Topic 3
Java Overview with Streams & Sockets

Part C

Object Streams

UDP and Datagrams

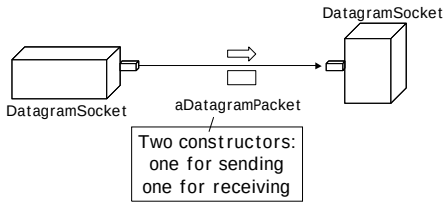
Datagram Networking with UDP

UDP

- An alternative to TCP/IP is UDP/IP
- IP provides the transfer of packets
- TCP guarantees delivery and sequence ordering of packets
- UDP packets may be lost
- But if they arrive, are guaranteed not to be corrupt

UDP and Java

- Class DatagramPacket
- Class DatagramSocket



Class DatagramPacket

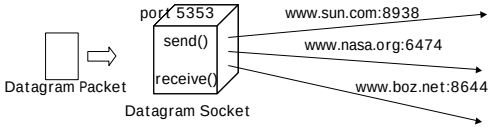
- DatagramPacket(byte ibuf[], int length)
  - for receiving **** bytes
  - ibuf is the byte array that will hold the packet contents
- DatagramPacket(byte ibuf[], int length, InetAddress iaddr, int port)
  - creates a packet of **** bytes from *** for transmission
  - a server must be listening at **** on ****

Class DatagramPacket

- InetAddress getAddress()
  - returns the InetAddress of packet
- int getPort()
  - returns port of packet
- byte[] getData()
  - extracts packet data into a byte array
- int getLength()
  - used to find the length of the actual UDP packet – usually less than buffer size

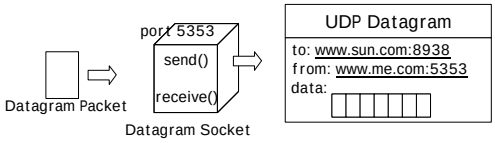
Class DatagramSocket

- Used to both send and receive DatagramPackets
- Like a TCP socket, it must be on a port



Class DatagramSocket

- Creates a UDP Datagram which is what actually gets sent



DatagramSocket Constructors

- DatagramSocket() throws SocketException
  - creates a DatagramSocket on randomly chosen port number
- DatagramSocket(int port) throws SocketException
  - creates a DatagramSocket on specific port number

DatagramSocket Methods

- void send(DatagramPacket p) throws IOException
  - sends packet to destination address specified in packet
- synchronized void receive(DatagramPacket p) throws IOException
  - receives a single UDP packet into DatagramPacket p. The packet can be queried for source and port.

Sending UDP

```
String host = www.sun.com;
int port = 8282;

byte[] message = {'a', 'b', 'c'};

DatagramSocket socket = new DatagramSocket(1459);
DatagramPacket packet = new DatagramPacket(message, 3,
                    InetAddress.getByName(host), port);
socket.send(packet);
```

Receiving UDP

```
int port = 8282;
byte[] incoming = new byte[10];

DatagramSocket socket = new DatagramSocket(port);
DatagramPacket packet = new DatagramPacket(
                                    incoming, incoming.length);
//wait until a packet shows up
socket.receive(packet);

//copy packet data into another byte array
byte[] msg = packet.getData();
```