بسمه تعالی

School of Computer Engineering

Advanced Computer Networks

Introduction: Basics of Computer Networks

Part 4

Seyed Hamed Rastegar

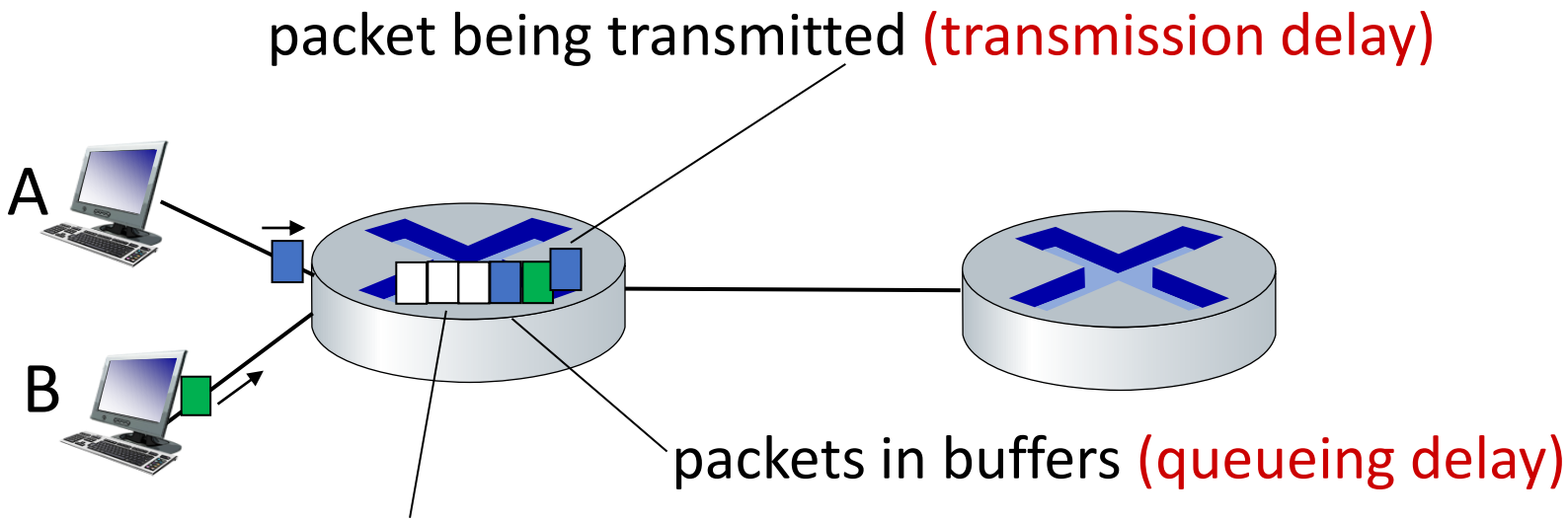
Fall 1401

Chapter 1: roadmap

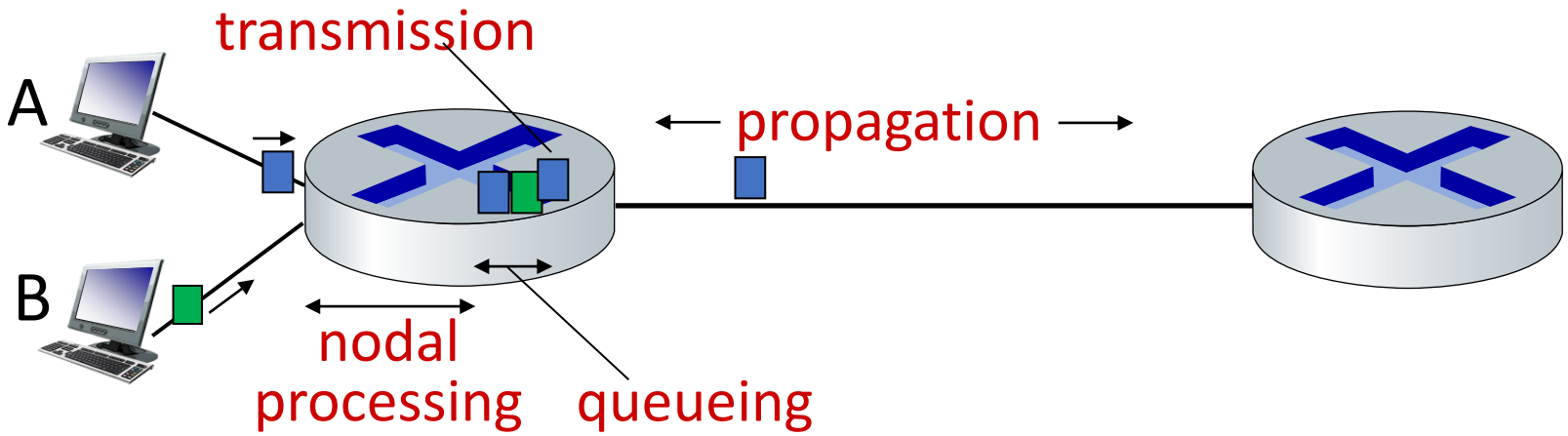
- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

How do packet delay and loss occur?

