

Chapter 1: roadmap

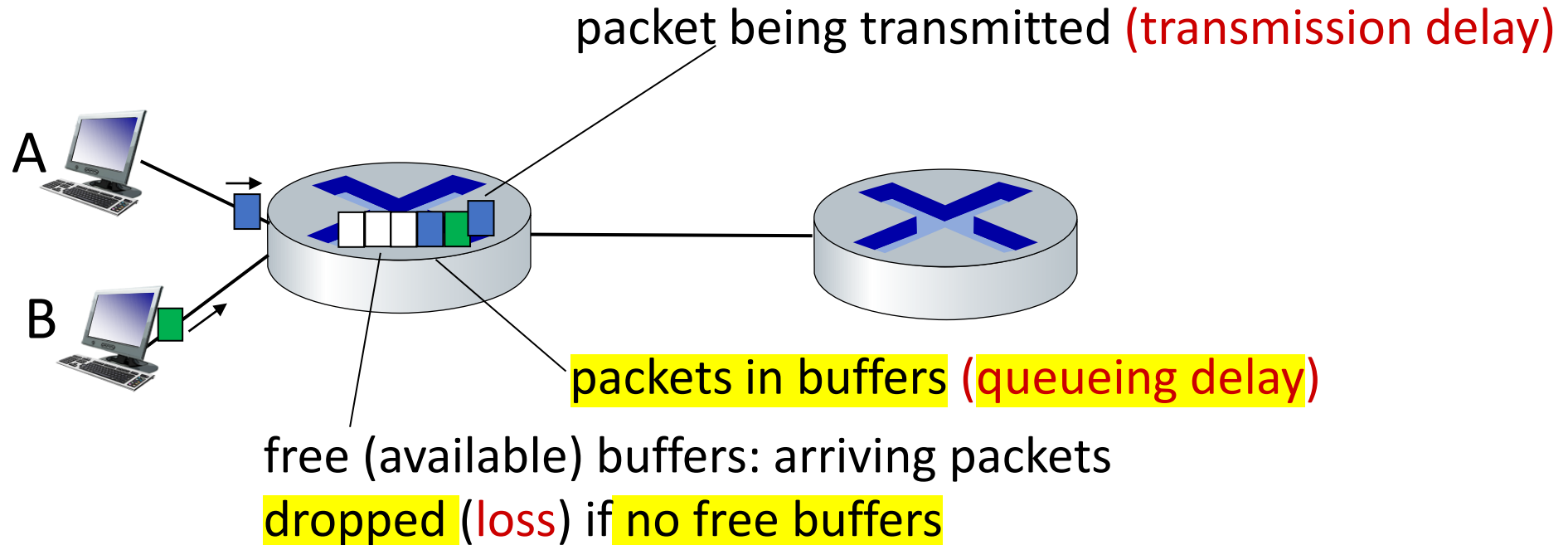
- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- **Performance: loss, delay, throughput**
- Security
- Protocol layers, service models
- History



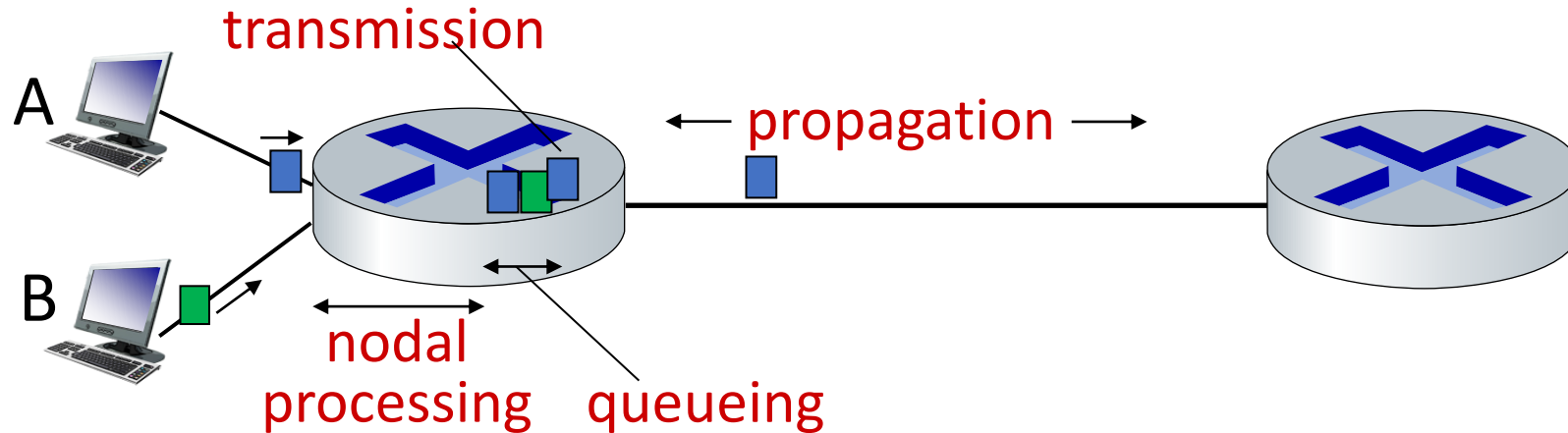
How do packet loss and delay occur?

