# Computer Networks

## Lecture 3: Introduction:

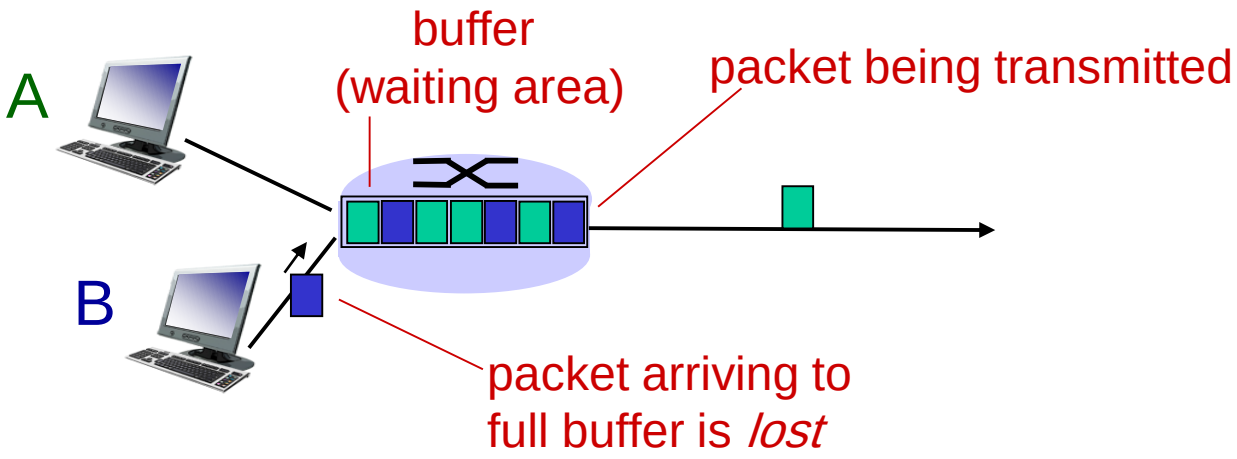
Delay, loss, throughput in networks
Protocol layers, service models Security


Dr. Hossam Mahmoud Moftah

Assistant professor – Faculty of computers and information – Beni Suef University

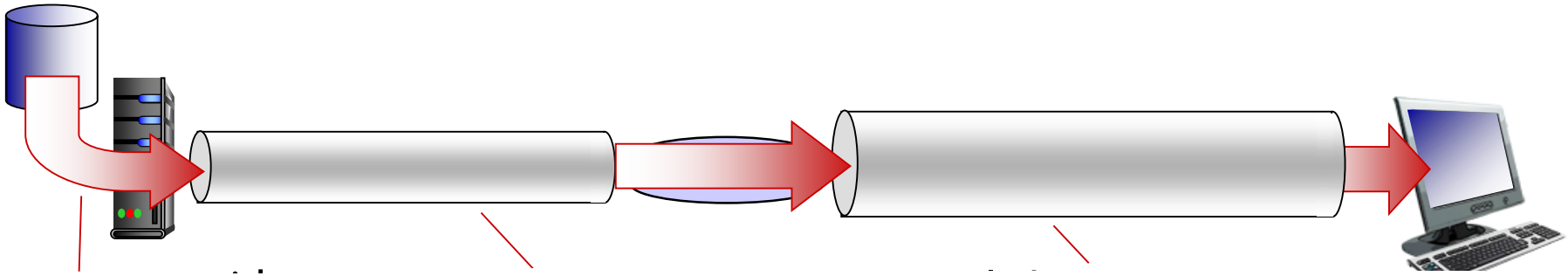
# Packet loss

- 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



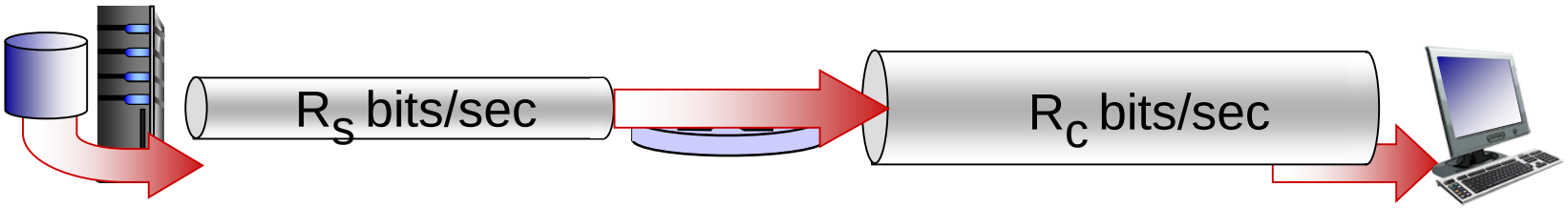
# Throughput

- *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
  - Usually, we mean average throughput

