CNT 4704
Analysis of Computer Communication Networks

Computer Networks and the Internet (Cont'd)

Lecture-3

Mesut Ozdag, Ph.D.

Department of Computer Science

Fall 2024

HOUSEKEEPING & ACKNOWLEDGEMENT

This class session is
being recorded

- An ample portion of the material is derived/borrowed from Copyrighted ppt. slides of J.F. Kurose, K.W. Ross 1996-2020. All Rights Reserved.
- Original material can be found on: https://gaia.cs.umass.edu/kurose_ross/ppt.htm

CHAPTER 1: 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

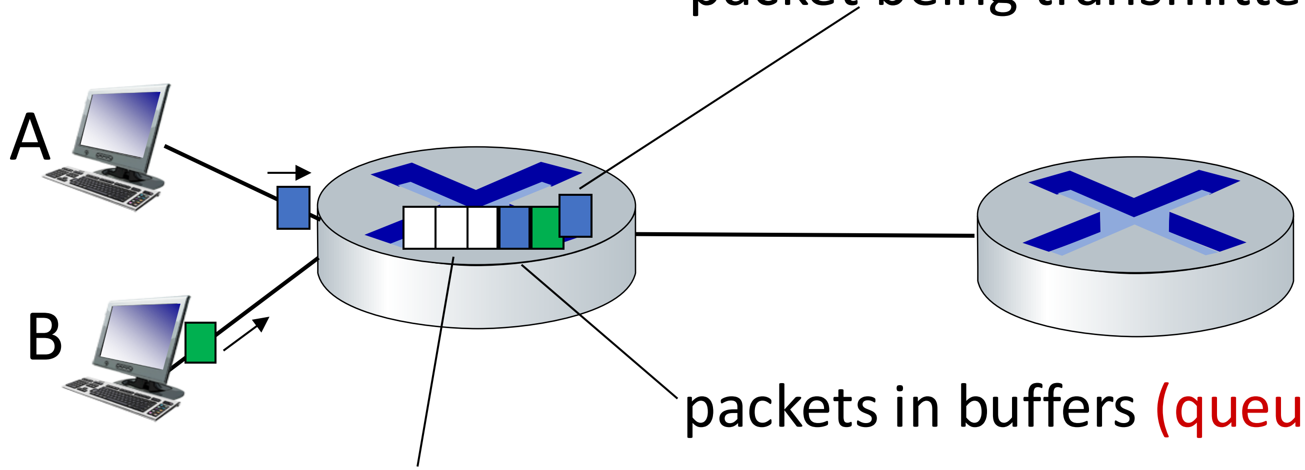
1.6 Networks under attack: security

1.7 History

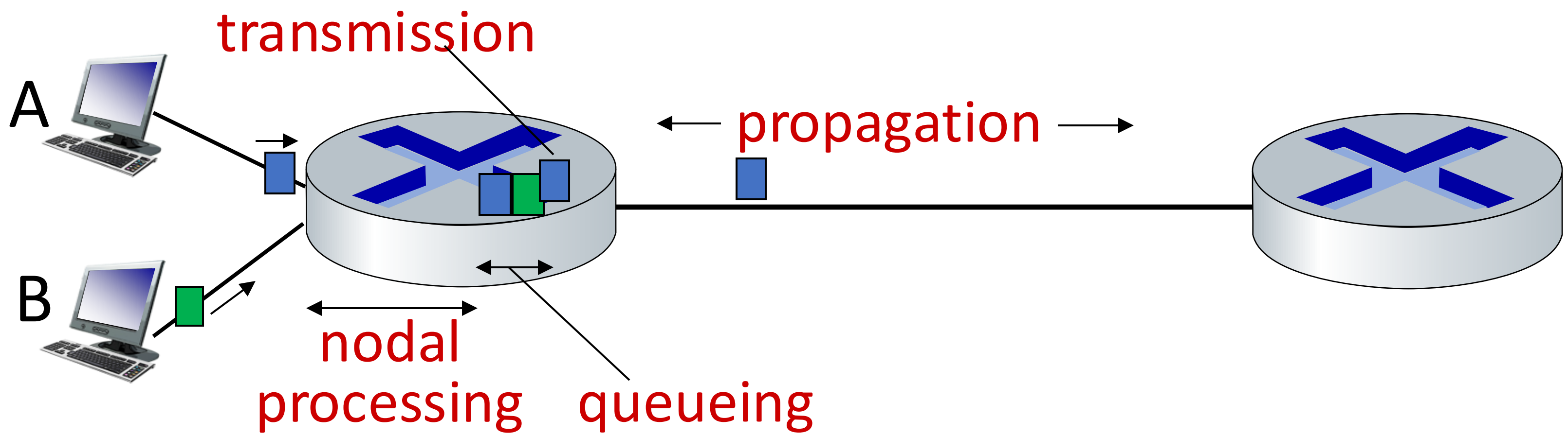
How do packet loss and delay occur?

