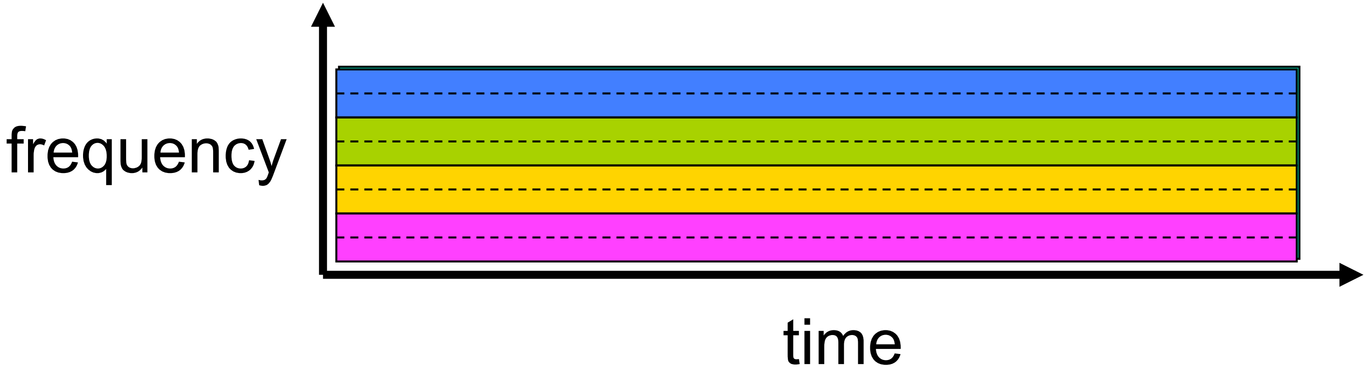
FDM and TDM

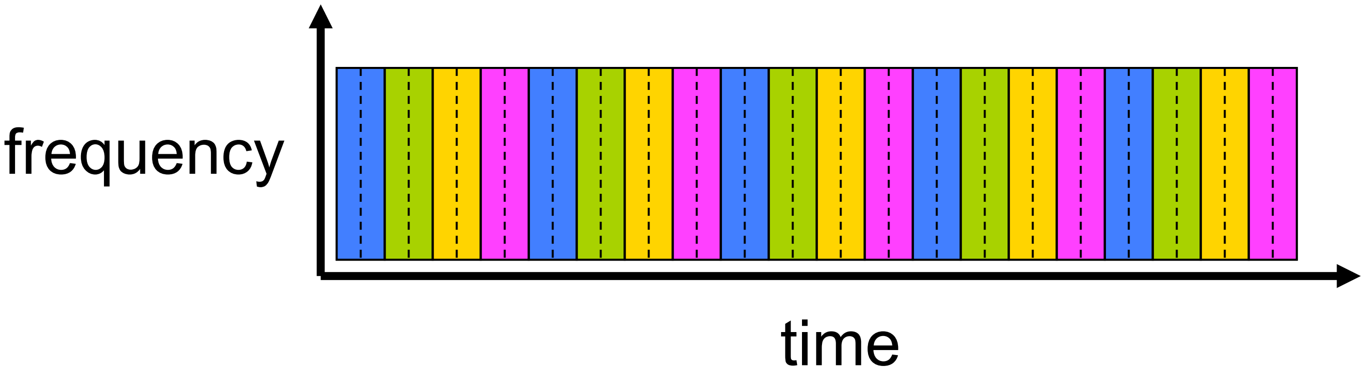
Example:

4 users

FDM



TDM



Network Core: Packet switching

each end-to-end data stream divided into *packets*

- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

Bandwidth division into “pieces”

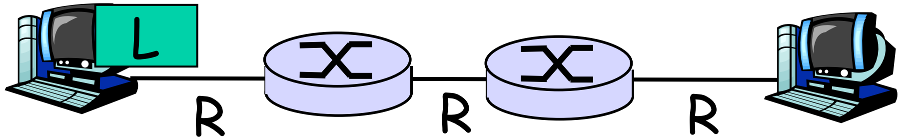
Dedicated allocation

Resource reservation

resource contention:

- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - switch receives complete packet before forwarding

Packet switching: store-and-forward



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

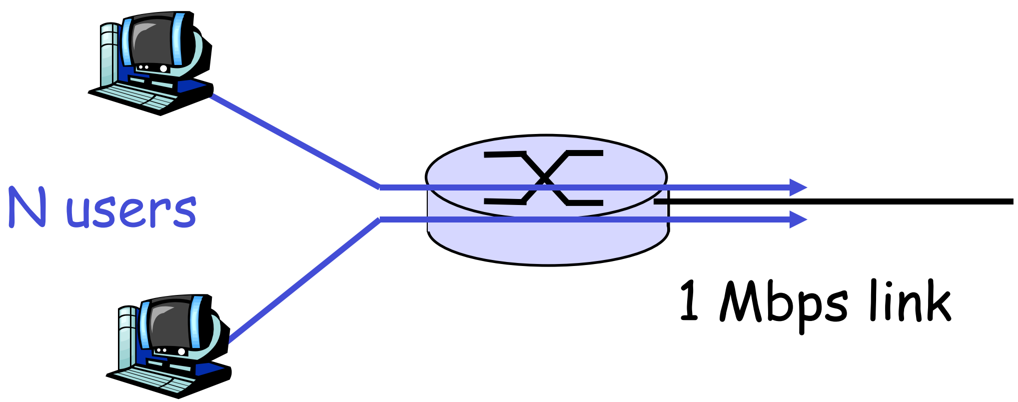
Example:

- L = 7.5 Mbits
- R = 1.5 Mbps
- transmission delay = 15 sec

Packet switching versus circuit switching (1)

Packet switching allows more users to use network!

