Last Lecture

Nuts-and-bolts description of the Internet

- The topology
  - The core
  - The edge
- The communication links

This Lecture

- How to send data from end to end: two switching methods
  - Circuit switching
  - Packet switching
- Packet loss and delay in a packet switched network

How is data transferred through a network?

Two *switching* methods:

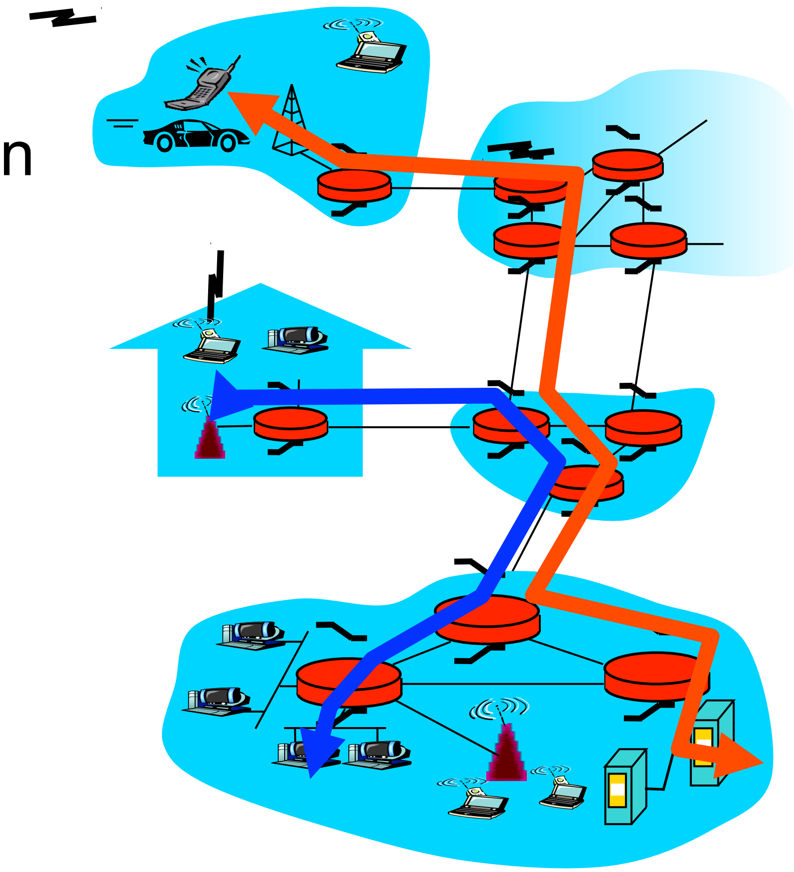
1. Circuit Switching: dedicated physical circuit is established, maintained, and terminated over a communication session (e.g. ISDN)

2. Packet Switching: data are transferred in packets (chunks of data of a fixed size), possibly go through different paths to reach the destination
(e.g. ATM, X.25, Frame Relay, Internet)

1. Circuit Switching

Three step process

- Source establishes connection to destination
  - Find path
  - Reserve resources
- Data exchanged (no need for destination address)
- Connection torn down
  - Resources released



Sharing a Link: Multiplexing

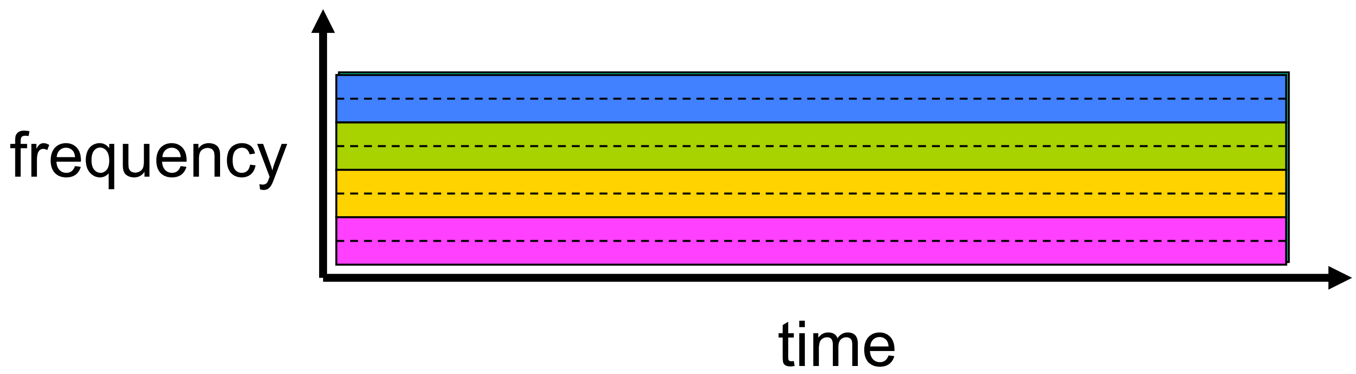
- To combine multiple signals (analog or digital) for transmission over a single line or medium.
- Multiplexing technologies:
  - Frequency Division Multiplexing **(FDM) :** each signal is assigned a different frequency range (e.g. FM radio).
  - Time Division Multiplexing **(TDM) :** each signal is assigned a fixed time slot in a "fixed" rotation.
  - Statistical Time Division Multiplexing **(STDM):** time slots are assigned to signals dynamically to make better use of bandwidth.
  - Wavelength Division Multiplexing **(WDM) :** each signal is assigned a particular wavelength; used in optical fiber.