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



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

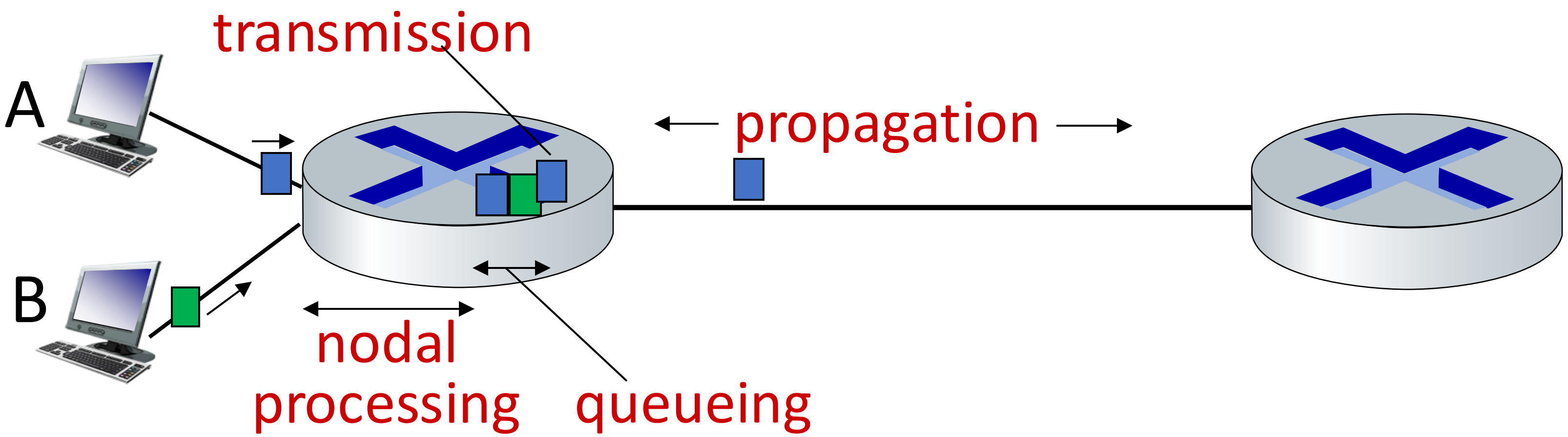
d_{proc}: nodal processing

- 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



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

d_{trans}: transmission delay:

- L: packet length (bits)
- R: link *transmission rate (bps)*
- $d_{trans} = L/R$

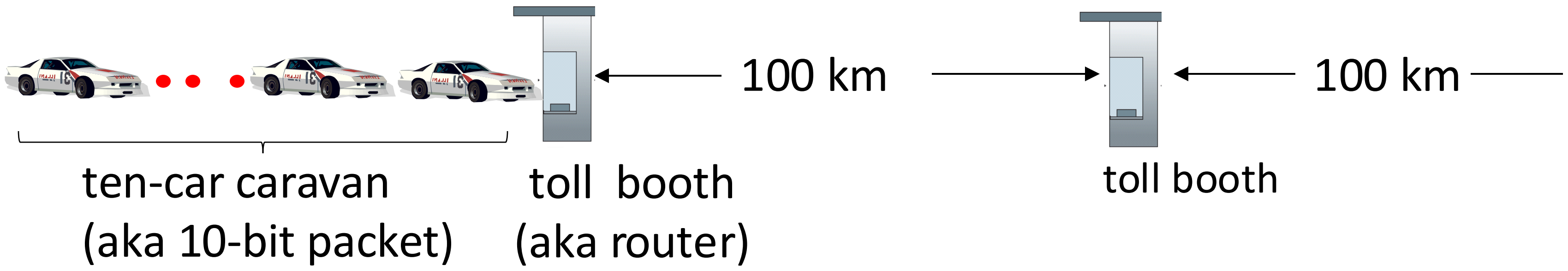
d_{prop}: propagation delay:

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

d_{trans} and d_{prop} *very* different

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

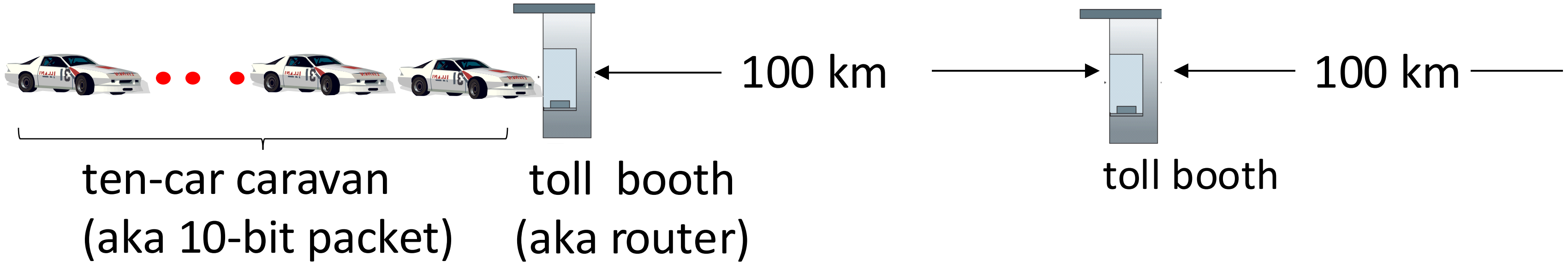
Caravan analogy



- Assumptions:
  - car ~ bit;
  - caravan ~ packet
  - Toll booth ~ link transmission
- toll booth takes 12 sec to service car (bit transmission time)
- cars "propagate" at 100 km/hr
- *Q:* How long until caravan is lined up before 2nd toll booth?

- time to "push" entire caravan through toll booth onto highway = 12sec*10cars = 120 sec (to transmit the caravan) (transmission time)
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr) = 1 hr (Propagation time)
- *A:* 62 minutes

Caravan analogy

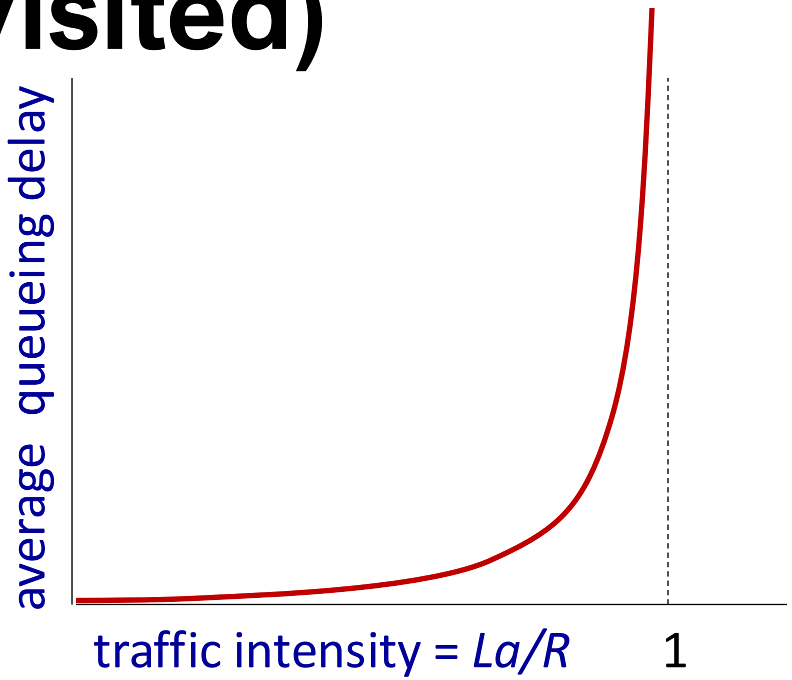


- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q: Will cars arrive to 2nd booth before all cars serviced at first booth?*

  A: Yes! after 7 min, first car arrives at second booth; three cars still at first booth

Packet queueing delay (revisited)

- *R:* link bandwidth (bps) (the transmission rate of bits)
- *L:* packet length (bits)
- *a:* average packet arrival rate
- *L . a == arrival rate of bits*



- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving is more than can be serviced - average delay infinite!

