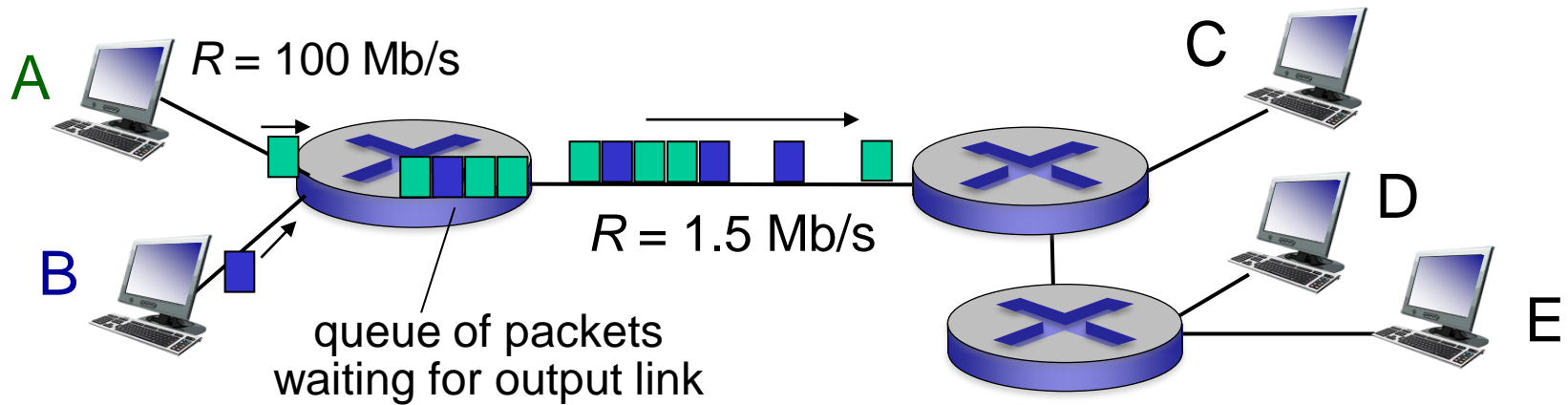


Overview of last class

Packet Switching: queueing delay, loss



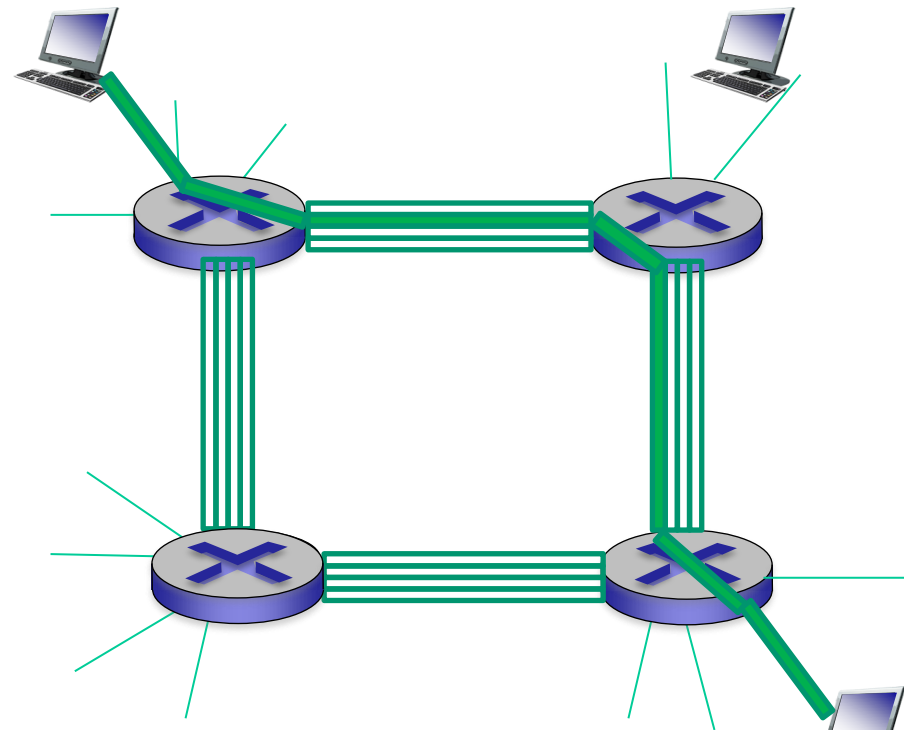
queuing and loss:

- 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

Alternative core: circuit switching

end-end resources allocated to, reserved for “call” between source & dest:

- in diagram, each link has four circuits.
 - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
 - circuit-like (guaranteed) performance
- circuit segment idle if not used by call (*no sharing*)
- commonly used in traditional telephone networks

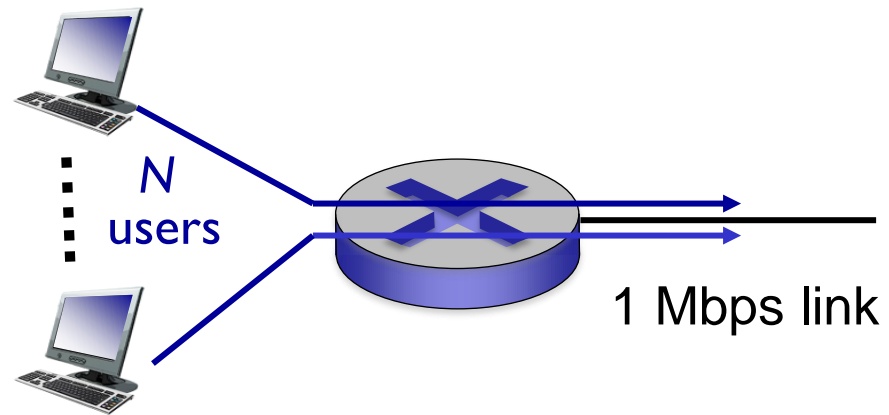


Packet switching versus circuit switching

packet switching allows more users to use network!

example:

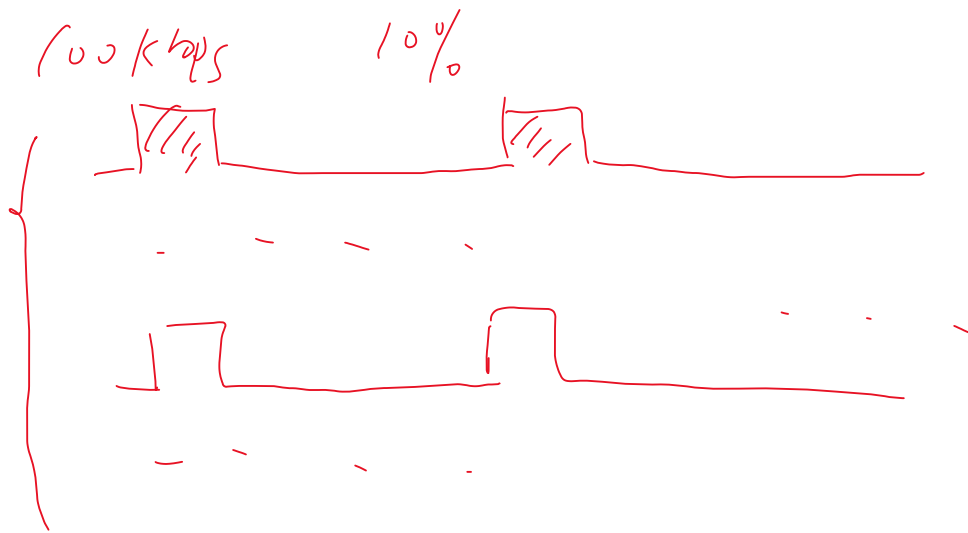
- 1 Mb/s link
- each user:
 - 100 kb/s when “active”
 - active 10% of time
- *circuit-switching*:
 - 10 users
- *packet switching*:
 - with 35 users, probability > 10 active at same time is less than .0004 *



Q: how did we get value 0.0004?

