (revisited)

- ❖  $R$ : link bandwidth (bps)
- ❖  $L$ : packet length (bits)
- ❖  $a$ : average packet arrival rate



- ❖  $La/R \sim 0$ : avg. queueing delay small
- ❖  $La/R \approx 1$ : avg. queueing delay large (**nature of traffic**)
- ❖  $La/R > 1$ : more “work” arriving than can be serviced, average delay infinite!

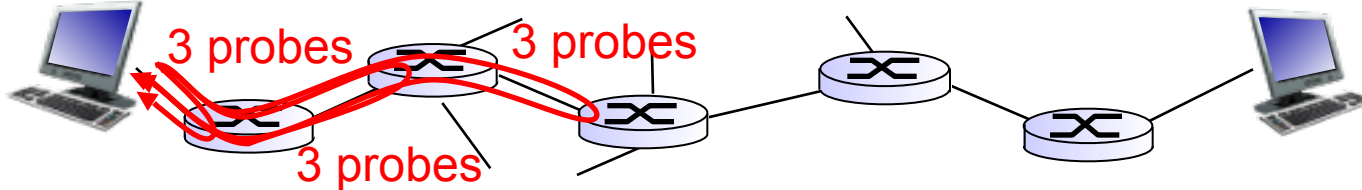
$La/R \sim 0$



$La/R \rightarrow 1$

# “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

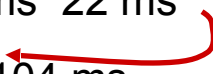
traceroute: gaia.cs.umass.edu to www.eurecom.fr

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



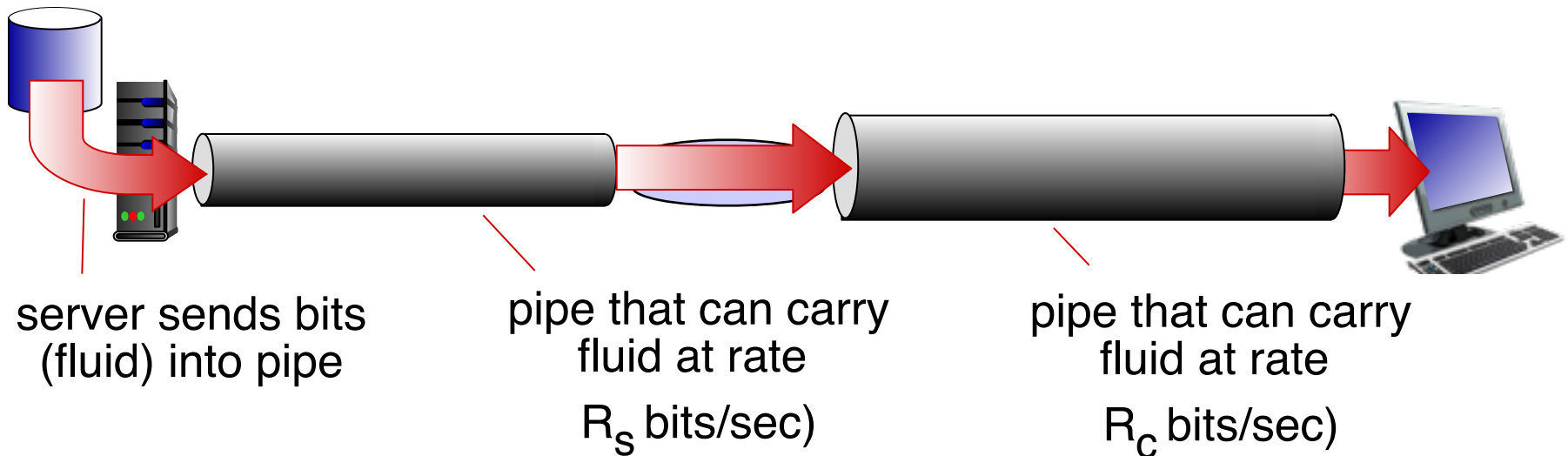
\* means no response (probe lost, router not replying)



\* Do some traceroutes from exotic countries at [www.traceroute.org](http://www.traceroute.org)

