

CSCI 491-01

Topics: Internet Programming

Fall 2008

Introduction II

Derek Leonard
Hendrix College

September 5, 2008

Original slides copyright © 1996-2007 J.F Kurose and K.W. Ross

Chapter 1: Roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network core

1.4 Delay, loss, and throughput in packet-switched networks

1.5 Protocol layers, service models

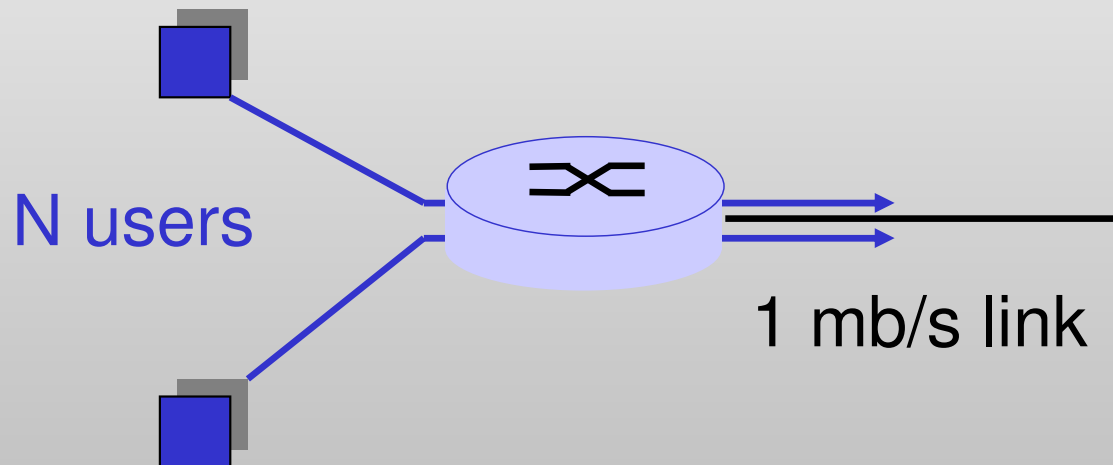
1.6 Networks under attack: security

1.7 History

Packet Switching vs. Circuit Switching

Packet switching allows more users to use network!

- 1 mb/s link
- Each user:
 - 100 kb/s when “active”
 - Active 10% of time
- Circuit-switching:
 - Supports 10 users
- Packet switching:
 - With 35 users, probability that more than 10 users are active is 0.0424%; with 50 users – 0.94%
 - Max 100 users (if perfectly unsynchronized)



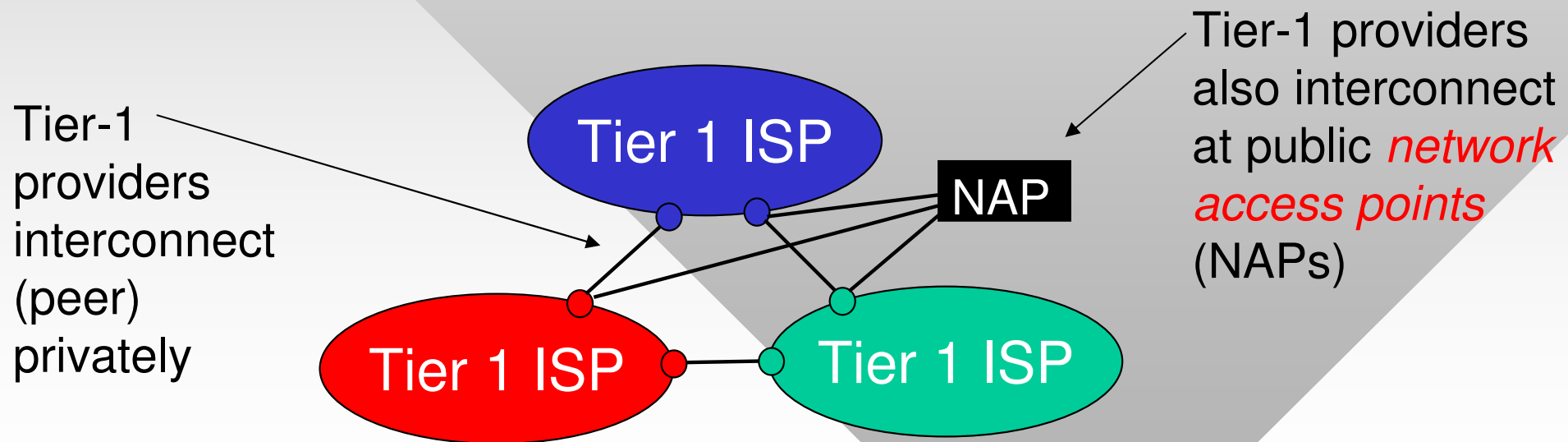
Packet Switching vs. Circuit Switching

Packet switching:

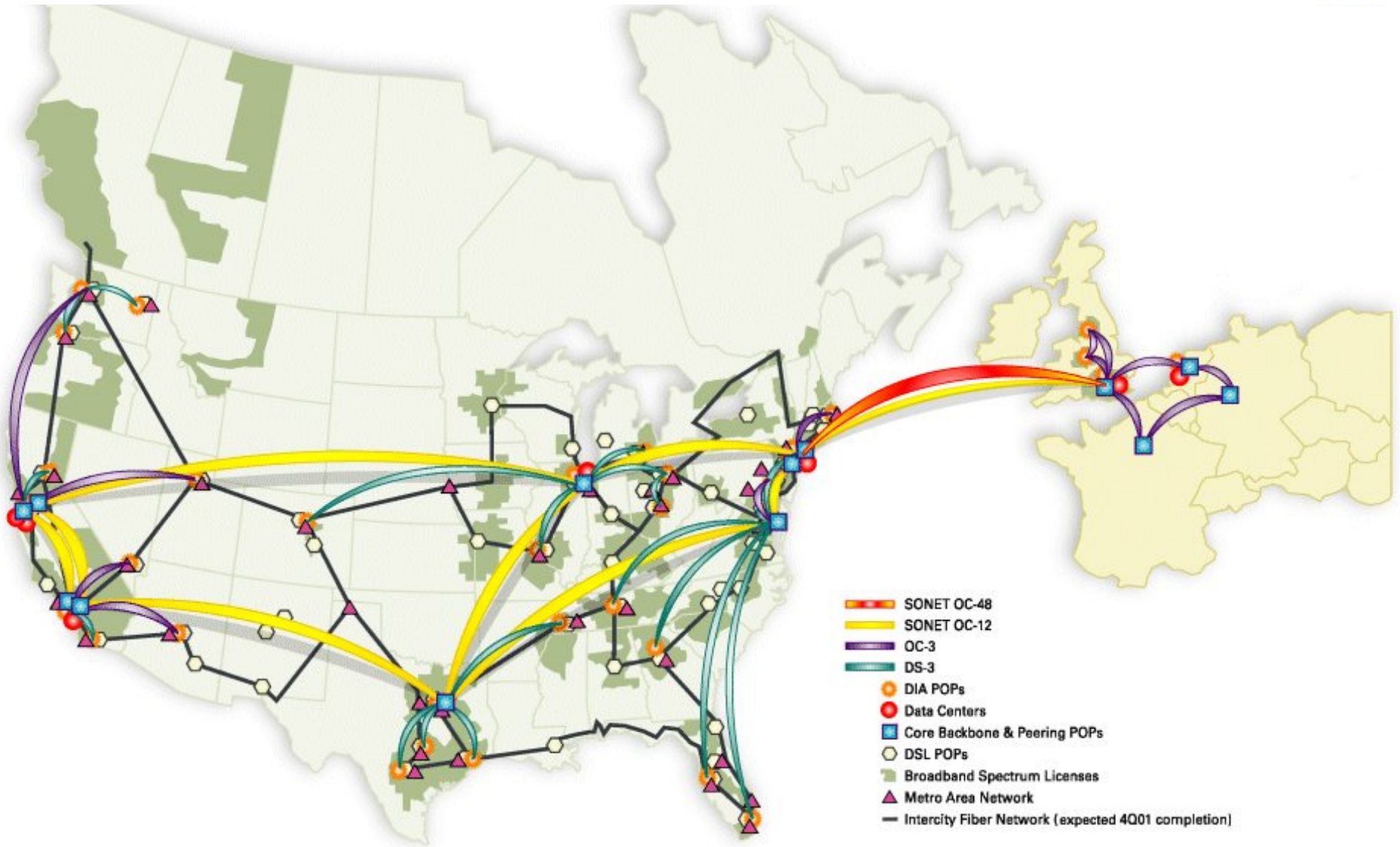
- Great for bursty data
 - Resource sharing
 - Simpler, no call setup
- But suffers from excessive congestion (packet delay and loss)
 - Protocols needed for reliable data transfer, congestion control
- **Q: How to provide circuit-like behavior?**
 - Bandwidth guarantees needed for audio/video apps
 - Still an unsolved problem (chapter 7)