Circuit Switching: FDMA and TDMA

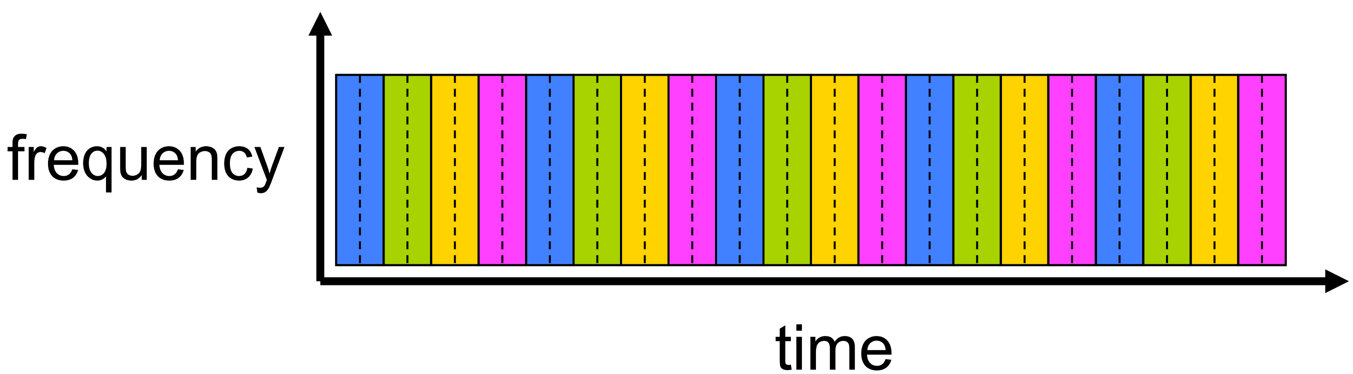
Example:

4 users

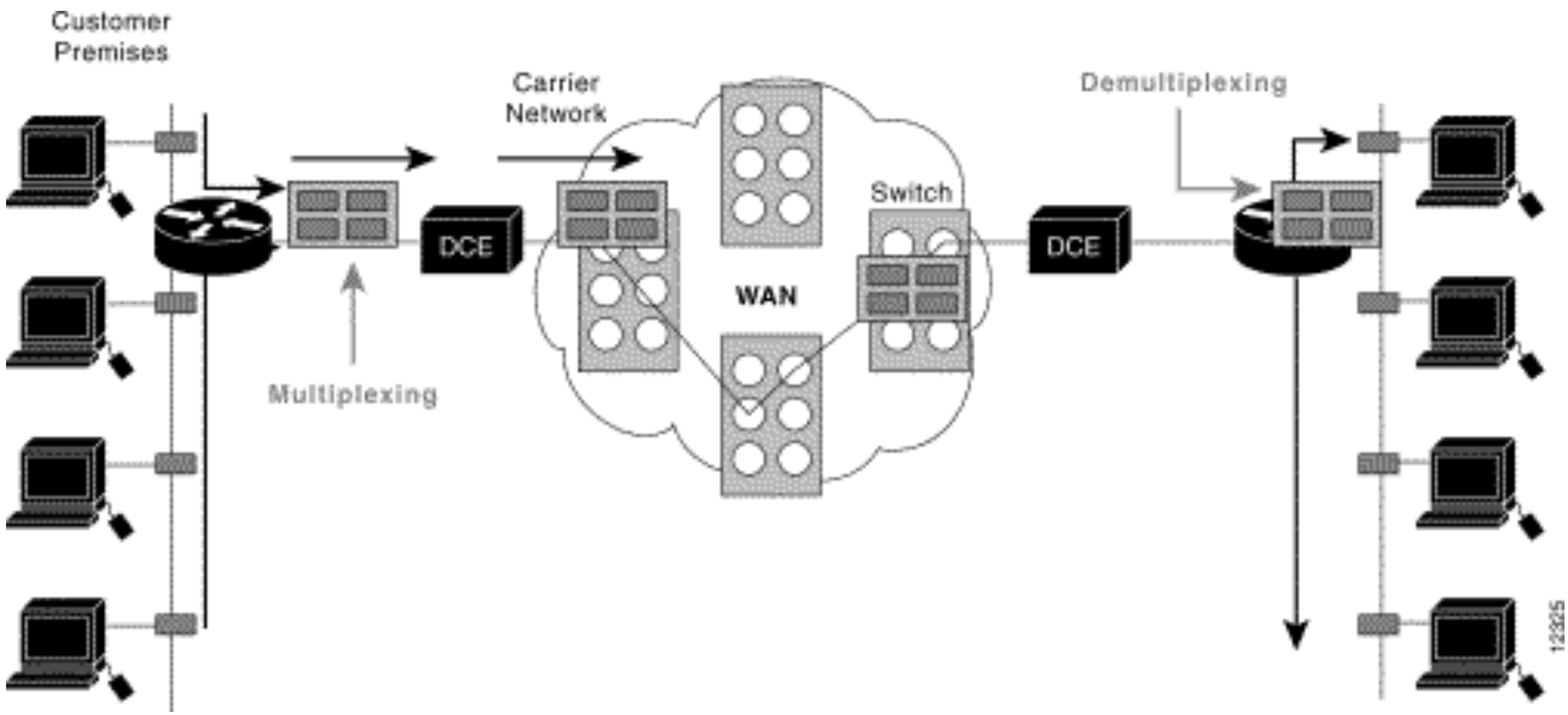
FDMA



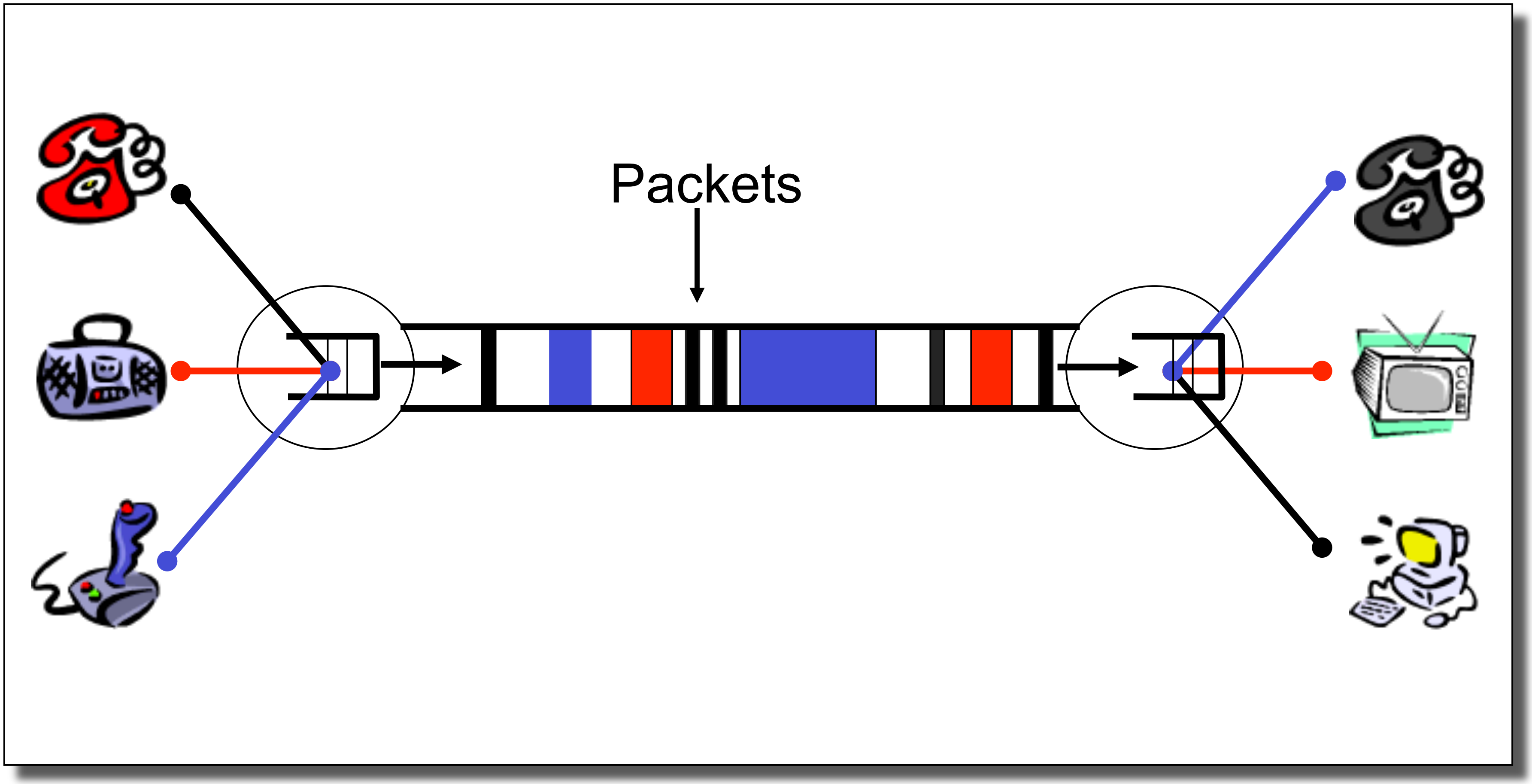
TDMA



2. Packet Switching



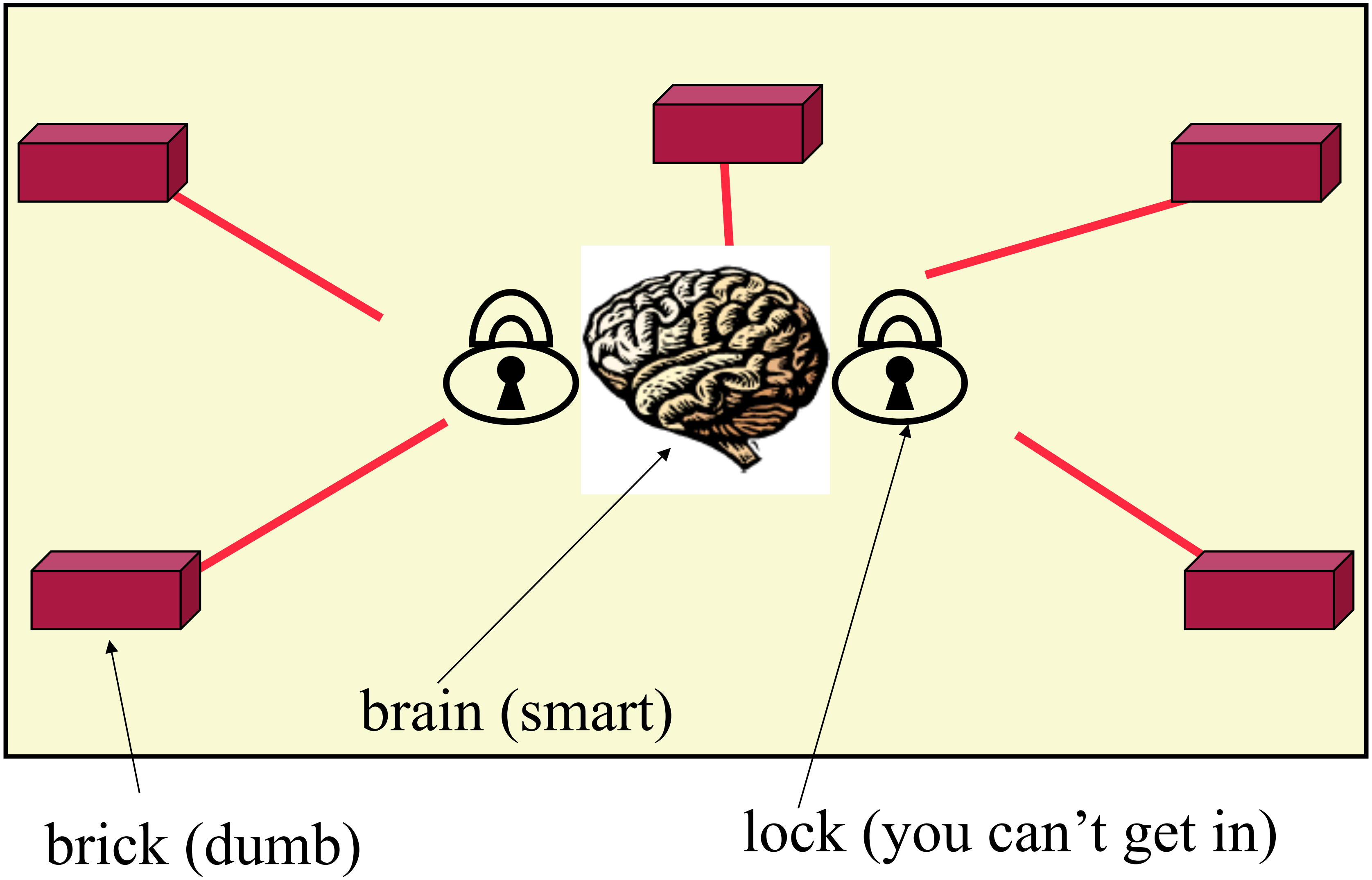
Packet Switching: Statistical Multiplexing



