CS380 — Project 5

February 2, 2015

Due: Wednesday, February 11, 2015 (80 points)

Description

In this project, you'll be implementing UDP on top of IPv4. To do this, you will create IPv4 packets as in project 3, then create UDP packets as the data for the IPv4 packets. These will then be transmitted to the server. The server will first check the IPv4 headers to ensure the IPv4 packet is constructed correctly, then check the IPv4 data to get the UDP packet and check the header and data.

Our programs will perform the following procedure:

(1) First, we'll do a "handshaking" step where you send a single IPv4 packet with 4 bytes of data hard-coded to `0xDEADBEEF`. Then, my server will respond with 2 bytes of raw data (not an IP packet, so you don't need to be able to parse incoming IP packets) which you should treat as an unsigned 16-bit integer corresponding to a port number.

(2) You will then send UDP packets with the destination port header field assigned to the number you received in the previous step. You should send packets as follows: send 12 total packets with data size starting at 2 bytes and doubling each time. The UDP source port can be set to anything you like but the data should be randomized[1].

(3) After sending each UDP packet, wait for the server response which will contain exactly 4 bytes hard-coded to `0xCAFEBABE` if your packet was constructed correctly. Print out the elapsed time in milliseconds since you sent the packet for each packet transmitted. After sending all 12 packets print the average RTT for the packets.

If there is an error in a packet that you send, instead of `0xCAFEBABE` I will send one of the following error codes:

Code	Reason
`0xBAADF00D`	Problem with IPv4 portion of packet (see project 3 to fix!)
`0xCAFED00D`	Incorrect destination port in UDP packet
`0xDEADC0DE`	Invalid UDP checksum
`0xBBADBEEF`	Incorrect UDP data length

The host to connect to is 76.91.123.97 and the port is 38005. It should be available by Wednesday, February 4. If you still need to test your project 3 solution to ensure you are sending IPv4 packets correctly, you can do that on port 38003.

Your IPv4 headers will have the same restrictions as project 3 except the protocol should be set for UDP instead of TCP. Keep in mind that the UDP checksum is calculated using the same method but different input. It includes the UDP data and a "pseudoheader" including part of the IPv4 header.

[1] You can use Java's Random class and the `nextBytes(byte[] b)` method.

Submission

Submit a single Java file, UdpClient.java to Blackboard. You should hardcode the above host and port information. Don't package your classes. If you use more than one class, include all of them in the single `UdpClient.java` file.