CS 3251 – Programming Assignment 1

Abhishek Shroff

September 23, 2010

1 Submission

This tar.gz file contains3 directories – *bin*, *doc*, and *src*.

The *doc* directory contains this readme, as well as the output file that shows the replies received from the TCP and UDP servers.

The *bin* directory contains binary files created as a result of executing 'make' in the *src* directory.

The *src* directory contains all the source files that are required for this project. It also contains a Makefile that contains targets for common builds. Apart from the standard first *all* target, tt contains a *clean* target that completely removes everything inside the *bin* directory, and a *debug* target that compiles everything in debug mode, that shows what is happening inside both ends of the program.

2 Protocol

2.1 TCP

The TCP part is fairly straightforward. The TCP client is not supposed to disconnect from the server before the end of the entire transaction. There will only be a performance penalty if it does. There is a simple request-reply paradigm that the suite follows.

The client first establishes a connection with the server. There is no additional handshake on top of TCP. Once the connection is established, the client sends a triple separated by greater than or equal to one space in the middle of each. The first two components must be integers, and the last component must be a symbol in {+, -, *, /, %}. The result of each operation is an integer that is relayed back to the client as well as stored on the server. Once the client receives the response, it procedes sending the next two components of the expression because the server will automatically treat the previous result as the first operand.

If at any point during the transaction, the client sends a request that does not consist of a triple, the server responds with the string *invalid*. The server tries to cope with denial of service by terminating the connection with invalid requests.

The client ends the transaction with the server by sending the *end*. This terminates the connection on the server side, and the server starts waiting for any more connections.

2.2 UDP

The UDP server responds to single stateless requests. It takes in a triple in the same format as the TCP part – *integer integer operation*, and replies back with the result.

A UDP client must send in a triple, get the response, and treat the response as the first operand of the next operation. Since this a datagram connection, there is a chance of packet loss. The client waits for a response from the server, and times out between 3 and 8 seconds. It allows for 5 maximum timeouts before it decides that the server is not available.

3 Limitations

The servers are fairly robust, and will not terminate on any input from the clients. That said, there are some limitations, which, if surpassed, will break this program.

1. The size of data inside a single request may not exceed 511 bytes.

2. All operations are performed on a closed set of integers, which means floating points are not supported by either input or as a result of a calculation. Floating points will be truncated.