Internet Structure: Network of Networks

- Roughly hierarchical
- **In the center:** “tier-1” ISPs (e.g., UUNet, BBN/Genuity, Sprint, AT&T), national/international coverage
 - Treat each other as equals
 - Form the **backbone** of the Internet



Tier-1 ISP: XO Communications (2001)

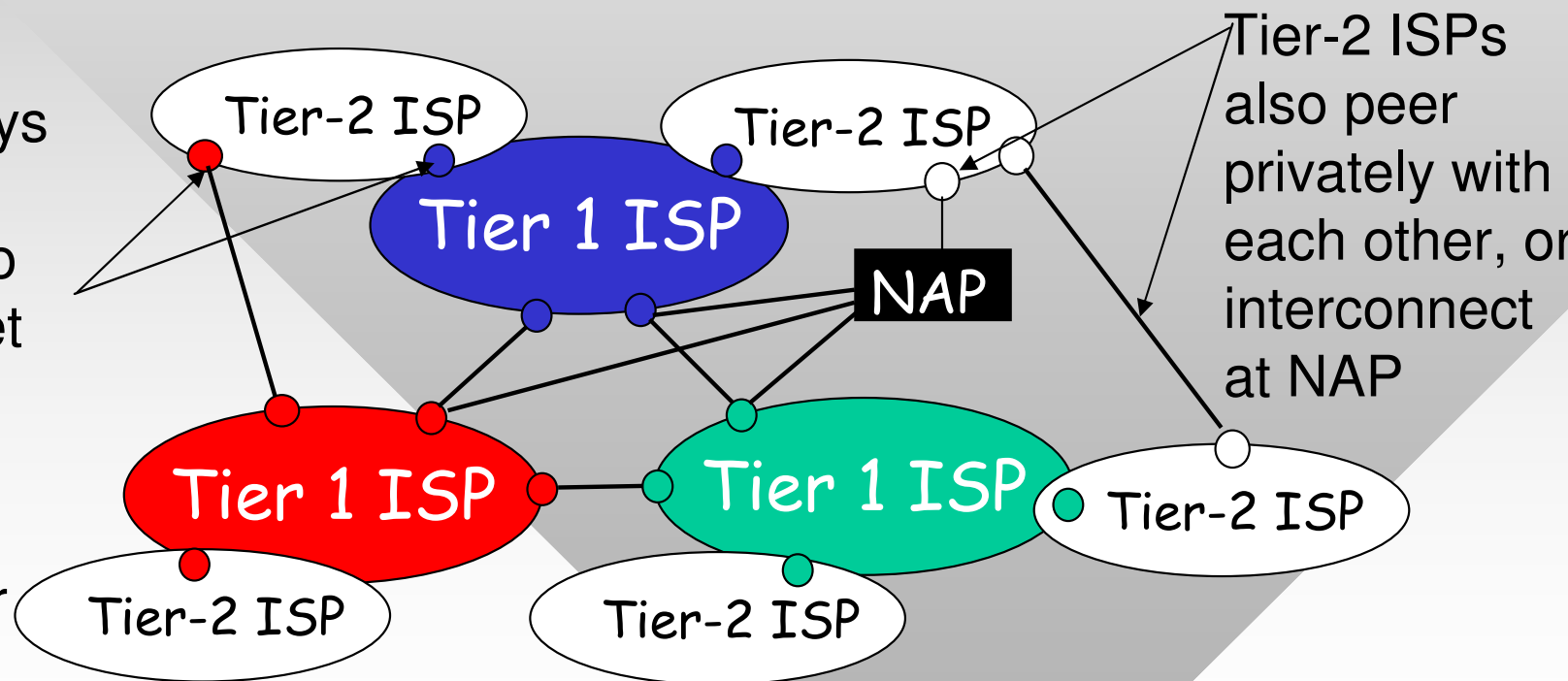


Internet Structure: Network of Networks

- “Tier-2” ISPs: smaller (often regional) ISPs
 - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet

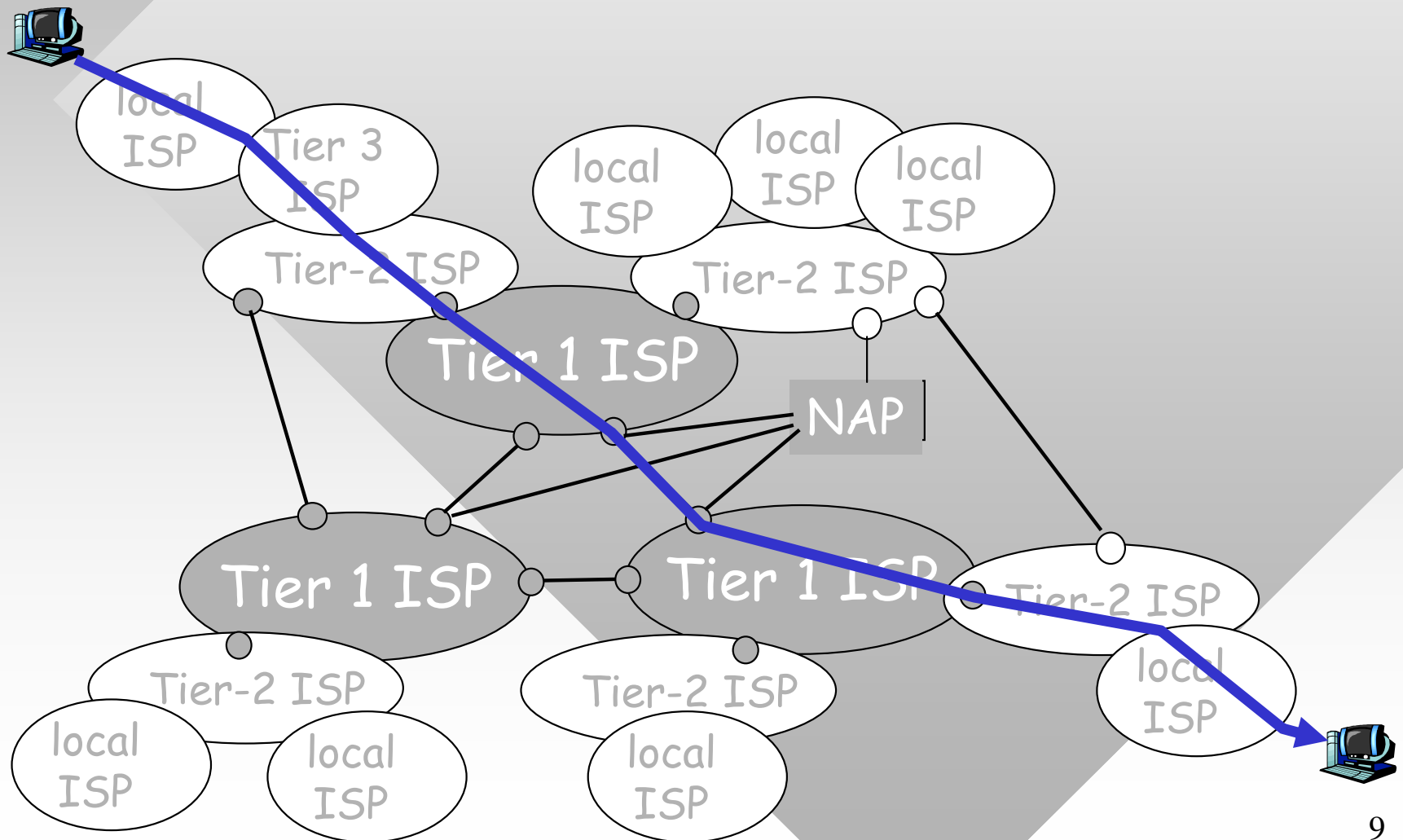
Tier-2 ISP is *customer* of tier-1 provider



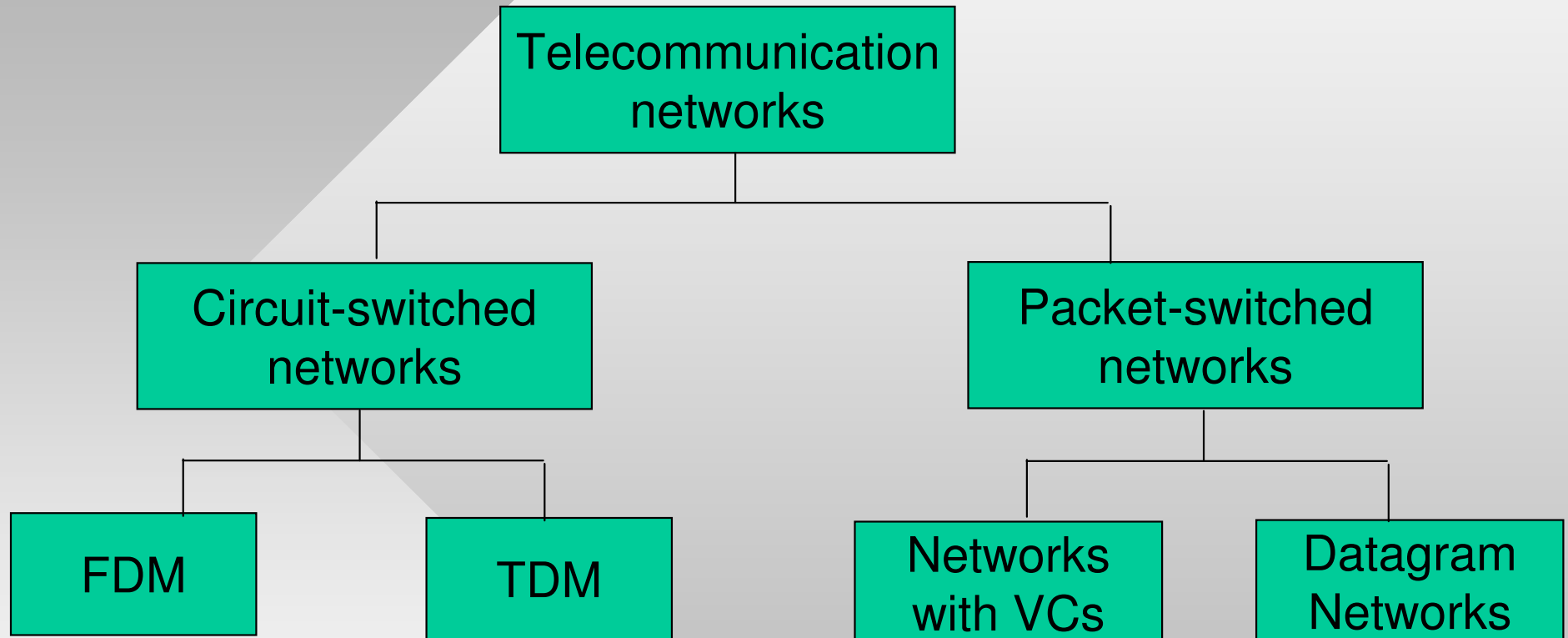
Tier-2 ISPs also peer privately with each other, or interconnect at NAP

Internet Structure: Network of Networks

- A packet passes through many networks!



Network Taxonomy



Internet is a datagram network

- It provides both connection-oriented (TCP) and connectionless services (UDP) to applications