- packets *queue* in router buffers, waiting for turn for transmission
  - queue length grows when arrival rate to link (temporarily) exceeds output link capacity
- packet *loss* occurs when memory to hold queued packets fills up



Packet delay: four sources



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

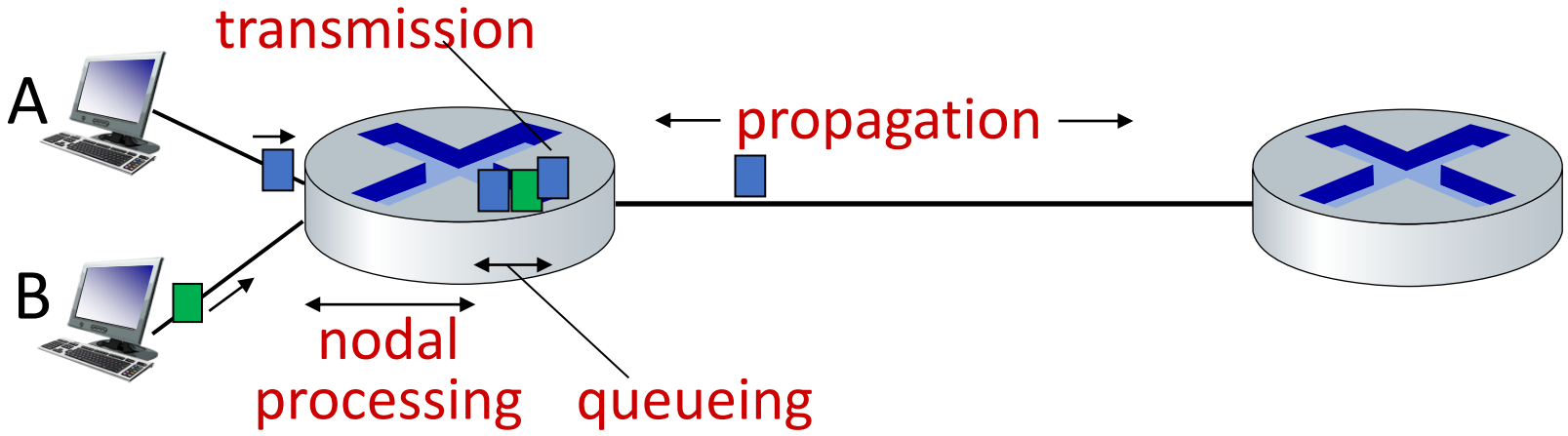
d_{proc}: nodal processing

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

d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router
- On the order of microseconds to milliseconds in practice

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)*
- $d_{trans} = L/R$

On the order of microseconds to milliseconds in practice

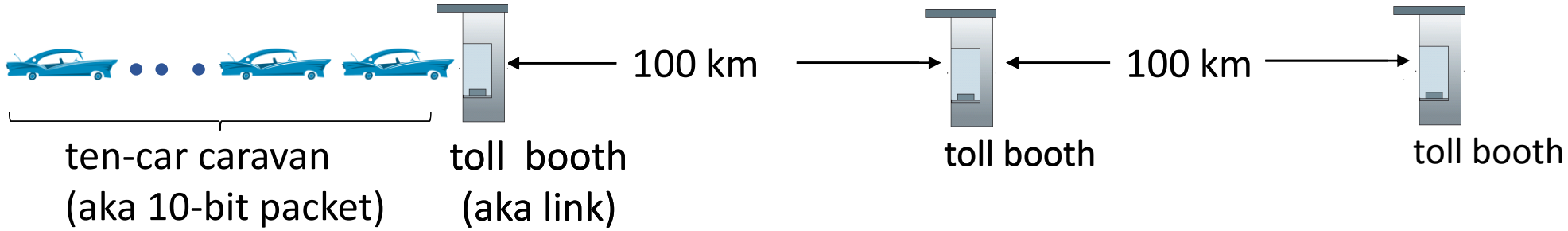
d_{prop}: propagation delay:

- d: length of physical link
- s: propagation speed (~$2x10^8$ m/sec)
- $d_{prop} = d/s$