La/R ~ 0

La/R -> 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.

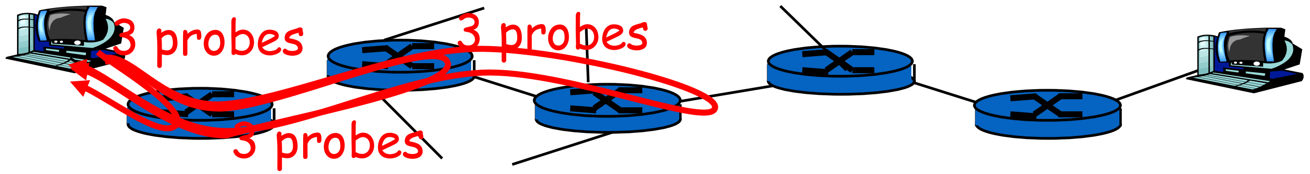
Running a Traceroute

Take the following steps to run a traceroute in Microsoft® Windows®:

1. Press Windows key + R to open the Run window.
2. Enter cmd and press Enter to open a Command Prompt.
3. Enter tracert, a space, then the IP address or web address for the destination site (for example: tracert www.lexis.com).
4. Press Enter.

Take the following steps to run a traceroute in Mac OS®:

1. Click the Spotlight icon in the Menu bar.
   Note: The icon looks like a magnifying glass.
2. Enter Network Utility into the Spotlight Search field.
3. Double-click Network Utility from the Top Hit list.
4. Click the Traceroute tab.
5. Enter the IP address or web address for the destination site and click Trace.



"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

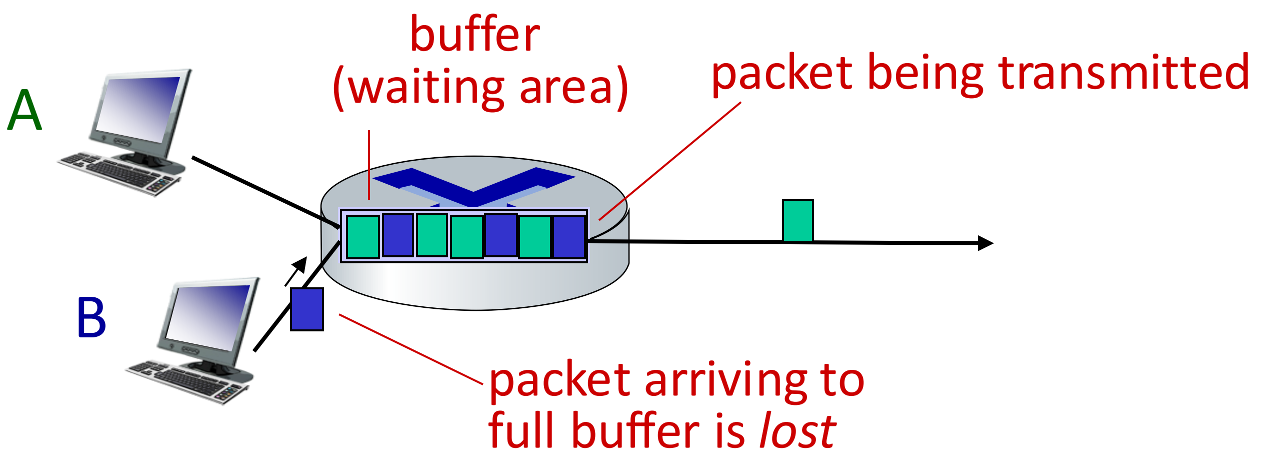
trans-oceanic link

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

Under Windows is "tracert"