Chapter 1: Roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network core

1.4 Delay, loss, and throughput in packet-switched networks

1.5 Protocol layers, service models

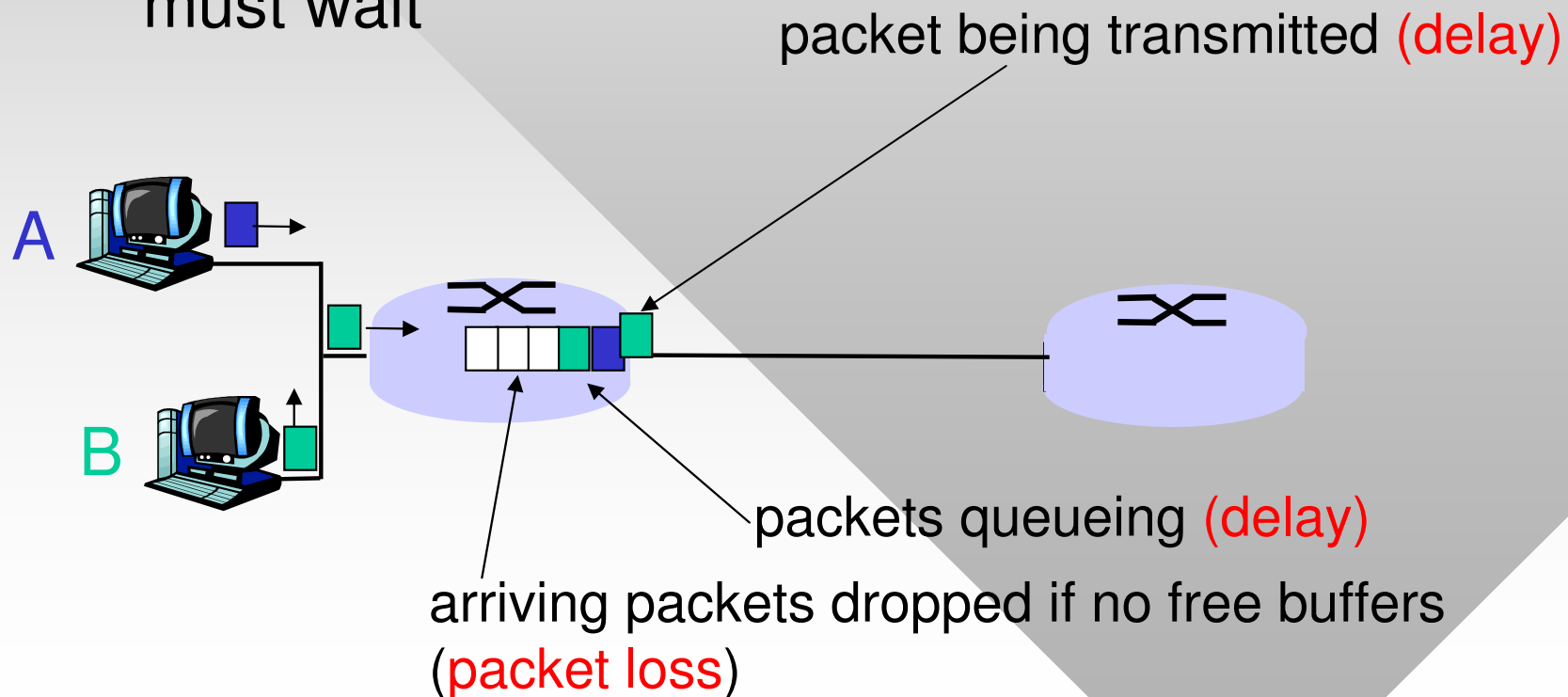
1.6 Networks under attack: security

1.7 History

How Do Loss and Delay Occur?

Packets *queue* in router buffers

- **If packet arrival rate exceeds output link capacity:**
 - Packets queue, wait for their turn
 - Analogy: 6 lanes of traffic merge into 1 – some vehicles must wait



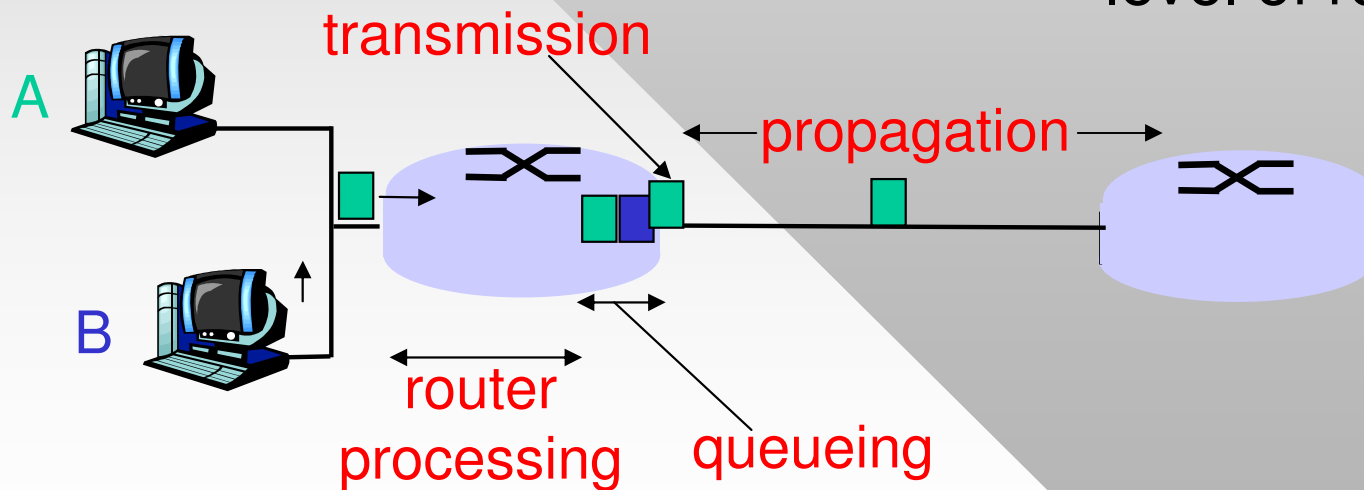
Four Sources of Packet Delay

1. Router processing delay:

- Check bit errors
- Determine output link
- Place packet in buffer

2. Queueing delay

- Time waiting at output link for transmission
- Depends on congestion level of router



Delay in Packet-Switched Networks

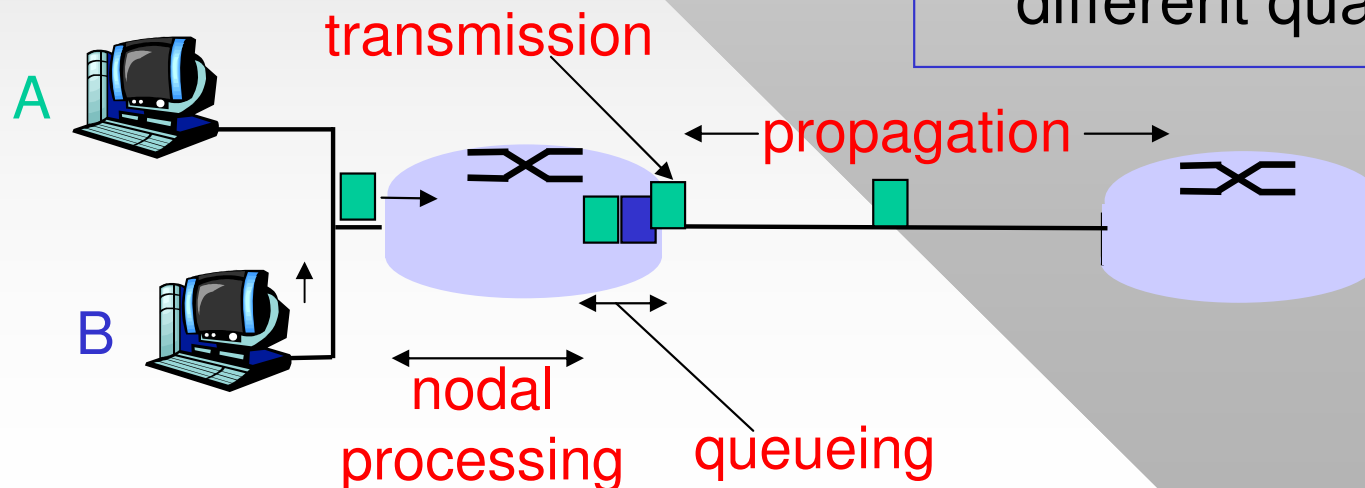
3. Transmission delay:

- R = link bandwidth (bps)
- L = packet length (bits)
- Time to send bits into link = L/R

