

Motivation, Terminology,  
Layered systems  
(and other random stuff)

# History and Motivation

- Early computers were highly centralized.
  - Single point of failure
  - User has to “go to” the computer.
- Proliferation of low cost computers made it possible to get past these 2 primary disadvantages (with a network).

# Motivation

- Sharing of resources is more efficient
- Price/Performance
- Use each piece of equipment for what it is best at
- Centralize administration
- Computers as communication tools

# Computer Networks are now everywhere

- PCs <-> Mainframes
- Automated Tellers
- Embedded Systems
- Communications Systems
- The Internet

# Networked Computers - Traditional Uses

- Communications (email)
- File exchange, disk sharing
- Sharing peripherals (printers, tape drives)
- Remote execution

# New(er) Uses for Networked Computers

- Information Sharing
- Entertainment, distributed games
- Commerce
- Automation of business processes
- Collaborative computing
- Homework Submission

# Wide variety of type of networks

- circuit switched - telephone system
- packet switched - TCP/IP (many other computer networks).
- we will focus on just a single (important) protocol suite.

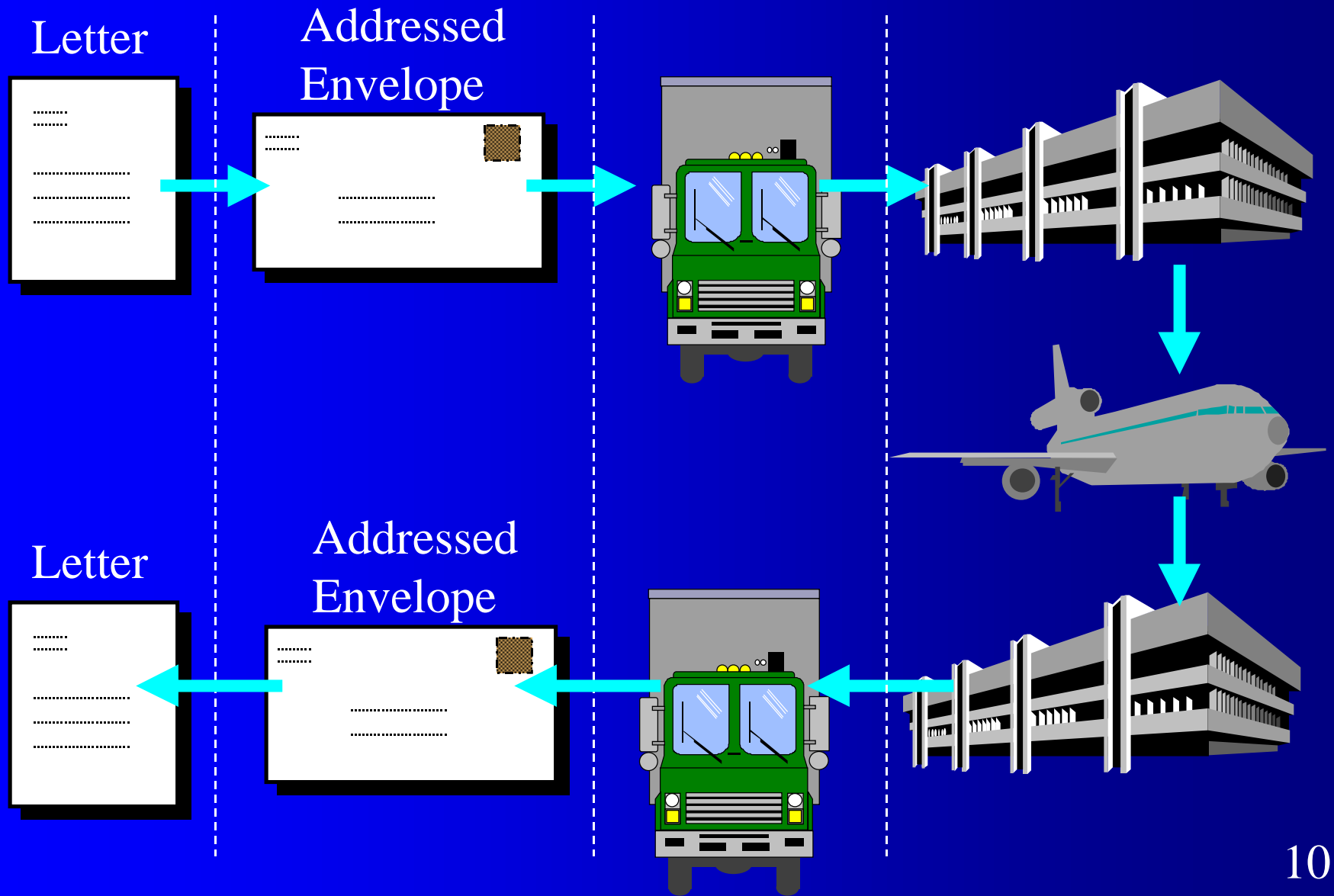
# Layering

- Divide a task into pieces and then solve each piece independently (or nearly so).
- Establishing a well defined interface between layers makes porting easier.
- Major Advantages:
  - ◆ Code Reuse
  - ◆ Extensibility