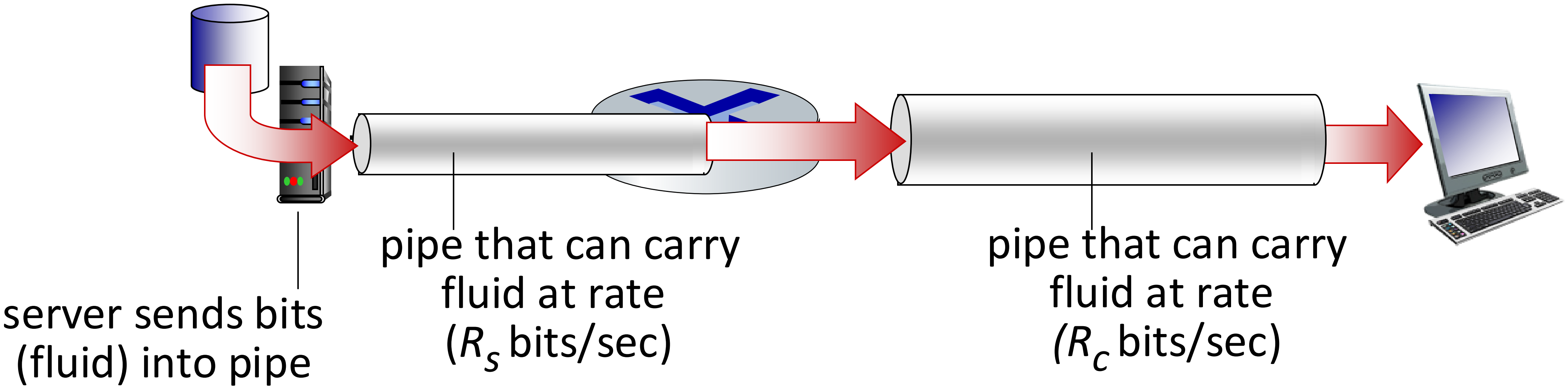
Packet loss

- 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



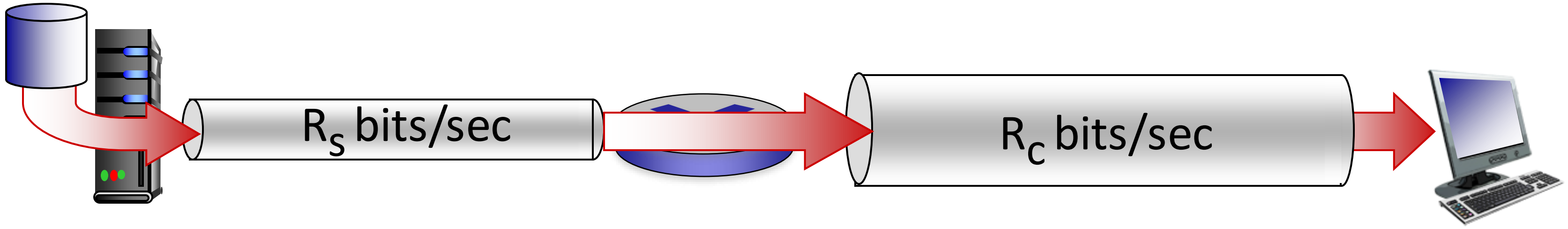
Throughput

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

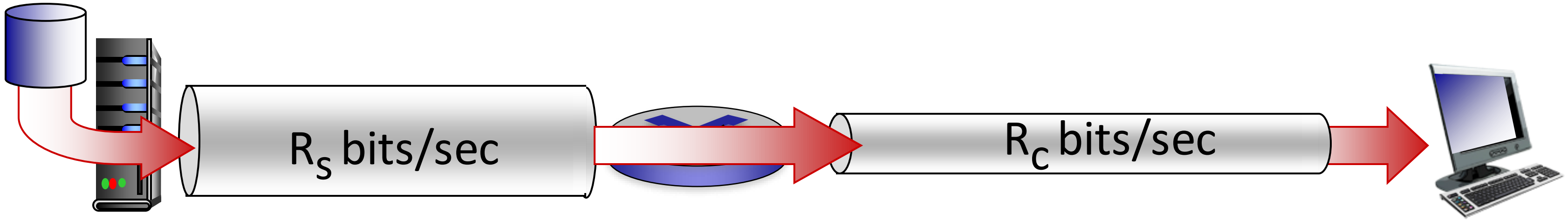


Throughput

$R_{server} < R_{client}$ What is average end-end throughput?



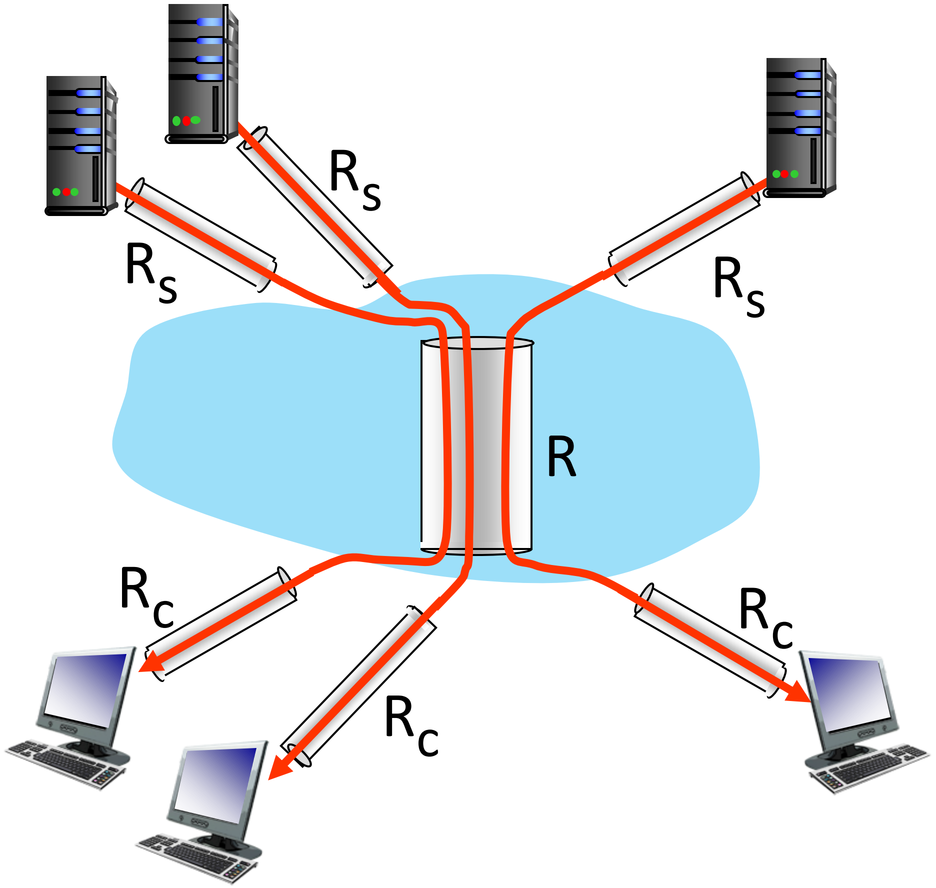
$R_s > R_c$ What is average end-end throughput?



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
- Ex.: Rs = 2Mbps, Rc=1Mbps, R=5Mbps

Throughput = 5/10Mbps = 500 Kbps

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

CHAPTER 1: 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

1.6 Networks under attack: security

1.7 History

Protocol "Layers"

Networks are complex, with many "pieces":

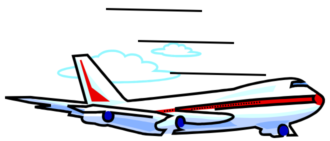
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