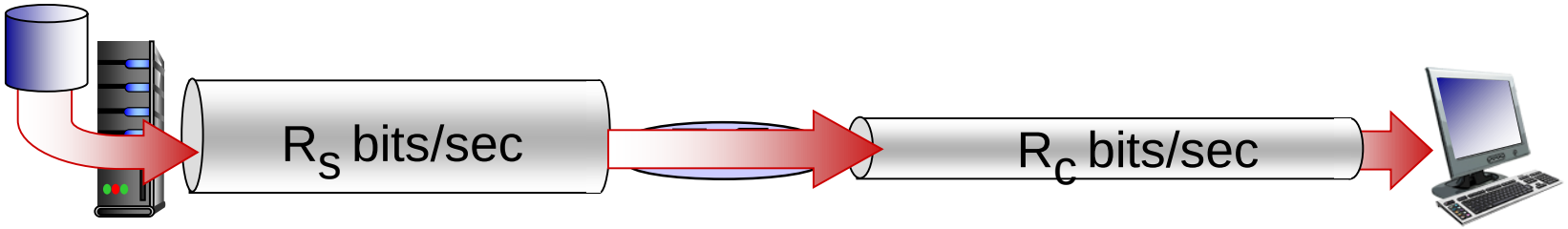


# Throughput (more)

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




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



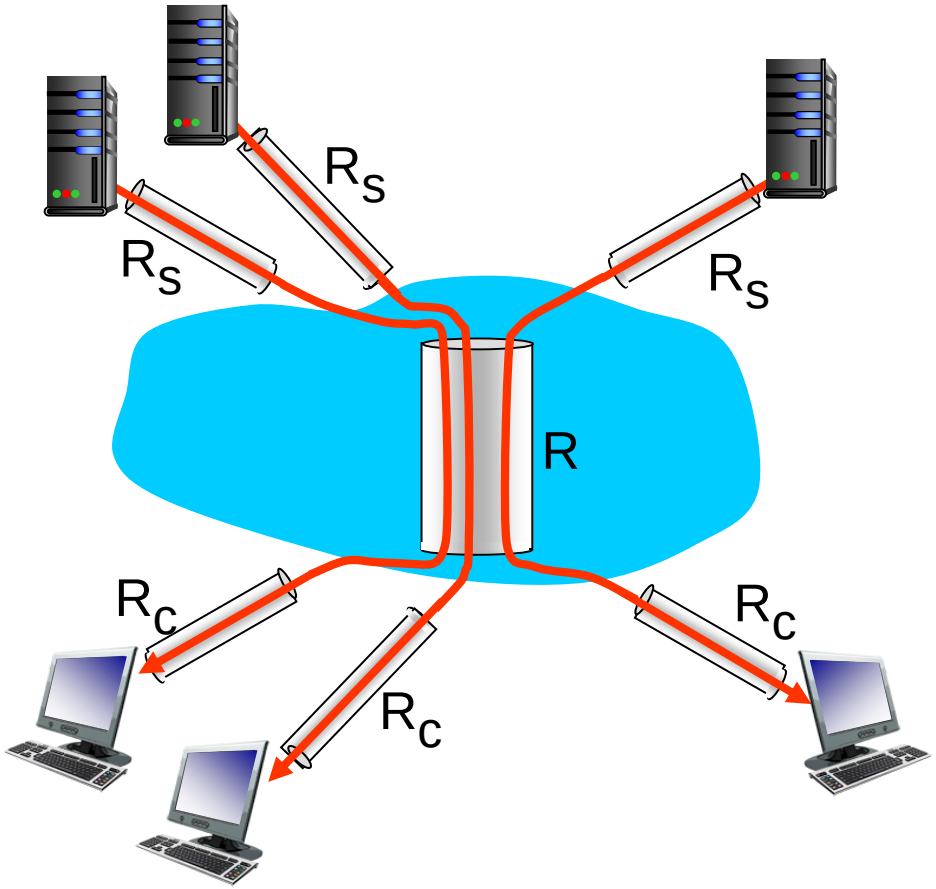

Throughput = min{Rs,Rc}

*bottleneck link*

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

# Throughput: Internet scenario

- per-connection end-end throughput:
  $\min(R_c, R_s, R/6)$
- in practice: bottleneck link is
  $R_c$ or $R_s$




