

# CMSC 132: Object-Oriented Programming II

---



## Networking

Department of Computer Science  
University of Maryland, College Park

1

## Advanced Programming Concepts

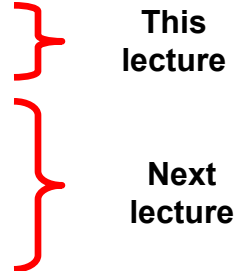
- Objected-oriented support in Java for
  - Exception handling
  - Streams
  - Threads
  - Graphics user interfaces (GUIs)
  - Networking
  
- Look at **networking** as example of OO design

2

# Overview

## ■ Networking

- Background
- Concepts & terms
- Java's objected-oriented view
- Java's networking API  
(Application Program Interface)
- Network applications



3

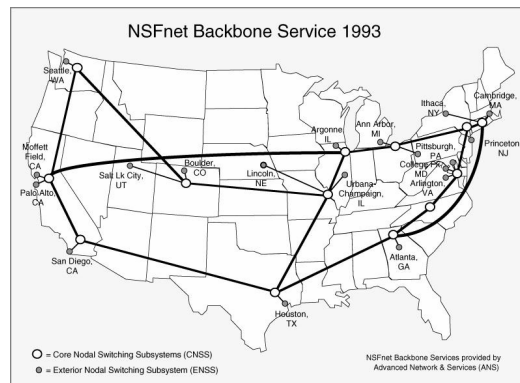
# Networking Background

## ■ Definition

- Set of computers using common protocols to communicate over connecting media

## ■ History

- 1969 ARPANET
- 1986 NSFnet
- 1995 Internet



4

## Networking Concepts

- Protocols
- Network model
- Internet addresses
- Ports
- Sockets
- URLs
- Connection (TCP) vs. packet oriented (UDP)
- Reliability

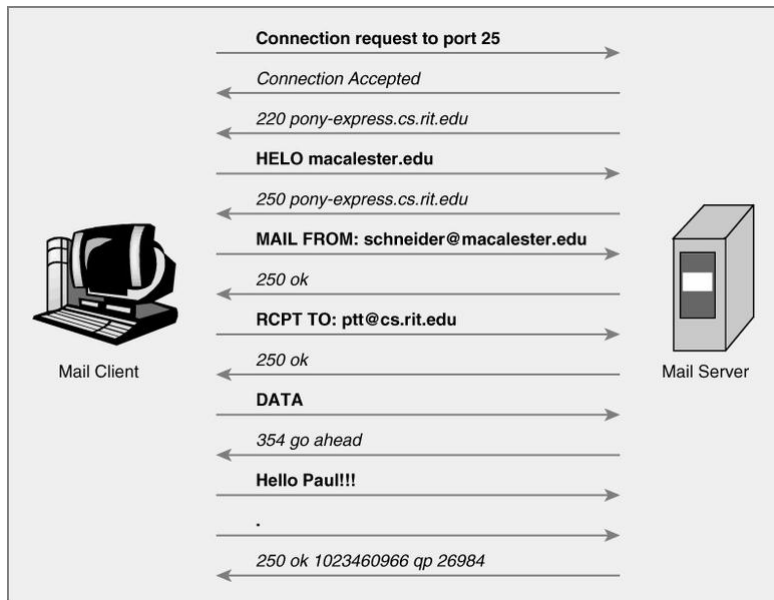
5

## Protocols

- Definition
  - Formal description of formats and rules
- Used for
  - Message formats
  - Sequence & order of actions
- Needed by computers to exchange information
- Vital for networking

6

## Protocols – Email Delivery



7

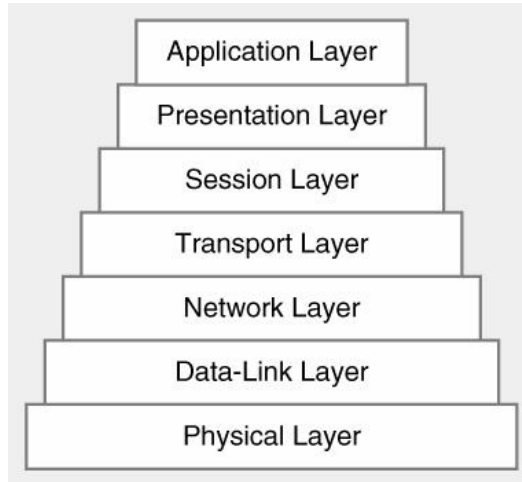
## Protocol – HTTP GET (Web Page)