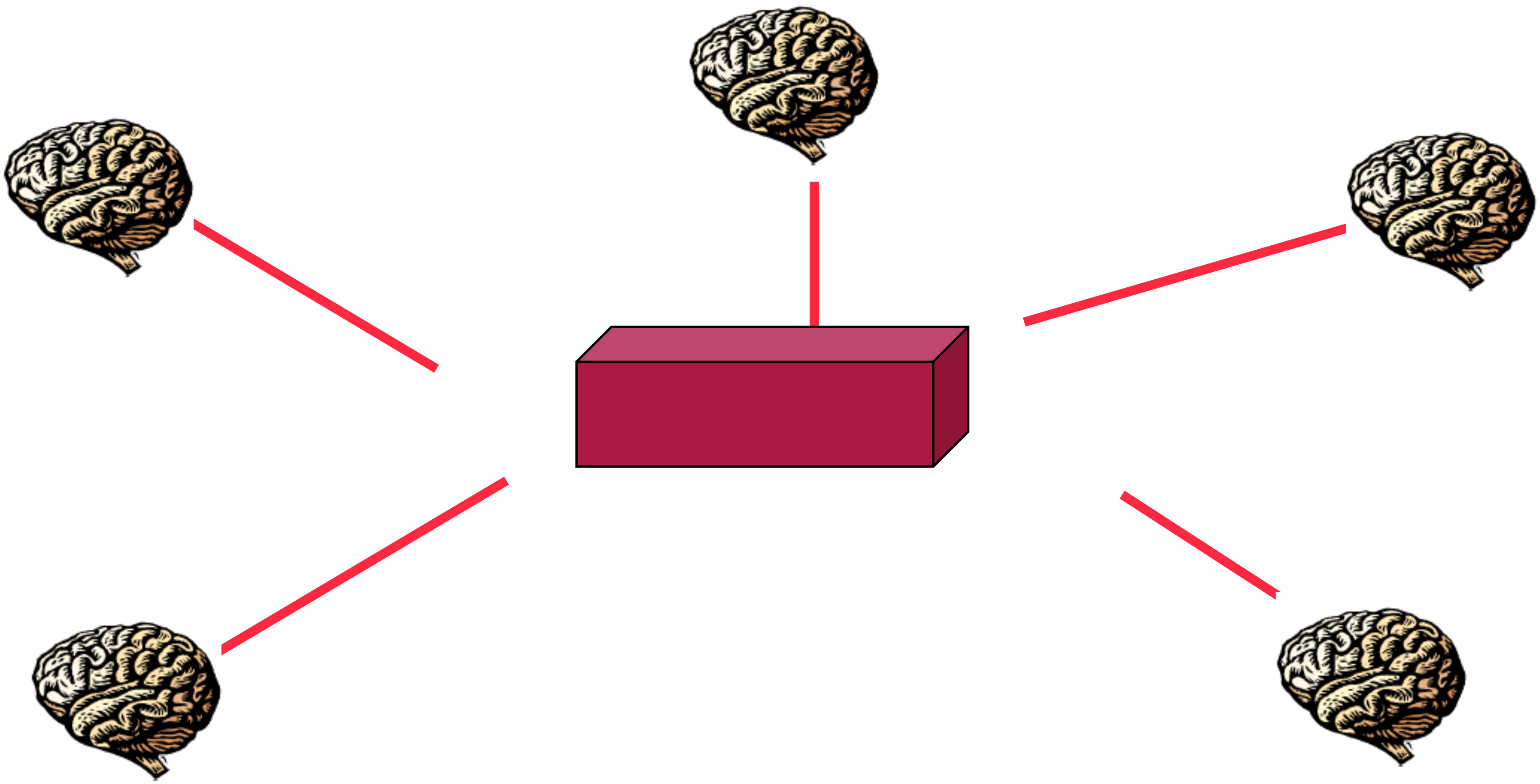
Packet Switching vs. Circuit Switching: In Theory

- Packet Switching
  - CS wastes bandwidth when data is sporadic
  - PS is statistically more efficient and less costly
  - CS takes time to establish the circuit
  - PS is simpler to implement
  - *Side Question: what about packet sizes? Small or Large?*
- Circuit Switching
  - PS is not suitable for real time application
  - A sudden surge of traffic could overflow router's buffers
  - PS could deliver packets in wrong order
  - CS is transparent (carrier does not need to know packet format)

Common View of the Telco Network (CS)



Common View of the IP Network (PS)



PS vs CS in Practice

Common assumptions about the Internet (That you found in many textbooks and research papers)

- IP *dominates* global communications
- Packet switching is more efficient than circuit switching
- Packet switching is robust
- IP (and PS) is simpler
- Quality of Service (QoS) *can* be realized over IP

IP Dominates Global Communications? NO

- [US-census 2002] **Revenues**: Satellite Telecom (5.7B), ISPs (18.7B), Radio/TV broadcast (48.5B), Cable Distribution (77.7B), Cellular & other wireless Telecom (96.5B), Wired telecom-carriers (237.6B).
- [Nielsen/NetRatings survey 2004 & others] **Percentage of US households having access**: Internet (75%), Cable/Pay TV (78%), TV (98%)
- [RHK Industry Reports 2002] **Public Telecom Infrastructure Expenditures**: Core routers (1.7B), Edge routers (2.4B), SONET/SDH/WDM (28.0B), Telecom Multi-Service Switches (4.5B)

PS is more efficient than CS? Yes, but …

- More efficient means better utilized (both in transmission lines and switching equipments)
- True for networks with scarce bandwidths
- However, does it really matter today?
  - Average utilization levels
    - ATT switched voice (33%), Internet backbones (15%)
    - Private lines networks (3-5%), LANs (1%)
  - Various Reasons
    - Internet traffic is asymmetric and bursty, links are symmetric
    - Operators tend to over-provision because PS networks behave very badly once congested (oscillation, routing loops, black holes, disconnections, etc)
    - Over-provision to ensure low delay (satisfy customers), it's more economical to add capacity in large increments

PS is more robust than CS? Not necessarily …

- Downtime per year:
  - Internet: 471min [Labovitz et al. 2000]
  - Phone networks: 5min [Kuhn 1997]
- Recover time
  - Internet: median 3min, frequently > 15min (due to slow BGP convergence time)
  - SONET/SDH rings: < 50ms (via pre-computed backup paths)
- Routing in the Internet
  - Routing info affected by user traffic, suffering from congestion (in-band routing)
  - Routing computation complex → overload processors
  - Probability of mis-configuring a router is high, one router's error affect the whole network

IP (and PS) is simpler?