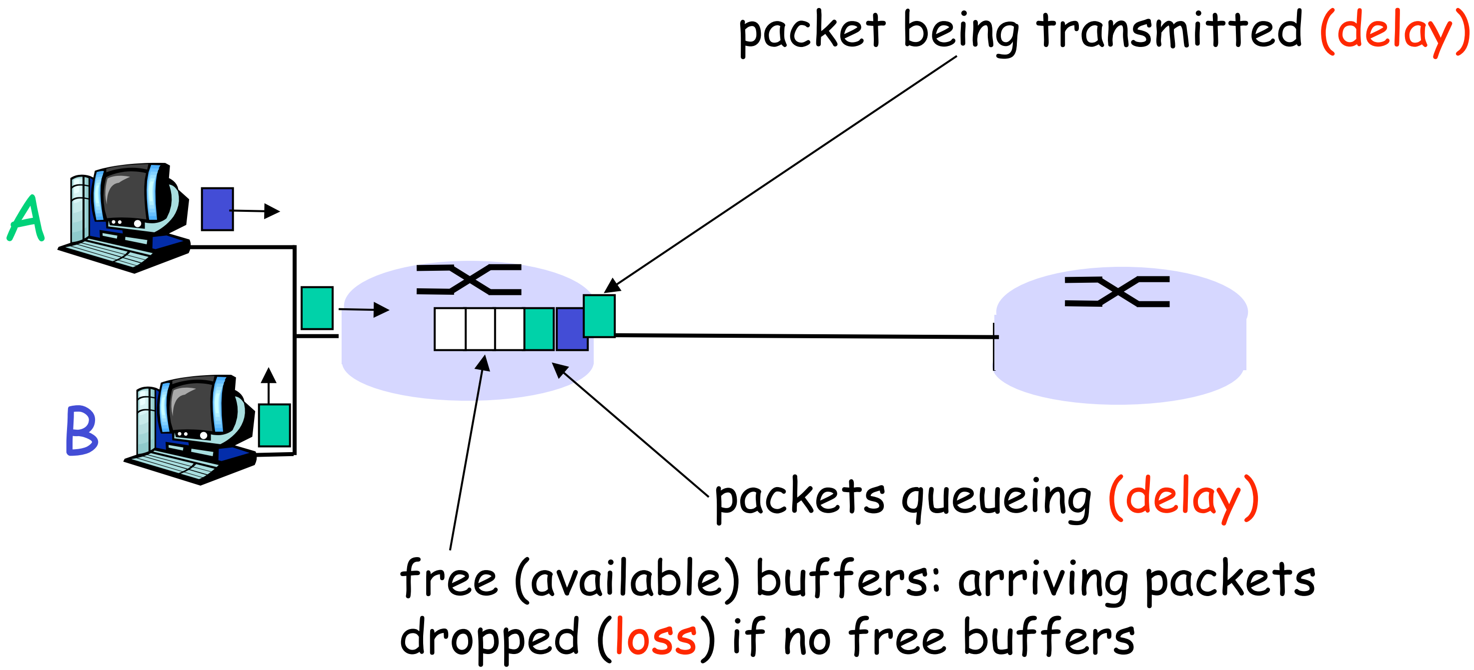
- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time



- *circuit-switching:*
  - 10 users simultaneously
- *packet switching:*
  - with 35 users, probability > 10 active at same time is less than .0004

How do loss and delay occur?

packets *queue* in router buffers
- packet arrival rate to link exceeds output link capacity
- packets queue, wait for their turn



Packet switching versus circuit switching (2)

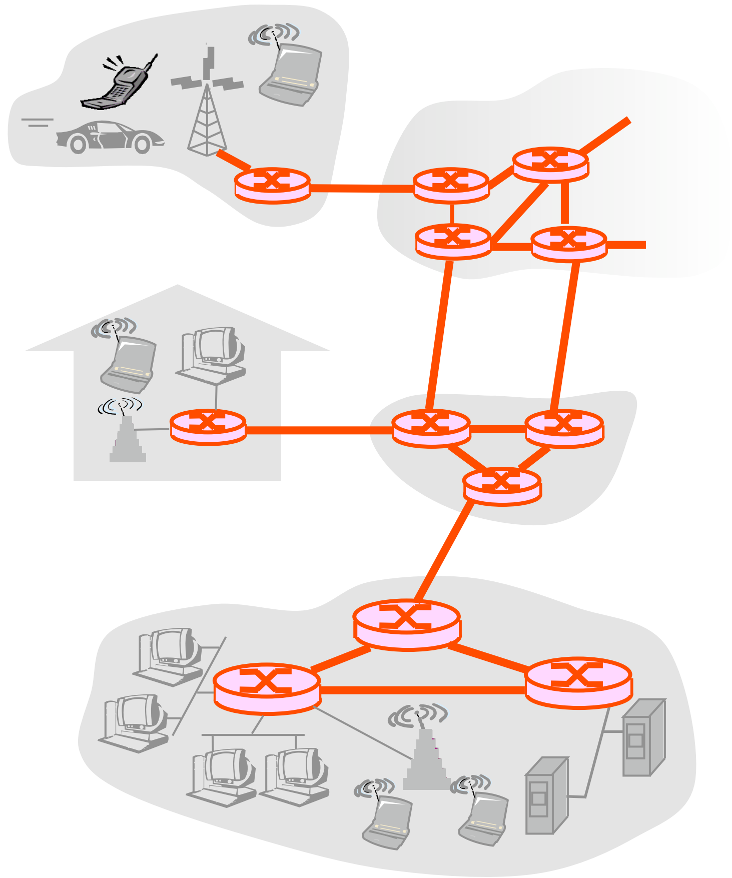
Is packet switching a “slam dunk winner?”

- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive 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

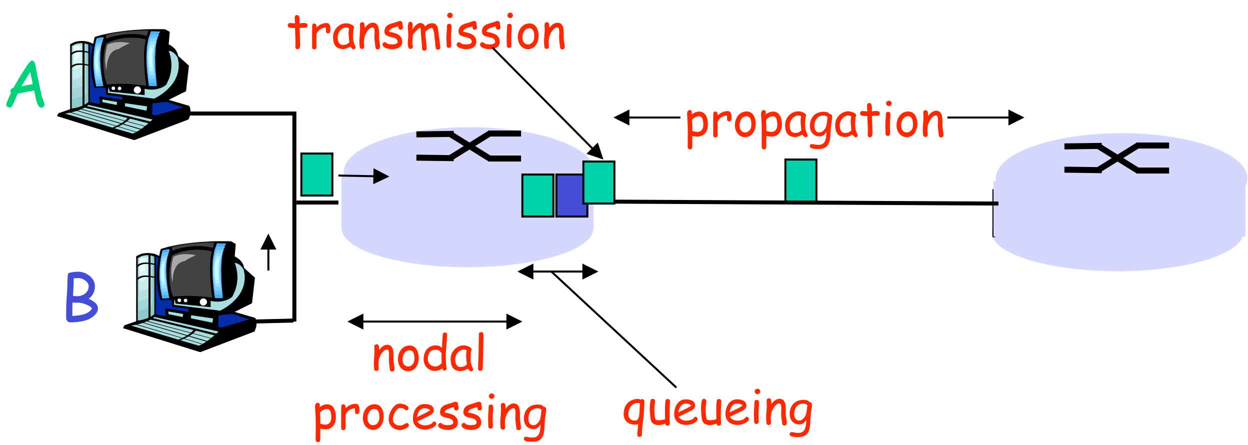
- mesh of interconnected routers

- *the* fundamental question: how is data transferred through the net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete "chunks"



Four sources of packet delay at each router (1)

- 1. processing:
  - check bit errors
  - determine output link
- 2. queueing
  - time waiting at output link for transmission
  - depends on congestion level of router



Four sources of packet delay at each router (2)

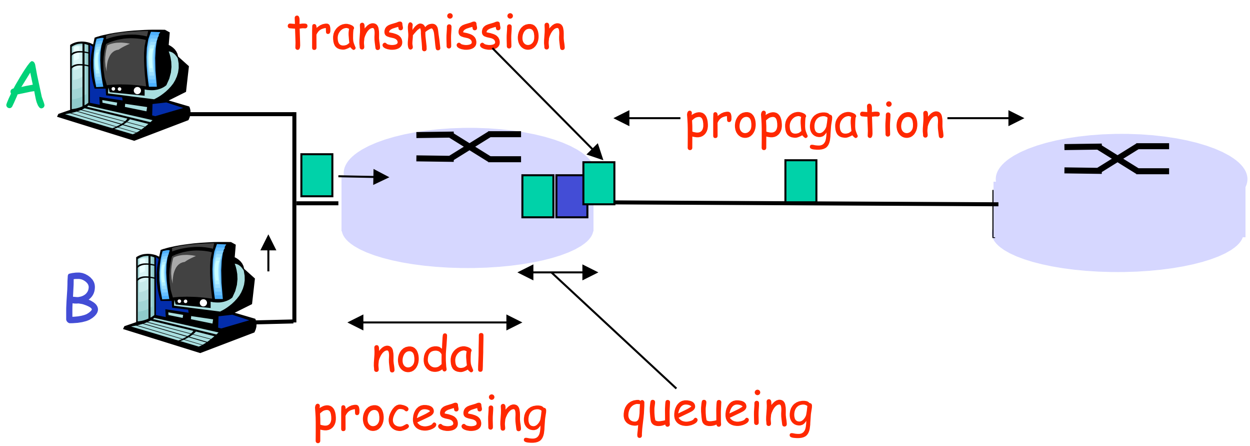
3. Transmission delay:

- R=link bandwidth (bps)
- L=packet length (bits)
- time to send bits into link = L/R