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

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

Example: organization of air travel



ticket (purchase)	ticket (complain)
baggage (check)	baggage (claim)
gates (load)	gates (unload)
runway takeoff	runway landing
airplane routing	airplane routing

airplane routing

airline travel: a series of steps, involving many services

Example: organization of air travel

ticket (purchase)	*ticketing service*	ticket (complain)
baggage (check)	*baggage service*	baggage (claim)
gates (load)	*gate service*	gates (unload)
runway takeoff	*runway service*	runway landing
airplane routing	*routing service*	airplane routing

layers: each layer implements a service

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

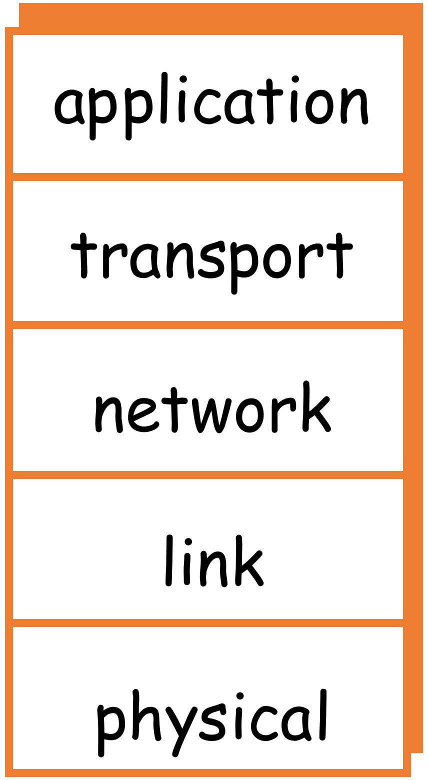
Why layering?

Dealing with complex systems:

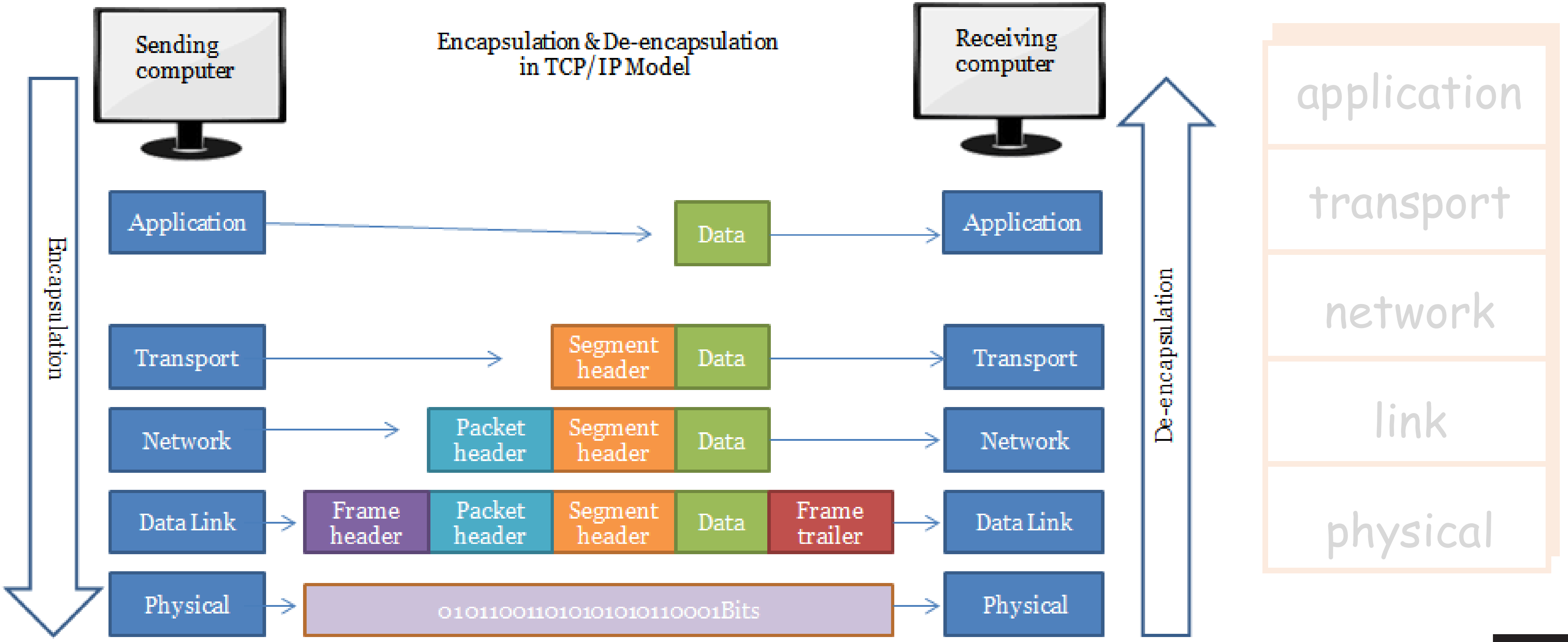
- explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change in layer's service *implementation*: transparent to rest of system
  - e.g., change in gate procedure doesn't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

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”