- **Client connects to server on port 80**  
GET /~pugh/index.html HTTP/1.0  
<blank line>
- **Server responses with HTTP headers**  
HTTP/1.1 200 OK  
Date: Mon, 20 Feb 2006 03:47:44 GMT  
Server: Apache  
Last-Modified: Wed, 15 Feb 2006 01:17:09 GMT  
ETag: "9b2b1c-948-1222af40"  
Accept-Ranges: bytes  
Content-Length: 2376  
Connection: close  
Content-Type: text/html; charset=ISO-8859-1
- **Followed by blank line, then contents of response**

8

## Network Model

- **Open Systems Interconnection (OSI) model**
  - Multiple layers (7)
  - One function each
  - Each layer relies on previous layer
- **Designed to reduce complexity using abstraction**



9

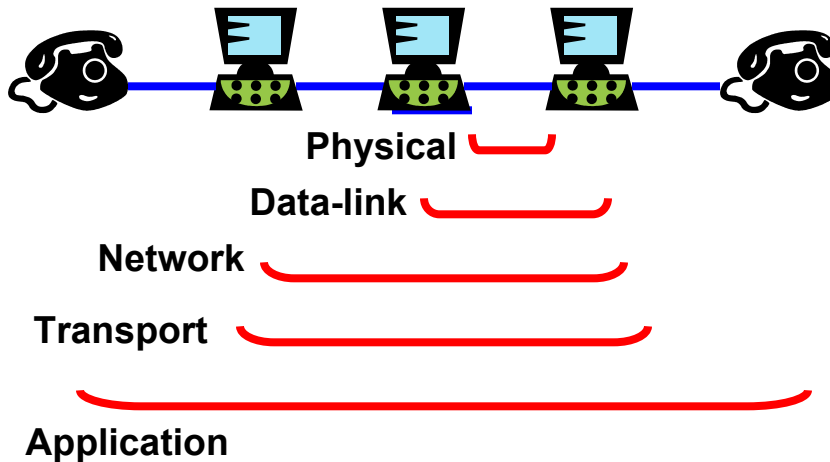
## Network Model – Layers

- **Physical layer**
  - Transmit data as 0's and 1's over connection
- **Data-link layer**
  - Between two physically connected computers
- **Network layer**
  - Between any two computers connected to network
- **Transport layer**
  - Deliver network data to application
- **Application layer**
  - Between two applications using network

10

## Network Model – VOIP Example

### ■ Voice over IP (VOIP)



11

## Internet (IP) Address

### ■ Unique address for machine on internet

- Get from ISP when connecting to internet
- Allows network to find your machine

### ■ Format

- 32-bit unsigned integer ⇒ 128.8.128.8
- Domain name ⇒ cs.umd.edu

### ■ Name and address for local machine

- Localhost
- 127.0.0.1

### ■ Machine can have multiple IP addresses

- Virtual machines

12

## Internet (IP) Address

### ■ Problem

- Running out of 32-bit IP addresses
- Caused by initial address allocation
  - Stanford & MIT given more IP addresses than China

### ■ Switching to 128-bit IP addresses in IPv6

- 1+ million addresses per square meter on Earth

13

## IP Address – DNS

### ■ Domain Name System (DNS)

- Protocol for translating domain names to IP addresses
  - Example: cs.umd.edu → 128.8.128.44
- Multiple DNS servers on internet
- DNS server may need to query other DNS servers
  - edu DNS server queries umd.edu server to find cs.umd.edu

14

## IP Address – DHCP

- **Dynamic Host Configuration Protocol (DHCP)**
  - Protocol used by networked computers to obtain
    - Unique IP addresses
    - Default router, subnet mask
    - IP addresses for DNS servers
  - DHCP server on local network
  - Used when computers first connect to network
  - Settings (leases) periodically refreshed

15

## IP Address – NAT

- **Network Address Translation (NAT)**
  - Rewriting source / destination IP addresses
    - As data passes through router
  - Permits sharing of single IP address among multiple computers
  - Used in home networking (**NAT box**)



16

## IP Address – NAT

### ■ NAT approach

- NAT box connects to both internet & private network
  - Obtains single public IP address from ISP
  - Assigns local IP address to computers on private network
- For data traveling to / from internet
  - NAT box merges and translates IP addresses
    - Local IP addresses ↔ public IP address
- From Internet
  - All private computers appear to have same IP address

17

## Ports

### ■ Abstraction to identify (refine) destination

- Provide multiple destinations at single IP address

### ■ Format

- Unsigned 16-bit integer (0 to 65,535)
- Ports 0 to 4096 often reserved & restricted

### ■ Many ports pre-assigned to important services

- 21 ftp (file transfer)
- 23 telnet (remote terminal)
- 25 SMTP (email)
- 80 http (web)
- ...