- Number of lines of codes in
  - Typical Tel. Switches: 3 millions, extremely complex switch: 16M
  - Cisco's IOS: 8 millions [more susceptible to attacks]
- Routers crash frequently, takes long time to reboot
- Hardware
  - A line card of a router: OC192 POS has 30M gates + 1 CPU + 300MB packet buffers + 2MB forwarding table + 10MB other state memory
  - Current trend makes routers more complex (multicast, QoS, access control, security, VPN, etc) – violation of E2E
  - A line card of a typical transport switch: ¼ number of gates, no CPU, no forwarding table, one on-chip state memory
- Density: highest transport switch capacity = 4 x highest router capacity, at 1/3 the price
  - WDM, DWDM push the difference further
- IP's "simplicity" does not scale!

QoS can be realized over IP?

- Belief: over-provisioning allows low e2e delay → guaranteeing QoS is possible

- After > 10 years of research, IntServ and DiffServ are still not good enough.

- Few financial incentive to provide QoS over IP
  - Watch out for VoIP, however.
  - On the other hand, current phone services are much better with very low price

Other measures

- Scalability
  - CS scales more or less linearly
  - When data rates increase, routers can't keep up
- Flexibility
  - IP is more flexible
  - Lead to high costs of end-systems
  - Need more sophisticated users [large organizations need a room of sys admin, just 1 phone operator]

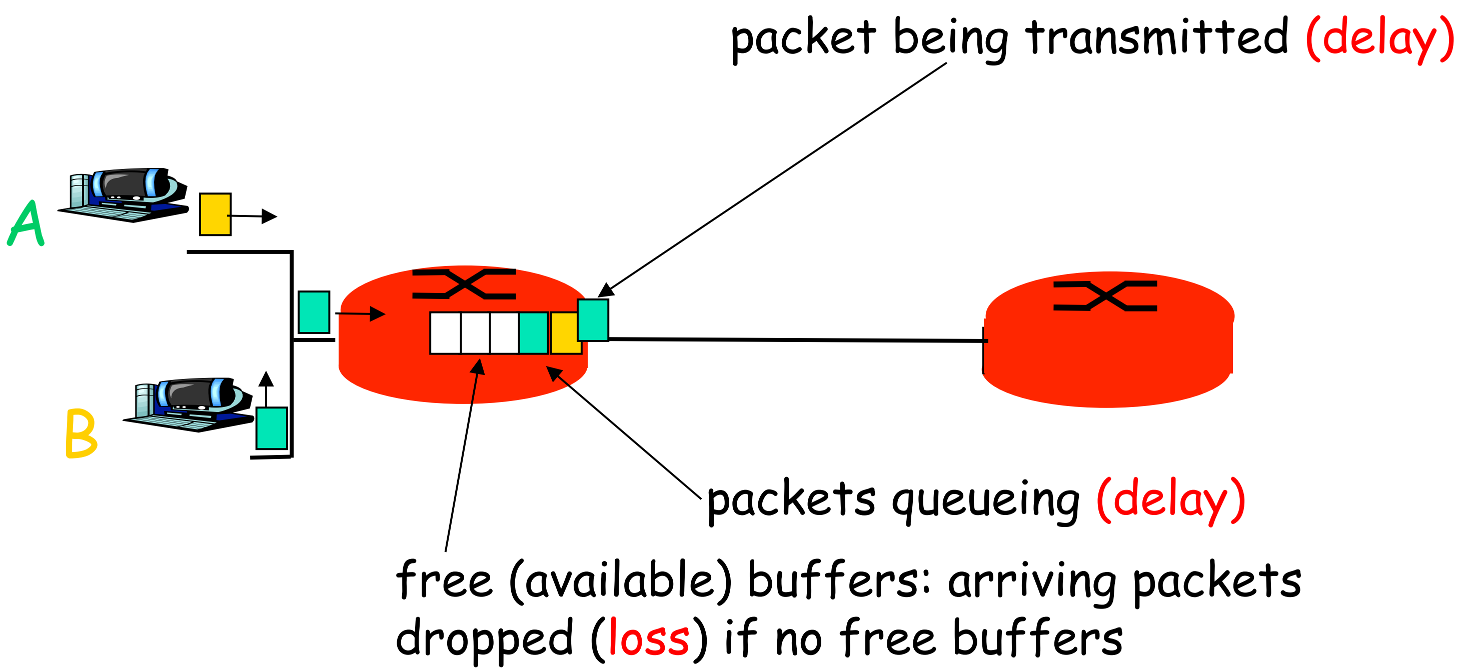
This Lecture

- How to send data from end to end: two switching methods
  - Circuit switching
  - Packet switching
- Packet loss and delay in a packet switched network

How do loss and delay occur?