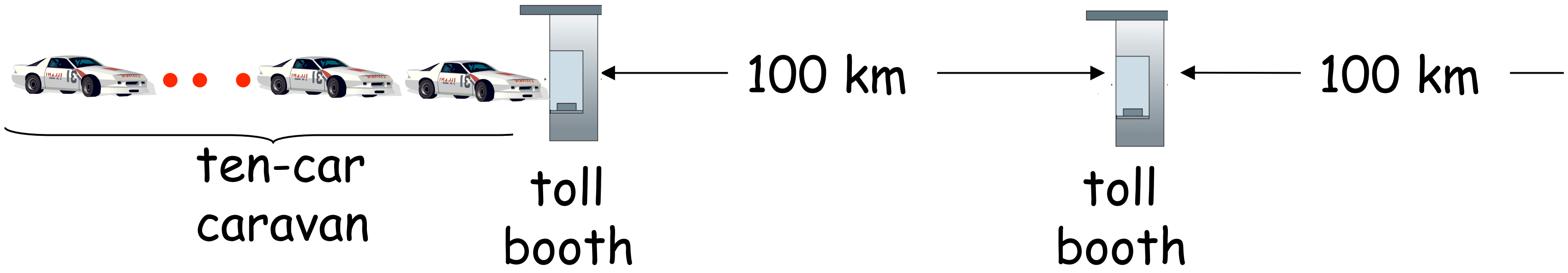
4. Propagation delay:

- d = length of physical link
- s = propagation speed in medium (~2bis3x10^8 m/sec)
- propagation delay = d/s

Note: d/s and L/R are *very* different quantities!



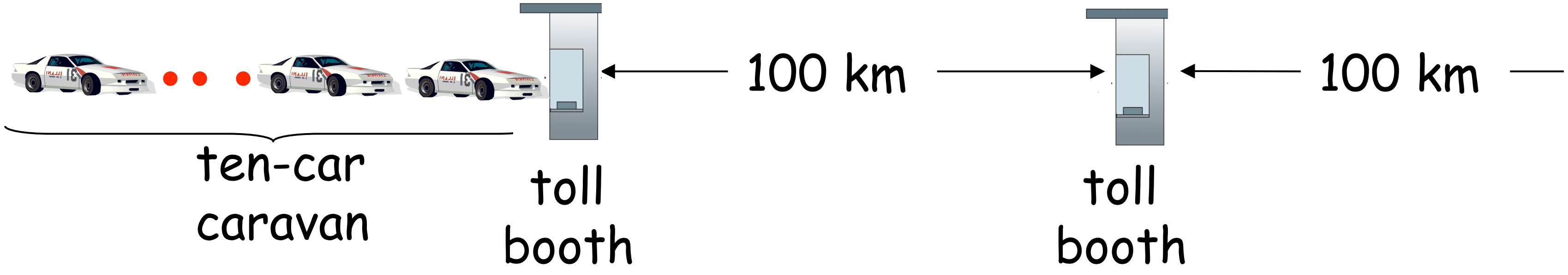
Caravan analogy (1)



- cars "propagate" at 100 km/hr (propagation rate)
- toll booth takes 12 sec to service car (transmission rate)
- 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*10 = 120 sec (transmission delay)
- Time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr (propagation delay)
- A: 62 minutes

Caravan analogy (2)



- Cars now “propagate” at 1000 km/hr
- Toll booth now takes 1 min to service a car
- Q: Will cars arrive to 2nd booth before all cars serviced at 1st booth?

- Yes! After 7 min, 1st car at 2nd booth and 3 cars still at 1st booth.

- 1st bit of packet can arrive at 2nd router before packet is fully transmitted at 1st router!

Total router (nodal) delay

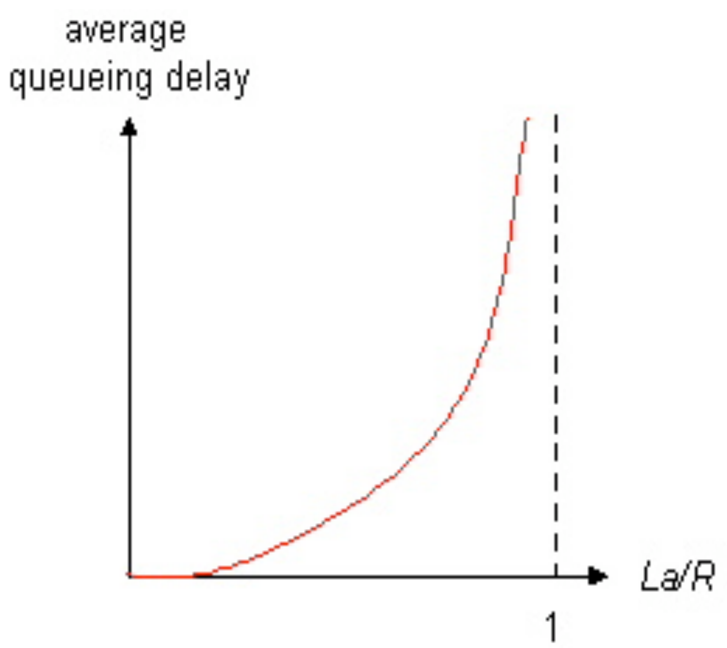
$$\boxed{d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}}$$