18

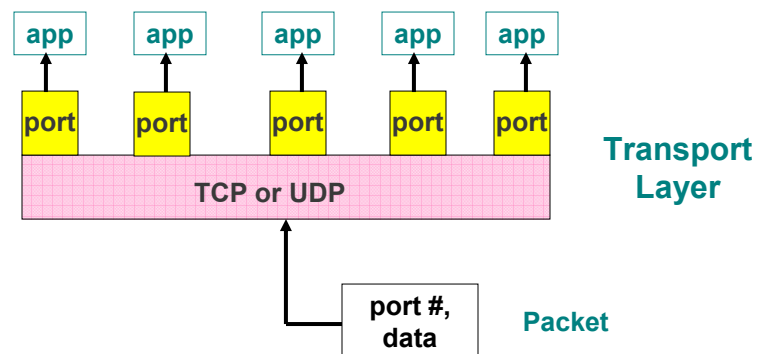
## Sockets

- **Application-level abstraction**
  - Represents network connection
  - Implemented in software
  - Supports both UDP and TCP protocols
- **History**
  - Introduced in Berkley UNIX in 1980s
  - Networking API

19

## Sockets

- **Socket is bound to port number**
  - Receives data packet
  - Relays to specific port



20

## Uniform Resource Locators (URLs)

### ■ Represent web resources

- Web pages
- Arbitrary files
- ...

### ■ Examples

- <http://www.cs.umd.edu/index.html>
- [ftp://www.cs.umd.edu/pub/doc/csd\\_policies.pdf](ftp://www.cs.umd.edu/pub/doc/csd_policies.pdf)
- <https://login.yahoo.com/>
- <file://dir/my.txt>

21

## Uniform Resource Locators (URLs)

### ■ Consists of

- Protocol
  - http
  - ftp
  - https (secure http)
  - file
  - ...
- IP address (or domain name)
- Port (optional)
  - <http://www.cs.umd.edu:80/>
- Reference to anchor (optional)

22

## Internet Connections

- Two types of connections
  1. Connection-oriented (TCP)
  2. Packet-oriented (UDP)

23

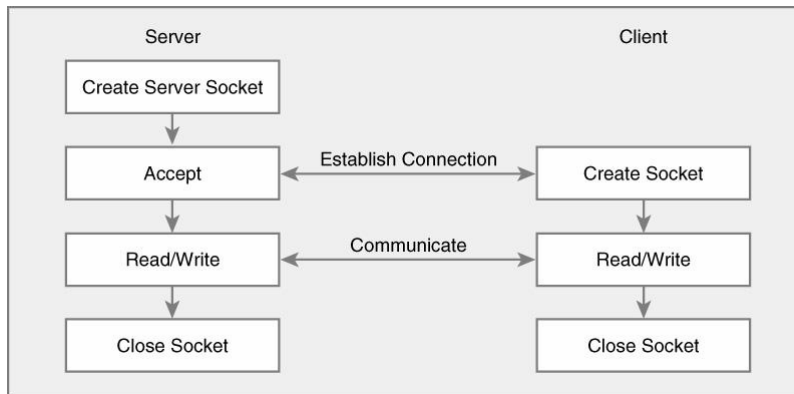
## Connection Oriented

- Approach
  - Reserve (single) communication channel
  - Send **stream** of data along channel
- Also called
  - Circuit switching
  - Stream oriented
- Example
  - Telephone call (current)

24

## Connection Oriented

### ■ Protocol



25

## Connection Oriented

### ■ Advantages

- Simpler scheme
- Easier to use
- Higher quality communication
  - Less likely to lose data (at network layer)

26

## Packet Oriented

### ■ Approach

- Break message up into **packets**
- Transmit packets separately
- Assemble packets at destination