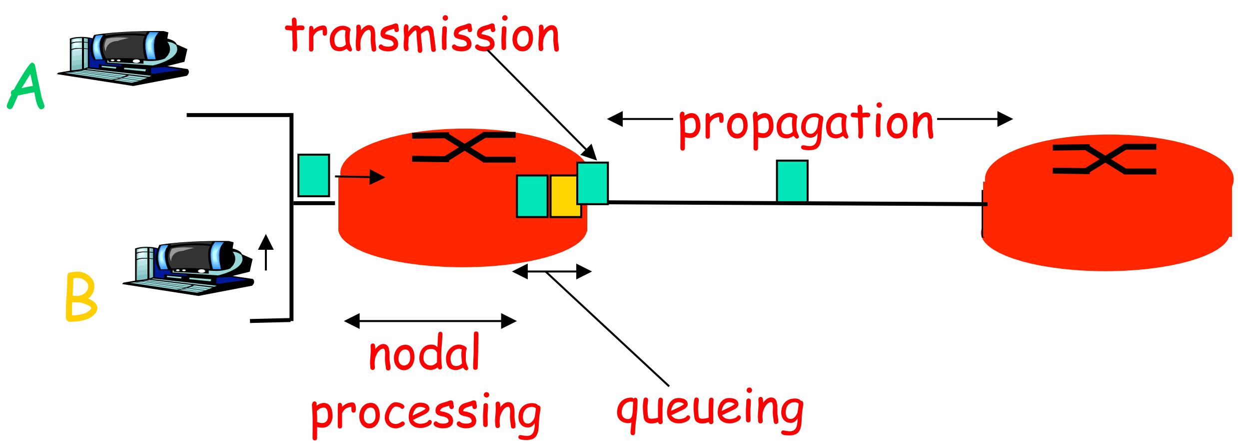
Packets *queued* in router buffers

- packet arrival rate to link exceeds output link capacity
- packets queued, wait for turn



Four sources of packet delay

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



Delay in packet-switched networks

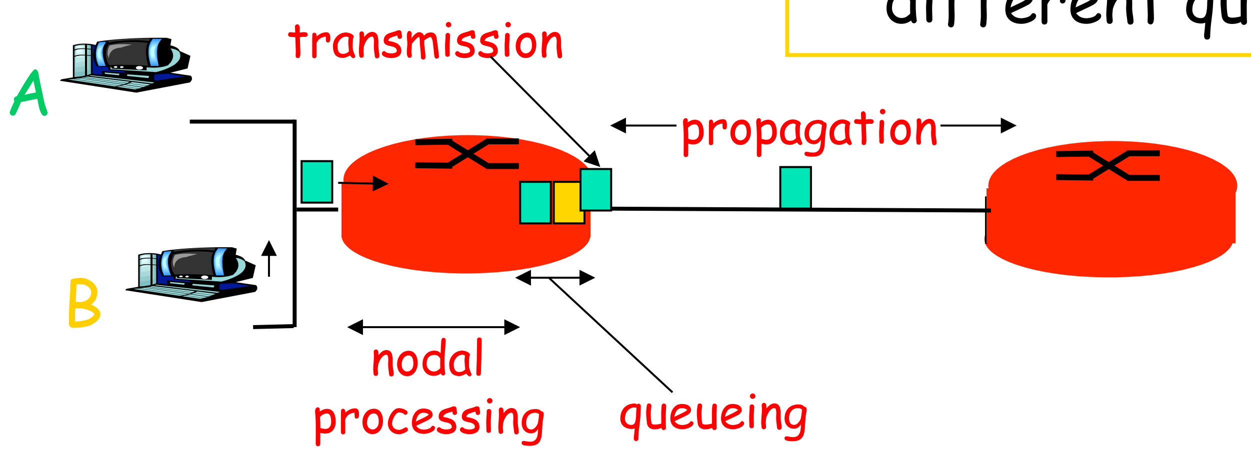
3. Transmission delay:

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

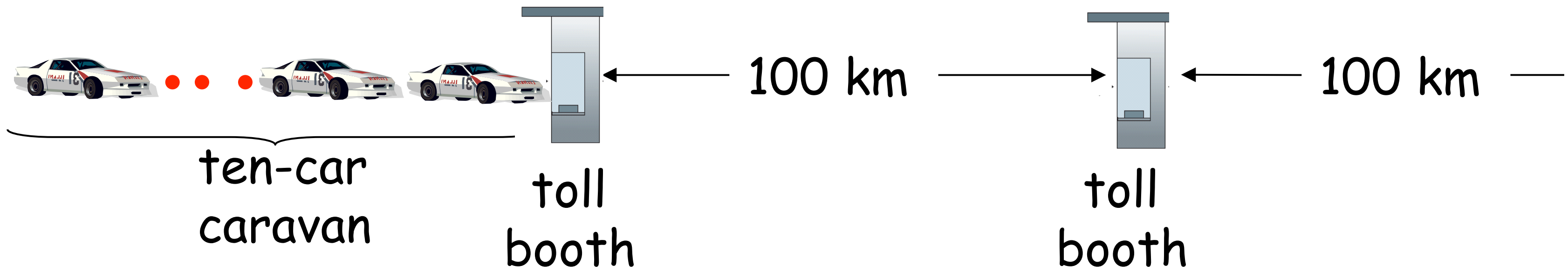
4. Propagation delay:

- d = length of physical link
- s = propagation speed in medium (~2×10^8 m/sec)
- propagation delay = d/s

Note: s and R are *very* different quantities!

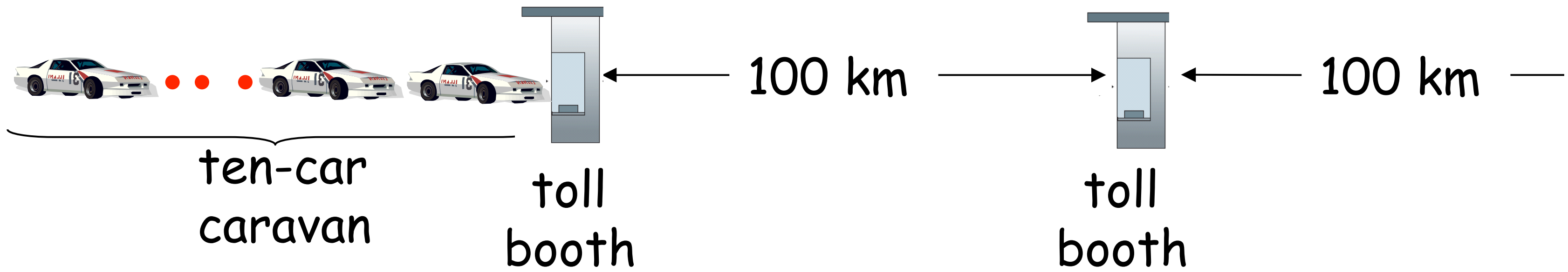


Caravan analogy



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

Caravan analogy (more)