In wide area networks, it is on the order of milliseconds

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

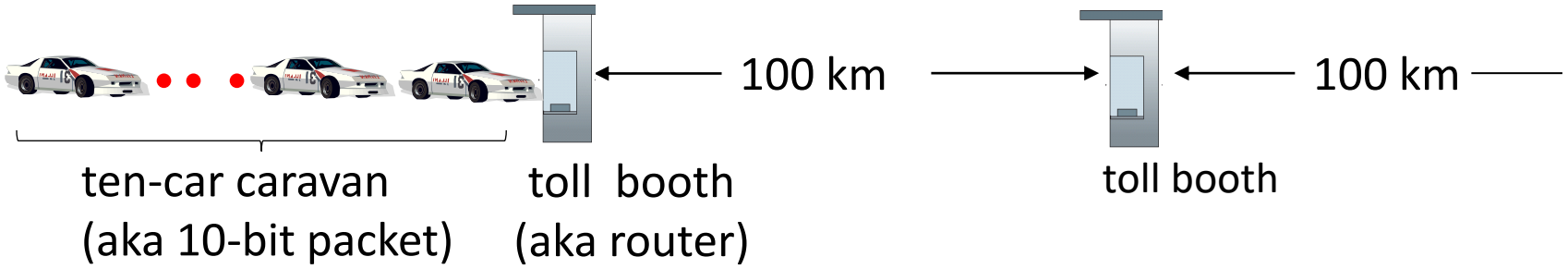
Caravan analogy



- car ~ bit; caravan ~ packet; toll service ~ link transmission
- toll booth takes 12 sec to service car (bit transmission time)
- "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 = 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



- 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

➢ This situation also arises in packet-switched networks—the first bits in a packet can arrive at a router while many of the remaining bits in the packet are still waiting to be transmitted by the preceding router.

Packet delay: four sources

- Applet: Propagation vs Transmission

Transmission versus Propagation Delay Interactive Animation (pearsoncmg.com)

Packet Nodal Delay: Remarks

- **The contribution of these delay components can vary significantly.**

- For example, d_{prop} can be negligible (for example, a couple of microseconds) for a link connecting two routers on the same university campus;
  - however, d_{prop} is hundreds of milliseconds for two routers interconnected by a geostationary satellite link, and can be the dominant term in d nodal.

- Similarly, d_{trans} can range from negligible to significant. Its contribution is typically negligible for transmission rates of 10 Mbps and higher (for example, for LANs);
  - however, it can be hundreds of milliseconds for large Internet packets sent over low-speed dial-up modem links.

- The processing delay, d_{proc} , is often negligible; however, it strongly influences a router's maximum throughput, which is the maximum rate at which a router can forward packets.

Packet queueing delay (revisited)

- The most complicated and interesting component of nodal delay is the queuing delay, d_{queue}.

- Unlike the other three delays, the queuing delay can vary from packet to packet.

  - For example, if 10 packets arrive at an empty queue at the same time, the first packet transmitted will suffer no queuing delay, while the last packet transmitted will suffer a relatively large queuing delay (while it waits for the other nine packets to be transmitted).

  - Therefore, when characterizing queuing delay, one typically uses statistical measures, such as average queuing delay, variance of queuing delay, and the probability that the queuing delay exceeds some specified value.

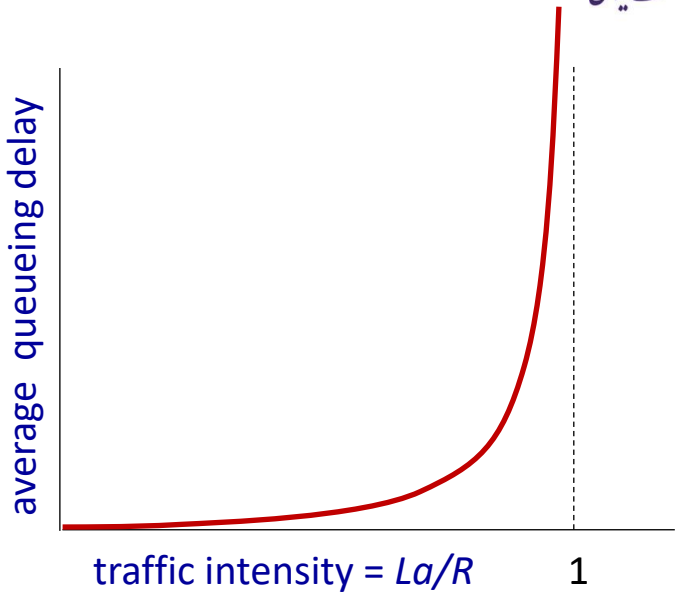
Packet queueing delay (revisited)

- When is the queuing delay large and when is it insignificant?
  - The answer to this question depends on
    - The rate at which traffic arrives at the queue,
    - The transmission rate of the link,
    - and the nature of the arriving traffic, that is, whether the traffic arrives periodically or arrives in bursts (and its randomness).
- So let's take a brief look at it.

Packet queueing delay (revisited)

- *a:* average packet arrival rate
- *L:* packet length (bits)
- *R:* link bandwidth (bit transmission rate)

$$\frac{L \cdot a}{R} : \frac{\text{arrival rate of bits}}{\text{service rate of bits}}$$ *"traffic intensity"*



