CMSC 132: Object-Oriented Programming II

Networking

Department of Computer Science
University of Maryland, College Park
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
Overview

Networking

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

This lecture

Next lecture
Networking Background

Definition

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

History

- 1969 ARPANET
- 1986 NSFnet
- 1995 Internet
Networking Concepts

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

Connection request to port 25

Connection Accepted

220 pony-express.cs.rit.edu

HELO macalester.edu

250 pony-express.cs.rit.edu

MAIL FROM: schneider@macalester.edu

250 ok

RCPT TO: ptt@cs.rit.edu

250 ok

DATA

354 go ahead

Hello Paul!!!

.

.

250 ok 1023460966 qp 26984
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
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
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
Network Model – VOIP Example

Voice over IP (VOIP)

Physical

Data-link

Network

Transport

Application
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
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
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
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
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)
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
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)
  - ...

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
Sockets

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

Transport Layer

Packet

TCP or UDP

Port #, data
Uniform Resource Locators (URLs)

- Represent web resources
  - Web pages
  - Arbitrary files
  - ...

- Examples
  - https://login.yahoo.com/
  - file://dir/my.txt
Uniform Resource Locators (URLs)

Consists of

- Protocol
  - http
  - ftp
  - https (secure http)
  - file
- IP address (or domain name)
- Port (optional)
- Reference to anchor (optional)
Internet Connections

Two types of connections

1. Connection-oriented (TCP)
2. Packet-oriented (UDP)
Connection Oriented

- **Approach**
  - Reserve (single) communication channel
  - Send stream of data along channel

- **Also called**
  - Circuit switching
  - Stream oriented

- **Example**
  - Telephone call (current)
Connection Oriented Protocol

Server

Create Server Socket

Accept

Read/Write

Close Socket

Client

Create Socket

Read/Write

Close Socket

Establish Connection

Communicate
Connection Oriented

Advantages

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

Server
Create Socket
Read/Write
Close Socket

Client
Create Socket
Read/Write
Close Socket

Communicate
Packet Oriented

Advantages

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

Network layer
- Internet Protocol (IP)

Transport layer
- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)
Internet

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>Internet Model</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td>APPLICATION</td>
<td>Telnet, FTP, etc.</td>
</tr>
<tr>
<td>PRESENTATION</td>
<td>TRANSPORT</td>
<td>TCP &amp; UDP</td>
</tr>
<tr>
<td>SESSION</td>
<td>INTERNET</td>
<td>IP</td>
</tr>
<tr>
<td>TRANSPORT</td>
<td>HOST TO NETWORK</td>
<td>Device Driver &amp; Card</td>
</tr>
<tr>
<td>NETWORK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA-LINK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHYSICAL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Internet Protocol (IP)

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

- Reliable layers
  - Data-link
- Unreliable layers
  - Physical
  - Network
- Can be either
  - Transport
    - Reliable ⇒ TCP
    - Unreliable ⇒ UDP
  - Application