packets *queue* in router buffers

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



Packet delay: four sources



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

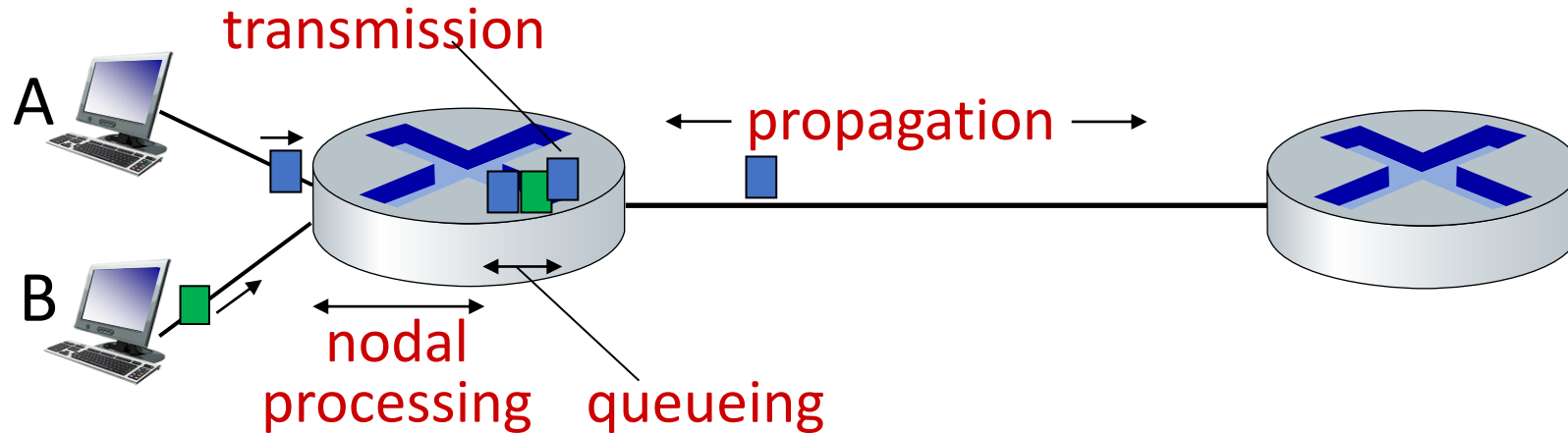
d_{proc} : processing delay

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

d_{queue} : queueing delay

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

Packet delay: four sources



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

d_{trans} : transmission delay:

- L : packet length (bits)
- R : link transmission rate (bps)

■ $d_{\text{trans}} = L/R$

d_{prop} : propagation delay:

- d : length of physical link
- s : propagation speed ($\sim 2 \times 10^8$ m/sec)

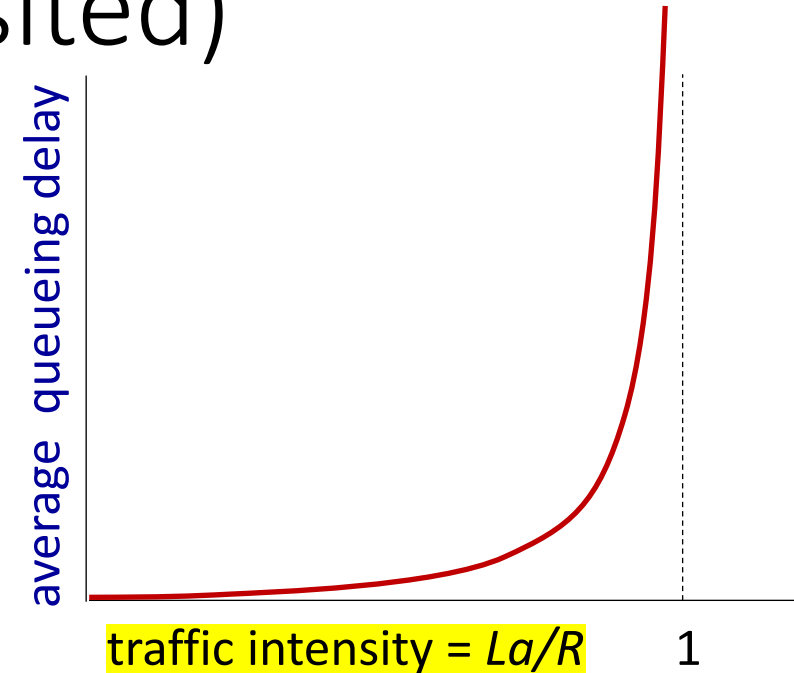
■ $d_{\text{prop}} = d/s$

d_{trans} and d_{prop}
very different

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

Packet queueing delay (revisited)

- R : link bandwidth (bps)
- L : packet length (bits)
- a : average packet arrival rate
- $La/R \sim 0$: avg. queueing delay small
- $La/R \rightarrow 1$: avg. queueing delay large
- $La/R > 1$: more “work” arriving is more than can be serviced - average delay infinite!

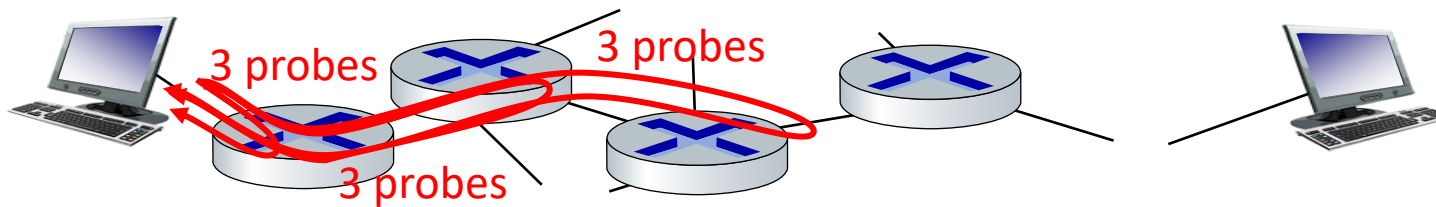


$La/R \sim 0$

$La/R \rightarrow 1$

“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 (with time-to-live field value of i)
 - router i will return packets to sender
 - sender measures time interval between transmission and reply



Real Internet delays and routes

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

Tracert in Windows!