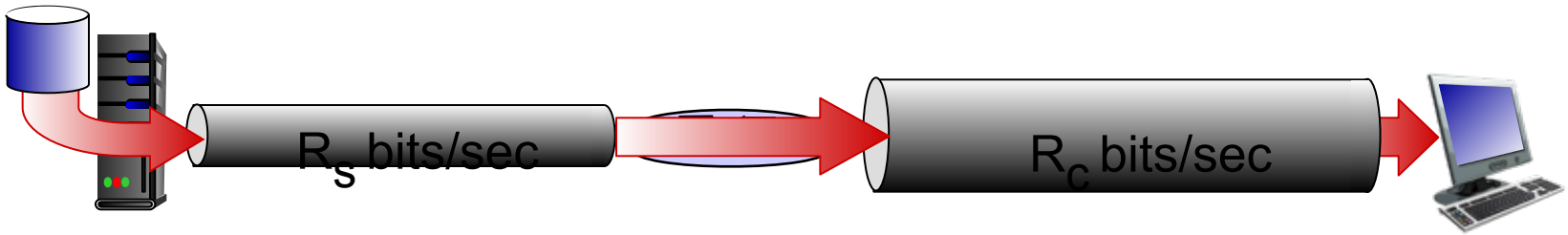
# Throughput

- ❖ *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

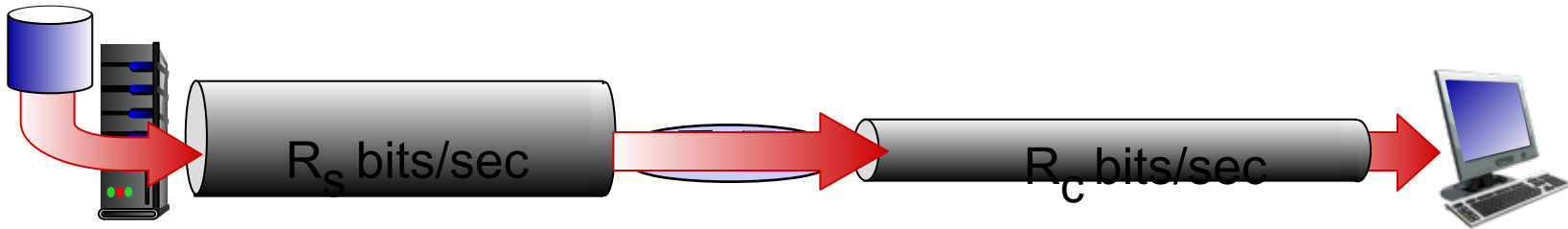


# Throughput (more)

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



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

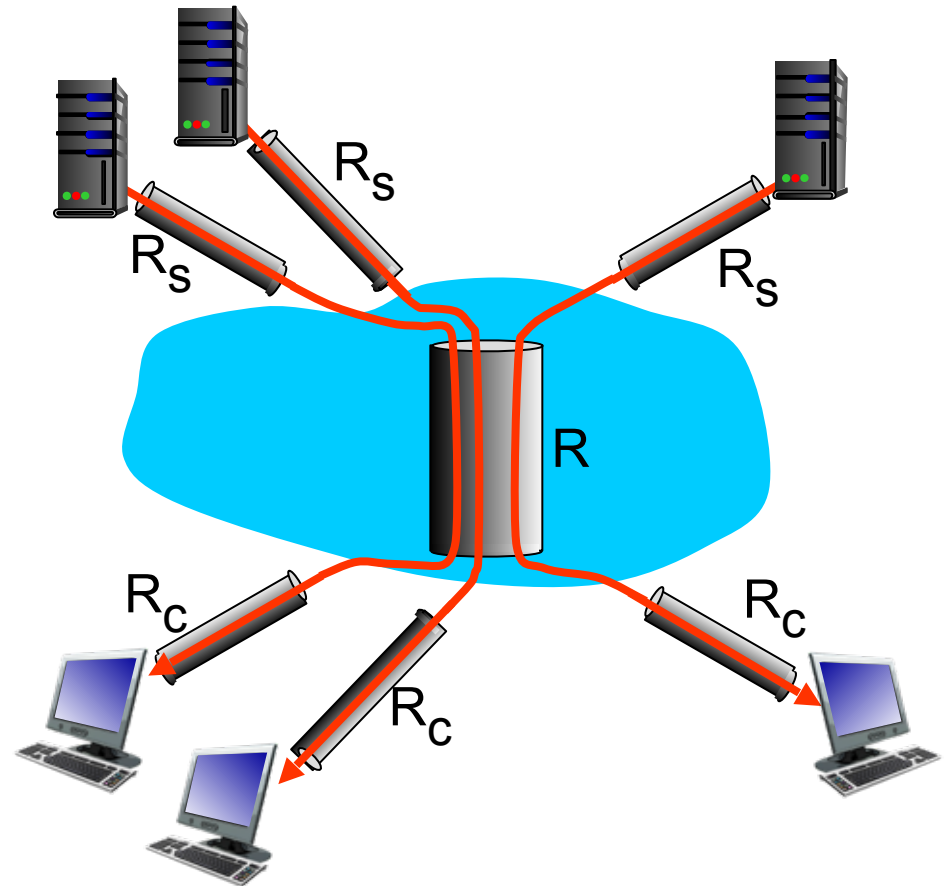


*bottleneck*

lin *link* path that constrains end-end  
throughput