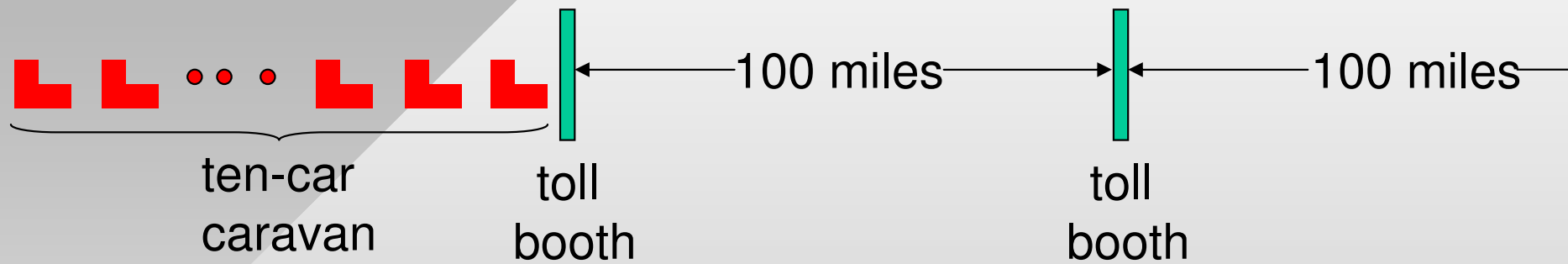
4. Propagation delay:

- d = length of physical link
- s = propagation speed in medium ($\sim 2 \times 10^8$ m/sec)
- Propagation delay = d/s

Note: s and R are very different quantities!

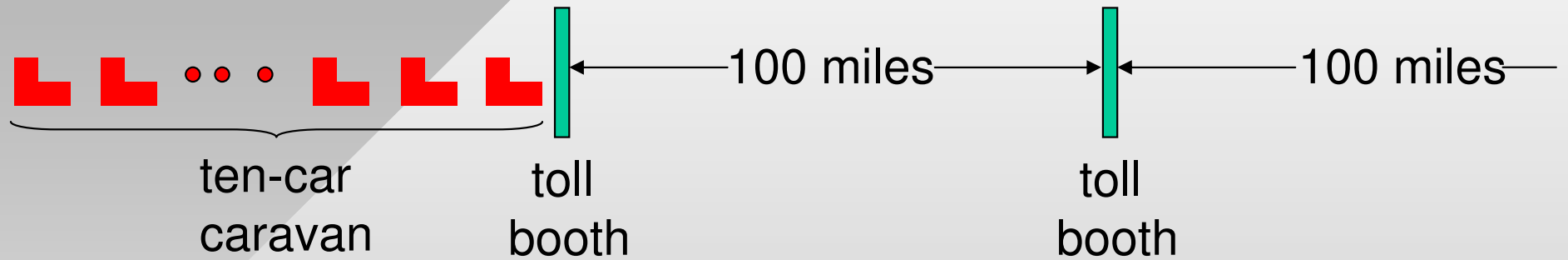


Caravan Analogy



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

Caravan Analogy (more)



- Cars now “propagate” at 1,000 mph
- Toll booth now takes 1 min to service a car
- **Q: Will cars arrive to 2nd booth before all cars are 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 from 1st router!

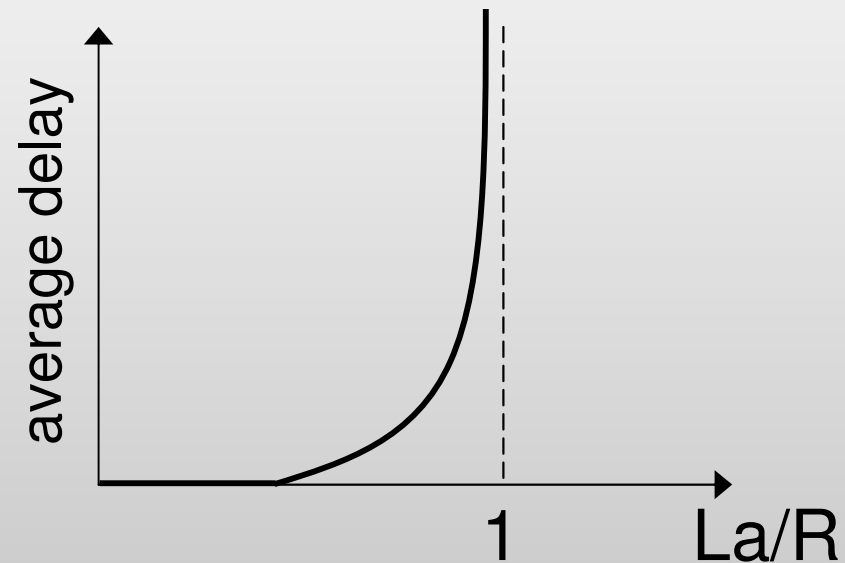
Nodal (Per-Router) Delay

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

- d_{proc} = processing delay
 - Typically a few microseconds or less, usually fixed for all packets
- d_{queue} = queuing delay
 - Depends on congestion, randomly varies between packets
- d_{trans} = transmission delay
 - Equals L/R , high for low-speed links, depends on packet size
- d_{prop} = propagation delay
 - A few microseconds to hundreds of msecs, depends on physical length of the link

Queueing Delay (Revisited)

- R = link bandwidth (bps)
- L = packet length (bits)
- a = **average** packet arrival rate (pkts/sec)
- Infinite buffer space



Traffic intensity = La/R

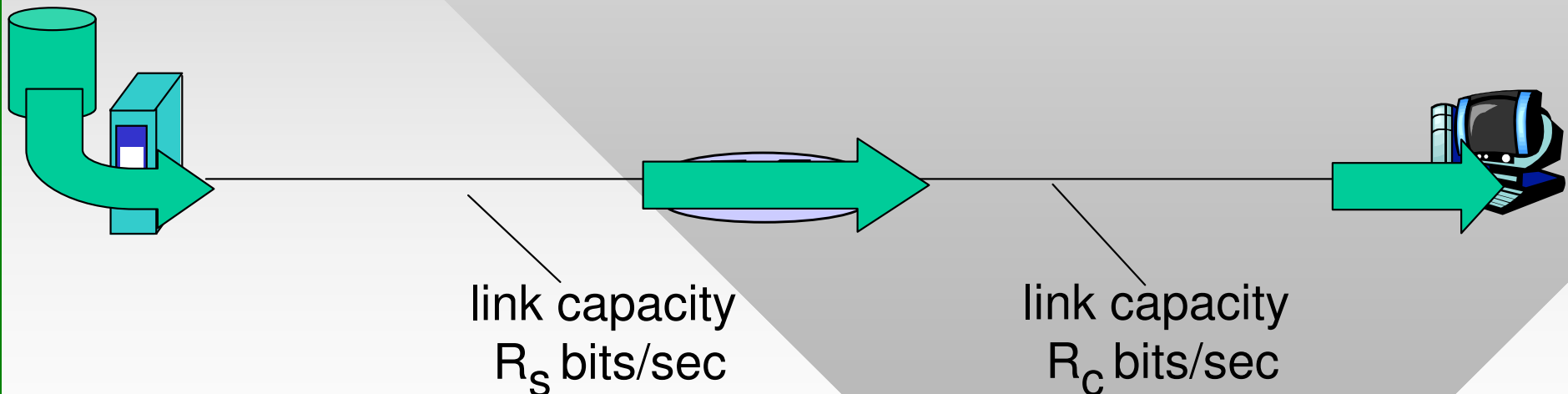
- $La/R \sim 0$: average queueing delay is small
- $La/R \rightarrow 1$: delays become large
- $La/R > 1$: more “work” arriving than can be serviced, average delay is **infinite!**

Packet Loss

- Queues (aka buffers) have **finite** capacity
- When packets arrive to a full queue, they are **dropped** (aka lost)
- Lost packet may be **retransmitted** by previous node, by the source (end system), or not retransmitted at all
- **Average packet loss**: fraction of lost data computed over a long period of time
 - Example: link capacity $R = 10$ mb/s and total arrival rate of traffic is 11 mb/s
 - What's the average loss rate on the link?