Q: what happens if > 35 users ?

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/



① Peak rate allocation

$$\frac{1 \text{ Mbps}}{100 \text{ kbps}} = 10 \text{ users}$$

Utilization low
delay $\rightarrow 0$

② Average rate allocation

$$\frac{1 \text{ Mbps}}{100 \text{ kbps} \times 10\%} = 100 \text{ users}$$

Utilization high
delay $\rightarrow \infty$

③ Effective bandwidth
35 users with overloading $p_{nb} < 0.0004$
(a proper number, under a QoS constraint)

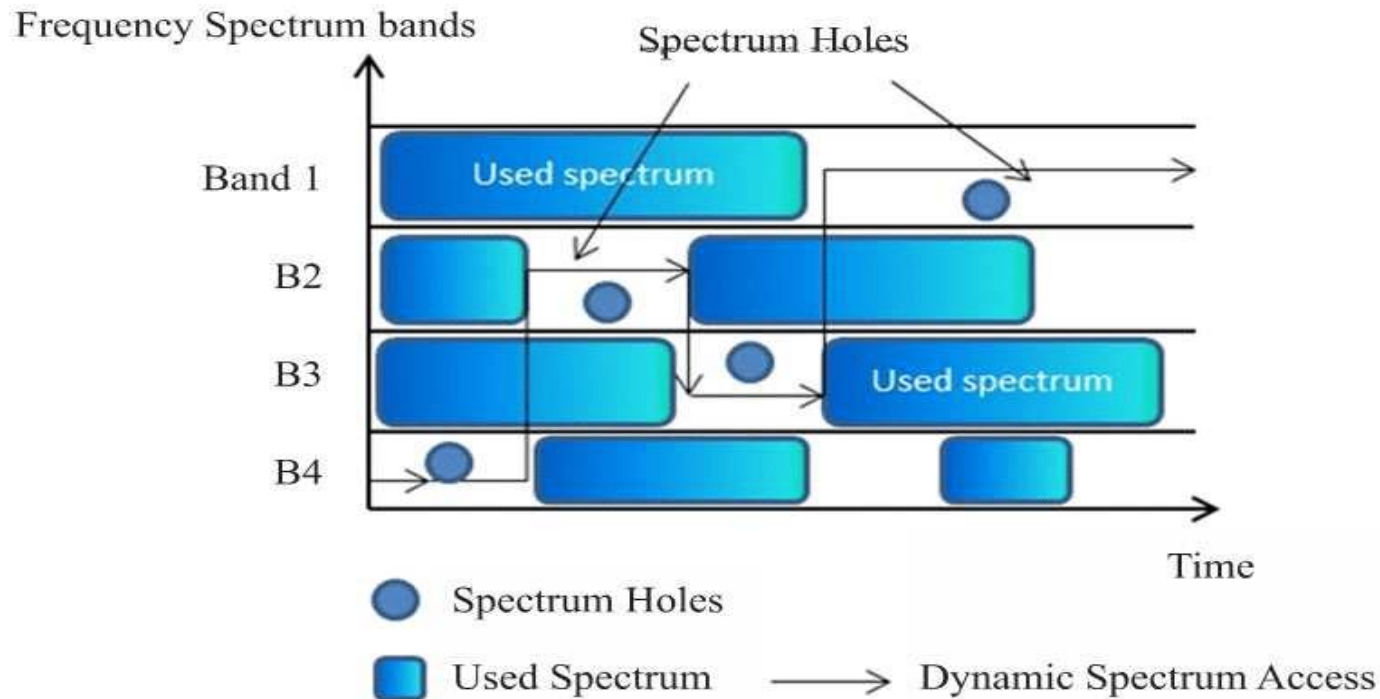
Utilization proper

Stochastic QoS guarantee

Dynamic Spectrum Sharing (Cognitive Radio)

- white spaces spectrum: frequencies allocated to a broadcasting service but not used locally
- Dynamic spectrum sharing to improve efficiency

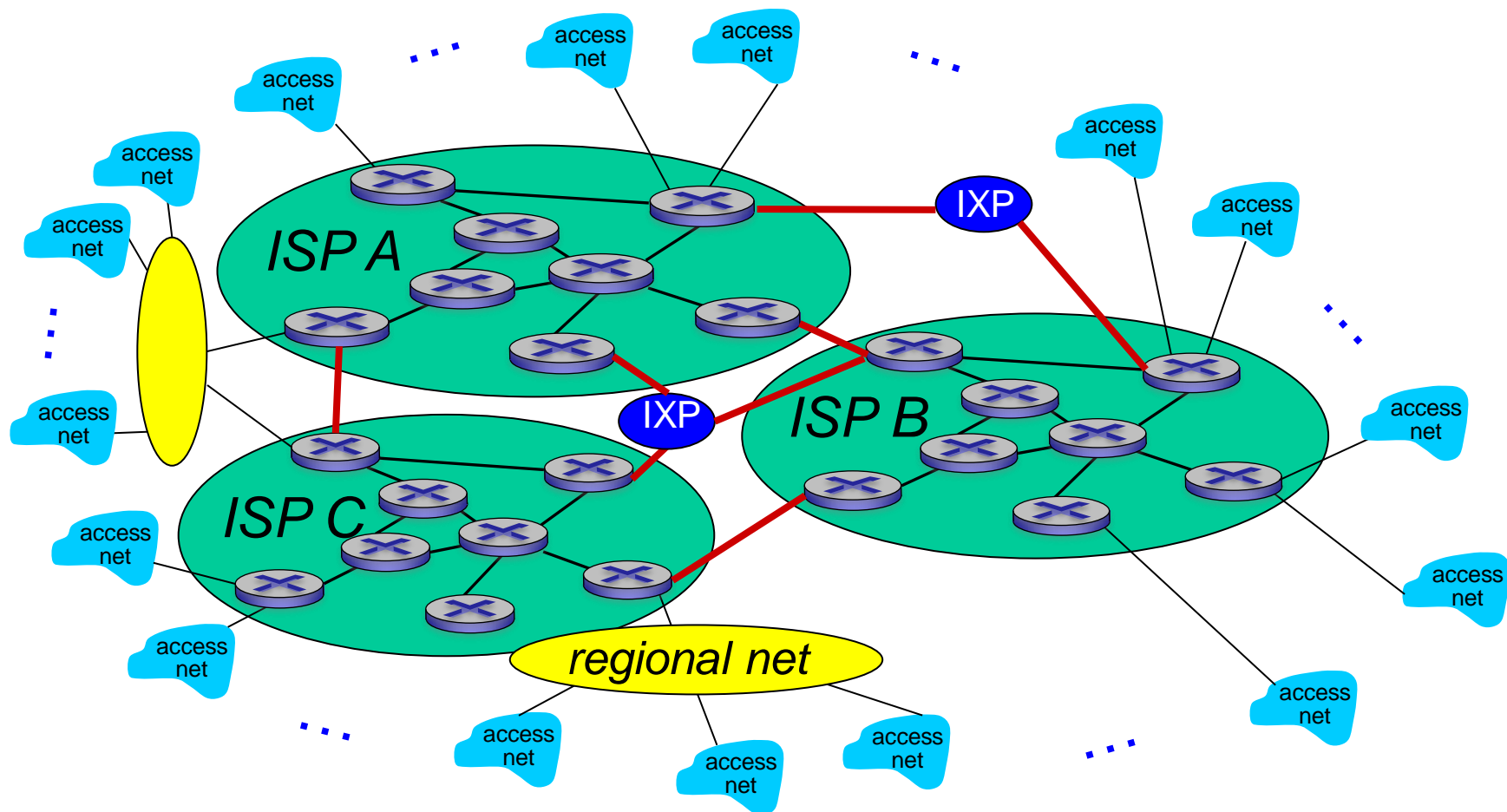
[https://en.wikipedia.org/wiki/White_spaces_\(radio\)#White_spaces_devices](https://en.wikipedia.org/wiki/White_spaces_(radio)#White_spaces_devices)