- 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!
  - See Ethernet applet at AWL Web site

Nodal delay

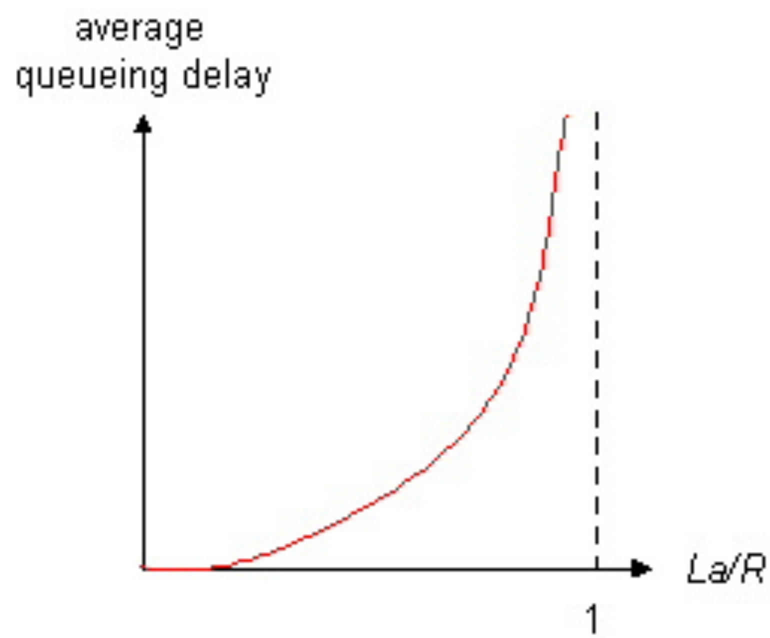
$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

- dproc = processing delay
  - typically a few microsecs or less
- dqueue = queuing delay
  - depends on congestion
- dtrans = transmission delay
  - = L/R, significant for low-speed links
- dprop = propagation delay
  - a few microsecs to hundreds of msecs

Queueing delay (revisited)

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

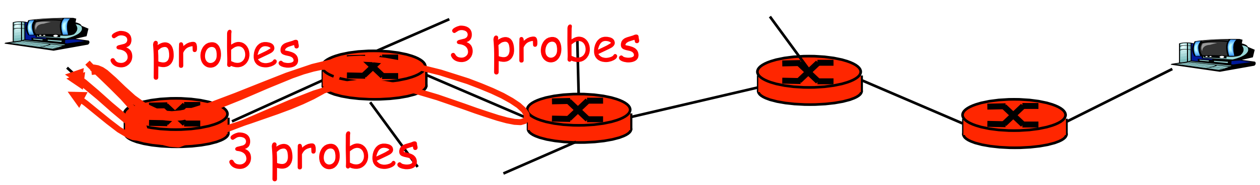
traffic intensity = La/R



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

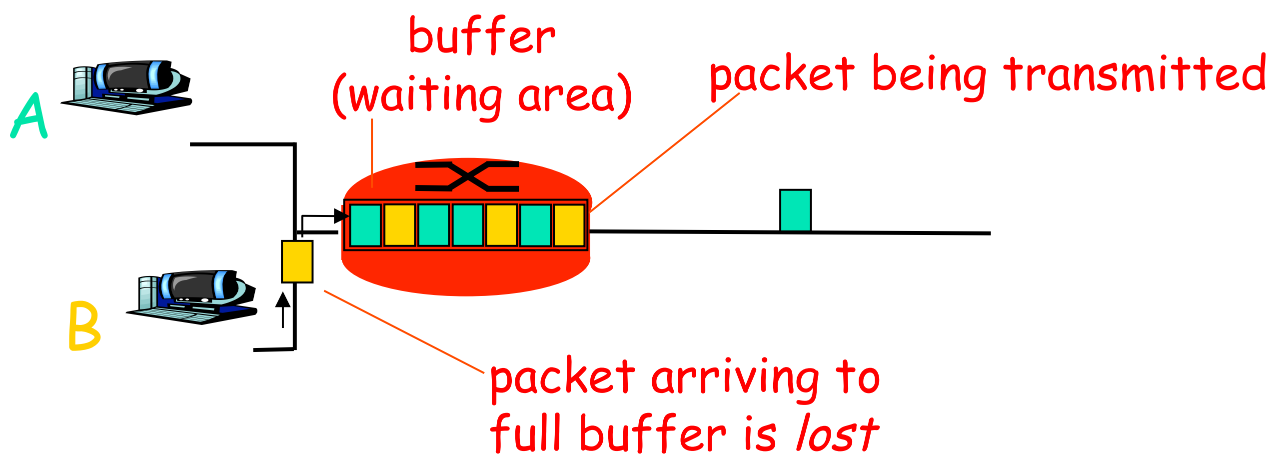
"Real" Internet delays and routes

- 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.