3 round trip delay measurements from
gaia.cs.umass.edu to cs-gw.cs.umass.edu

3 delay measurements
to border1-rt-fa5-1-0.gw.umass.edu

trans-oceanic link

looks like delays
decrease! Why?

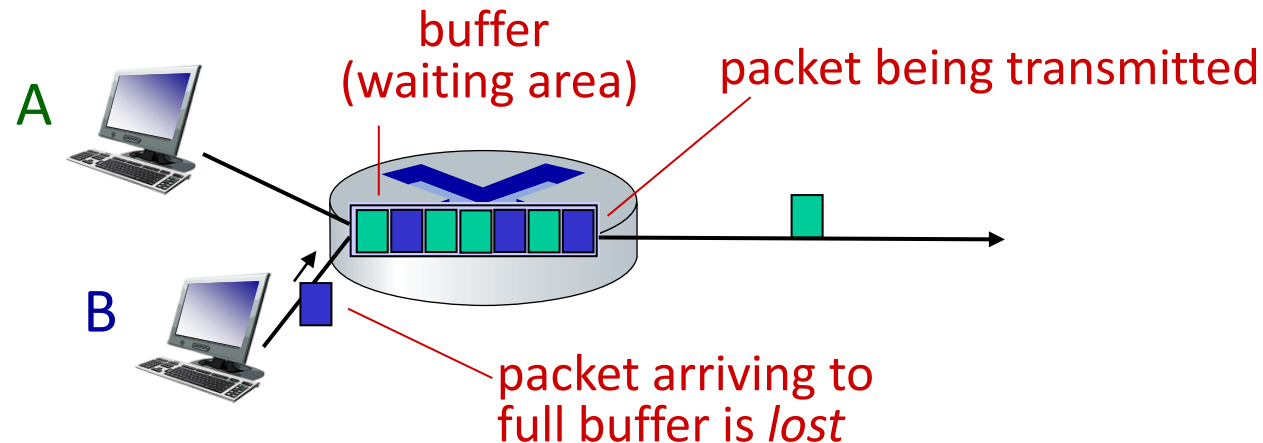
* means no response (probe lost, router not replying)

```
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
```

* Do some traceroutes from exotic countries at www.traceroute.org

Packet loss

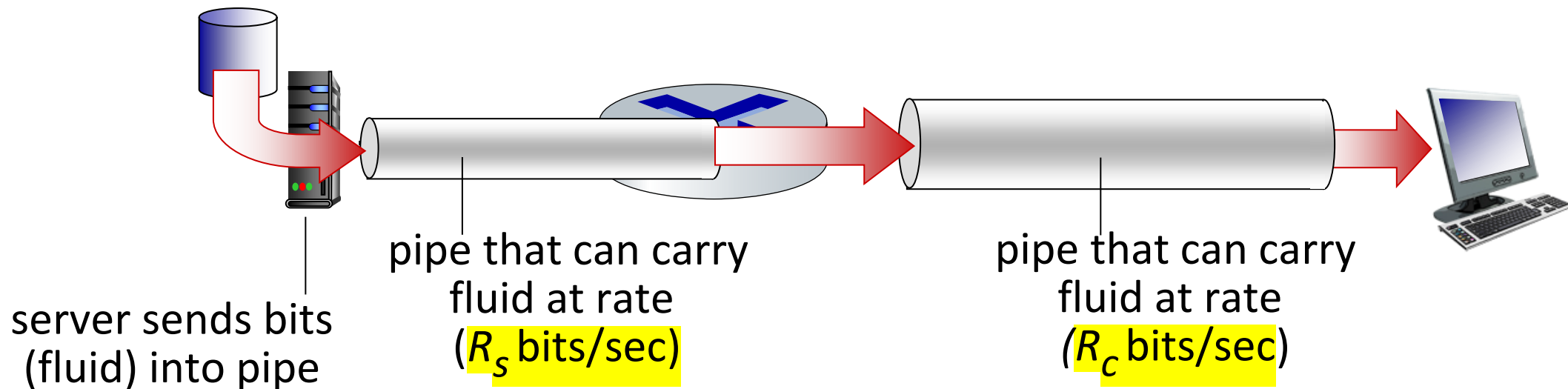
- queue (aka buffer) preceding link in buffer has **finite capacity**
- packet arriving to **full queue dropped** (aka lost)
- lost packet may be **retransmitted** by previous **node**, by **source end system**, or not at all



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

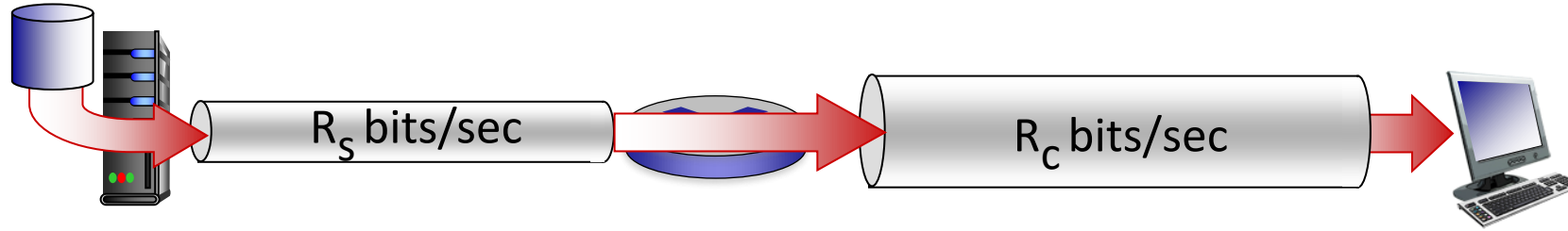
Throughput

- *throughput*: rate (bits/time unit) at which bits are being sent from sender to receiver
 - *instantaneous*: rate at given point in time
 - *average*: rate over longer period of time