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

# Measurements of network performance:

- Bandwidth – capacity of the system
- Throughput – number of bits that can be pushed through time
- Delay (latency) – delay incurred by a bit from start to finish
- Loss – number of bits lost (dropped) in the network
- What is the difference between Throughput and bandwidth?
  - Bandwidth is the maximum amount of data that can travel through a 'channel'.
  - Throughput is how much data actually does travel through the 'channel' successfully. This can be limited by a ton of different things including latency, and what protocol you are using.

# 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"

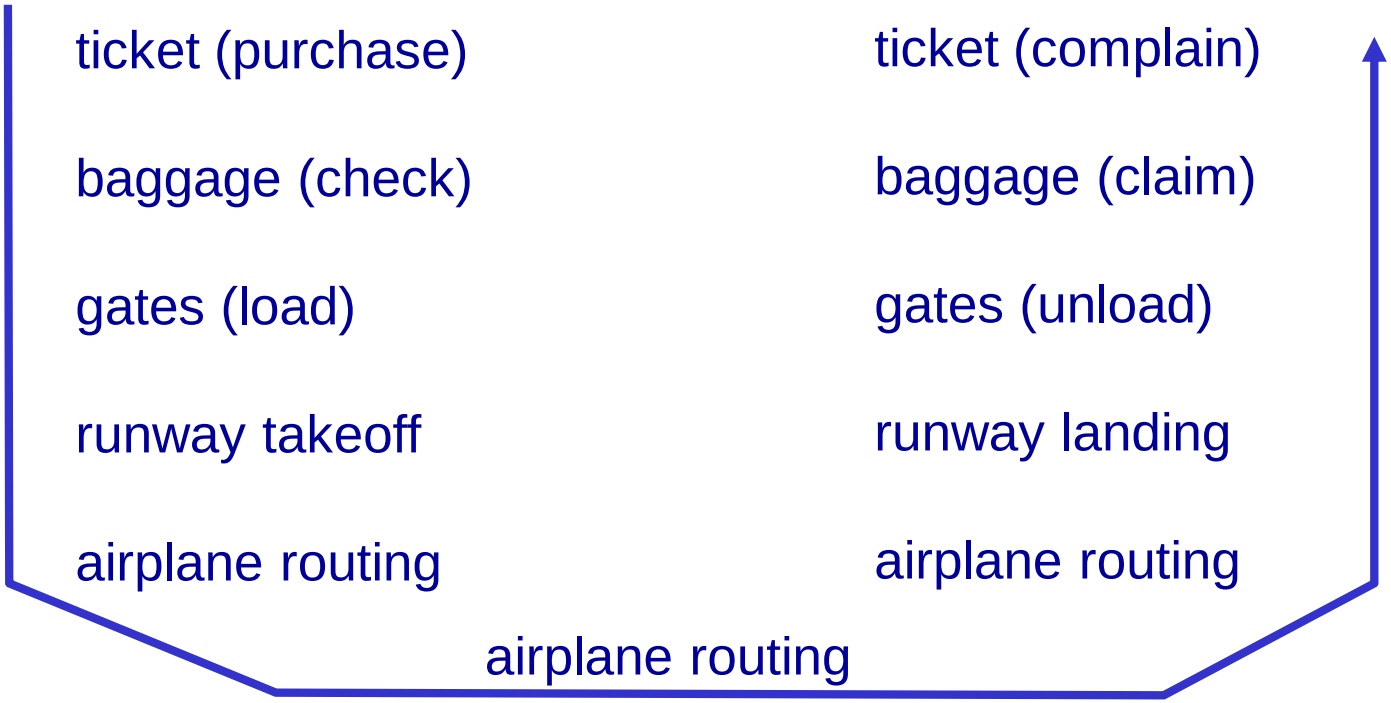
*Networks are complex, with many "pieces":*