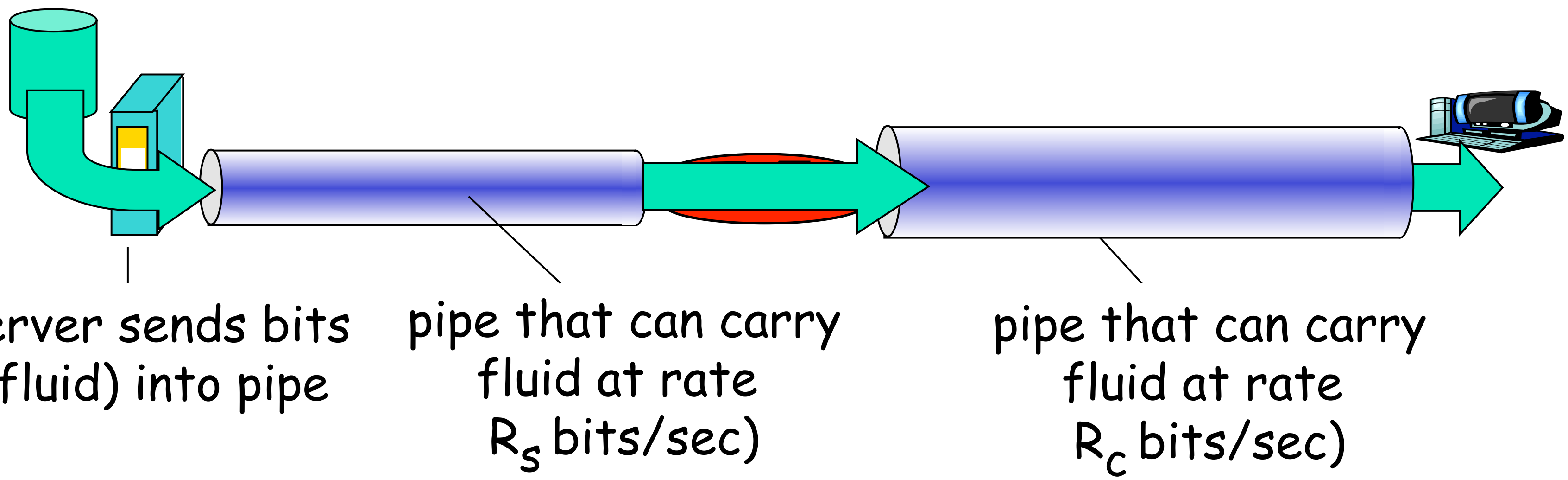
Packet loss

- 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



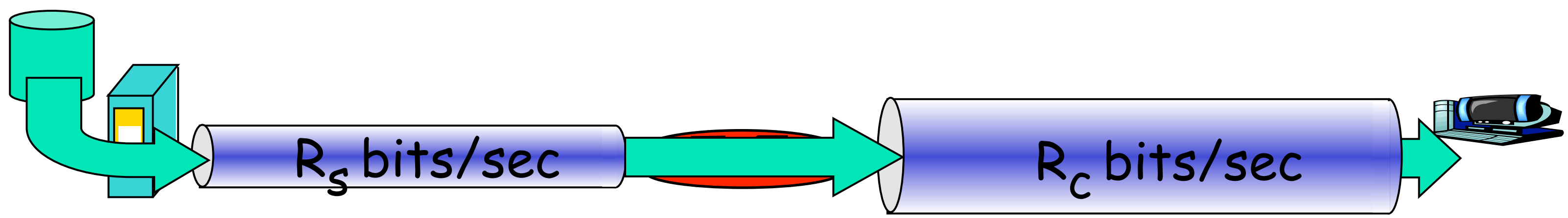
Throughput

- 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

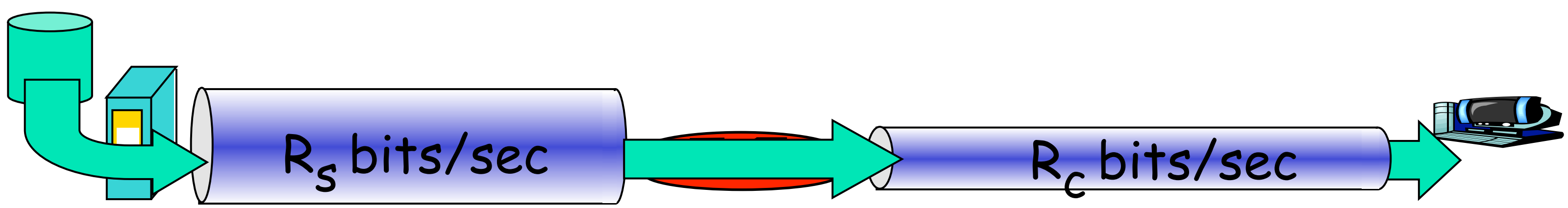


Throughput (more)

- Rs < Rc What is average end-end throughput?



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



bottleneck link

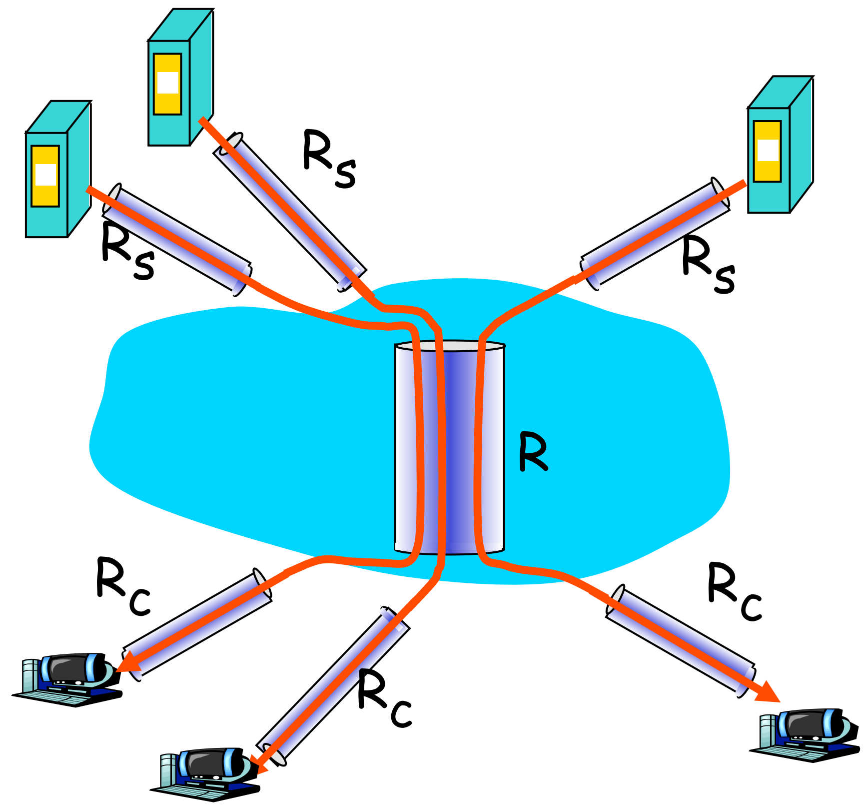
link on end-end path that constrains end-end throughput

Throughput: Internet scenario

- per-connection end-end throughput is

$$\min\{R_s, R_c, \frac{R}{10}\}$$

- in practice: R_c or R_s is often bottleneck



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