§24.7: UDP Packets: Connectionless Client/Server

2 April 2007
CMPT167
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Review last time

- IP, TCP vs. UDP
- TCP server
  - ServerSocket object (in java.net)
    - .accept()
  - Socket object
    - .getInputStream()
- TCP client
- Servers and multithreading

Quiz 5 (10 minutes)

- What is object serialization? What is needed for our objects to be serializable? [4]
- What kind of stream is used to output a serialized object? What method is used to actually do the output? [3]
- What is buffered I/O, and why do we care? [4]
- Contrast TCP with UDP. [4]
- Write Java code to open the file "input.dat" for reading, move to the 200th byte, and read an integer stored in binary format. (multiple possible solutions) [5]

Quiz 5: answers #1-2

- What is object serialization? What is needed for our objects to be serializable?
  - Formatting the data in an object for I/O on a stream
  - Declare that the class implements Serializable
  - All instance variables must be Serializable, too
    - Unless declared transient
- What kind of stream is used to output a serialized object? What method is used to actually do the output?
  - ObjectOutputStream, .writeObject()

Quiz 5: answers #3-4

- What is buffered I/O, and why do we care?
  - I/O not immediately committed, but placed in intermediate buffer until ready for flush
  - Often better throughput from combining multiple writes
- Contrast TCP with UDP.
  - TCP: connection-oriented, more overhead
  - UDP: connectionless, no guarantee of delivery, packets may arrive out of order or duplicated

Quiz 5: answers #5

- Write Java code to open the file “input.dat” for reading, move to the 200th byte, and read an integer.
  - RandomAccessFile file =
    new RandomAccessFile(“input.dat”, “r”);
  - file.seek(200);
  - int myint = file.readInt();

What's on for today

- Connectionless client/server networking with UDP
- Receiving a UDP packet (server)
- Sending a UDP packet (client)

Connectionless client/server

- TCP is connection-oriented
- UDP is connectionless
  - Send data one packet at a time
    - Similar to envelopes through CanadaPost
    - Fragment larger data into multiple packets
  - Packets might:
    - Not arrive at all
    - Arrive out of order
    - Get duplicated
  - Less overhead, better latency and possibly better throughput

Receiving a UDP packet

- DatagramSocket (in java.net):
  - sock = new DatagramSocket(*port*);
  - Block/wait for a packet
    - sock.receive(packet);
- DatagramPacket:
  - Data payload:
    - byte data[] = new byte[100];
    - packet = new DatagramPacket(data, data.length);
  - Read packet:
    - packet.getData(), packet.getLength()
    - packet.getAddress(), packet.getPort()

Sending a UDP packet

- Prepare payload:
  - String msg = “Hello, World!”;
  - data = msg.getBytes();
- Package payload:
  - packet = new DatagramPacket(
    data, *length*, *hostname*, *port*);
- Send packet:
  - socket.send(packet);

TODO

- Lab5 due Wed 11Apr:
  - File I/O
  - Store inventory and point-of-sale system
  - Worth 60 points
- Last day for submitting late labs is Fri 13Apr
- Last day of classes is Mon 16Apr
- Final exam is Fri 20Apr 2-4pm