Internet structure: network of networks

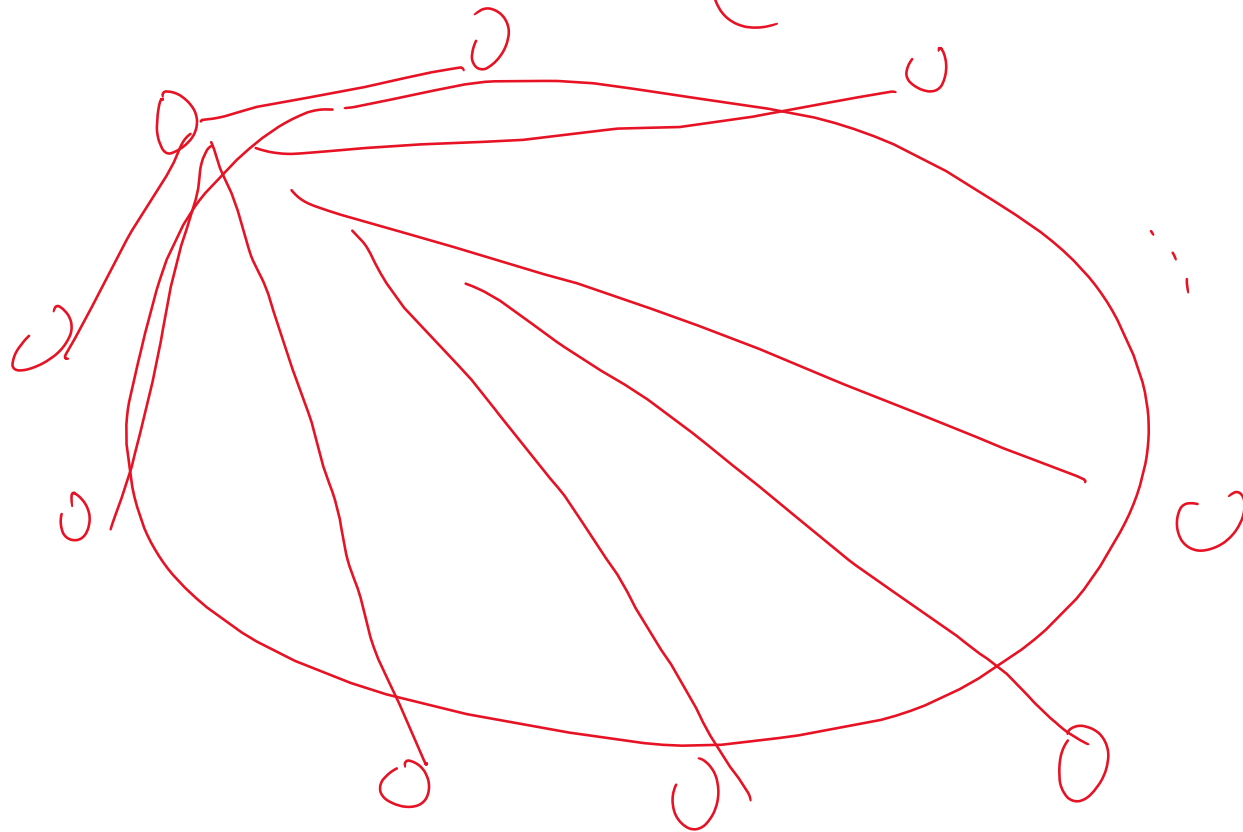
Hierarchical Structure: Scalability

shared resource

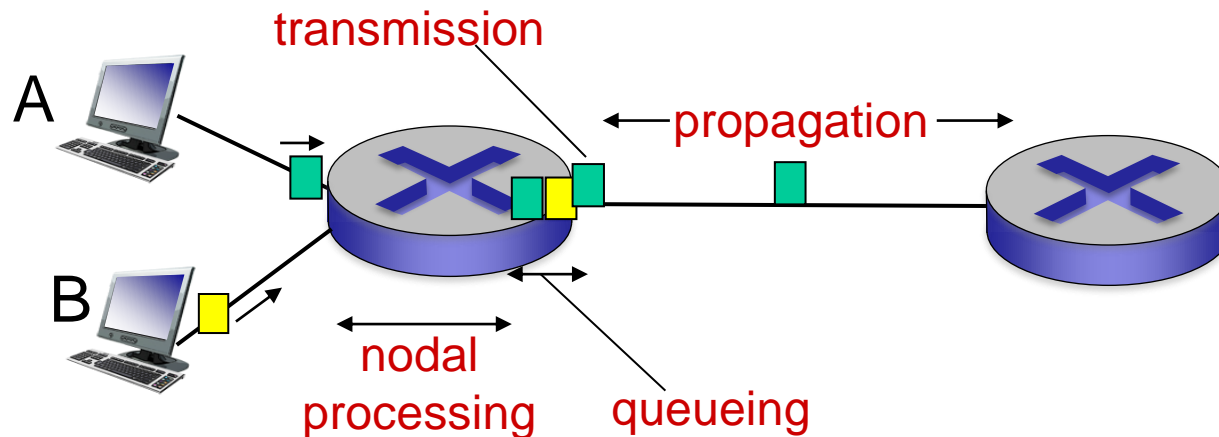


Dedicated resource allocation

(N^2) scalability



Four sources of packet delay



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

d_{trans} : transmission delay:

- L : packet length (bits)
- R : link bandwidth (bps)