# 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



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

# Introduction: 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”:*

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

*Question:*

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

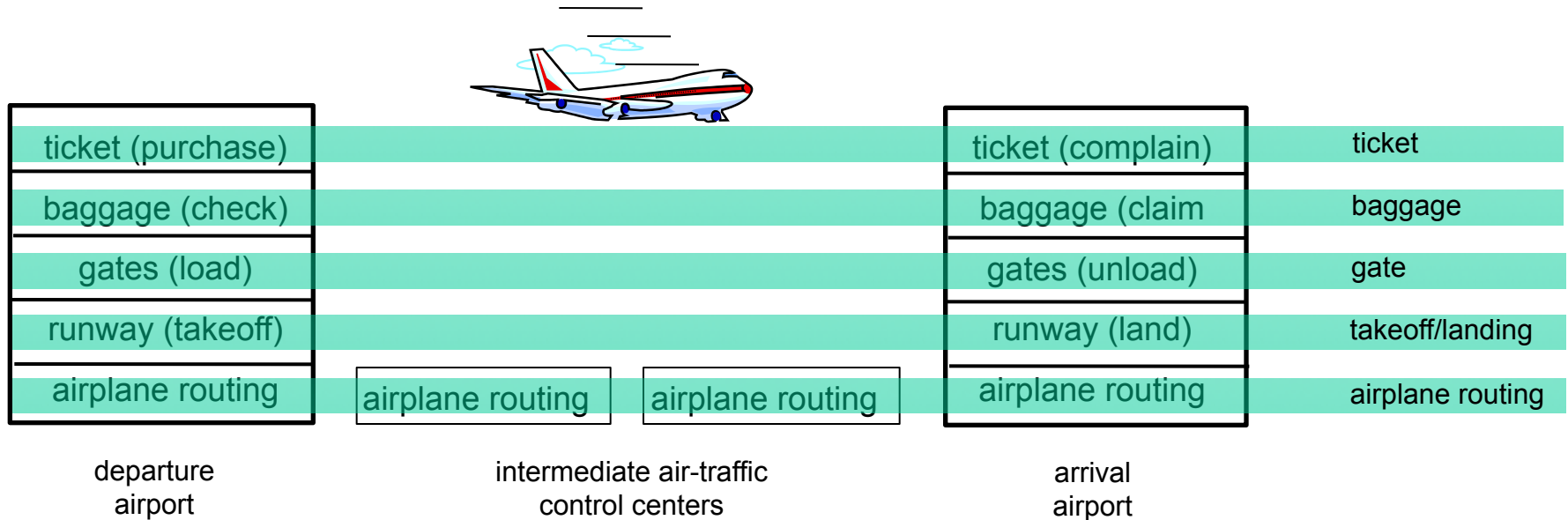
.... or at least our  
discussion of  
networks?