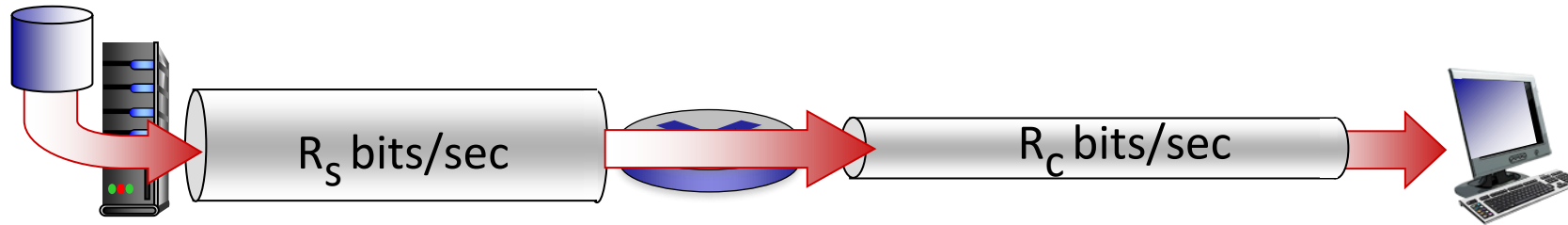


Throughput

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



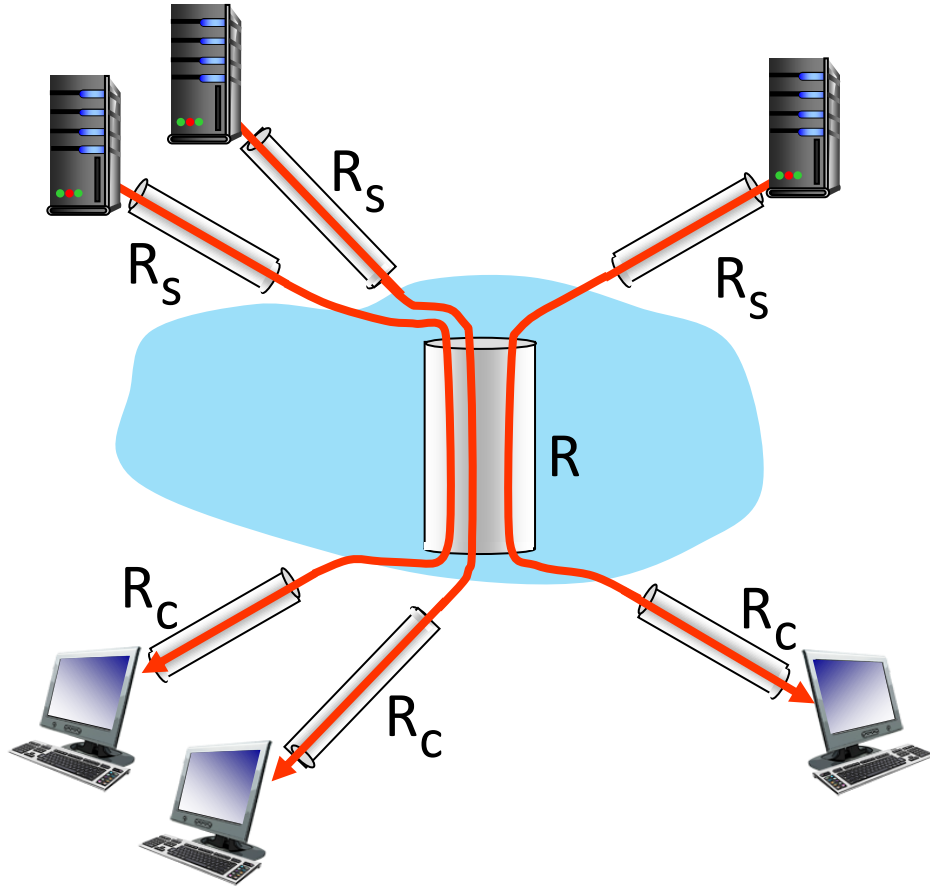
$R_s > R_c$ What is average end-end throughput?



bottleneck link

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

Throughput: network scenario



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

- per-connection end-end throughput:
 $\min(R_c, R_s, R/10)$
- in practice: R_c or R_s is often bottleneck

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

Chapter 1: roadmap

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- **Security**
- Protocol layers, service models
- History



Network security

- **field of network security:**
 - how bad guys can **attack computer networks**
 - 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**” 😊
 - security considerations in all layers!