- d_{proc} = processing delay
  - typically a few microsecs or less
- d_{queue} = queuing delay
  - depends on congestion
- d_{trans} = transmission delay
  - = L/R, significant for low-bandwidth links
- d_{prop} = propagation delay
  - a few microsecs to hundreds of msecs

Queueing delay (revisited)

- R=link bandwidth (bps)
- L=packet length (bits)
- a=average packet arrival rate (packets/sec)
- La=average bit arrival rate (bps)

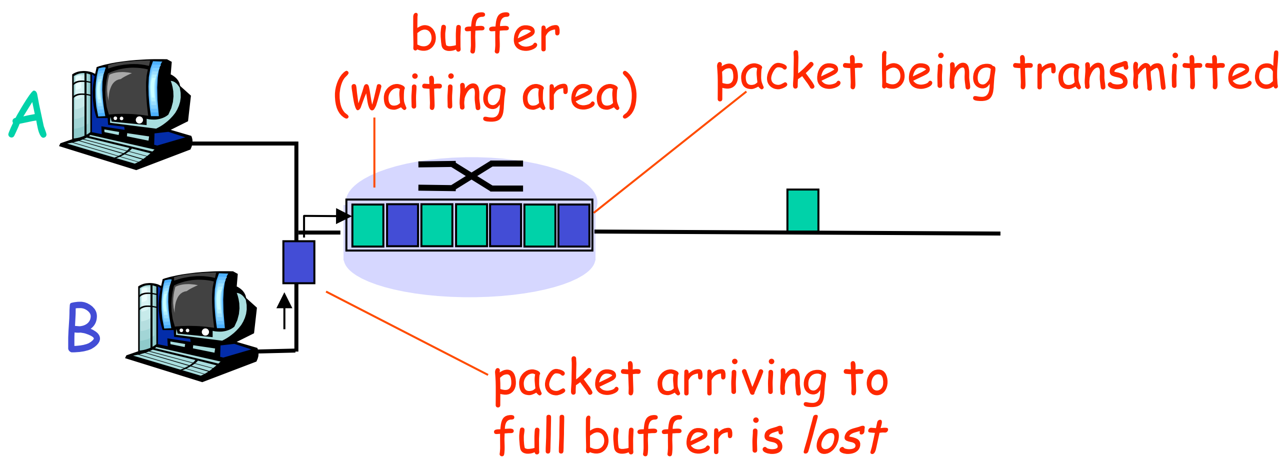
traffic intensity = La/R



- La/R > 1: more "work" arriving than can be serviced, average delay may grow infinitely!
- La/R ~ 0: average queueing delay small (close to zero)
- La/R ~ 1: delays become large and larger

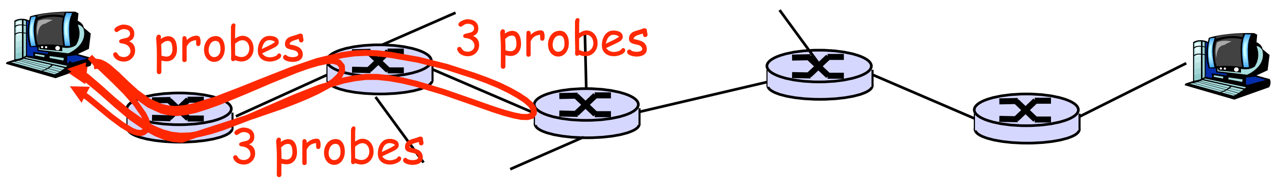
Packet loss

- queue (buffer) preceding link in 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



"Real" Internet delays and routes

- What do "real" Internet delay & loss look like?
- **`Traceroute`** program: provides delay measurement from source to router along end-to-end Internet path towards destination. For all *i:*
  - source sends three copies of packet i that will reach router *i* on path towards destination
  - router *i* will return the three copies of packet i to sender
  - sender times interval between transmission and reply (round-trip delay).



“Real” Internet delays and routes

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)