- *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

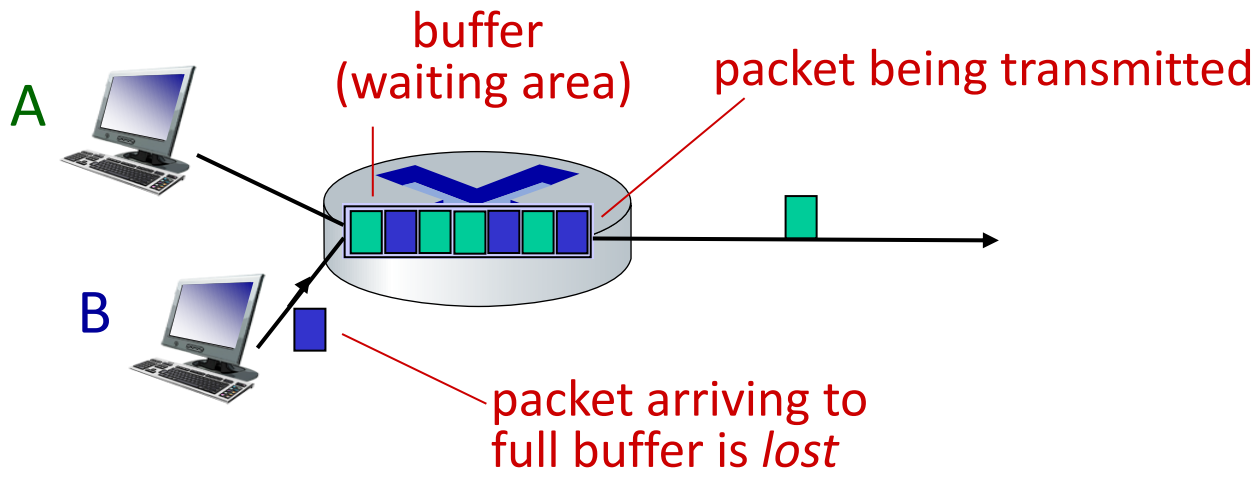
La/R -> 1

Packet Loss

- In our discussions above, we have assumed that the queue is capable of holding an infinite number of packets.
- In reality a queue preceding a link has finite capacity, although the queuing capacity greatly depends on the router design and cost.
- Because the queue capacity is finite, packet delays do not really approach infinity as the traffic intensity approaches 1.
- Instead, a packet can arrive to find a full queue. With no place to store such a packet, a router will drop that packet; that is, the packet will be lost.

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



Packet Loss

- From an end-system viewpoint, a packet loss will look like a packet having been transmitted into the network core but never emerging from the network at the destination.
- The fraction of lost packets increases as the traffic intensity increases.
- Therefore, performance at a node is often measured not only in terms of delay, but also in terms of the **probability of packet loss**.
- A lost packet may be retransmitted on an end-to-end basis in order to ensure that all data are eventually transferred from source to destination

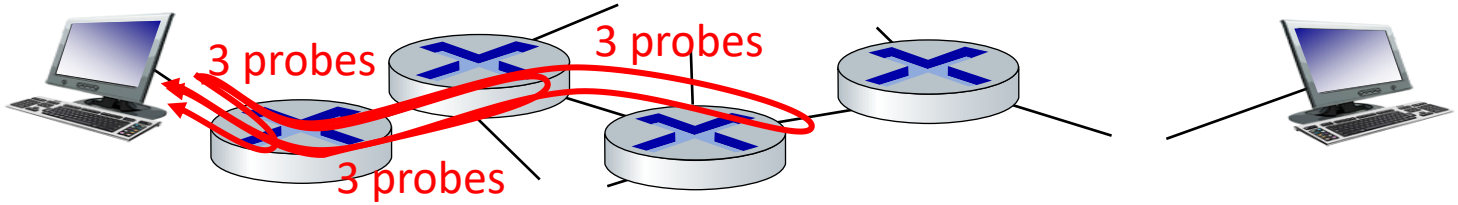
How do packet delay and loss occur?

Applet: Queueing and Loss

Queuing and Loss Interactive Animation (pearsoncmg.com)

“Real” Internet delays and routes

- what do “real” Internet delay & loss look like?
- **traceroute** program (RFC 1393) : 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

3 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
```

Real Internet delays and routes

traceroute: Another example

```
C:\Users\Hamed>tracert ece.ut.ac.ir

Tracing route to ece.ut.ac.ir [80.66.179.158]
over a maximum of 30 hops:

  1    27 ms    14 ms    11 ms  aaa-computer.ipm.ir [192.168.134.2]
  2    16 ms    19 ms    33 ms  192.168.222.2
  3    15 ms    17 ms    15 ms  192.168.222.1
  4    18 ms    20 ms    16 ms  192.168.63.5
  5    12 ms    18 ms    24 ms  80.66.176.237
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12    28 ms    13 ms    17 ms  science.ut.ac.ir [80.66.179.158]

Trace complete.
```

```
C:\Users\Hamed>tracert cs.ipm.ir

Tracing route to cs.ipm.ir [94.184.211.118]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  192.168.1.1
  2    26 ms    27 ms    27 ms  172.19.6.100
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8    42 ms    40 ms    42 ms  85.15.4.98
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11    42 ms   104 ms    55 ms  94.184.211.118

Trace complete.
```

Real Internet delays and routes

traceroute: Another example

```
C:\Users\Hamed>tracert cs.umass.edu

Tracing route to cs.umass.edu [128.119.240.136]
over a maximum of 30 hops:

  1     1 ms     2 ms     1 ms  aaa-computer.ipm.ir [192.168.134.2]
  2     5 ms     2 ms     2 ms  192.168.222.2
  3    13 ms    26 ms    21 ms  192.168.222.1
  4    23 ms    19 ms    16 ms  192.168.63.1
  5     4 ms     5 ms     5 ms  nia-sw-150-20.ipm.edge-2.iranet.ir [194.225.150.20]
  6    17 ms    12 ms    26 ms  l-1-v101.lct-ro-151-1.ipm.edge-1.iranet.ir [194.225.151.1]
  7   121 ms   103 ms   124 ms  185.233.140.65
  8     *        *        *     Request timed out.
  9   111 ms   110 ms   117 ms  hu0-0-0-2.br03.frf05.pccwbtn.net [63.218.230.45]
 10   108 ms   103 ms   104 ms  hu0-0-0-2.br03.frf05.pccwbtn.net [63.218.230.45]
 11     *        *        *     Request timed out.
 12   110 ms   134 ms   117 ms  be2814.ccr42.ams03.atlas.cogentco.com [130.117.0.141]
 13   200 ms   202 ms   203 ms  be12266.ccr42.par01.atlas.cogentco.com [154.54.56.174]
 14   211 ms   201 ms   200 ms  be2318.ccr32.bio02.atlas.cogentco.com [154.54.61.117]
 15   250 ms   202 ms   207 ms  be2332.ccr42.dca01.atlas.cogentco.com [154.54.85.245]
 16   202 ms   202 ms   205 ms  be2807.ccr42.jfk02.atlas.cogentco.com [154.54.40.109]
 17   210 ms   203 ms   202 ms  be3472.ccr32.bos01.atlas.cogentco.com [154.54.46.33]
 18   203 ms   212 ms   207 ms  be2731.rcr51.orh01.atlas.cogentco.com [154.54.41.130]
 19   265 ms   270 ms   278 ms  38.104.218.14
 20   262 ms   261 ms   263 ms  69.16.0.8
 21   266 ms   267 ms   267 ms  69.16.1.0
 22   282 ms   270 ms   270 ms  core1-rt-et-8-3-0.gw.umass.edu [192.80.83.109]
 23   277 ms   264 ms   264 ms  n5-rt-1-1-et-0-0-0.gw.umass.edu [128.119.0.8]
 24   262 ms   261 ms   263 ms  cics-rt-xe-0-0-0.gw.umass.edu [128.119.3.32]
 25   262 ms   266 ms   264 ms  mailsrv.cs.umass.edu [128.119.240.136]

