Nguyen Phan Yen Ngan – 14597

Chapter 2: Socket Addresses – TCP/IP SOCKETS in JAVA

2.1 Socket Addresses

- Recall that a client must specify the IP address of the host running the server program when it initiates communication. The network infrastructure then uses this destination address to route the client’s information to the proper machine.

- The InetAddress abstraction represents a network destination, encapsulating both names and numerical address information.

2.2 TCP Sockets

- Java provides two classes for TCP:

Socket and ServerSocket. An instance of socket represents one end of a TCP connection. A TCP connection is an abstract two-way channel whose ends are each identified by an IP address and port number. Before being used for communication, a TCP connection must go through a setup phase, which starts with the client’s TCP sending a connection request to the server’s TCP.

2.3 UDP Sockets

- UDP provides an end-to-end service different from that of TCP.

- 2 functions: 1) it adds another layer of addressing (ports) to that of IP, and 2) it detects some forms of data corruption that may occur in transit and discards any corrupted messages.

- Different with TCP: For example, UDP sockets do not have to be connected before being used.

- Another difference between UDP sockets and TCP sockets is the way that they deal with message boundaries: UDP sockets preserve them. This makes receiving an application message simpler, in some ways, than it is with TCP sockets.