Distributed Information Processing

5th Lecture

Eom, Hyeonsang (엄현상)
Department of Computer Science & Engineering
Seoul National University
Outline

- Communication
  - Layered Protocols
  - Middleware Protocols
  - Types of Communication

- Communication Methods

- Q&A
Message Passing Requirements

Agreements Needed at a Variety of Levels

- Meaning of the Bits Being Sent
  - Character coding: e.g., EBCDIC and ASCII
- Number of Volts for a 1-bit
- Indication of the Last Bit of the Message
- Detection of Damaged or Lost Messages
- Lengths of Numbers, Strings, and Others
- Representations

Agreements from the low-level details of bit transmission to the high-level details of how information is to be expressed
Layered Protocols

ISO OSI (Open Systems Interconnection) Reference Model

- Designed to Allow Open Systems to Communicate
  - Open system is prepared to communicate with any other by using standard rules that govern the format, contents, and meaning of messages
    - Protocols: such rules formalized
      - Connection oriented
      - Connectionless

- Useful for Understanding Computer Networks

Protocol Suite (or Stack)

- Collection of Protocols Used in a System
Layers, interfaces, and protocols in the OSI model
Illustration: Layered Protocols (2)

A typical message as it appears on the network

[Diagram showing layered protocols with headers from data link, network, transport, session, presentation, and application layers, with bits that actually appear on the network at the bottom]
Middleware Protocols

An adapted reference model for networked communication

Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007 Prentice-Hall, Inc. All rights reserved
Types of Communication

- Synchronize at request submission
- Synchronize at request delivery
- Synchronize after processing by server

Viewing middleware as an intermediate (distributed) service in application-level communication

Persistent vs. Transient Communication
Asynchronous vs. Synchronous Communication
Communication Methods

- **RPC (Remote Procedure Call)**
  - Communication by Calling Remote Procedures
    - Definition of service interface
    - Lack of ability to create new object instances
    - Lack of support for remote object references

- **RMI (Remote Method Invocation)**
  - Communication by Calling Methods of a Remote Object
    - Implementation of a remote interface
    - Creation of new object instances
    - Support for remote object references
Communication Methods (Cont’d)

- **Socket**
  - Communication of Messages and Data between Processes
    - Use of a raw communication channel
    - Definition of a low-level message protocol
    - Definition of data transmission format

- **Distributed Event-Based Systems**
  - Communication via Event Subscription and Notification
    - Support for heterogeneity
    - Support for asynchronous communication
Middleware Approaches

- Location Transparency

- Applications
- RMI, RPC and events
- Request reply protocol
- External data representation
- Operating System

Coulouris, Dollimore and Kindberg  Distributed Systems: Concepts and Design  Edn. 4  © Pearson Education 2005
Method Invocation

- Local vs Remote Invocation
Distributed Object Model

A Remote Object and Its Remote Interface
Distributed Object Model (Cont’d)

Instantiation of Remote Objects

Coulouris, Dollimore and Kindberg   Distributed Systems: Concepts and Design   Edn. 4   ©  Pearson Education 2005
RMI Components

- **Proxy**
  - Forwarding Messages to a Remote Object and Receiving the Reply
  - Making RMI transparent to clients

- **Dispatcher**
  - Receiving the Request and Selecting the Appropriate Skeleton Method

- **Skeleton**
  - Implementing Methods in the Remote Interface
    - Unmarshalling arguments and invoking the method
RMI Components (Cont’d)

Illustration

Translating between Local and Remote Object References and Creating Remote Object References
RPC Components

Illustration