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

*Question:*

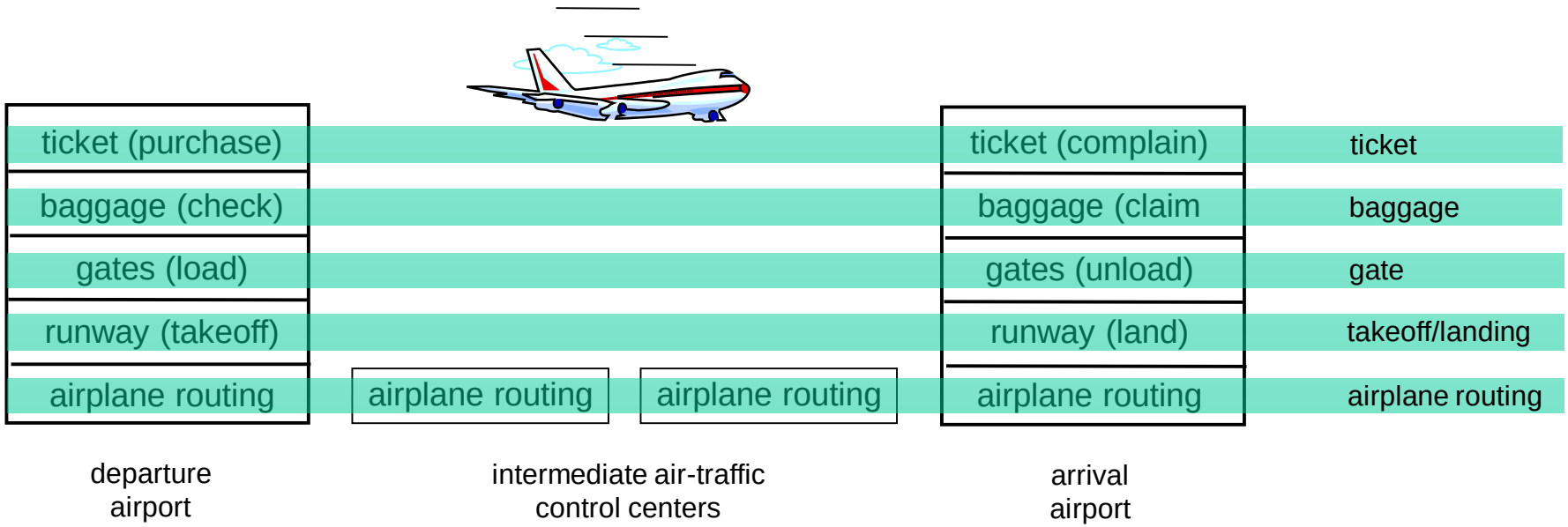
is there any hope of *organizing* structure of network?