Layered protocol (service) architecture

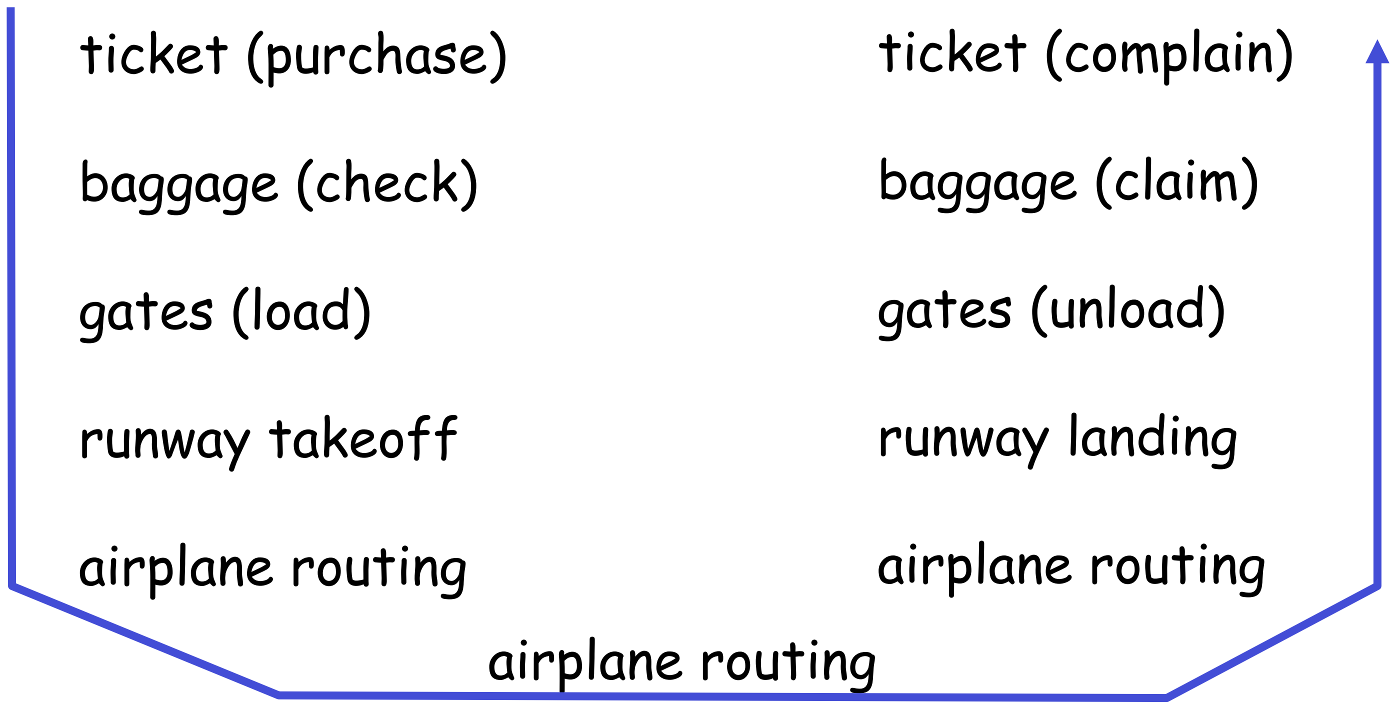
The Internet is complex!

- many “pieces”:
  - hosts
  - access network
  - routers
  - links of various media
  - applications
  - protocols

Question:

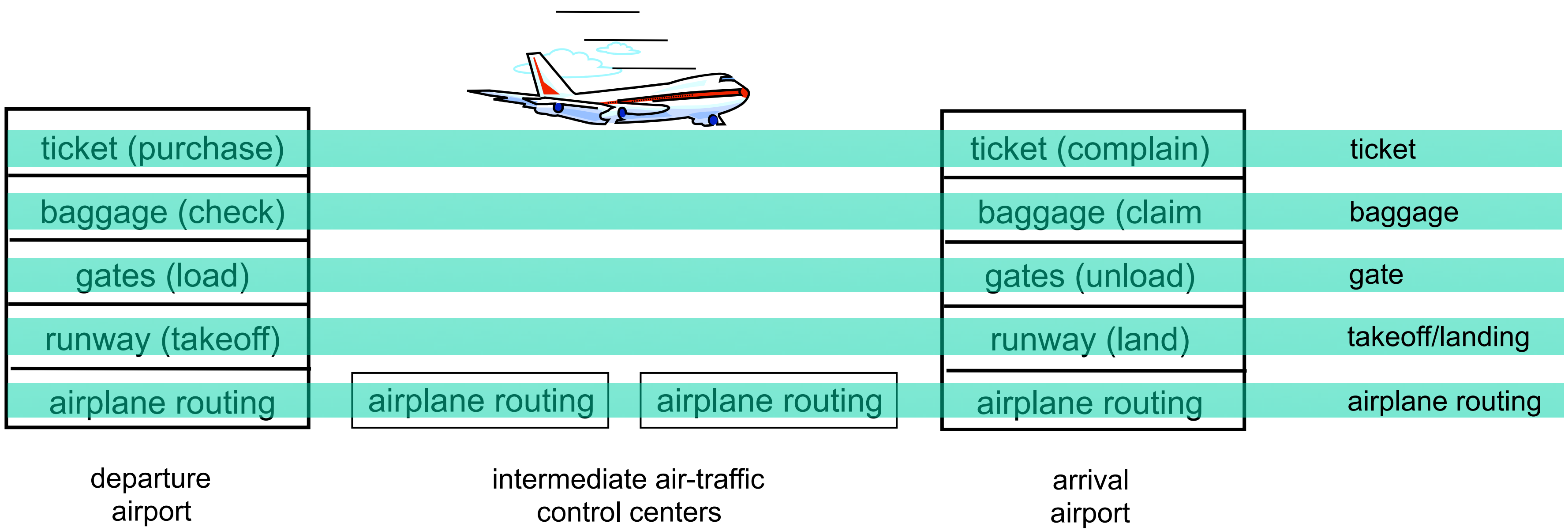
Is there any hope of *organizing* a structure of the Internet a so-called ***network (service) architecture***?

