Assignment 2B Report

By Jose Eduardo Soto SID#5673

The strategy to solve the problem was to continue from the Assignment 2A code that was already developed.

The first task was to figure out HOW to add to the code already produced. Eventually the author learned what could be done. In every line of code that required to use the TCP connection was replaced with a new module called *ReliableDataTransfer(RDT).* The responsibility of the module was to implement the GBN and SR algorithms.

There are several portions of the FTP application that needed to change. The Client and Server now both are required to state if it will use the GBN or SR algorithms. Rather than extending the *Socket* module provided, the author decided the *RDT* module be composed of a socket.

Producing the *RDT* module ran into multiple challenges:

1. The first challenge that occupied the author was how to make the Client and Server connect. Recall that *connect* function now had to be defined by the *RDT.* It relates with the *accept* and *bind* functions in its requirement and level of difficulty. The authors occupation ran longer than scheduled. After some consideration, the author decided to instantiate the socket as a SOCK_STREAM/TCP type and use the provided *sendto* and *recvfrom* functions.
2. The second challenge was to utilize the timer module provided. The author could not find a way to utilize the module to the implementation. Therefore, *settimeout* was used.

Once past the two challenges. The author utilized the time left to produce the GBN and SR algorithms.

GBN chosen first to be implemented. The code was implemented but debugging did not prove productive and still requires more time to complete. As a consequence, SR was not implemented.

There is no working software. But the usage is simple. Start two seperate terminals and run *python3 server-ftp.py* and walk through the prompt. *Python3 client-ftp.y* to begin the dialoge. Once begun the program runs.