

# CS380 — Project 3

January 19, 2015

Due: Wednesday, January 28, 2015 (80 points)

## Description

In this project, you will be implementing IPv4. I will have a server running that will accept IPv4 packets, verify the checksum, and reply with “good” if the packet is formatted properly, it can correctly verify the checksum, and the amount of data sent matches the expected size of the packet.

If the server detects a problem with one of your packets, it will respond with a line of text indicating what it thinks the problem is. If you receive an indication from the server that a packet was bad, the server will then close the connection, so don’t try to send any more packets. Instead, restart your program and make a new connection for future packets.

In the following table, I will tell you whether or not to correctly implement a header field. If I say do not implement, simply fill that field with all zeros.

Version	Implement
HLen	Implement
TOS	Do not implement
Length	Implement
Ident	Do not implement
Flags	Implement assuming no fragmentation
Offset	Do not implement
TTL	Implement assuming every packet has a TTL of 50
Protocol	Implement assuming TCP for all packets
Checksum	Implement
SourceAddr	Implement with an IP address of your choice
DestinationAddr	Implement using the provided IP address
Options/Pad	Ignore (do not put in header)
Data	Implement

You must correctly implement the checksum algorithm as provided in the book in section 2.4. It is provided in C, so you must translate to Java. Although it is calculated in 16-bit chunks, you should use `int` in your intermediate calculations so that you can easily check for overflow. Be careful with this algorithm, Java does not provide support for unsigned math by default so make sure that you use zero-extension and not sign-extension when moving from `byte` to `short` or `int` or `short` to `int`.

Send the packets as raw bytes through the socket’s output stream. The host will be the same as the previous project: 76.91.123.97 on port 38003. The server should be running by Wednesday, January 21.

For more information about the implementation of IPv4, see your textbook, the RFC specification at <https://tools.ietf.org/html/rfc791>, or the Wikipedia article about IPv4: <https://en.wikipedia.org/wiki/IPv4>.

Your program must send packets in the following way when run: you will send 12 total packets. The data size will start at 2 bytes, then double each time. The contents of the data do not matter. You can either fill the data with zeros or randomize the contents.

## **Submission**

Submit a single Java file, `Ipv4Client.java` to Blackboard. You should hardcode the above host and port information. Do not use packaging for your classes. If you use more than one class, combine them all in to the same `Ipv4Client.java` file.