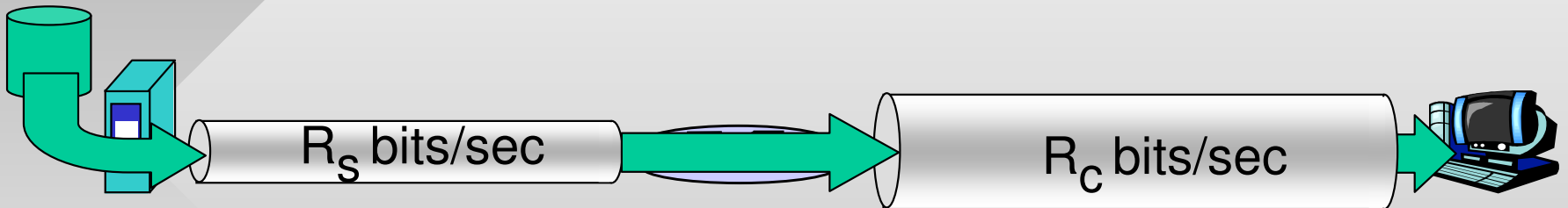
Throughput

- **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

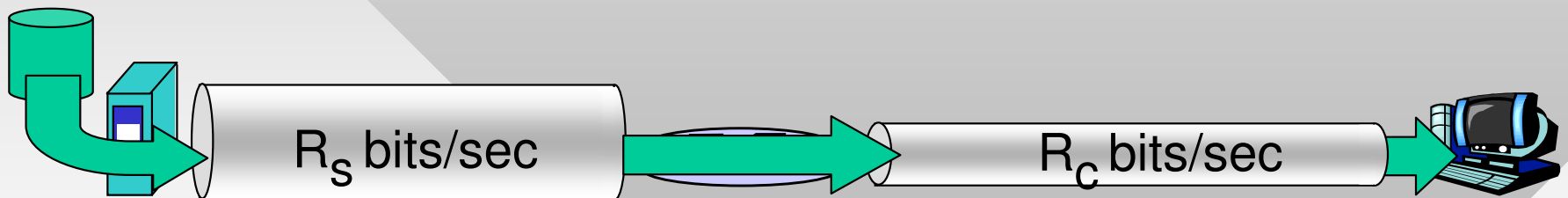


Throughput (more)

- $R_s < R_c$ 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

Chapter 1: Roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network core

1.4 Delay, loss, and throughput in packet-switched networks

1.5 Protocol layers, service models

1.6 Networks under attack: security

1.7 History

Protocol “Layers”

Networks are complex!

- Many “pieces”
 - Hosts
 - Routers
 - Links of various media
 - Applications
 - Protocols
 - Hardware, software
- Modular organization is desirable

Question:

Is there any hope of *organizing* the structure of the network?

Solution: Layered structure

- Upper layers rely on lower layers for service
- Layers talk to similar layers on the other end-host

Layered Organization

New York

Boss (idea)

Assistant (type)

Mailroom (package)

FedEx

Truck driver

Los Angeles

Boss (make decision)

Assistant (read)

Mailroom (receive)

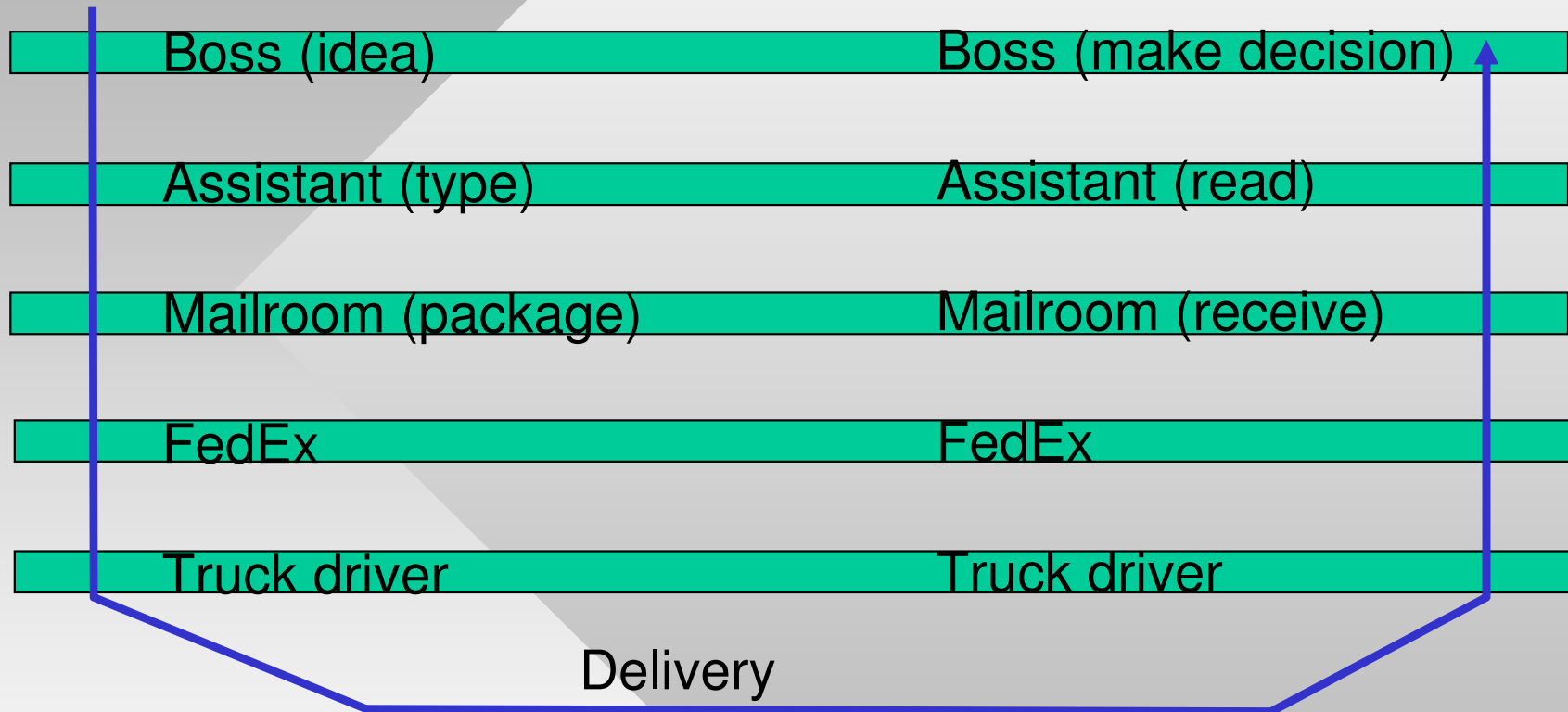
FedEx

Truck driver

Delivery

- Information travels **down** the protocol stack on the sender side and **up** on the receiver side

Layering



Layers: each layer implements a service

- Via its own internal-layer actions
- Relying on services provided by the layer below

Why Layering?

