Problem Set 1: Internetworking

Each student should submit his own report: NO TEAM work.

Due date: 11:59PM Eastern, January 26th, 2013.

Late submissions will result in a 10% penalty per day (e.g., 2.5 days late result in 25% penalty)

1. Internetworking

Describe in detail all the steps that your internet browser goes through when you click on a web page such as http://www.northeastern.edu/. You should describe which protocols are invoked (e.g., TCP, ARP, DNS, ethernet), their parameters (e.g., port numbers, addresses), network entities (e.g., DNS server, default gateway/router) and the network stack structure.

Provide screen dumps (or packets listing) from a packet sniffer such as wireshark to confirm your description.

Hints: clear your machines arp tables before clicking on the web page link, use information from ipconfig/ifconfig, route, etc.

2. Sockets Communication

Write a simple Java client-server broadcast chat application. The server listens at a specified UDP port, and waits for GREETING messages from remote clients. Once a greeting has been received the client may send a MESSAGE command to the server, which will forward the contents in an INCOMING message to every remote point that has previously sent a GREETING (including this sending client).

The client creates a socket which it will maintain throughout its lifetime, and is capable of receiving and sending packets from/to the server.

The types and format of messages to be exchanged in the application are:

1. GREETING: Greets the server. Client to server only. The server should register the client who sent this message as active.

2. MESSAGE: Sends some text to the server for further distribution. Contains the text of the message to send. Client to server only.

3. INCOMING: The server has received some text, and is passing it along with the sender's ip and port. Contains the IP address and port of sender as well as the text of the message. Server to client only.

4. Any other message (e.g. ICMP errors) must be ignored.

A sample run of your application must work as follows:

- server$ java ChatServer 9090 runs the server on port 9090
  Server Initialized... Server is left running
- user1$ java ChatClient server-ip 9090 runs the client and GREETs the server.
  ] Prompt message from user
  ] Hello World! Sends a message
  <From w.x.y.z:aa>: Hello World! Every client sees this, where w.x.y.z:aa is the sender's ip address and port.
  ]

Note the following:

- The messages do not need to be authenticated, confirmed nor encrypted.
- There are no usernames whatsoever.
- The server sends the message to everyone who has GREETED him. There is no mechanism to make the server "forget" or logout clients.
- There is no limit to the number of clients the server supports.

3. Grading Policy

For Question 1, be comprehensive and specific about all the protocols involved. For Question 2, here are some guidelines on how it will be graded (modulo some final adjustments):

- code compiles: 10 points
- code working with one client: 15 points
- code working with multiple clients: 15 points
- code quality: 20 points
  Exception Handling: 5
  Modularity: 5
  Variable Names: 3
  Comments: 3
  Readability, Indentation, etc: 4