▪ $d_{\text{trans}} = L/R$ ← d_{trans} and d_{prop} →
very different

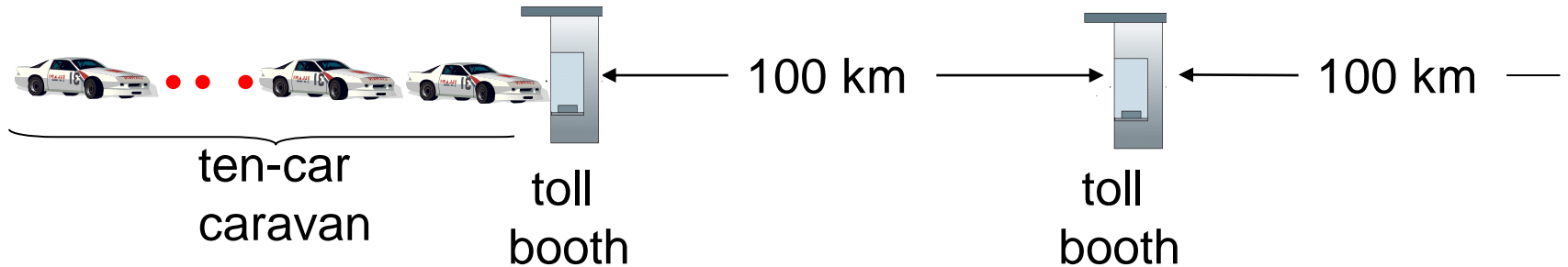
d_{prop} : propagation delay:

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

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

* Check out the Java applet for an interactive animation on trans vs. prop delay

Caravan analogy

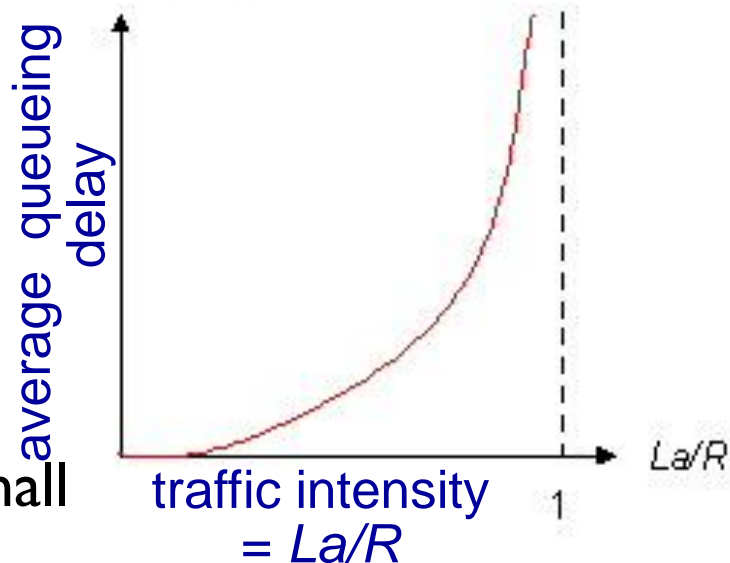


- cars “propagate” at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- car ~ bit; caravan ~ packet
- Q: How long until caravan is lined up before 2nd toll booth?
- time to “push” entire caravan through toll booth onto highway = $12 * 10 = 120$ sec
- time for last car to propagate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

Queueing delay (revisited)

Delay
loss
Throughput

- R : link bandwidth (bps)
- L : packet length (bits)
- a : average packet arrival rate
- $La/R \sim 0$: 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!