# Layering Example: Federal Express

- Letter in envelope, address on outside
- FedX guy adds addressing information, barcode.
- Local office drives to airport and delivers to hub.
- Sent via airplane to local FedX office
- Delivered to right office
- Delivered to right person

# FedX Layers



# Layered Software Systems

- Network software
- Operating systems
- Windowing systems

# Exercise

- Come up with an example of a layered system.
- Describe the interface and/or peer-to-peer protocols for your example.



# OSI Reference Model

The International Standards Organization (ISO) proposal for the standardization of the various protocols used in computer networks (specifically those networks used to connect open systems) is called the Open Systems Interconnection Reference Model (1984), or simply the OSI model.

# OSI Reference Model

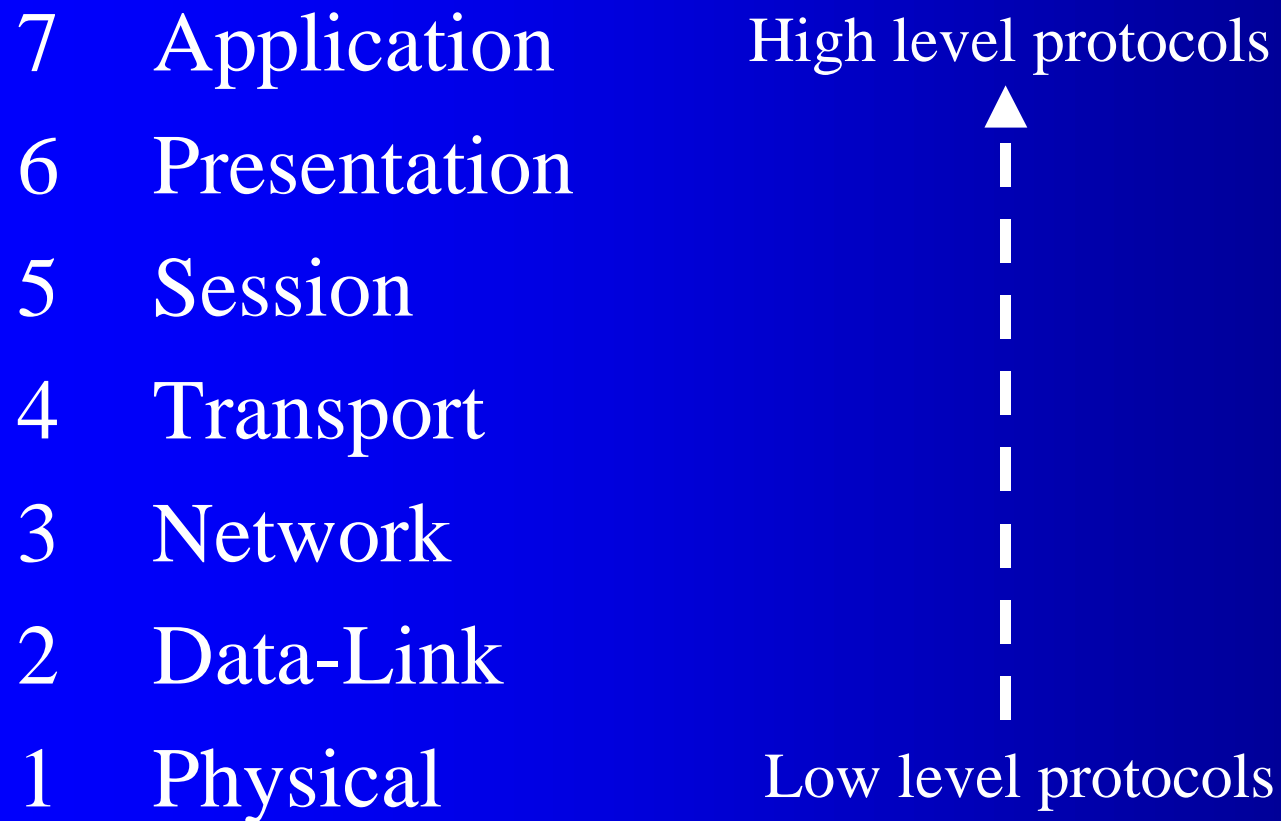
Although the OSI model is a just a model (not a specification), it is generally regarded as the most complete model (as well it should be - nearly all of the popular network protocol suites in use today were developed before the OSI model was defined).