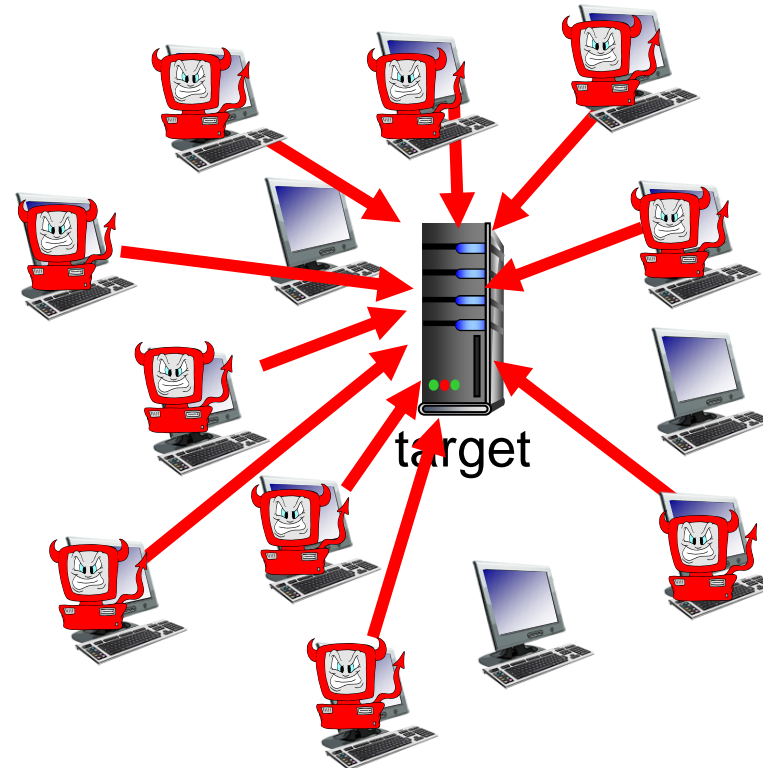
Bad guys: malware

- 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

Bad guys: denial of service

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

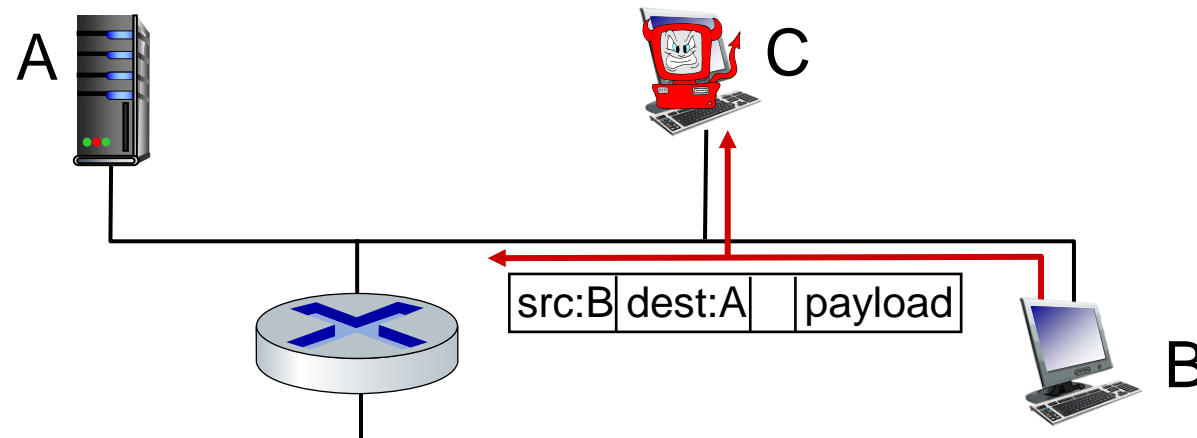
1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



Bad guys: packet interception

packet “sniffing”:

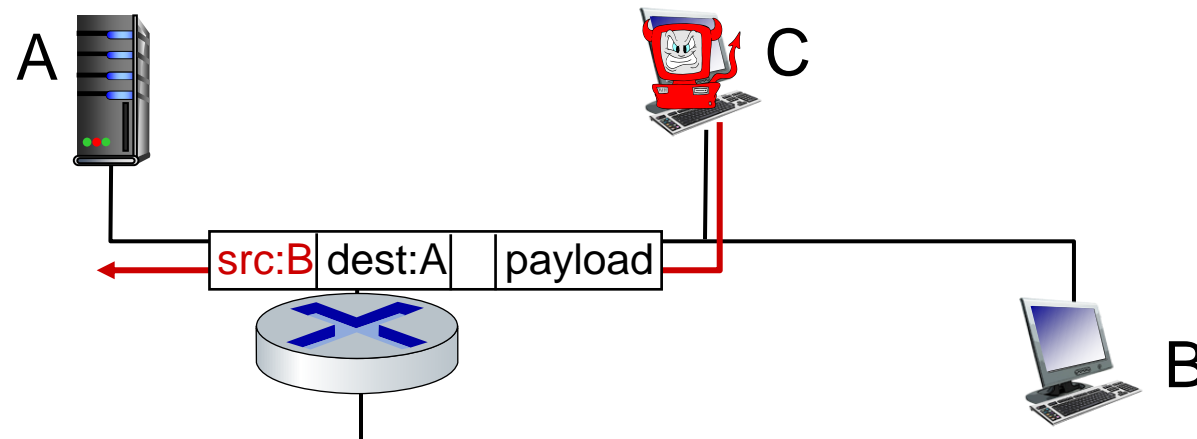
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



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

Bad guys: fake identity

IP spoofing: send packet with false source address



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

Chapter 1: roadmap

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History



Protocol “layers” and reference models

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

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

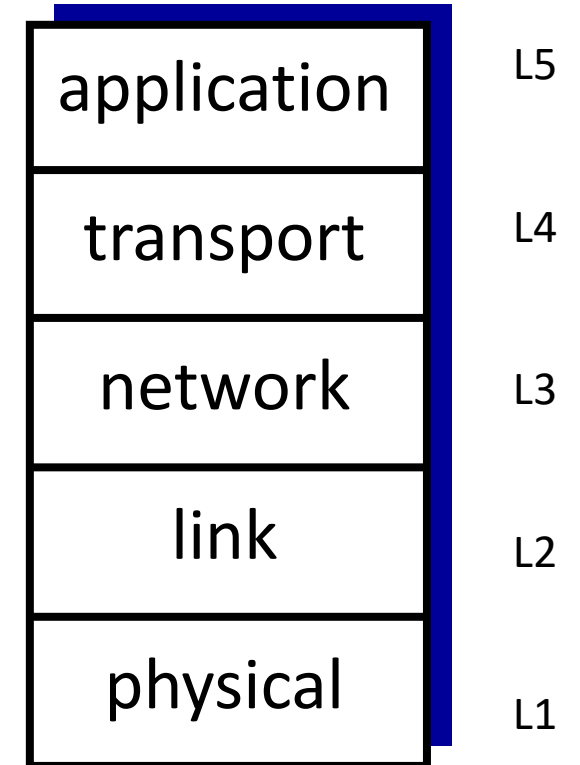
Question:

is there any hope of
organizing structure of
network?