# 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



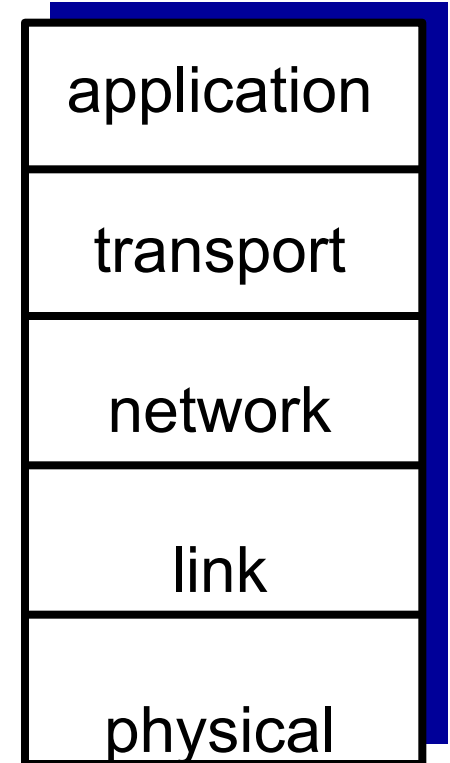
# Why layering?

dealing with complex systems:

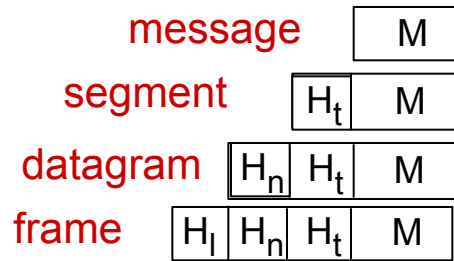
- ❖ explicit structure allows identification, relationship of complex system's pieces
- ❖ modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- ❖ layering considered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

# Internet protocol stack

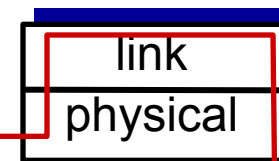
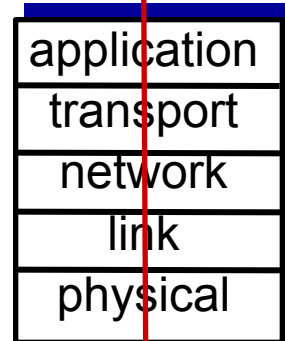
- ❖ *application*: supporting network applications
  - FTP, SMTP, HTTP
- ❖ *transport*: process-process data transfer
  - TCP, UDP
- ❖ *network*: routing of datagrams from source to destination
  - IP, routing protocols
- ❖ *link*: data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi)
- ❖ *physical*: bits “on the wire”



# Encapsulation

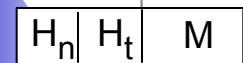
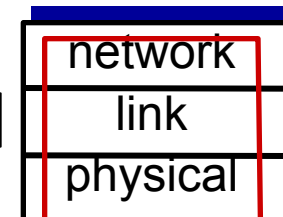
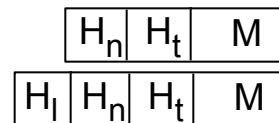
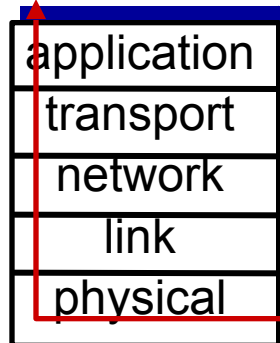
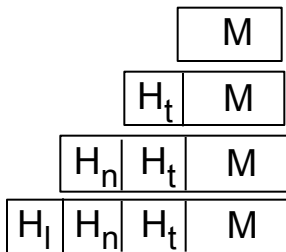


*source*



switch

*destination*



router

# Introduction: 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