Peak rate allocation

Average rate allocation

Effective bandwidth allocation



$La/R \sim 0$

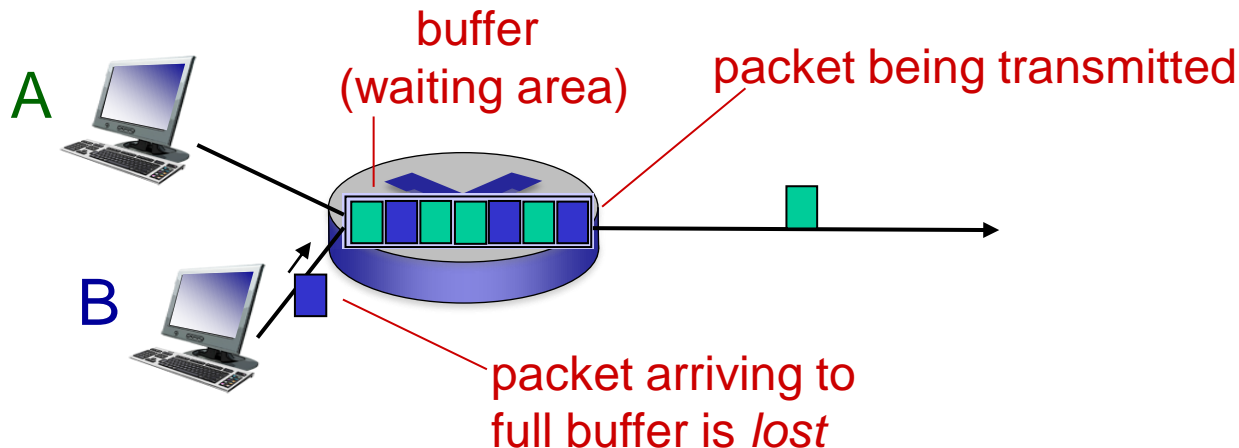


$La/R \rightarrow 1$

* Check online interactive animation on queuing and loss

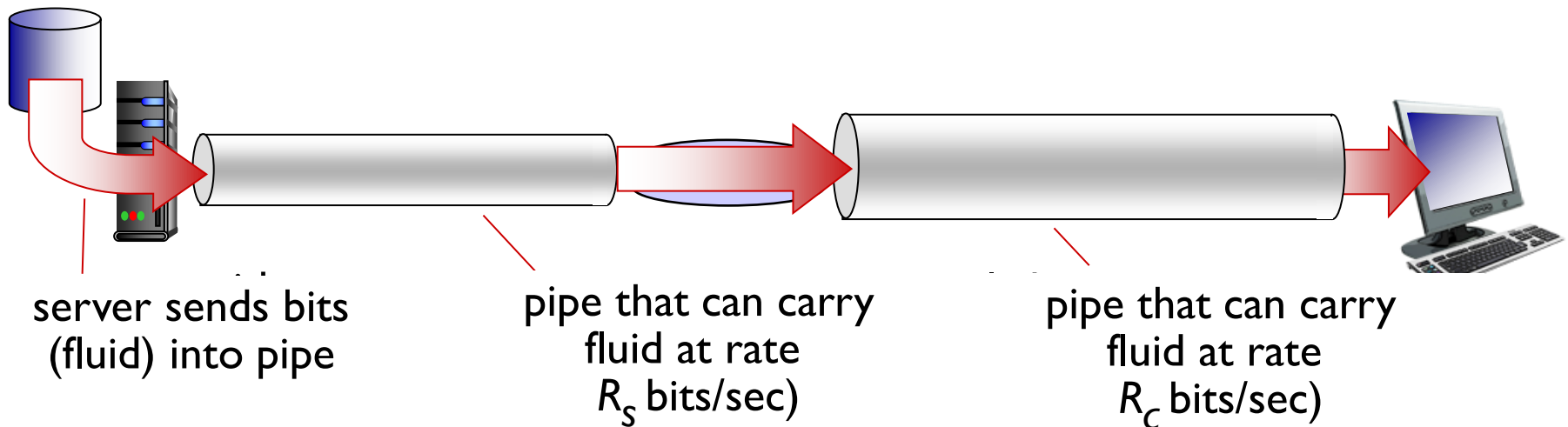
Packet loss

- 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



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



Instantaneous

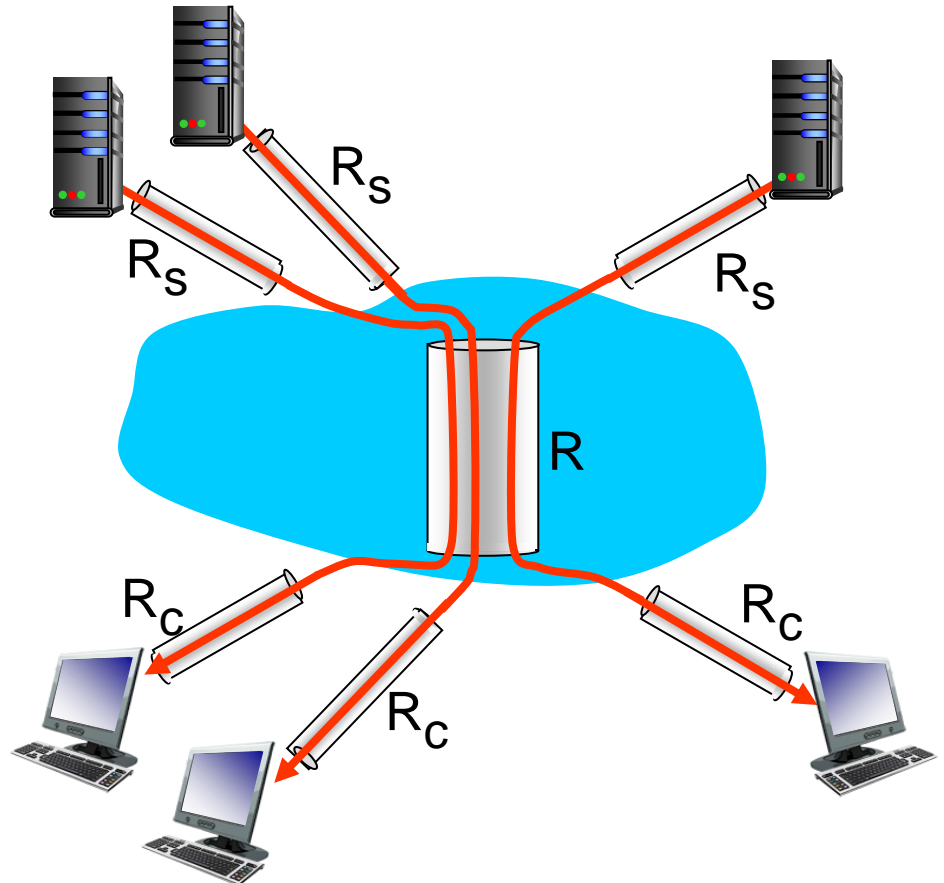
rate

Average
rate



Throughput: Internet scenario

- per-connection end-end throughput:
 $\min(R_c, R_s, R/I O)$
- in practice: R_c or R_s is often bottleneck



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

* Check out the online interactive exercises for more
examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Class Today

Chapter 1: 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

1.6 networks under attack: security

1.7 history

Protocol “layers”

*Networks are complex,
with many “pieces”:*

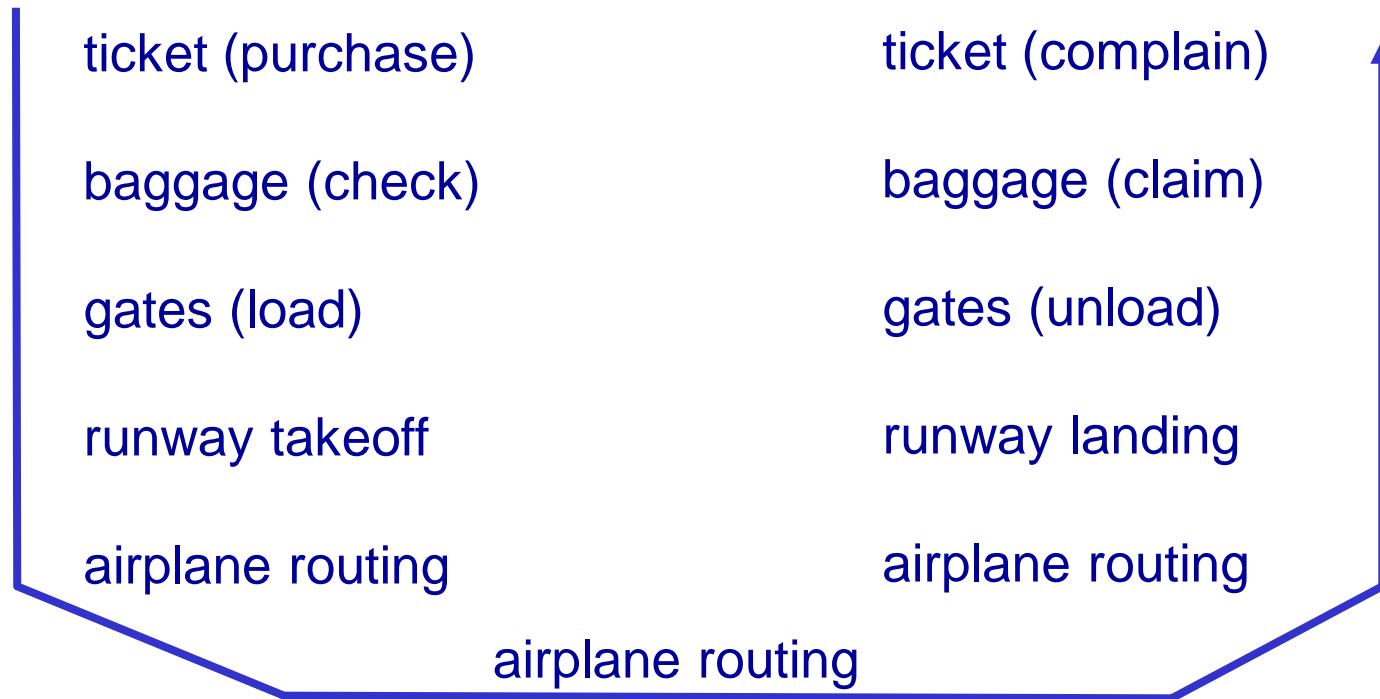
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

is there any hope of
organizing structure of
network?