# Organization of air travel




- a series of steps

# Layering of airline functionality

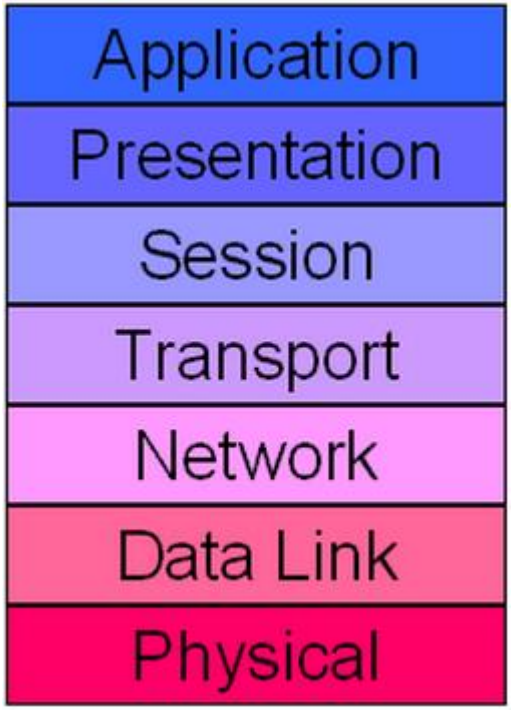



*layers:* each layer implements a service

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

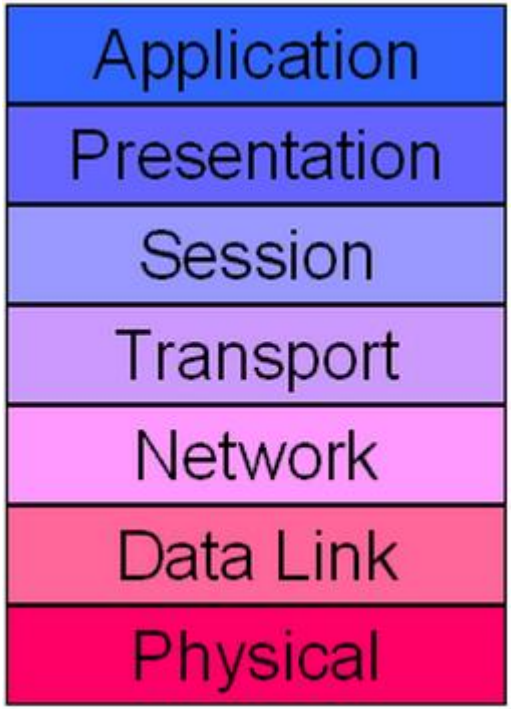
# ISO/OSI reference model

- *application:* supporting network applications
  - FTP, 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 (use MAC address)
  - Ethernet, 802.111 (WiFi), PPP
- *physical:* bits "on the wire"

