**Java Networking**
(lectures programs)

**Sun OnLine Documentations**

**Chapter 15 (of 2nd Edition)**

**TCP Sockets**

**Simple Client/Server:**

- **JabberServer:**

  A simple server that just echoes back whatever the client sends:

  1. create a server socket
     
     ```java
     ServerSocket s = new ServerSocket(PORT);
     ```
  
  2. accept a client connection
     
     ```java
     Socket socket = s.accept();
     ```
  
  3. get socket input stream
     
     ```java
     in = socket.getInputStream();
     ```
  
  4. get socket output stream
     
     ```java
     out = socket.getOutputStream();
     ```
  
  5. read
     
     ```java
     String str = in.readLine();
     ```
  
  6. write
     
     ```java
     out.println(str);
     ```

- **JabberClient:**

  A simple client that just sends lines to the server and reads lines that the server sends:

  1. create a client socket connected to the server
     
     ```java
     Socket socket = new Socket(addr, PORT);
     ```
  
  2. get socket input stream
     
     ```java
     in = socket.getInputStream();
     ```
3. get socket output stream
   `out = socket.getOutputStream();`
4. write
   `out.println(str);`
5. read
   `String str = in.readLine();`

**Multi-Threaded Client/Server:**

- **MultiJabberServer:**

  A server that uses multithreading to handle any number of clients:

  1. For each client connection accepted in socket, create a thread to handle it:
     ```java
     new ServeOneJabber(socket);
     ```
  2. Each thread will get in/out streams from socket to read/write.

- **MultiJabberClient:**

  Client that tests the MultiJabberServer by starting up multiple clients:

  1. Create many clients up to MAX_THREADS and sleep 100 ms after the creation of each client.

     ```java
     while(true) {
       if (JabberClientThread.threadCount() < MAX_THREADS) {
         new JabberClientThread(addr);
         Thread.currentThread().sleep(100);
       }
     }
     ```

  2. Each client sends 25 messages to the server and exits.

     ```java
     for (int i = 0; i < 25; i++) {
       out.println("Client " + id + ": " + i);
       String str = in.readLine();
       System.out.println(str);
     }
     out.println("END");
     ```
UDP Sockets

Example Client/Server:

- **Dgram:**

  A utility class to convert back and forth between Strings and DatagramPacket.

  ```java
  DatagramPacket toDatagram (String s, InetAddress destIP, int destPort)
  String toString (DatagramPacket p)
  ```

- **ChatterServer:**

  A server that echoes received datagrams:

  ```java
  DatagramSocket socket = new DatagramSocket(PORT);
  socket.receive(dp);
  String rcvd = String(dp.getData(), 0, dp.getLength()) +
                  dp.getAddress() + dp.getPort();
  socket.send(dp);
  ```

- **ChatterClient:**

  Tests the ChatterServer by starting 10 clients, each of which sends/recieves 25 datagrams to the server. The code of each client is outlined as:

  ```java
  DatagramSocket s = new DatagramSocket();
  s.send (dp);
  s.receive (dp);
  ```