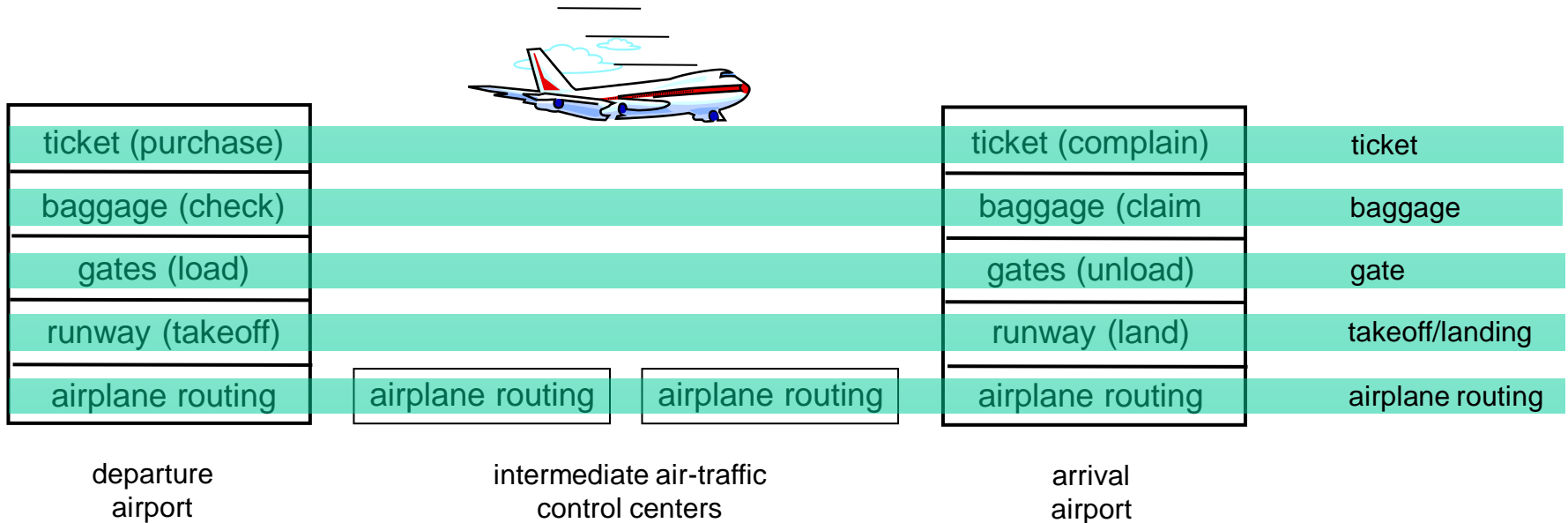
.... or at least our
discussion of networks?

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:

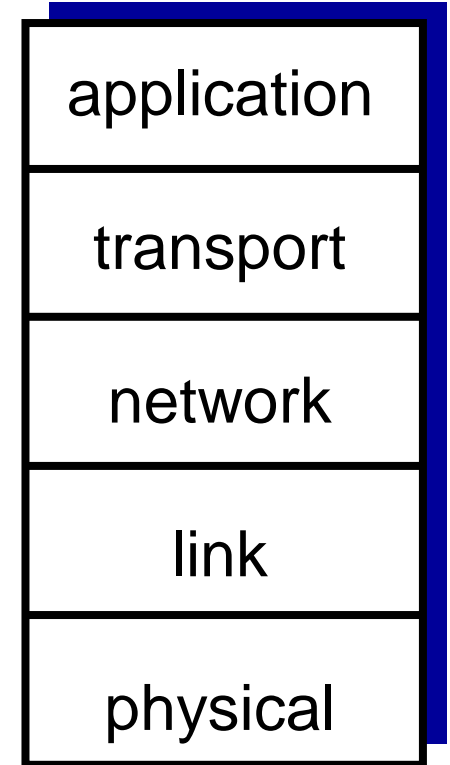
- explicit structure allows identification, relationship of complex system's pieces
 - layered *reference model* for discussion
- **modularization (modularity)** eases maintenance, updating of system
 - change of implementation of layer's service transparent to rest of system
 - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?
 - Redundant functions; how adapt to new networking context
 - Cross-layer

TCP over Wireless

- TCP (classic case): traffic control to avoid congestion
 - Long delay or loss → implicitly indicate network congested
→ reduce traffic rate
- TCP over wireless
 - Long delay or loss → normally due to bad wireless channel
→ reduce rate helpful?

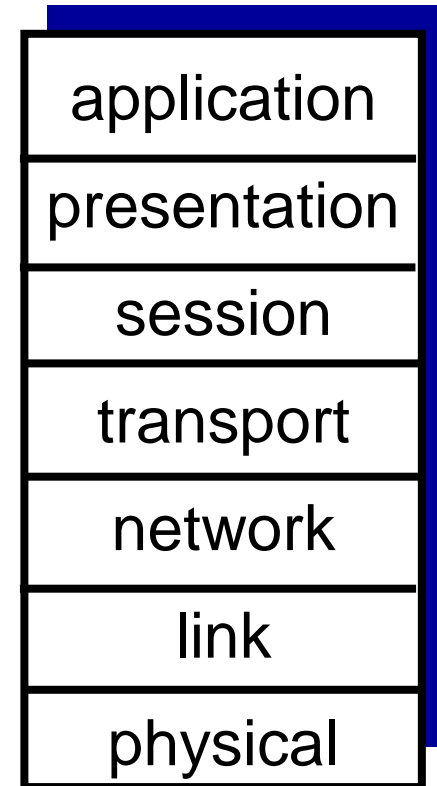
Internet protocol stack

- *application*: supporting network applications
 - SMTP, HTTP, DNS, FTP
- *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”

