Networking Support In Java

Fawzi Emad
Chau-Wen Tseng

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
- Java’s object-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
- Reliability
- Connection vs. packet oriented
- TCP vs. UDP

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

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)**
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

---

Internet (IP) Address

- **Domain Name System (DNS)**
  - DNS servers on internet
  - Can look up IP address associated with name
  - DNS server may need to query other DNS servers
    - edu DNS server queries umd.edu server to find cs.umd.edu

- **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

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
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)
Reliability

- **Reliable**
  - Data guaranteed to
    - Arrive in order
  - More overhead
  - Slower

- **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

Ways To Connect

1. Connection-oriented
2. Packet-oriented
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

![Connection Oriented Protocol Diagram]

- Server:
  - Create Server Socket
  - Accept
  - Read/Write
  - Close Socket

- Client:
  - Establish Connection
  - Create Socket
  - Communicate
  - Read/Write
  - Close Socket
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

- Advantages
  - Can share communication channel
  - Higher utilization of channels
  - 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 Protocol (IP)

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

![Diagram of network communication](image)

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

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)