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

Internet protocol stack

- **application:** supporting network applications
 - IMAP, 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, 802.11 (WiFi), PPP
- **physical:** bits “on the wire”

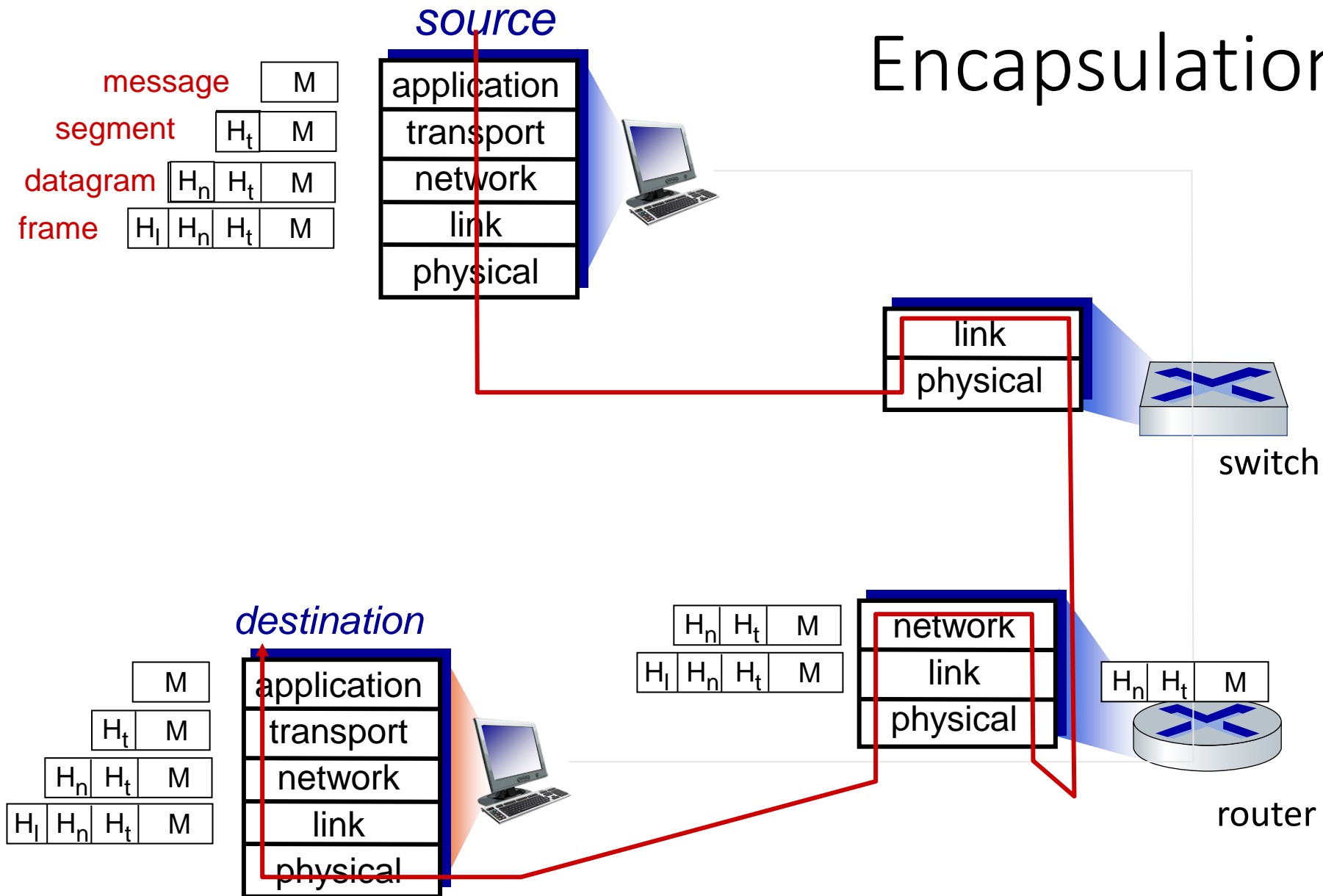


Why layering?

dealing with complex systems:

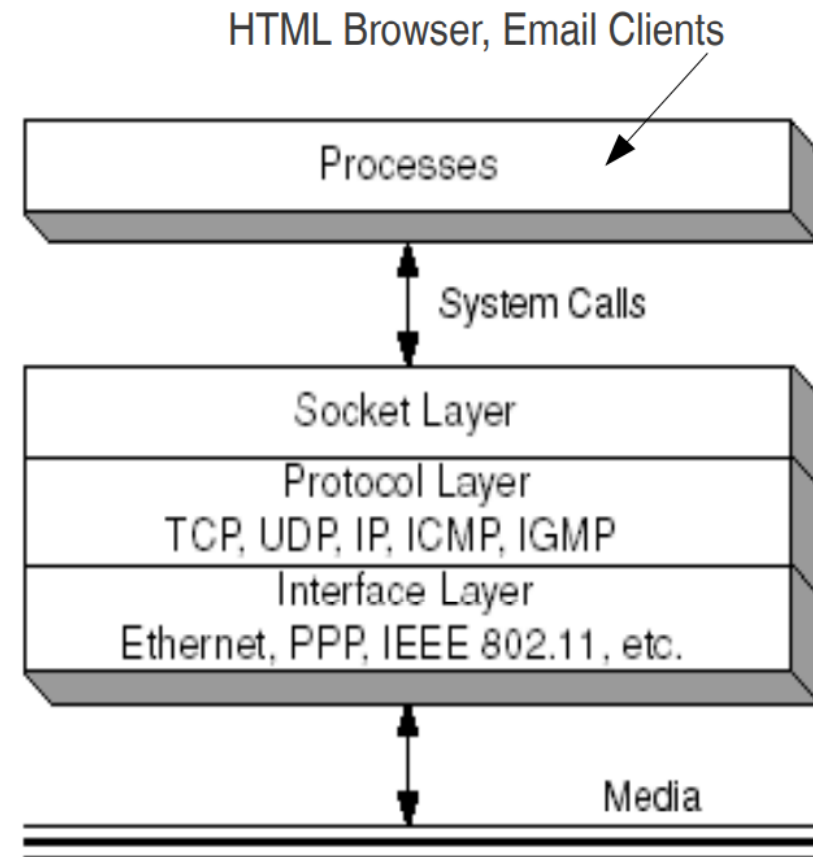
- explicit structure allows identification, relationship of complex system's pieces
 - layered *reference model* for discussion
- modularization eases maintenance, updating of system
 - change in *layer's service implementation*: transparent to rest of system
 - e.g., change in one layer procedure doesn't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

Encapsulation



Networking Code Organization

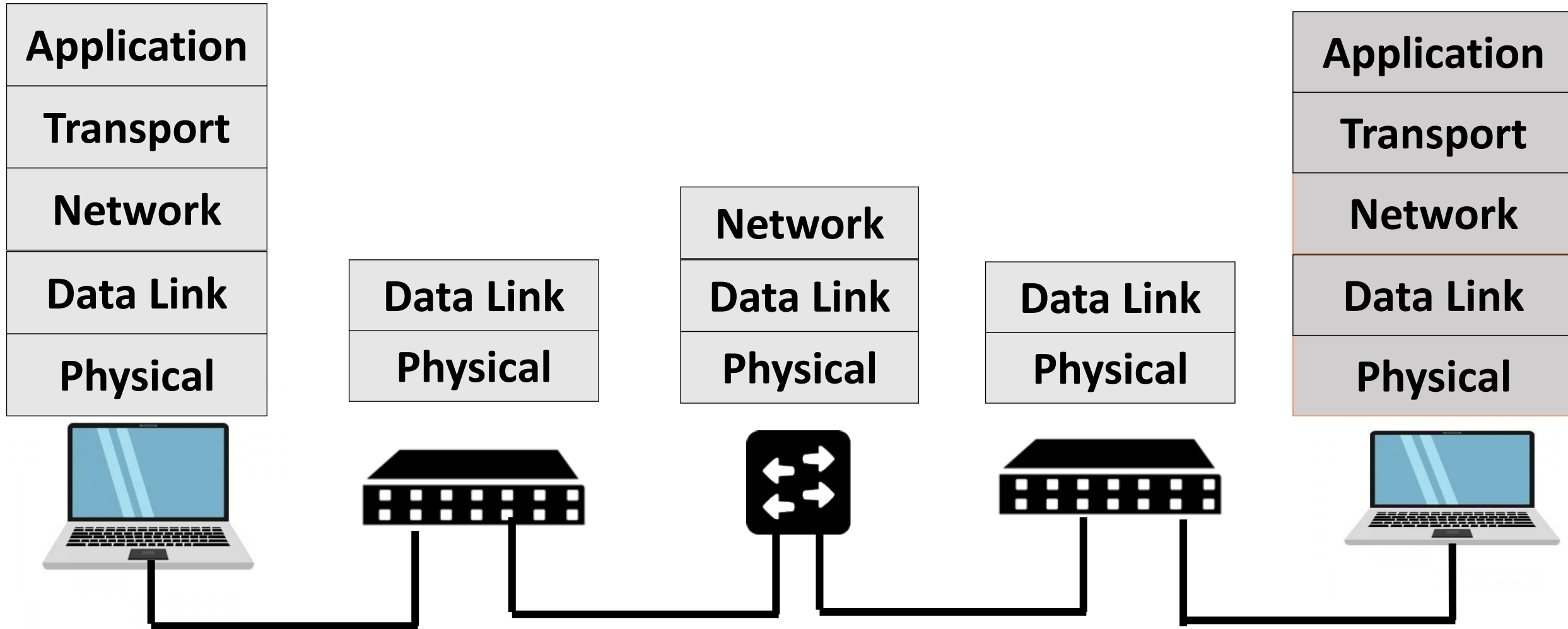
- Applications are mostly implemented as a user space processes.
- The network protocols are implemented in system kernel.
 - Socket layer
 - Protocol Layer
 - Interface layer



Network Configurations Files

- When a host is configured to **boot locally**, network configuration **parameters are stored in files**.
 - When the system boots up, **parameters are read from the files** and used to **configure the daemons** and the **network interface**.
 - A parameter may be changed by editing the corresponding configuration file.
- **Examples in Linux**
 - /etc/hosts
 - /etc/services
 - /etc/network/interfaces