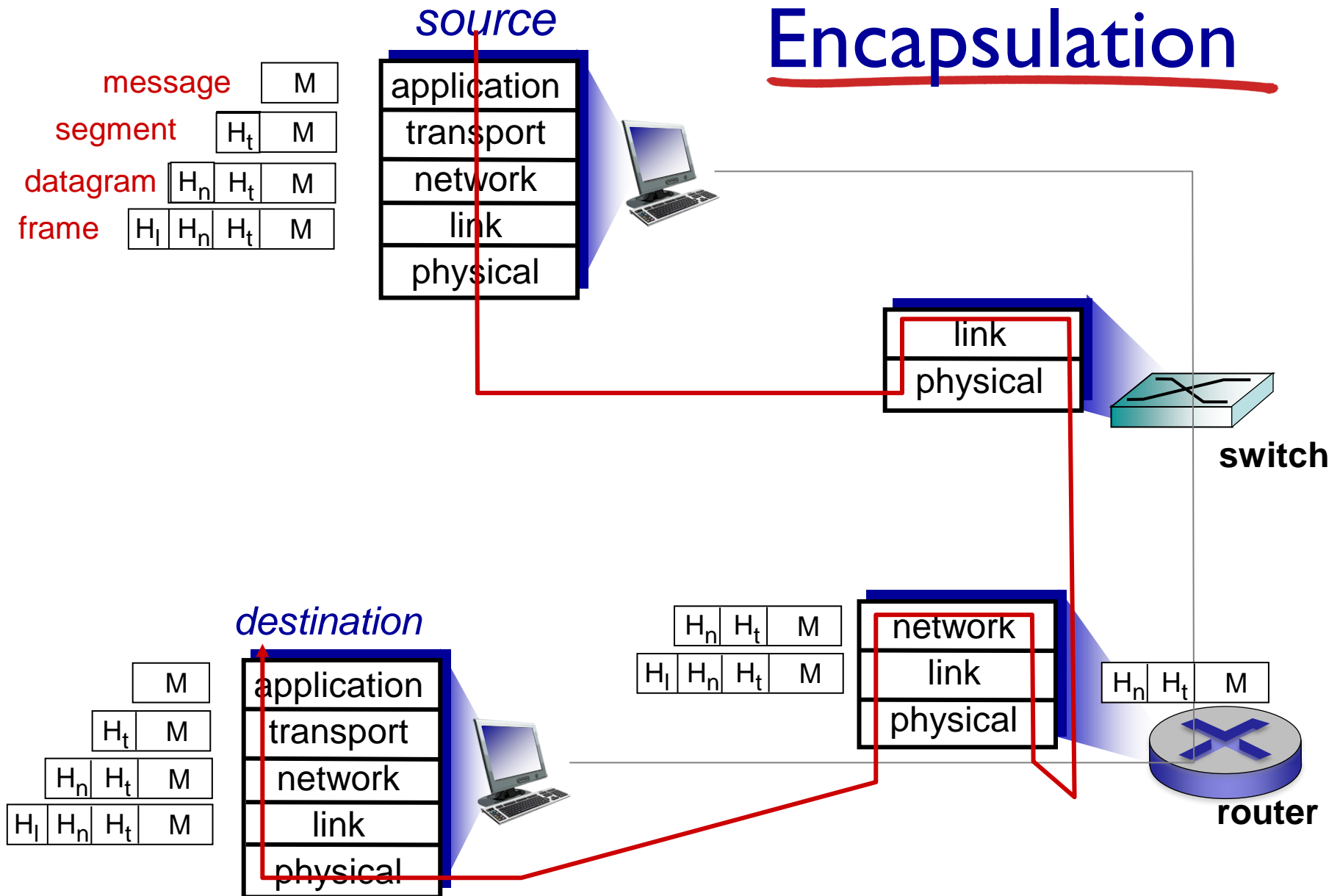


ISO/OSI reference model

- **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



Chapter 1: 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

1.6 networks under attack: security

1.7 history

Network security

- **field of network security:**
 - how bad guys can attack computer networks:
confidentiality, authentication, integrity, access control
 - how we can defend networks against attacks
 - how to design architectures that are immune to attacks
- **Internet not originally designed with (much) security in mind**
 - *original vision*: “a group of mutually trusting users attached to a transparent network” 😊
 - Internet protocol designers playing “catch-up”
 - security considerations in all layers!

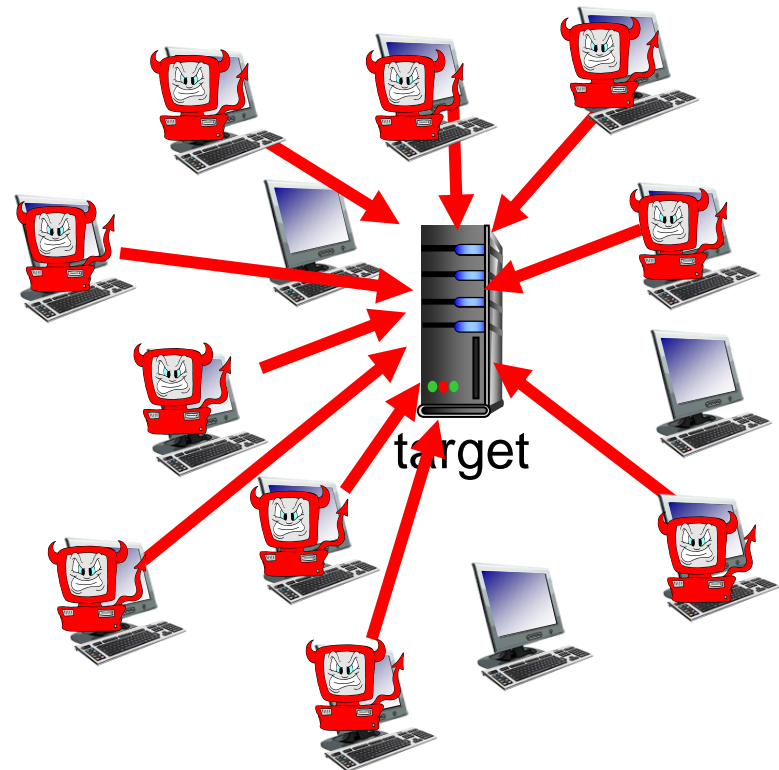
Bad guys: put malware into hosts via Internet

- malware can get in host from:
 - *virus*: self-replicating infection by receiving/executing object (e.g., e-mail attachment)
 - *worm*: self-replicating infection by passively receiving object that gets itself executed
- **spyware malware** can record keystrokes, web sites visited, upload info to collection site
- infected host can be enrolled in **botnet**, used for spam. DDoS attacks
 - Bot: is a software application that runs automated tasks (scripts) to imitate human activity on the Internet (https://en.wikipedia.org/wiki/Internet_bot)

Bad guys: attack server, network infrastructure

Denial of Service (DoS): attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic

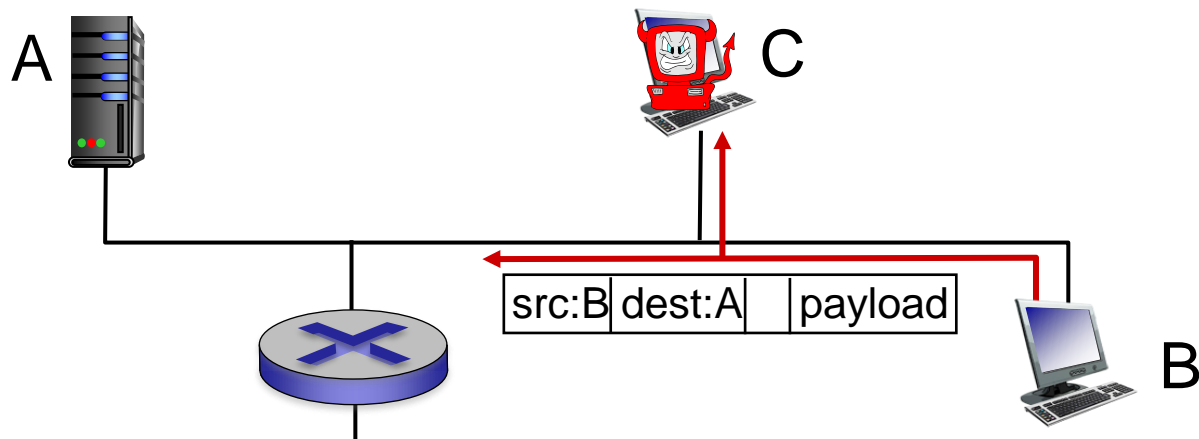
1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



