Network Programming [CACS355]

Er.Sital Prasad Mandal

BCA- VI
Mechi Campus
Bhadrapur, Jhapa, Nepal

(Email : info.sitalmandal@gmail.com)
https://networkprogam-mmc.blogspot.com/

Text Book

- Elliotte Rusty Harold, “Java Network Programming” O’Reilly, 2014
- David Reilly, Michael Reilly, “ Java Netwoking Programming and DistributedComputing”

6. Sockets for Clients

1. Using Sockets
   i. Investigating Protocols with Telnet
   ii. Reading from Servers with Sockets
   iii. Writing to Servers with Sockets
2. Constructing and Connecting Sockets
   i. Basic Constructors
   ii. Picking a Local Interface to Connect From
   iii. Constructing Without Connecting
   iv. Socket Addresses
   v. Proxy Servers
3. Getting Information About a Socket
   i. Closed or Connected?
   ii. toString()

4. Setting Socket Options
   i. TCP_NODELAY
   ii. SO_LINGER
   iii. SO_TIMEOUT

Datagrams

1. Before data is sent across the Internet from one host to another using TCP/IP, it is split into packets of varying but finite size called *datagrams*.
2. Datagrams range in size from a few dozen bytes to about 60,000 bytes.
3. Packets larger than this, and often smaller than this, must be split into smaller pieces before they can be transmitted.

Packets Allow Error Correction

1. If one packet is lost, it can be retransmitted without requiring redelivery of all other packets.

2. If packets arrive out of order they can be reordered at the receiving end of the connection.

Abstraction

1. Datagrams are mostly hidden from the Java programmer.
2. The host's native networking software transparently splits data into packets on the sending end of a connection, and then reassembles packets on the receiving end.
3. *Instead, the Java programmer is presented with a higher level abstraction called a socket.*

Sockets

1. A socket is a reliable connection for the transmission of data between two hosts.
2. Sockets isolate programmers from the details of packet encodings, lost and retransmitted packets, and packets that arrive out of order.
3. There are limits. Sockets are more likely to throw IOExceptions than files.

Socket Operations

There are four fundamental operations a socket performs. These are:

- Connect to a remote machine
- Send data
- Receive data
- Close the connection
- Blind to port
- Listen for incoming data
- Accept connection from remote machines on the bound port

Reference

http://www.cafeaulait.org/slides/sd2000east/sockets/09.html
https://slideplayer.com/slide/5776239/
https://www.javatpoint.com/socket-programming
https://slideplayer.com/slide/6501980/