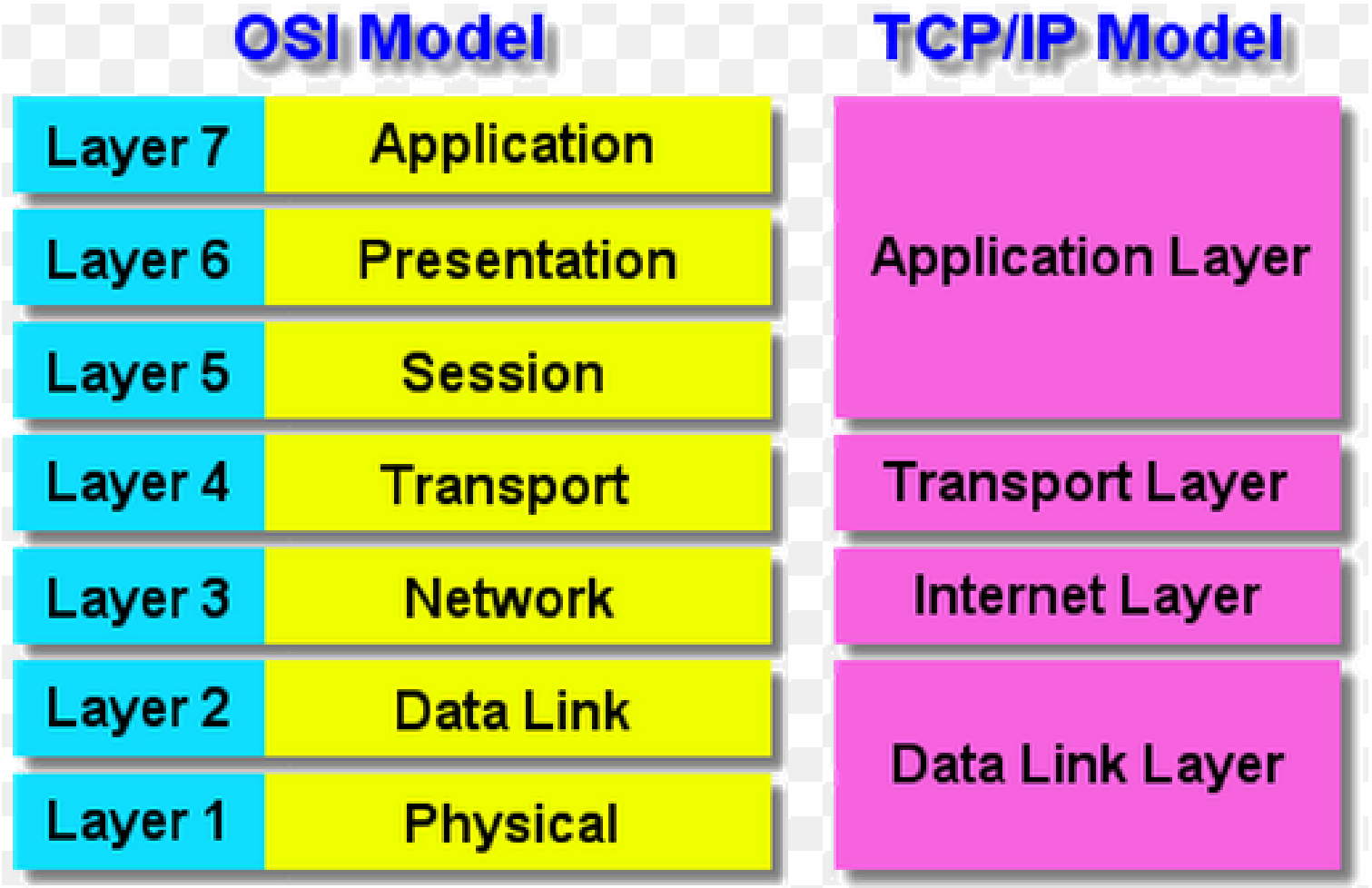


# ISO/OSI reference model

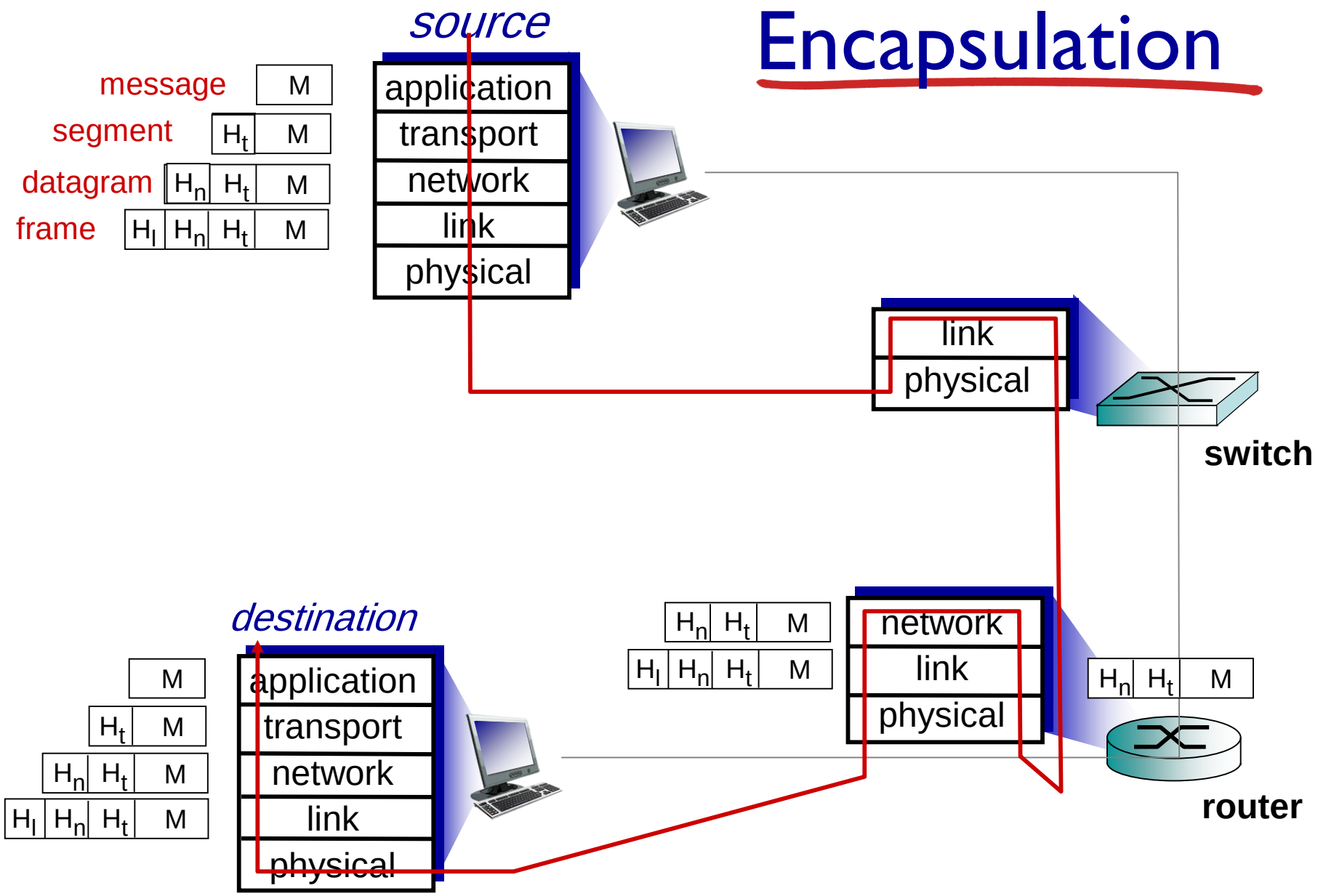
- *presentation:* allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session:* synchronization, checkpointing, recovery of data exchange