# OSI <-> Network Software

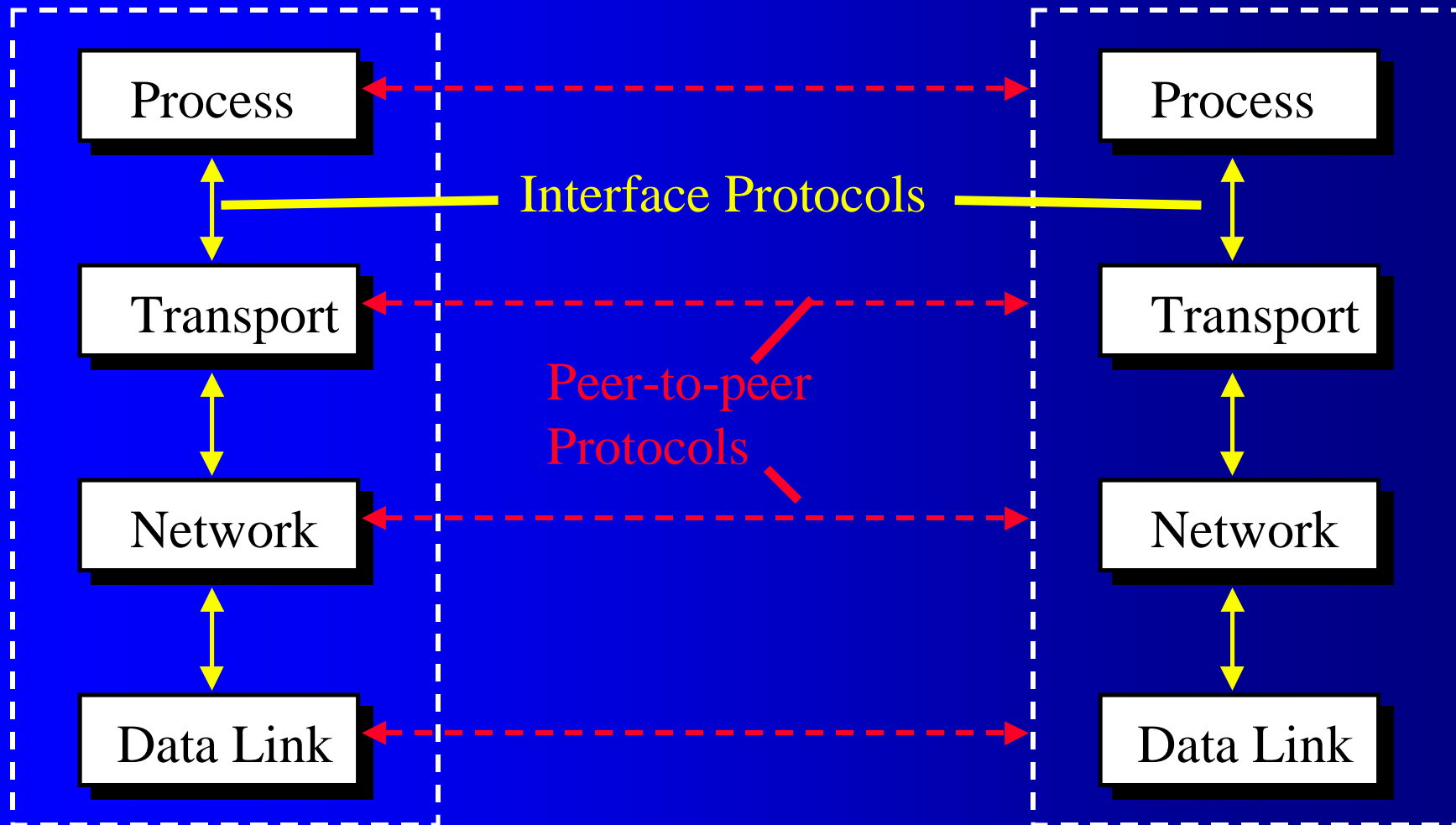
Although this course is about network programming (and not about networking in general), an understanding of a complete network model is essential.

We will look at the OSI Reference Model in detail later in the course.

# OSI 7 Layer Model:



# Simplified Network Model



# Programs & Processes

- A *program* is an executable file.
- A *process* or *task* is an instance of a program that is being executed.

# Client - Server

- A *server* is a process not a machine !
- A server waits for a request from a client.
- A client is a process that sends a request to an existing server and (usually) waits for a reply.

# Client - Server Examples

- Server returns the time-of-day.
- Server returns a document.
- Server prints a file for client.
- Server does a disk read or write.
- Server records a transaction.

# Servers

- Servers are generally more complex (more interesting).
- 2 Basic types of servers:
  - ◆ *Iterative* - server handles one client at a time.
  - ◆ *Concurrent* - server handles many clients at a time.
- We will study the differences later.