

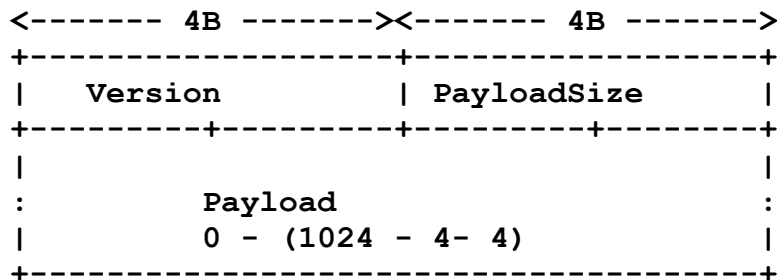
Goal:

For this Lab, we want write a simple UDP client that will send a message to an existing UDP server. The goal of this lab is to gain experience with the Java/Android networking API, UPD, Wireshark, and Eclipse.

The client we want to build will send a single UDP packet across the network to a server. The packet is specified below. The Payload is interpreted by the server as a NULL terminated String of 1 byte characters and displayed to the screen. Two UDP Servers are provided, a Java and C implementation, respectively. Your client must work with both servers.

The Packet:

The packet layout is as follows (not to scale!):



Strings in the payload MUST be NULL terminated

Version and PayloadSize are each 4 byte signed integers. The maximum size of the entire packet is 1024 bytes. The Payload has a size of between 0 to 1016 (1024-4-4), inclusive. For this assignment Version must be set to 1 for the server to process the packet correctly.

Servers:

The Java server is here: barney.cs.pacificu.edu (64.59.233.211) port 12345.
The C server is here: barney.cs.pacificu.edu (64.59.233.211) port 12346.

Existing Code:

Download the project SimpleUDPSendLab and import that project into Eclipse. This project includes two files, SimpleUDPPacket.java and SimpleUDPClientLab.java. The Packet class represents the UDP data your client will send and the ClientLab file is the client containing the main() method. The first step you must perform is to Refactor the package name to **edu.pacificu.cs.cs360.PUNetID.Lab** You will receive a warning but you may press Continue.

SimpleUDPPacket.java

A number of private member variables and getter/setter methods are provided in SimpleUDPPacket. Your need to write the code for **buildPacket()**, which must store the Packet into the byte array (rawPacket) for later transmission across the wire. buildPacket() must return the size of the packet. Return 0 if any errors occur.

Later, we will write **parsePacket()** that will take a byte array read from the network and parse the data into the private member variables of the class.

SimpleUDPClientLab.java

You need to write the **main()** method in this class. This method must read the Internet Address and port to connect to from the command line arguments (**args[0]** and **args[1]**, respectively). This method must use the following code to read a line of input from the keyboard. Build a UDP packet to send that line of input to the Internet Address:port provided.

The client needs to create a **DatagramSocket**, a **SimpleUDPPacket**, and **DatagramPacket**. Finally, send the **DatagramPacket** via the **DatagramSocket**.

```
BufferedReader bufferedReader = new BufferedReader( new  
    InputStreamReader(System.in));  
String text = bufferedReader.readLine();
```

Wireshark:

Wireshark allows you to inspect all the network traffic that traverses a particular network interface (you will be using **eth0**). If you run Wireshark while your UDP client is sending packets to the server you should be able to see the contents of your packet.

Hint:

byte[] String.getBytes() Encodes this String into a sequence of bytes using the platform's default charset, storing the result into a new byte array.

Package Names:

All of your labs for this course must be in the package:
edu.pacificu.cs.cs360.PUNetID.Lab

All of your projects for this assignment must be in:
edu.pacificu.cs.cs360.PUNetID.Lab.UPDLab

Grading:

Demonstrate your running code to me before 2:15pm on Feb 8, 2012. Ideally, demonstrate the running code before you leave the lab today.

If you have time in class:

Build a UDP server to replace the instructor's.