Benefits of layered organization:

- Sufficient to specify only the **relationship** between the system's pieces
 - Instead of defining the internal structure of each layer
 - Complexity reduced by **separately** standardizing individual components
- Modularization eases maintenance and upgrade
 - Change of implementation of layer's service transparent to the rest of system
 - For example, change in FedEx truck routing doesn't affect other layers

Internet Protocol Stack

- **Application:** supports network applications (ch 2)
 - FTP, SMTP, HTTP
- **Transport:** inter-process data transfer
 - TCP, UDP (ch 3)
- **Network:** routing of datagrams from source to destination (ch 4)
 - IP, routing protocols
- **Link:** data transfer between neighboring network elements
 - PPP, Ethernet (ch 5)
- **Physical:** bits “on the wire”
 - Not covered in this class

application (5)

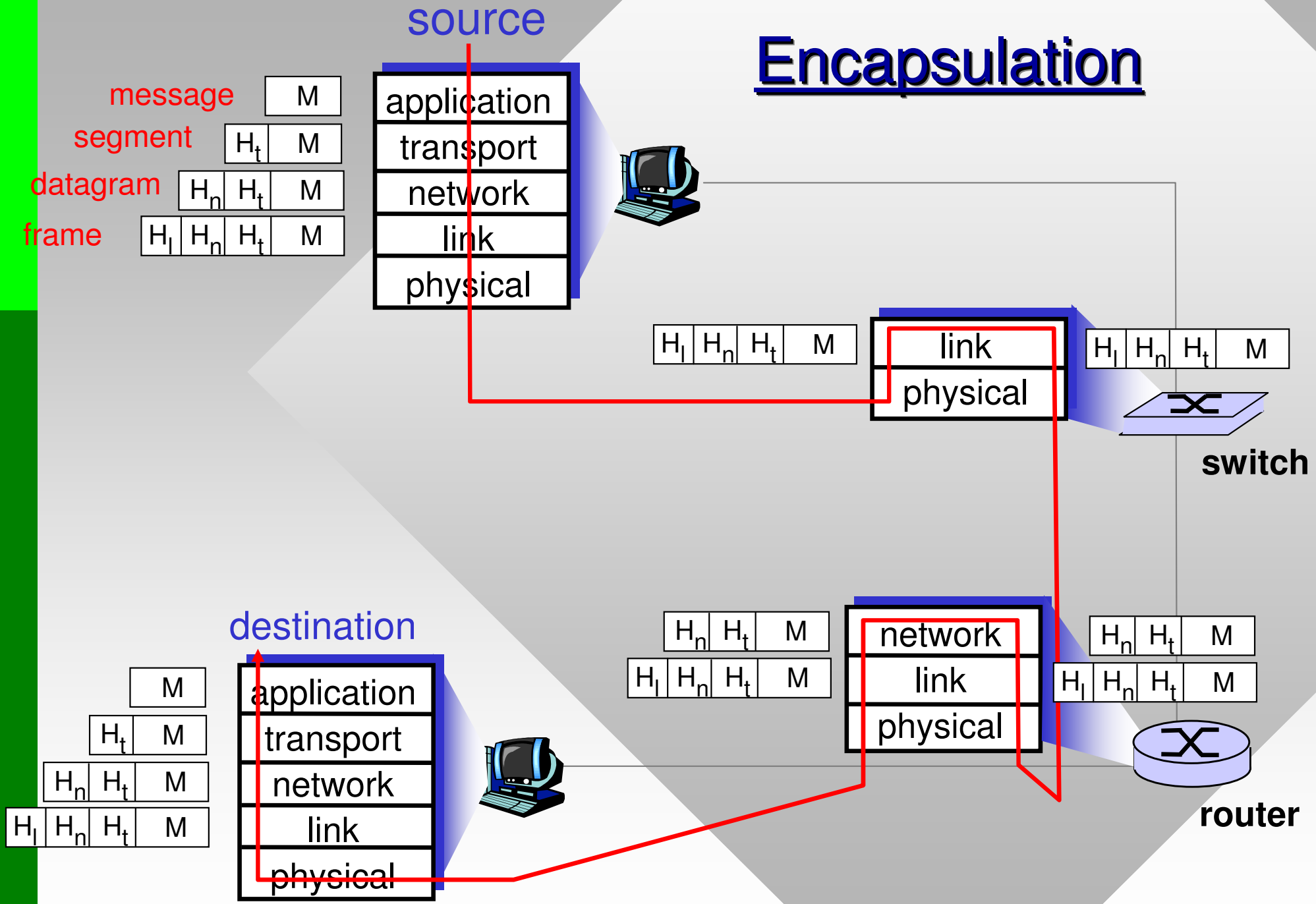
transport (4)

network (3)

link (2)

physical (1)

Encapsulation



Chapter 1: Roadmap

- 1.1 What *is* the Internet?
- 1.2 Network edge
- 1.3 Network core
- 1.4 Delay, loss, and throughput in packet-switched networks
- 1.5 Protocol layers, service models
- 1.6 Networks under attack: security
- 1.7 History

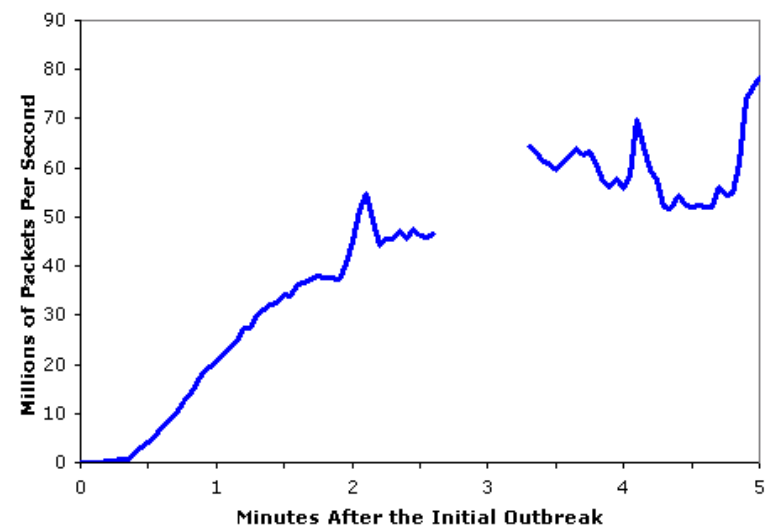
Network Security

- **The field of network security is about:**
 - 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!