Trace complete.
```

In Iran

Leaving Iran

PCC Global ISP (In Europe)

Cogent Comm. ISP (In North America)

UMass

Other types of delays

- In addition to processing, transmission, and propagation delays, there can be additional significant delays in the end systems.

- For example, an **end system** wanting to transmit a packet into a shared medium (e.g., as in a WiFi or cable modem scenario) may purposefully delay its transmission as part of its protocol for sharing the medium with other end systems.

- Another important delay is media **packetization delay**, which is present in Voice-over-IP (VoIP) applications.
    - In VoIP, the sending side must first fill a packet with encoded digitized speech before passing the packet to the Internet. This time to fill a packet—called the packetization delay—can be significant and can impact the user perceived quality of a VoIP call.

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

- If the file consists of *F* bits and the transfer takes *T* seconds for Host B to receive all *F* bits, then **the average throughput** of the file transfer is *F*/*T* bits/sec.

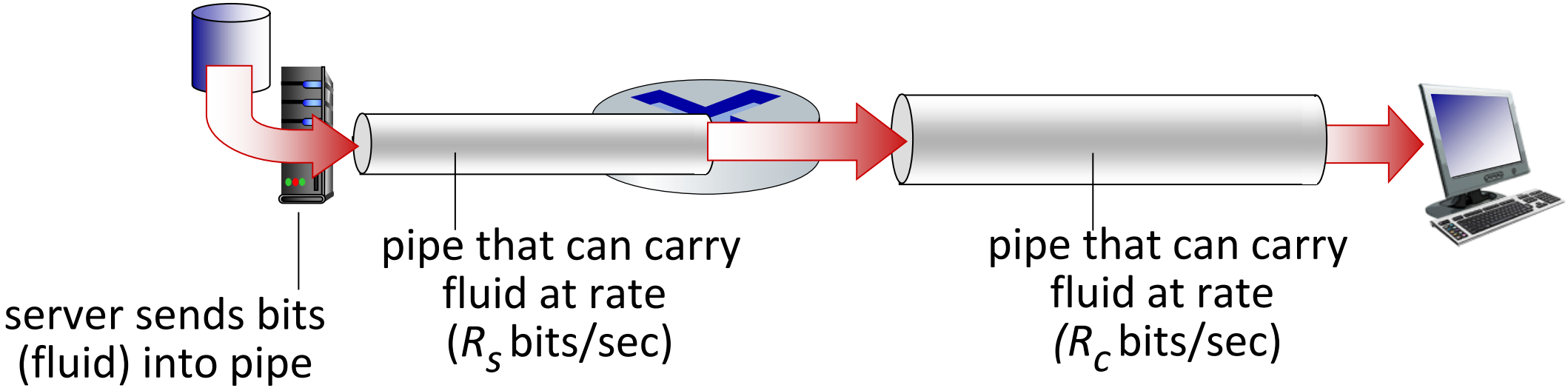
  - For some applications, such as Internet telephony, it is desirable to have a low delay and an instantaneous throughput consistently above some threshold (for example, over 24 kbps for some Internet telephony applications and over 256 kbps for some realtime video applications).

  - For other applications, including those involving file transfers, delay is not critical, but it is desirable to have the highest possible throughput.

- **Note***:* Throughput usually is interpreted as the *successful* transmission rate.

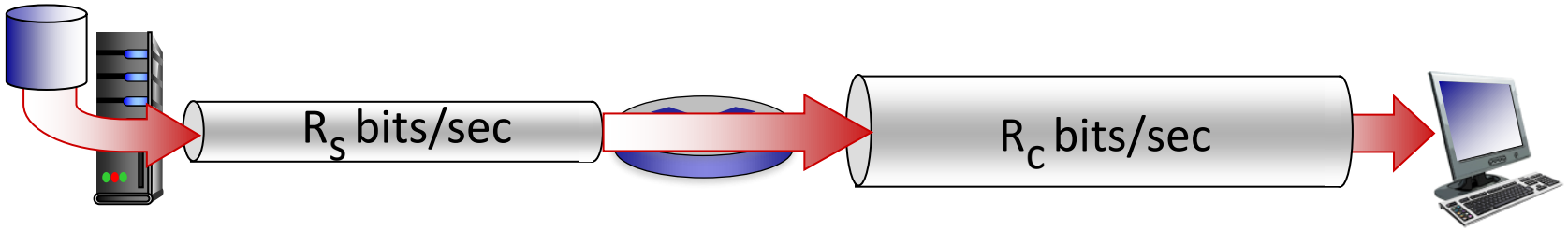
Throughput

- *throughput*

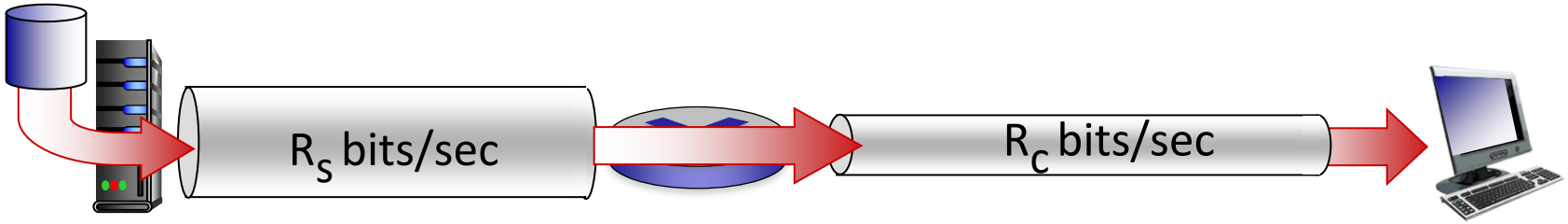


Throughput

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



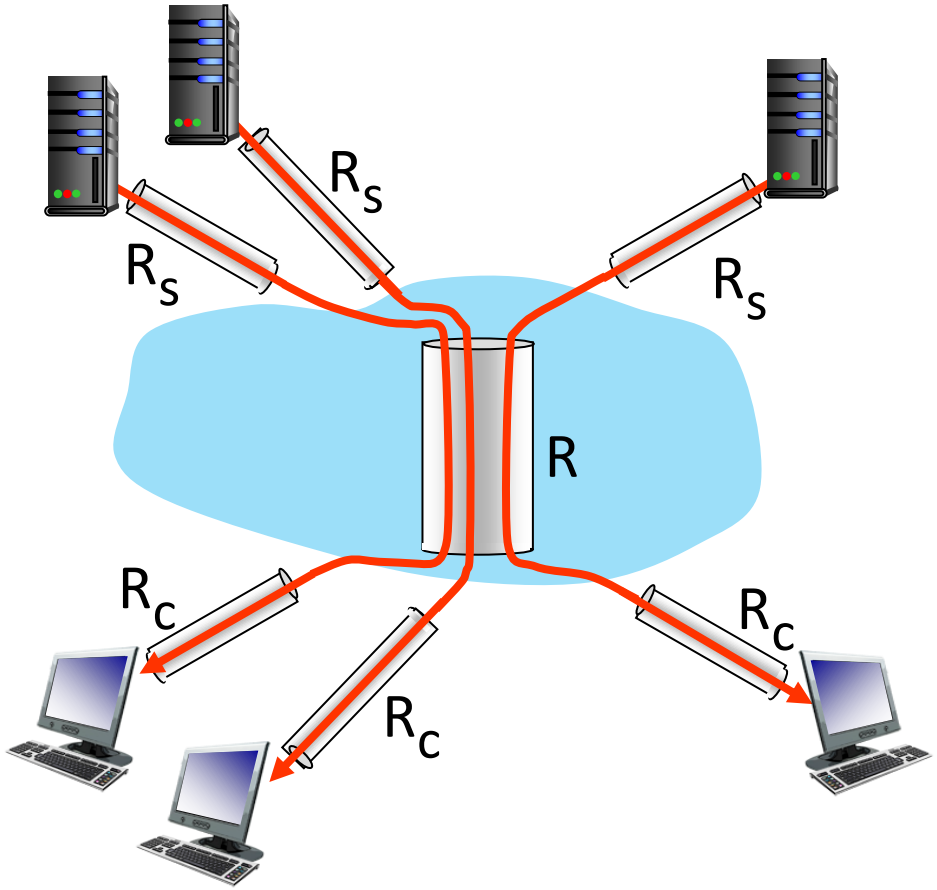
$R_s > R_c$ What is average end-end throughput?



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

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

- 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!
- We now need to think about:
  - how bad guys can attack computer networks
  - how we can defend networks against attacks