# TCP/IP reference model



# Encapsulation

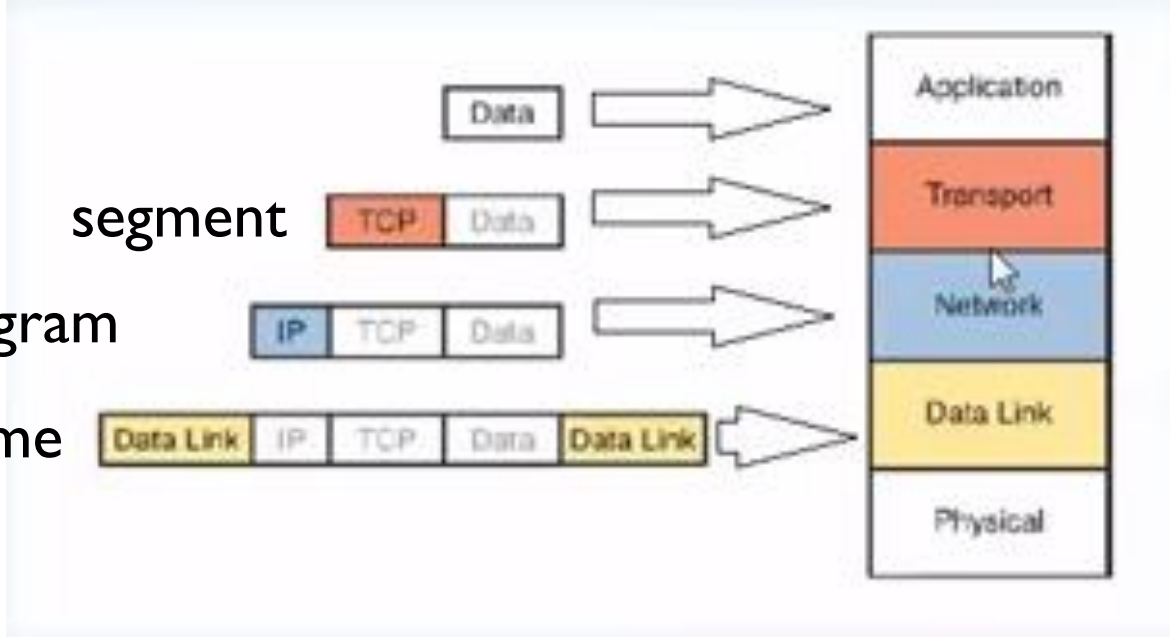


# Encapsulation

segment

Packet or datagram

frame



# 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
  - 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: put malware into hosts via Internet

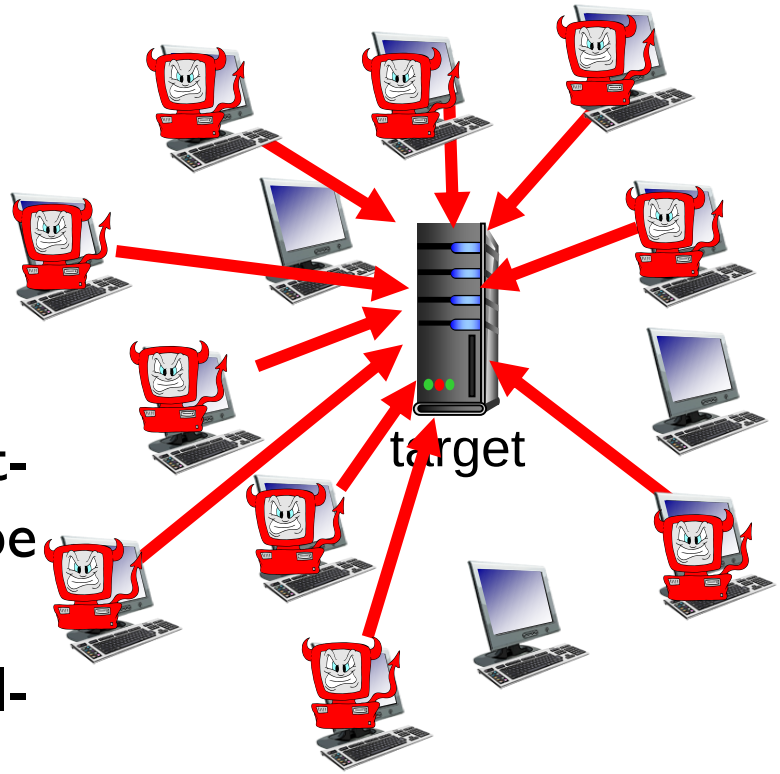
- malware can get in host from:
  - *virus:* self-replicating infection by receiving/executing object (e.g., e-mail attachment)
  - *worm:* self-replicating infection by passively receiving object that gets itself executed (It does not need human action)
- spyware malware can record keystrokes, web sites visited, upload info to collection site