Malware Propagation

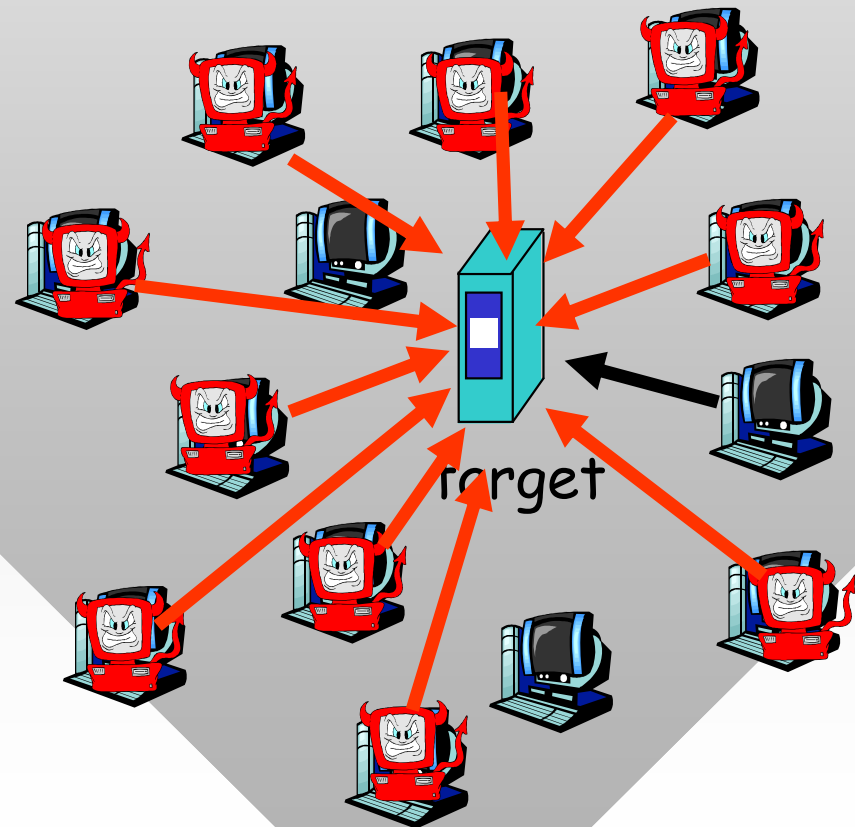
- **Trojan horse**
 - Hidden part of some otherwise useful software
 - Today often on a Web page (Active-X, plugin)
- **Virus**
 - infection by receiving object (e.g., e-mail attachment), actively executing
 - self-replicating: propagate itself to other hosts, users
- **Worm:**
 - infection by passively receiving object that gets itself executed
 - self-replicating: propagates to other hosts, users

Sapphire Worm: aggregate scans/sec in first 5 minutes of outbreak (CAIDA, UWisc data)



Attacking Network Infrastructure

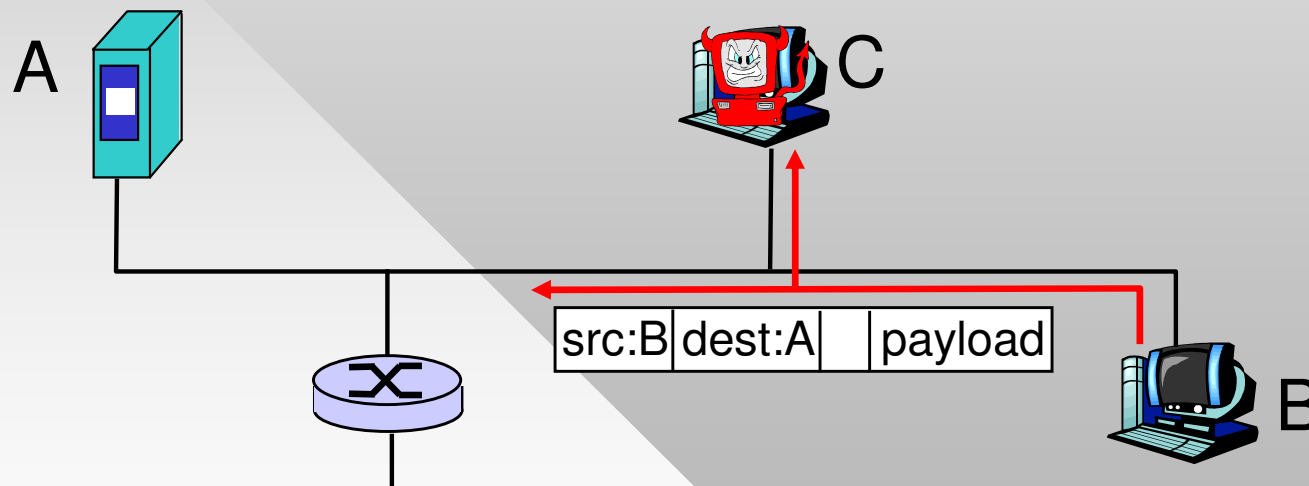
- **Denial of service** (DoS): attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic
 1. select target
 2. break into hosts around the network (create a botnet)
 3. send packets toward target from compromised hosts



Snooping on Users

Packet sniffing:

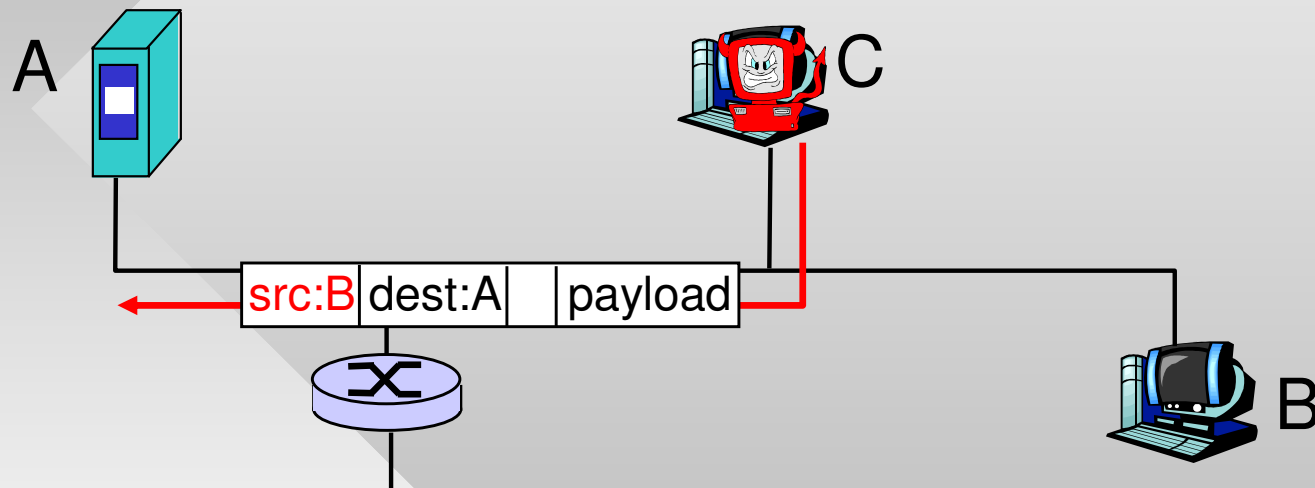
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



- Wireshark software (on Linux lab machines) is a free packet-sniffer

Posing as Somebody Else

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



Next Time

- Read the History section if you're interested
- Chapter 2: Application Layer
 - Read it!
- Reminder: Quiz on September 12 covering the problems from Chapter 1.
 - P1-P26 on pages 69-75