Analogy 1: Organization of air travel



- Structured into a series of steps on both ends

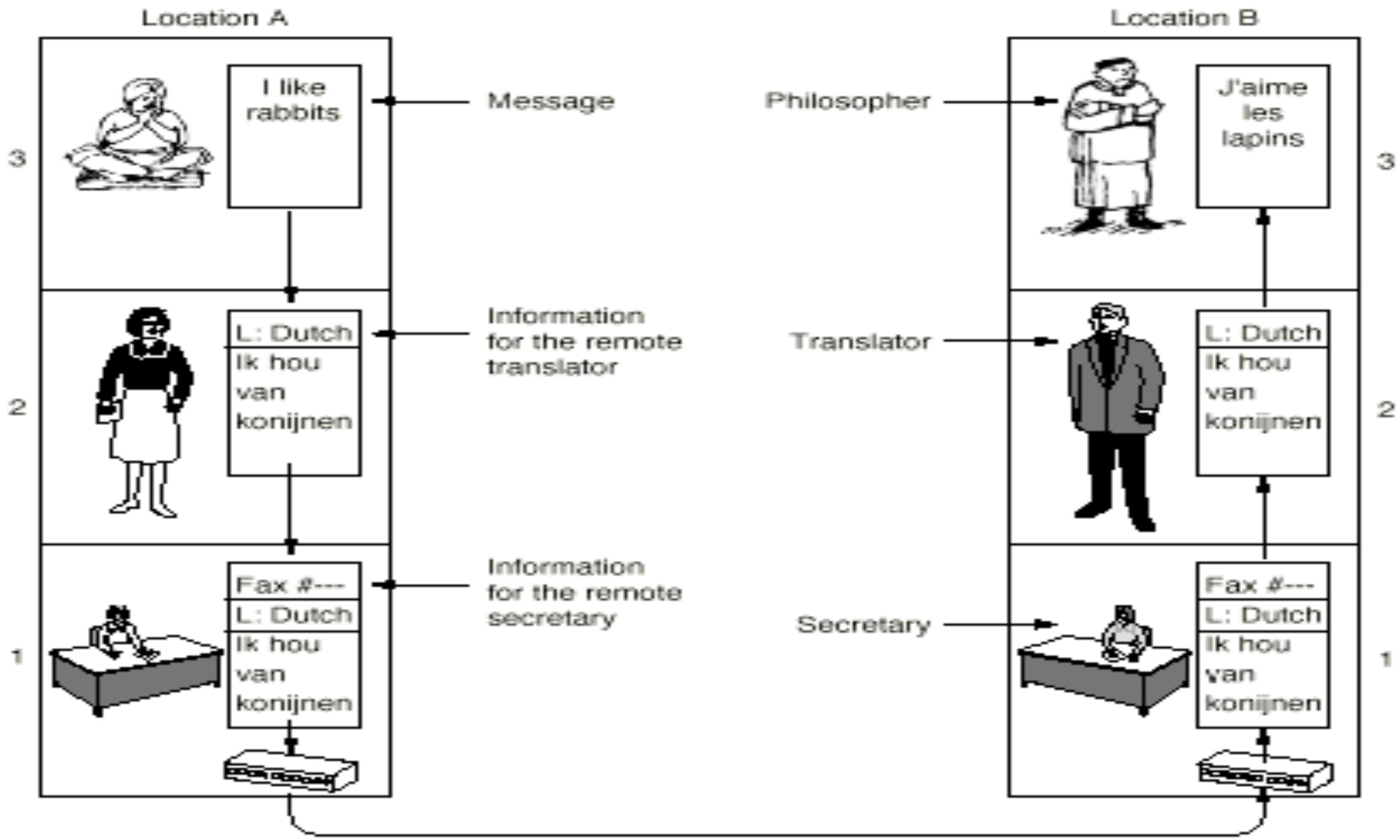
Layering of airline functionality



Layers: each layer implements a service

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

Analogy 2: The philosopher-translator-secretary architecture



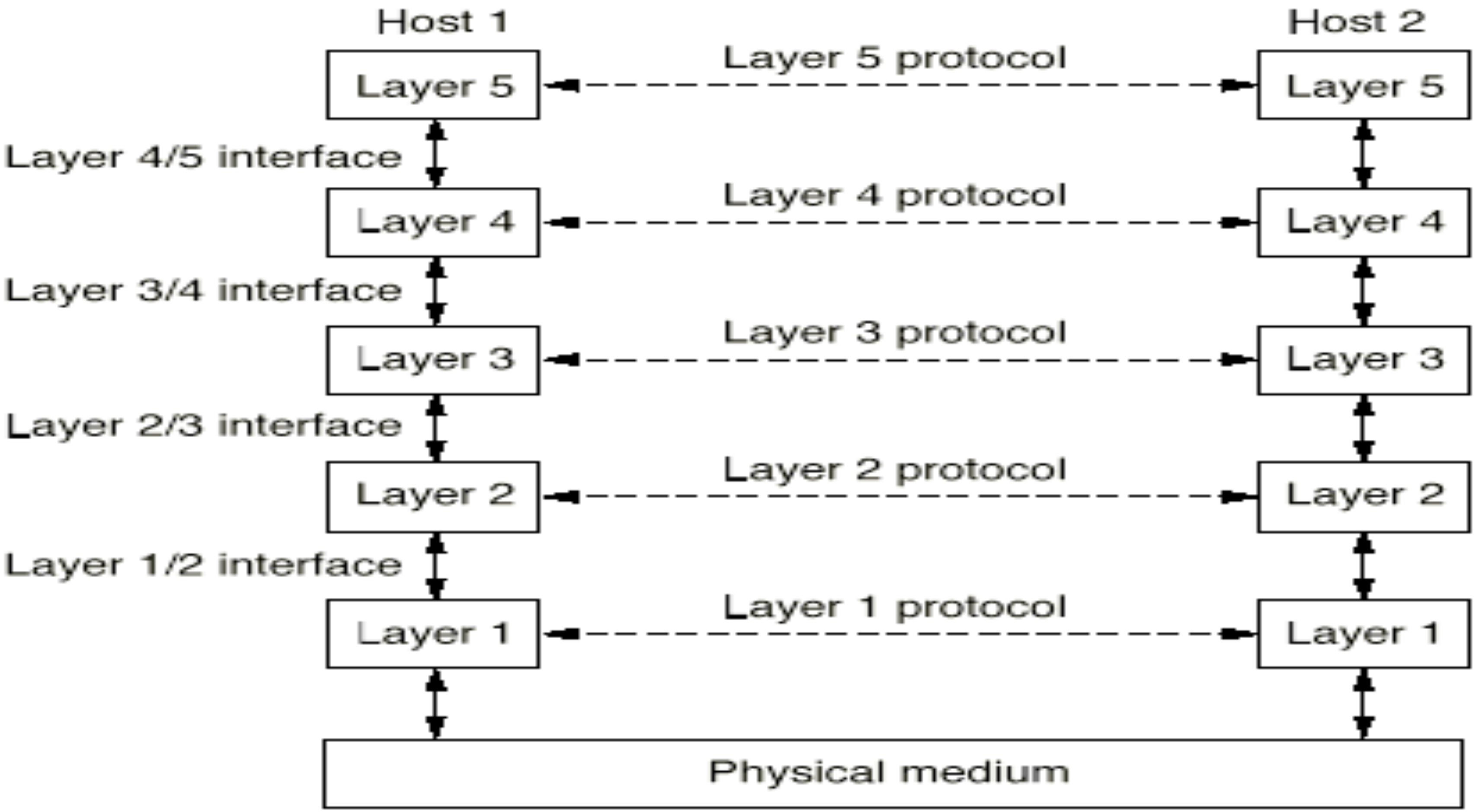
Why layering?

Dealing with complex systems:

- explicit structure allows identification and relationship of complex system's pieces
  - layered reference model