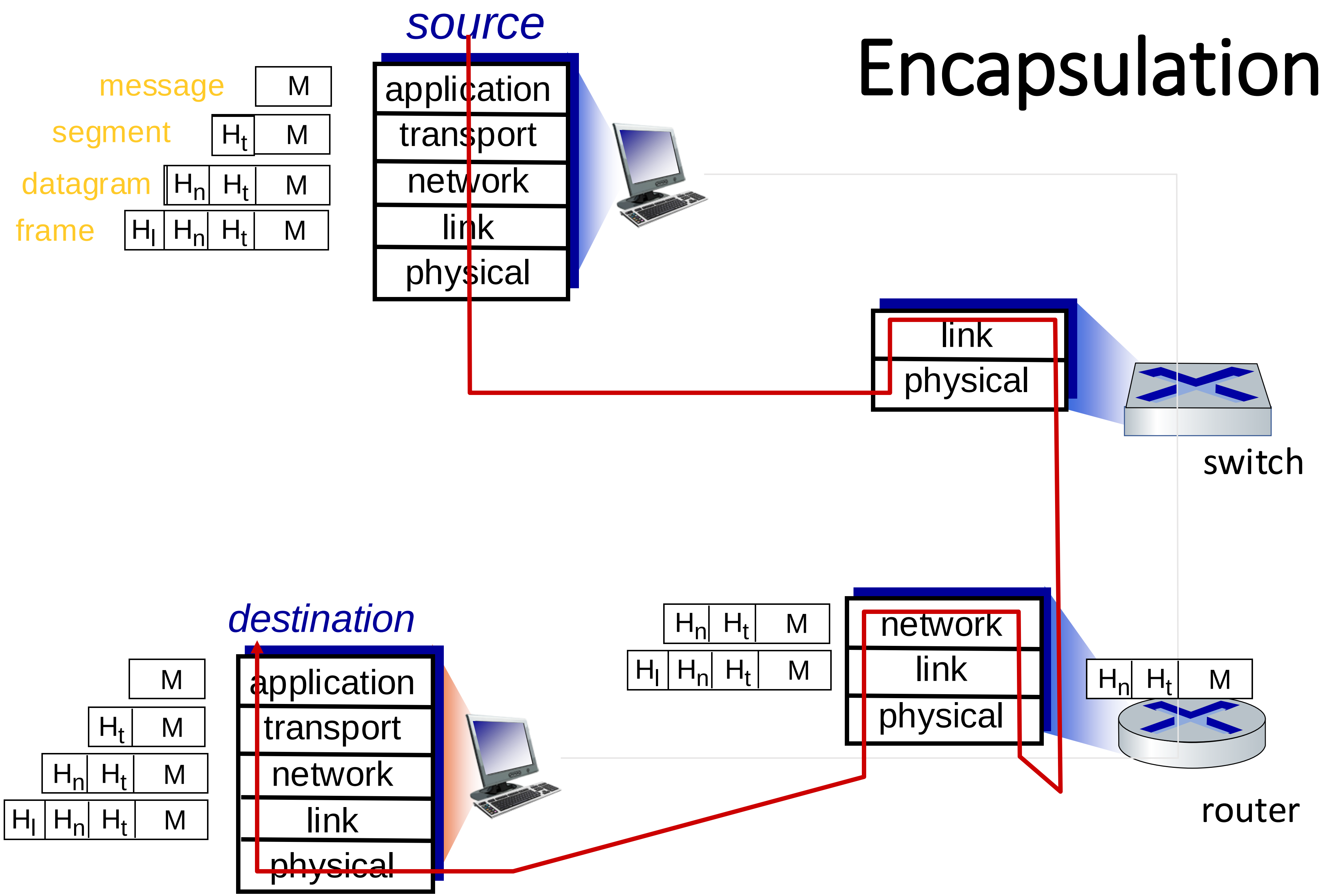


Servicing, layering and encapsulation



Courtesy of https://embeddedgeeks.com

Encapsulation



CHAPTER 1: 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

1.6 Networks under attack: security

1.7 History

Network Security

- Attacks on Internet infrastructure:
  - infecting/attacking hosts: malware, spyware, worms, unauthorized access (data stealing, user accounts)
  - denial of service: deny access to resources (servers, link bandwidth)

    - 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” ☺
  - Internet protocol designers playing “catch-up”
  - Security considerations in all layers!

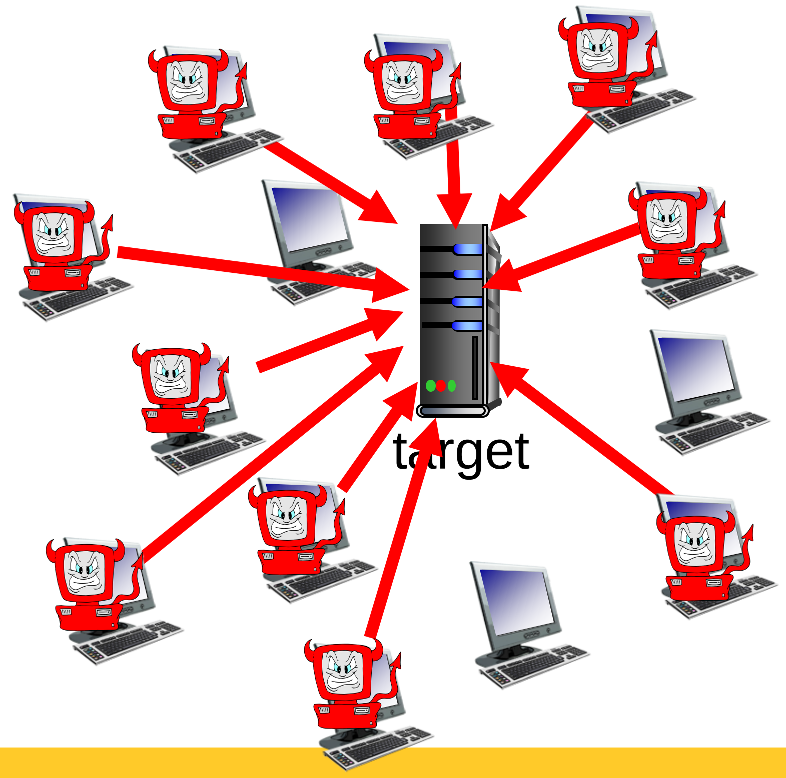
What can bad guys do: malware?

- **Spyware:**
  - infection by downloading web page with spyware
  - records keystrokes, web sites visited, upload info to collection site
- **Virus**
  - infection by receiving object (e.g., e-mail attachment), actively executing
  - self-replicating: propagate itself to other hosts, users
- **Worm:**
  - infection by passively receiving object that gets itself executed
  - self- replicating: propagates to other hosts, users
- infected host can be enrolled in **botnet**, used for spam or distributed denial of service (DDoS) attacks

Denial of service attacks

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

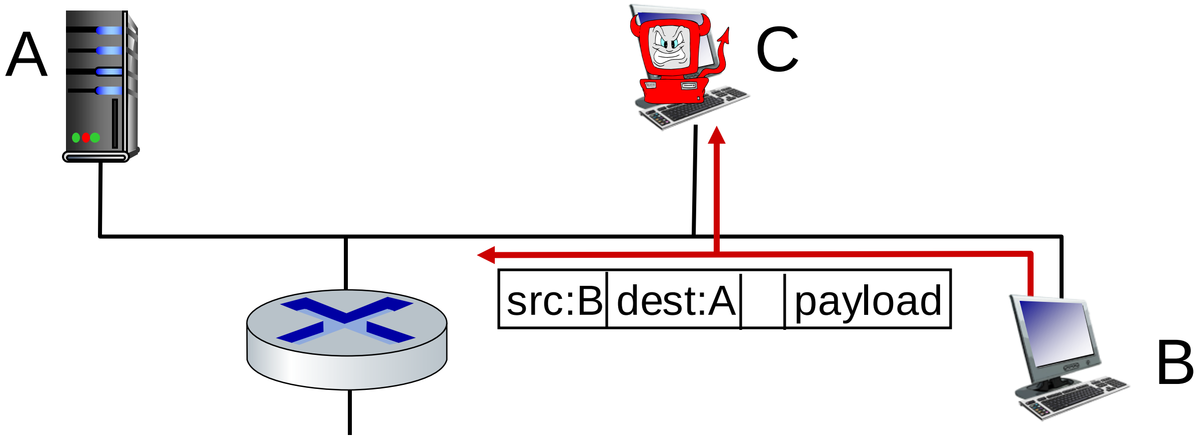
1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



