Distributed Computing

CMSC 433
Distributed Computing

- Computations running across multiple machines
- Computations that shouldn’t fail just because one machine fails
Communications

• Generally, TCP and UDP
• TCP: stream-based two way communication
• UDP: one way packet communication
  – no built in acknowledgement
• Multicast UDP: broadcast packet communication to local network
IP terminology

- IP address
  - hostname

- Port
  - port space for UDP and TCP are separate
  - each service typically has a standard port for initiating communication
TCP Server Sockets

• bound to a particular port
• accept() waits for a connection request on that port
  – returns a standard Socket for further communication
  – ServerSocket can be immediately reused
InetAddress

• `InetAddress.getByName(h)`
  – static factory for InetAddress
• `InetAddress.getLocalHost()`
  – gets local host name (does DNS)
• `InetAddress.getByName(“127.0.0.1”)`
  – gets local loopback address, no DNS
Socket

- new Socket(InetAddress a, int port)
  - creates a TCP socket connection to a server socket at the specified address and port
- For each socket, you can call
  inputStream()  
  outputStream()
Streams

- You should all know InputStream, OutputStream, Reader, Writer
- DataOutputStream adds writeInt, writeLong, writeDouble, writeUTF
- DataInputStream provides matching read methods
Object Streams

• ObjectOutputStream and ObjectInputStream allow objects to be sent
  – any serializable object

• Any object that implements Serializable can be serialized
  – if all of its fields are serializable
Object graphs

• Writing to an object output stream writes a graph of objects
  – writing a map sends all the elements
  – writing one node of a graph sends all connected nodes

• Reconstructed into an isomorphic graph
potential problems

• You write a class that received doesn’t have
  – error

• You write a different version of a class than the receiver has
  – maybe OK
RMI

- Remote method invocation
- Provides remote proxies for objects
  - invoke method on local proxy/stub
  - method is invoked over the network, using object serialization
  - wait for response
remote code base

• RMI provides an extension to object serialization
• allows you to provide a URL from which your classes can be downloaded in received doesn’t have them