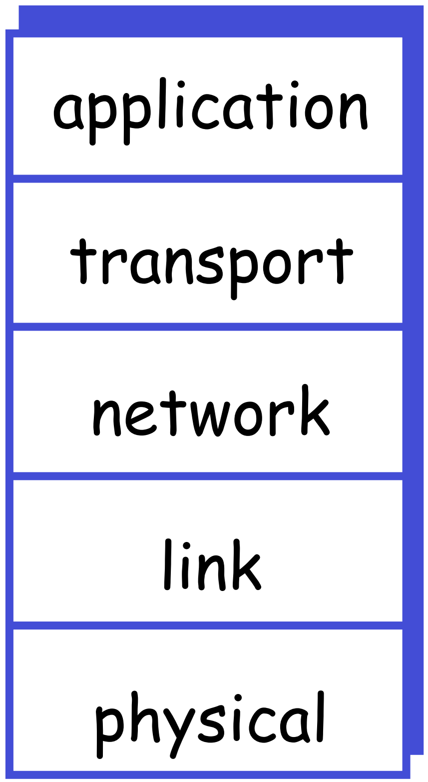
- modularization eases maintenance and updating of system
  - change of implementation of layer's service transparent to rest of system

Network Architecture: A set of layers and protocols



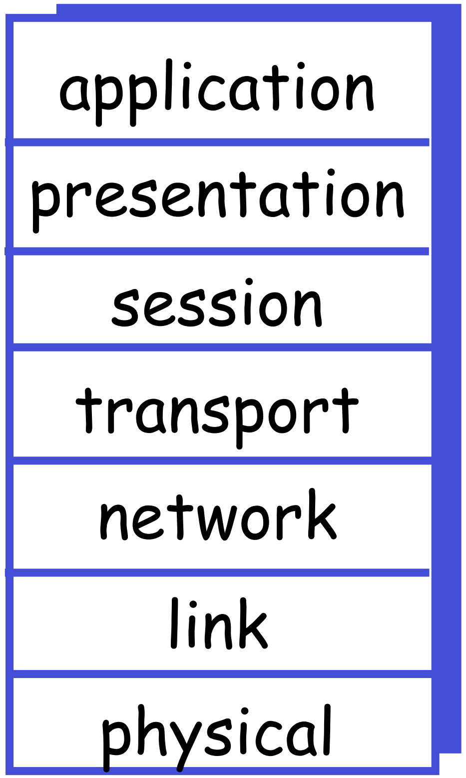
Internet (TCP/IP) protocol stack

- application: supporting network applications
  - FTP, SMTP, HTTP, DNS, Security
- 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
- 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?



Physical path data takes and the concept of Encapsulation