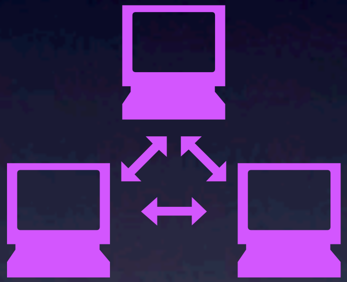
CISC 275: Introduction to Software Engineering

Lab 6: Introduction to Java Networking



Charlie Greenbacker
University of Delaware
Fall 2010

Overview

- Networking review
  - Client-Server model
  - Sockets & port numbers
- Code walkthrough of turn-based chat program
  - Server & client applications
  - Full code available on website
- Sample execution
- Lab exercise

Networking review

- Client-Server model:
  - Relationship between two computer programs
    - Either on the same computer or separate computers on the same network
  - Client program makes service requests to server program
  - Data is transmitted back and forth across a network connection
  - Server listens for incoming connections, client requests to connect, connection is established and data exchange begins

Networking review

- Socket:
  - One end of a two-way communication link between two programs running on a network
  - Socket classes represent connection between a server program and a client program
    - java.net.ServerSocket: server listens for incoming connection requests
    - java.net.Socket: client sends data to server, server receives data from client, and vice-versa

Networking review

- Port number:
  - Combined with IP address, specifies endpoint of network connection

Code walkthrough (server)

```
public class ChatServer {
    public static void main(String[] args) throws IOException {
        int portNumber = 9876;  // each team should choose a
                                // unique number

        ServerSocket serverSocket = null;
        Socket clientSocket = null;
        PrintWriter socketOut = null;
        BufferedReader socketIn = null, consoleIn = null;
        String toClient = "", fromClient = "";
```

Code walkthrough (server)

```
// display server IP address for client to input

System.out.println("Server IP address: " +
    InetAddress.getLocalHost().getHostAddress());

System.out.println("(client must enter this IP address to
    connect)\n");
```

Code walkthrough (server)

```
// listen for inbound connection

try {
    serverSocket = new ServerSocket(portNumber);
    System.out.print("Listing for inbound connection on " 
        "port " + portNumber + "... ");
} catch (IOException e) {
    System.err.println("Could not listen on port: " +
        portNumber + ". error: " + e);
    System.exit(1);
}
```

Code walkthrough (server)

```
 // accept connection request

try {
     clientSocket = serverSocket.accept();
     System.out.println("Connection established!");
 } catch (IOException e) {
     System.err.println("Accept failed. error: " + e);
     System.exit(1);
 }
```

Code walkthrough (server)

```
 // instantiate I/O objects

socketOut = new PrintWriter(clientSocket.getOutputStream(),
     true);

 socketIn = new BufferedReader(new
     InputStreamReader(clientSocket.getInputStream()));

 consoleIn = new BufferedReader(new
     InputStreamReader(System.in));
```

Code walkthrough (server)

```java
// loop conversation until "bye" is received
while (!fromClient.equalsIgnoreCase("bye")) {
    System.out.print("you: ");
    toClient = consoleIn.readLine();
    socketOut.println(toClient);

    fromClient = socketIn.readLine();
    System.out.println("them: " + fromClient);
}
```

Code walkthrough (server)

```
        // close I/O connections
        socketOut.close();
        socketIn.close();
        consoleIn.close();
        clientSocket.close();
        serverSocket.close();
    }
}
```

Code walkthrough (client)

```java
public class ChatClient {
   public static void main(String[] args) throws IOException {
       int portNumber = 9876;  // each team should choose
                               // a unique number

       Socket socket = null;
       PrintWriter socketOut = null;
       BufferedReader socketIn = null, consoleIn = null;
       String fromServer = "", toServer = "";
```

Code walkthrough (client)

```
// prompt user to input IP address of server
System.out.print("Enter server IP address (obtain from "
    + " server):");
consoleIn = new BufferedReader(new
    InputStreamReader(System.in));
String serverIP = consoleIn.readLine();
System.out.println("");
```