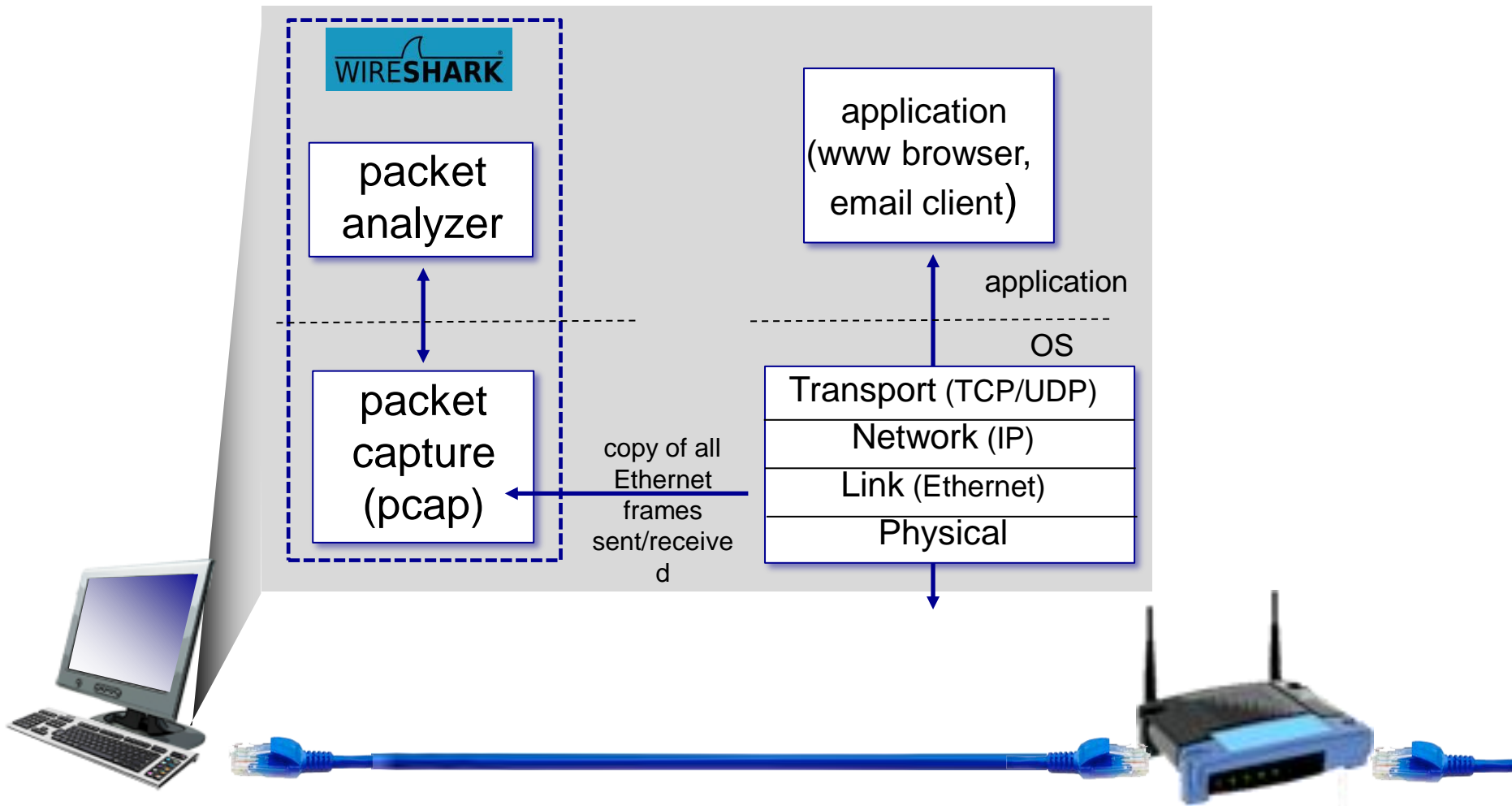
Bad guys can sniff packets

packet “sniffing”:

- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by

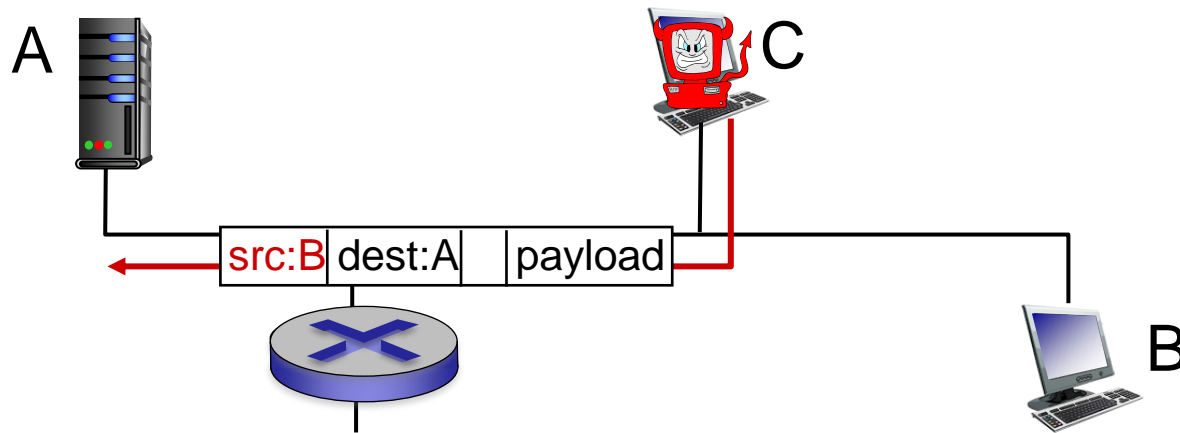


- wireshark software used for end-of-chapter labs is a (free) packet-sniffer



Bad guys can use fake addresses

IP spoofing: send packet with false source address



... lots more on security (throughout, Chapter 8)

Chapter 1: 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

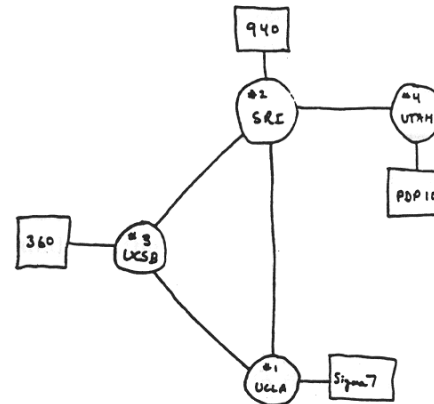
1.6 networks under attack: security

1.7 history

Internet history

1961-1972: Early packet-switching principles

- **1961:** Kleinrock - queueing theory shows effectiveness of packet-switching
- **1964:** Baran - packet-switching in military nets
- **1967:** ARPAnet conceived by Advanced Research Projects Agency
- **1969:** first ARPAnet node operational
- **1972:**
 - ARPAnet public demo
 - NCP (Network Control Protocol) first host-host protocol
 - first e-mail program
 - ARPAnet has 15 nodes



THE ARPA NETWORK

Internet history

1972-1980: Internetworking, new and proprietary nets

- **1970:** ALOHAnet satellite network in Hawaii (MAC)
- **1974:** Cerf and Kahn - architecture for interconnecting networks
- **1976:** Ethernet at Xerox PARC
- **late70' s:** proprietary architectures: DECnet, SNA, XNA
- **late 70' s:** switching fixed length packets (ATM precursor)
- **1979:** ARPAnet has 200 nodes

Cerf and Kahn' s internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

**define today' s Internet
architecture**

Smartness at Edge

Internet history

1980-1990: new protocols, a proliferation of networks

- **1983:** deployment of TCP/IP
- **1982:** smtp e-mail protocol defined
- **1983:** DNS defined for name-to-IP-address translation
- **1985:** ftp protocol defined
- **1988:** TCP congestion control
- new national networks: CSnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

Internet history

1990, 2000 's: commercialization, the Web, new apps

- early 1990' s: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
 - hypertext [Bush 1945, Nelson 1960' s]
 - HTML, HTTP: Berners-Lee
 - 1994: Mosaic, later Netscape
 - late 1990' s: commercialization of the Web

late 1990' s – 2000' s:

- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

Internet history

Challenges in the new century

- ~5B devices attached to Internet (2016)
 - smartphones and tablets, IoT
- aggressive deployment of broadband access
- increasing ubiquity of high-speed wireless access
- emergence of online social networks:
 - Facebook: ~ one billion users
- service providers (Google, Microsoft) create their own networks
 - bypass Internet, providing “instantaneous” access to search, video content, email, etc.
- e-commerce, universities, enterprises running their services in “cloud” (e.g., Amazon EC2)
- The impact of AI on Internet

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
- security
- history

you now have:

- context, overview, “feel” of networking
- more depth, detail *to follow!*

internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control