Sniff, modify, delete your packets

Packet sniffing:

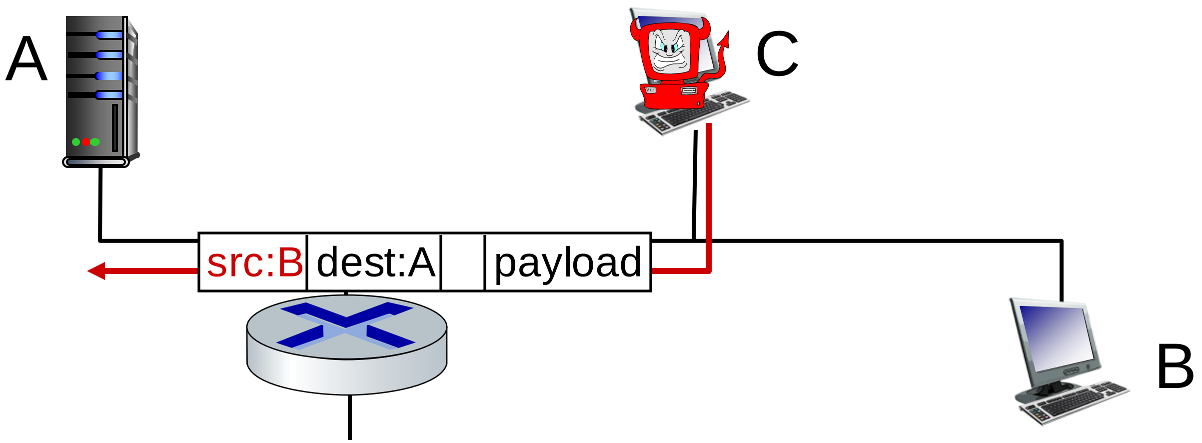
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



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

Masquerade as you

- *IP spoofing:* send packet with false source address



… lots more on security (throughout, Chapter 8)

CHAPTER 1: 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

1.6 Networks under attack: security

1.7 History

Internet history

Broken down to 5 main epochs:

1961-1972: Early packet-switching principles

1972-1980: Internetworking, new and proprietary nets

1980-1990: new protocols, a proliferation of networks

1990, 2000s: commercialization, the Web, new applications

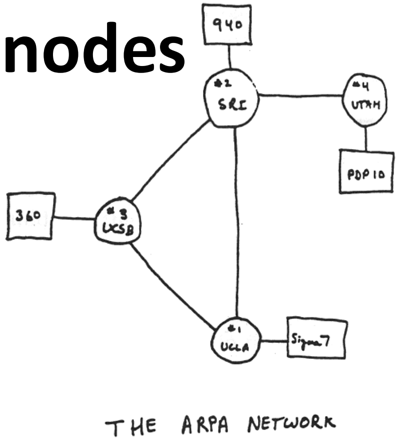
2005-present: more new applications, Internet is "everywhere"

Internet history

1961-1972: Early packet-switching principles

- 1961: **Kleinrock** - queueing theory shows effectiveness of packet-switching (Grad. Student published paper)
- 1964: Baran - packet-switching in **military nets**
- 1967: ARPAnet conceived by Advanced Research Projects Agency (world 1st Packet-switching network, the Internet's ancestor)
- 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**



Internet history

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 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 routing
- decentralized control

define today's Internet architecture

Internet history

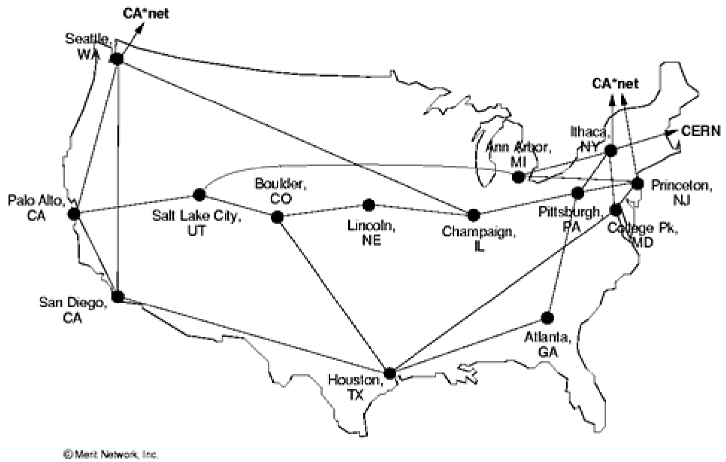
1980-1990: new protocols, a proliferation of networks

- 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

Next Chapter

- new national networks: CSnet, BITnet, NSFnet, Minitel
- **100,000 hosts** connected to confederation of networks

NSFNET T1 Network 1991



Internet history

1990, 2000s: commercialization, the Web, new applications

- early 1990s: 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 1990s: commercialization of the Web

late 1990s – 2000s:

- 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

2005-present: more new applications, Internet is "everywhere"

- ~18B devices attached to Internet (2017)
  - rise of smartphones (iPhone: 2007)
- aggressive deployment of broadband access
- increasing ubiquity of high-speed wireless access: 4G/5G, WiFi
- emergence of online social networks:
- Facebook: ~ 2.5 billion users
- service providers (Google, FB, Microsoft) create their own networks
  - bypass commercial Internet to connect "close" to end user, providing "instantaneous" access to search, video content, …
- enterprises run their services in "cloud" (e.g., Amazon Web Services, Microsoft Azure)