  - how to design architectures that are immune to attacks

Bad guys: put malware on hosts

- We attach devices to the Internet because we want to receive/send data from/to the Internet.
- This includes all kinds of good stuff, including Web pages, e-mail messages, MP3s, telephone calls, live video, search engine results, and so on.
- But, unfortunately, along with all that good stuff comes malicious stuff—collectively known as **malware**—that can also enter and infect our devices.
- Once malware infects our device it can do all kinds of devious things, including deleting our files; installing spyware that collects our private information, such as social security numbers, passwords, and keystrokes, and then sends this (over the Internet, of course!) back to the bad guys.

Bad guys: put malware on hosts

- Much of the malware out there today is **self-replicating**:
  - once it infects one host, from that host it seeks entry into other hosts over the Internet, and from the newly infected hosts, it seeks entry into yet more hosts.
  - In this manner, self-replicating malware can spread exponentially fast.
- Malware can spread in the form of a **virus** or a **worm**.
- Viruses are malware that **require** some form of **user interaction** to infect the user's device.
- The classic example is an e-mail attachment containing malicious executable code.
  - If a user receives and opens such an attachment, the user inadvertently runs the malware on the device.

Bad guys: put malware on hosts

- Typically, such email viruses are self-replicating:
  - Once executed, the virus may send an identical message with an identical malicious attachment to, for example, every recipient in the user's address book.

- Worms are malware that can enter a device **without** any explicit user interaction.
  - For example, a user may be running a vulnerable network application to which an attacker can send malware.

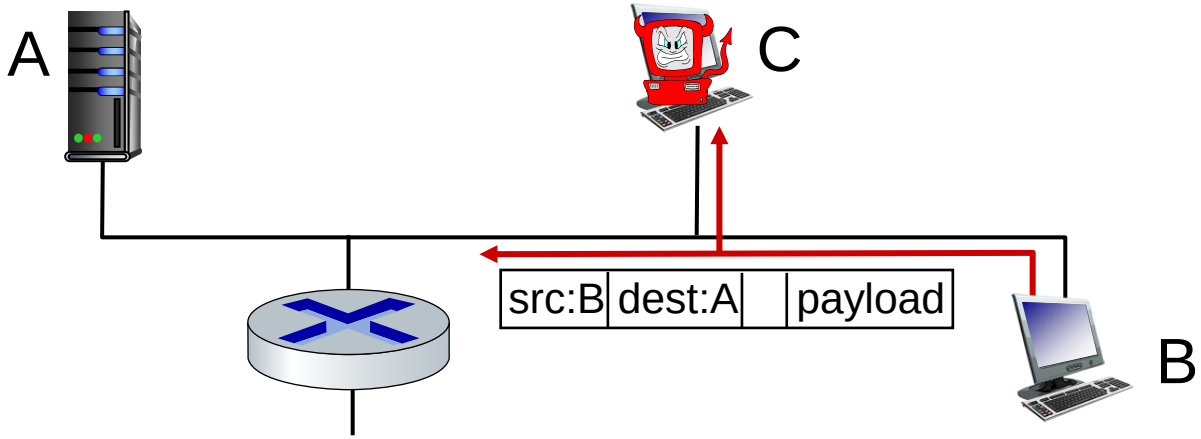
- In some cases, without any user intervention, the application may accept the malware from the Internet and run it, creating a worm.

- The worm in the newly infected device then scans the Internet, searching for other hosts running the same vulnerable network application. When it finds other vulnerable hosts, it sends a copy of itself to those hosts