Code walkthrough (client)

```java
// connect to server
try {
    System.out.print("Trying to connect to server... ");
    socket = new Socket(serverIP, portNumber);
    System.out.println("Connection established!");
} catch (UnknownHostException e) {
    System.err.println("Don't know about host at: " +
        serverIP + ". error: " + e);
    System.exit(1);
} catch (IOException e) {
    System.err.println("Couldn't get I/O for the connection"
        + " at: " + serverIP + ". error: " + e);
    System.exit(1);
}
```

Code walkthrough (client)

```
// instantiate I/O objects
socketIn = new BufferedReader(new
    InputStreamReader(socket.getInputStream()));
socketOut = new PrintWriter(socket.getOutputStream(), true);
```

Code walkthrough (client)

```
// loop conversation until "bye" is entered
while (!toServer.equalsIgnoreCase("bye")) {
    fromServer = socketIn.readLine();
    System.out.println("them: " + fromServer);

    System.out.print("you: ");
    toServer = consoleIn.readLine();
    socketOut.println(toServer);
}
```

Code walkthrough (client)

```
        // close I/O connections
        socketIn.close();
        socketOut.close();
        consoleIn.close();
        socket.close();
    }
}
```

Sample execution

- Can run both server & client on same machine or on separate machines (far more interesting)
- Easiest to compile & run from command line instead of from inside Eclipse
  - Open two terminal windows if running both on same machine
- Following sample execution will show full process, starting from compilation, in two separate windows displayed in columns
  - Server on left, client on right

Sample execution

Sample execution

- javac *.java

-

Sample execution

- javac *.java
-

-

Sample execution

- javac *.java
- java ChatServer

-

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
```

-

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
```

- java ChatClient

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
```

- java ChatClient

```
Enter server IP address (obtain from server):
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
        (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
```

- java ChatClient

```
Enter server IP address (obtain from server):
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
```

- java ChatClient

```
Enter server IP address (obtain from server):
      128.175.13.74
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
    (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
    Connection established!

you:
```

- java ChatClient

```
Enter server IP address (obtain from server):
    128.175.13.74

Trying to connect to server...
    Connection established!
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
    (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
    Connection established!
```

```
you: hello
```

- java ChatClient

```
Enter server IP address (obtain from server):
    128.175.13.74

Trying to connect to server...
    Connection established!
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
      Connection established!

you: hello
```

- java ChatClient

```
Enter server IP address (obtain from server):
      128.175.13.74

Trying to connect to server...
      Connection established!

them: hello

you:
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
      Connection established!

you: hello
```

- java ChatClient

```
Enter server IP address (obtain from server):
      128.175.13.74

Trying to connect to server...
      Connection established!

them: hello

you: hi
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
      Connection established!

you: hello

them: hi

you:
```

- java ChatClient

```
Enter server IP address (obtain from server):
      128.175.13.74

Trying to connect to server...
      Connection established!

them: hello

you: hi
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
      Connection established!

you: hello

them: hi

you: a/s/l?
```

- java ChatClient

```
Enter server IP address (obtain from server):
      128.175.13.74

Trying to connect to server...
      Connection established!

them: hello

you: hi
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
      Connection established!

you: hello

them: hi

you: a/s/l?
```

- java ChatClient

```
Enter server IP address (obtain from server):
      128.175.13.74

Trying to connect to server...
      Connection established!

them: hello

you: hi

them: a/s/l?

you:
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
      Connection established!

you: hello

them: hi

you: a/s/l?
```

- java ChatClient

```
Enter server IP address (obtain from server):
      128.175.13.74

Trying to connect to server...
      Connection established!

them: hello

you: hi

them: a/s/l?

you: bye
```

Sample execution

- javac *.java
- java ChatServer

```
Server IP address: 128.175.13.74
      (client must enter this IP address to connect)

Listing for inbound connection on port 9876...
      Connection established!

you: hello

them: hi

you: a/s/l?

them: bye

[process terminated]
```

- java ChatClient

```
Enter server IP address (obtain from server):
      128.175.13.74

Trying to connect to server...
      Connection established!

them: hello

you: hi

them: a/s/l?

you: bye

[process terminated]
```

Lab exercise (in pairs)

- Download & run the code on two computers
  - One partner will run server, the other runs client
  - Both machines must be on the <u>same network</u>
- Be sure to choose a random high port number
  - And modify the code for both partners!
- Launch the programs & have a chat
  - Compile, start server, start client, enter server IP address into client, exchange messages
  - Don't submit – show me output before you leave