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 now have:

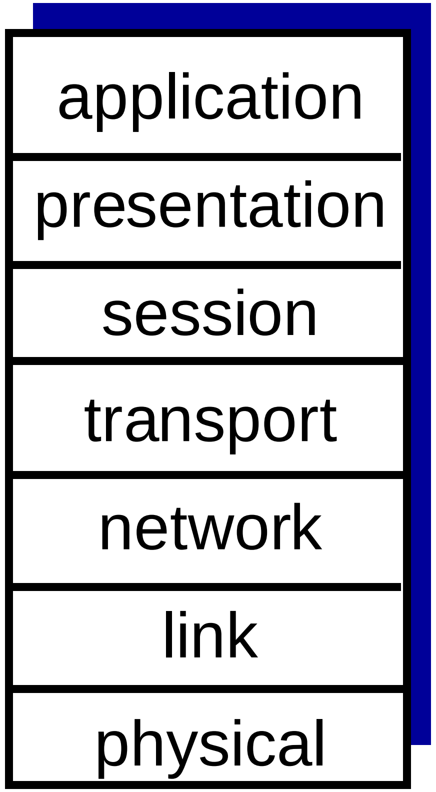
- context, overview, vocabulary, “feel” of networking
- more depth, detail, *and fun* to follow!

Additional Chapter 1 slides

ISO/OSI reference model

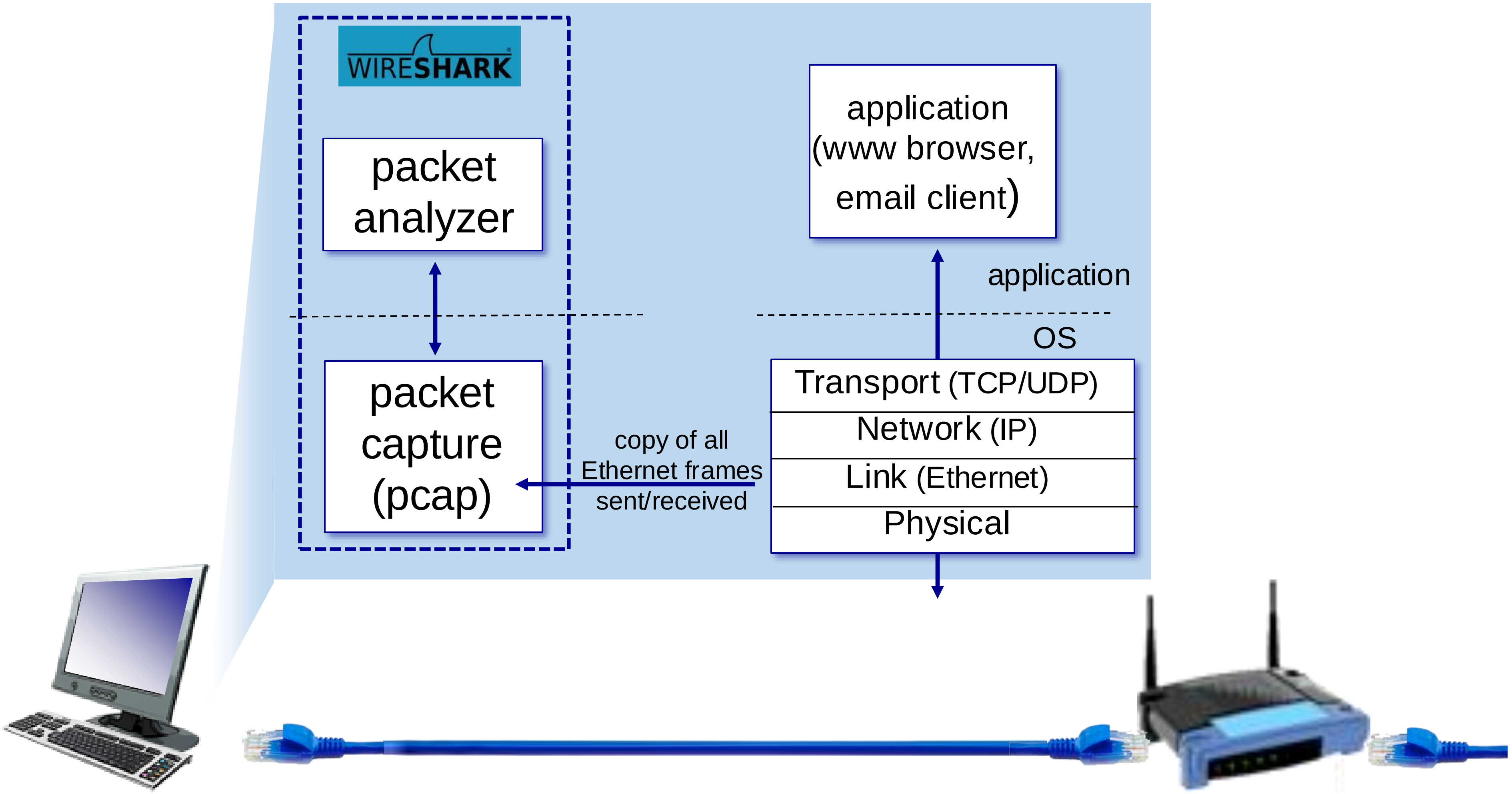
Two layers not found in Internet protocol stack!

- *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

Wireshark


WIRESHARK
packet analyzer
packet capture (pcap)
copy of all Ethernet frames sent/received
application (www browser, email client)
application
OS
Transport (TCP/UDP)
Network (IP)
Link (Ethernet)
Physical

Wireshark introduction ..

- Check the view drop-down for interface personalization
- Time, source, IP or MAC, Protocol, Info (can be the most important depending on your application)
- Filters:
  - It turns green on valid filters upon writing
  - Looking for a certain protocol packets, i.e. tcp, http
  - It can be case sensitive in some commands that required text

Wireshark introduction ..

- Filters examples:
  - Tcp, http, … lists packts related to the said protocol
  - http.request.method == "GET“
  - ip.addr == ‘ip you’re looking for’
  - Check the “Hypertext Transfer Protocol” section

Questions?