Bad guys: packet interception

packet "sniffing":

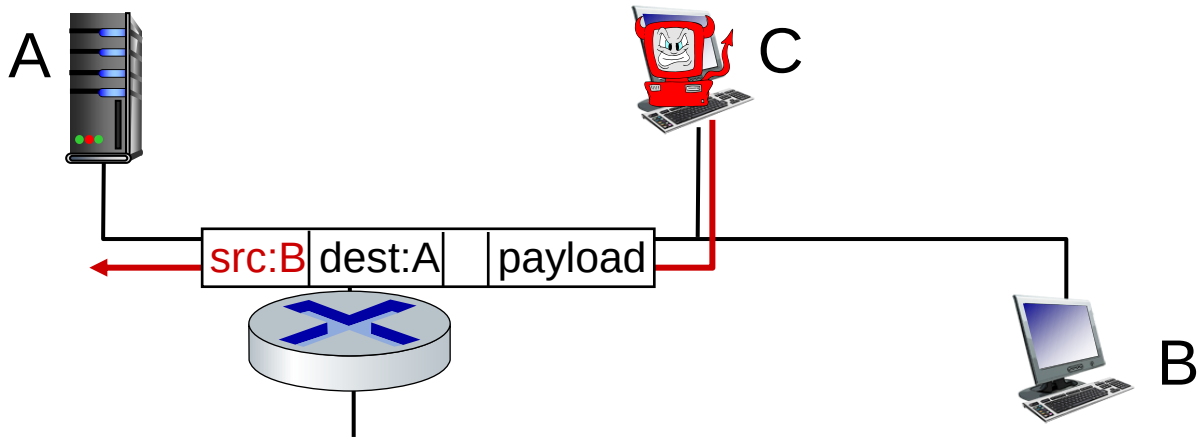
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by.
- works passively.
- How to defense against?
  - **A good choice**: Cryptography



Wireshark software is a (free) packet-sniffer

Bad guys: fake identity

IP spoofing: injection of packet with false source address



- How to defense against?
  - **A solution**: "*end-point authentication*", that is, a mechanism that will allow us to determine with certainty if a message originates from where we think it does.

Bad guys: denial of service

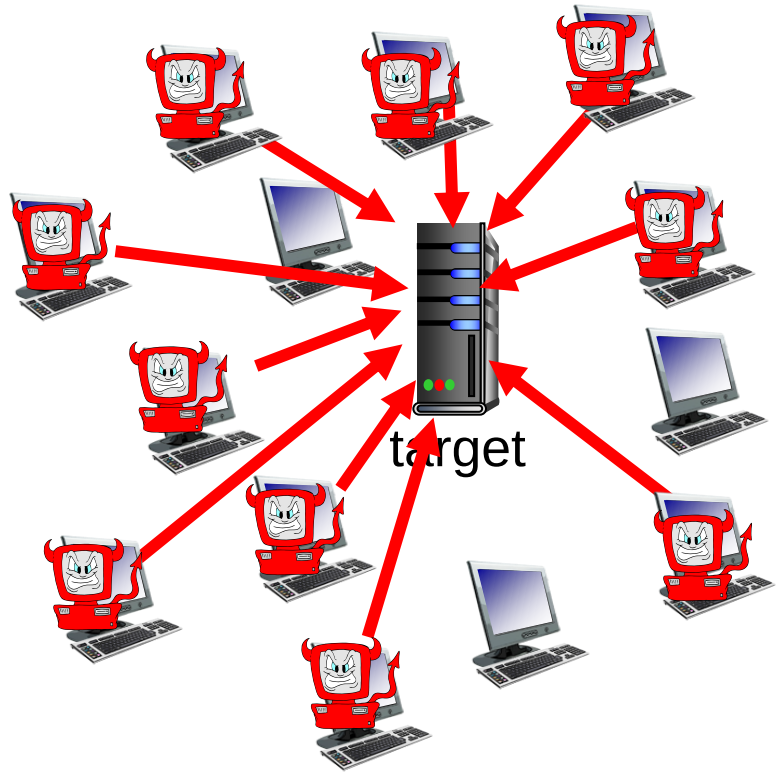
Denial of Service (DoS): attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic.

- Internet DoS attacks are extremely common, with thousands of DoS attacks occurring every year.

- Most Internet DoS attacks fall into one of three categories:
  - **Vulnerability attack**: This involves sending a few well-crafted messages to a vulnerable application or operating system running on a targeted host.
  - **Bandwidth flooding**: The attacker sends a deluge of packets to the targeted host
  - **Connection flooding**: The attacker establishes a large number of half-open or fully open TCP connections at the target host.

Bad guys: denial of service

- Suppose the bandwidth to a server is *R* bps.
- Can a single host make a bandwidth flooding attack?
- It might be simply detectable specially when the *R* is very large.
- Bandwidth flooding attack can be done via a distributed DoS (DDoS) attack, where the attacker controls multiple sources and has each source blast traffic at the target.



Lines of defense:

- authentication: proving you are who you say you are
  - cellular networks provides hardware identity via SIM card; no such hardware assist in traditional Internet
- confidentiality: via encryption
- integrity checks: digital signatures prevent/detect tampering
- access restrictions: password-protected VPNs
- firewalls: specialized “middleboxes” in access and core networks:
  - off-by-default: filter incoming packets to restrict senders, receivers, applications
  - detecting/reacting to DOS attacks