# Bad guys: attack server, network infrastructure

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

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

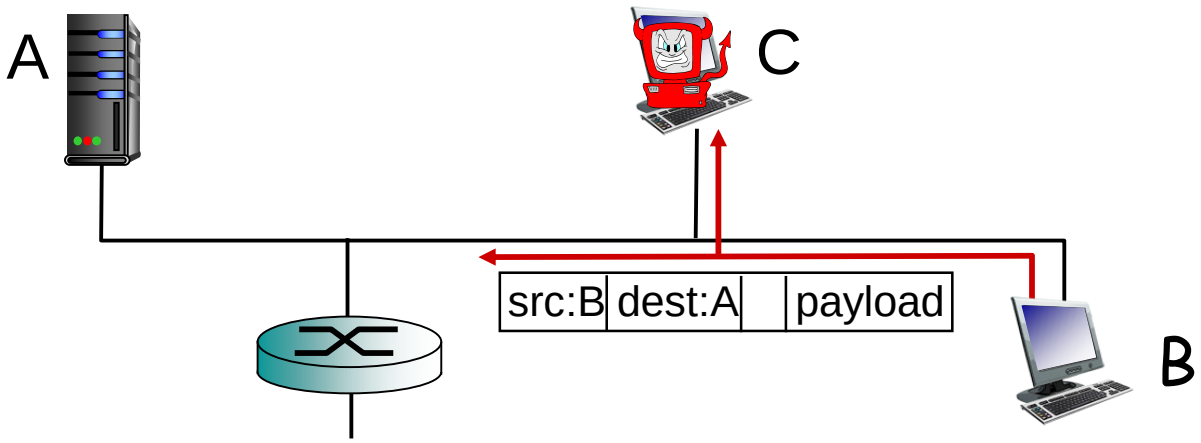
A botnet is a collection of Internet-connected programs. it could be used to send spam email or participate in distributed denial-of-service attacks.



# Bad guys can sniff packets

*packet "sniffing":*

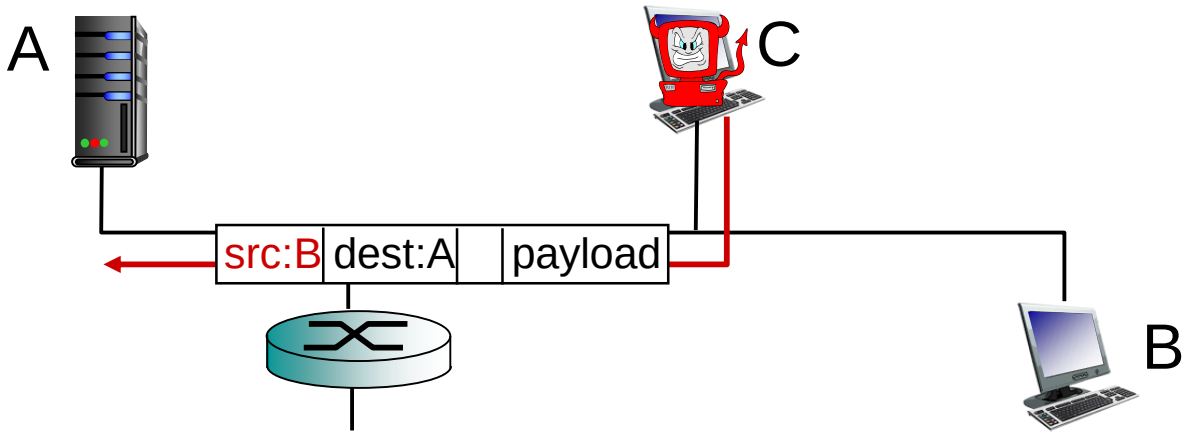
- broadcast media (shared ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by




- wireshark software 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)*

# The End