Communication between two remote Machine



Top Down Vs Bottom Up Approach



Chapter 1: summary

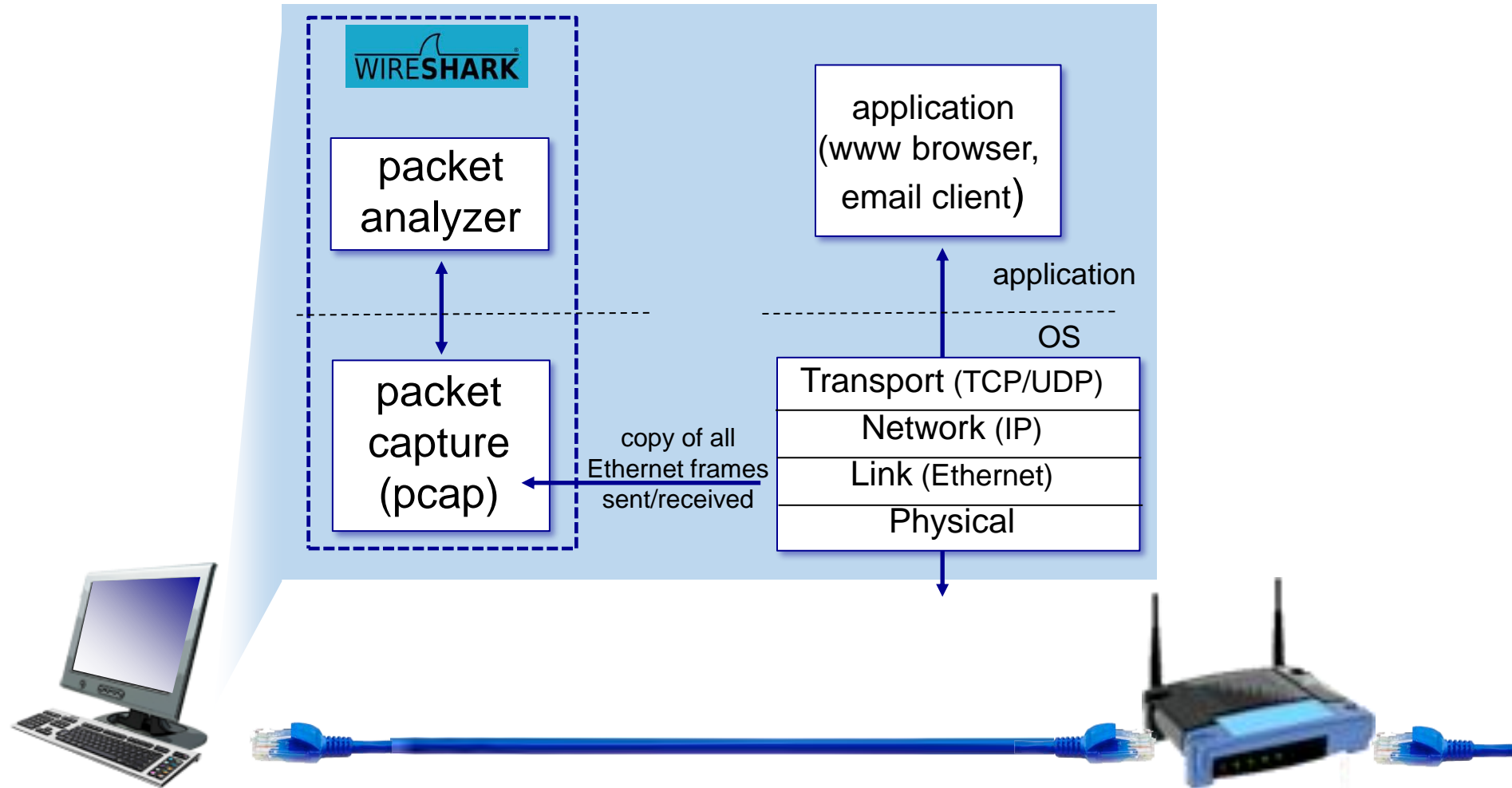
We've covered a “ton” of material!

- Internet overview
- what's a protocol?
- network edge, access network, core
 - packet-switching versus circuit-switching
 - Internet structure
- performance: loss, delay, throughput
- layering, service models
- security
- **History – You can read!**

You now have:

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

Wireshark



Assignment 0

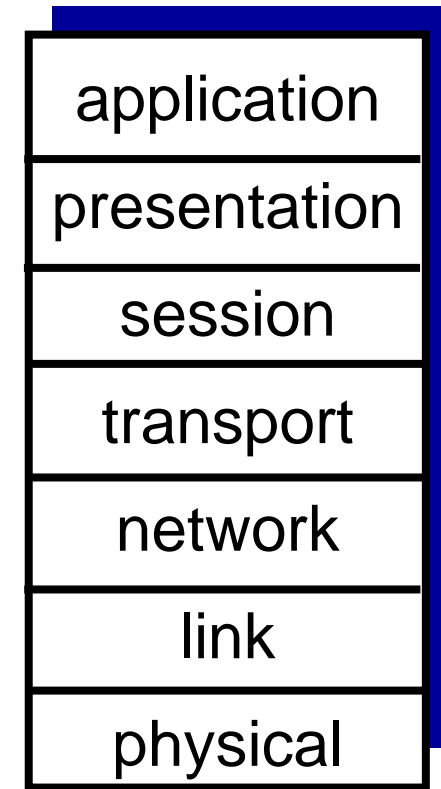
- Wireshark Installation

Additional Chapter 1 slides

ISO/OSI reference model

Two layers not found in Internet protocol stack!

- *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?



The seven layer OSI/ISO reference model