### ■ Also called

- Packet switching
- Connectionless

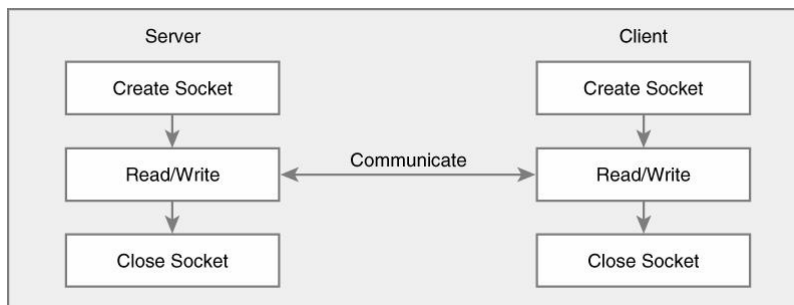
### ■ Example

- US Mail
- VOIP (Voice over IP)

27

## Packet Oriented

### ■ Protocol



28

## Packet Oriented

### ■ Advantages

- Higher utilization of channels
  - Can share communication channel
  - Can utilize multiple channels at once
- Can reroute around failed channels

29

## Internet

### ■ Network layer

- Internet Protocol (IP)

### ■ Transport layer

- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)

30

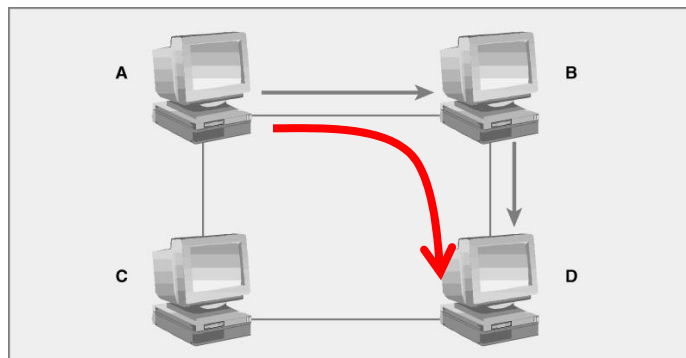
# Internet

OSI Model	Internet Model	Examples
APPLICATION	APPLICATION	Telnet, FTP, etc.
PRESENTATION		
SESSION		
TRANSPORT	TRANSPORT	TCP & UDP
NETWORK	INTERNET	IP
DATA-LINK	HOST TO NETWORK	Device Driver & Card
PHYSICAL		

31

## Internet Protocol (IP)

- Packet oriented
- Packets **routed** between computers
- Unreliable



32

## User Datagram Protocol (UDP)

- Packet oriented
- Message split into datagrams
- Send datagrams as packets on network layer
- Unreliable but fast
- Application must deal with lost packets
- Examples
  - Ping
  - Streaming multimedia
  - Online games

33

## Transmission Control Protocol (TCP)

- Connection oriented
- Message split into datagrams
- Send datagrams as packets on network layer
- Provides illusion of reliable connection
  - Extra messages between sender / recipient
  - Resend packets if necessary
  - Ensure all packets eventually arrive
  - Store packets and process in order
  - Provides warning if packets are lost

34

## Transmission Control Protocol (TCP)

- **Reliable but slower**
- **Application can treat as reliable connection**
  - **Despite unreliability of underlying IP (network)**
- **Examples**
  - **ftp (file transfer)**
  - **telnet (remote terminal)**
  - **http (web)**

35

## Reliability – Reliable

- **What is reliability?**
- **Reliable**
  - **Data guaranteed to**
    - **Arrive in order (if it arrives)**
  - **More overhead**
  - **Slower**
- **Reliability does not guarantee arrival**
  - **But will provide warning if data does not arrive**

36

## Reliability – Unreliable

### ■ Unreliable

- Data **not** guaranteed to
  - Arrive ⇒ lost data
  - Arrive in order ⇒ out of order data
- Less overhead
- **Faster!**
- Transfers responsibility to higher layer
  - Extra work for higher layer
  - Compensate with **timeouts**
    - Estimate packet lost if longer than average round trip

37

## Reliability

### ■ Reliable layers

- Data-link

### ■ Unreliable layers

- Physical
- Network

### ■ Can be either

- Transport
  - Reliable ⇒ TCP
  - Unreliable ⇒ UDP
- Application

38