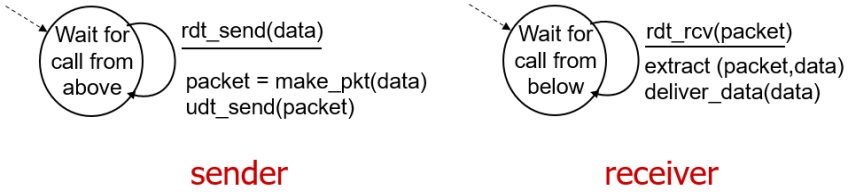
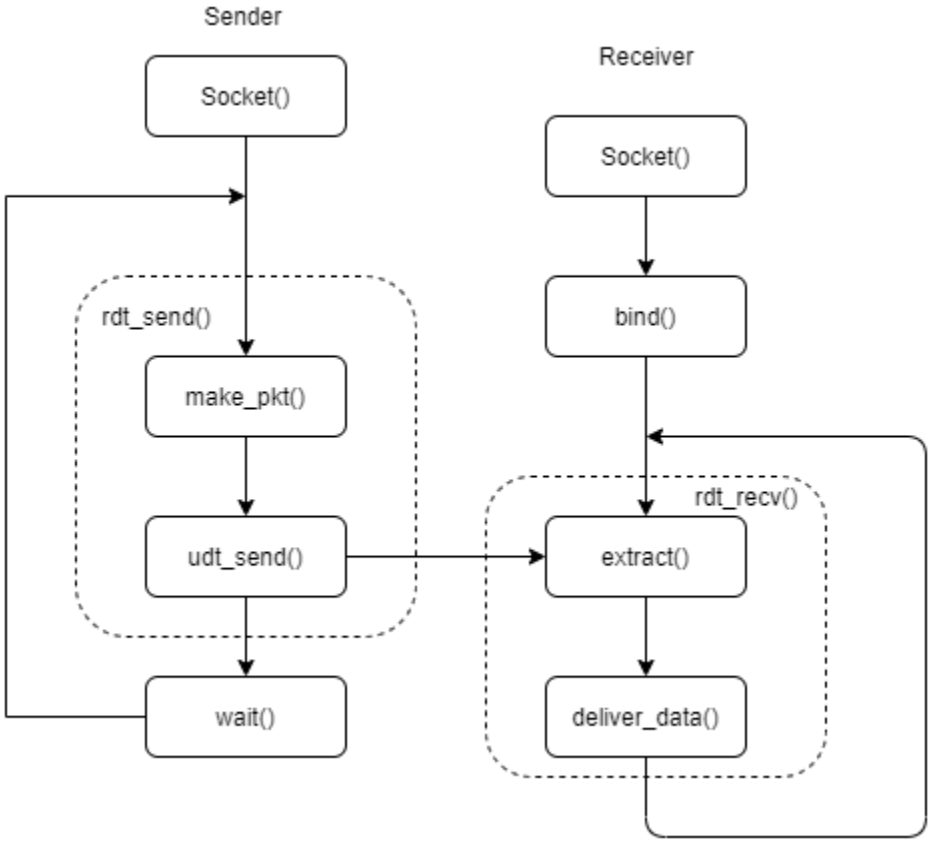
ECE 355 Course Project 1

Project description

The objective of this project is to implement the reliable transfer protocol rdt 1.0 in the textbook. The figure below shows the state diagram of the protocol. They describe the functions that are needed for the sender and receiver, respectively. You will need to utilize the socket programming to send a packet from the sender to the receiver. You can use either UDP or TCP socket. But the UDP socket is recommended due to its simple management.





Sender:

On the Sender side, you need to create a sender program. The sender can act like the client in the socket programming. It creates a socket to be connected to the receiver.

The main function needs to send the following sentence, “This is a test”, to the receiver. It sends one letter at a time. It calls the rdt_send() function, which then calls the make_pkt() to create a packet, and uses udt_send() to send it out. The udt_send () is the function that creates an UDP segment and send it to the receiver through the socket channel.

In the wait() function, please wait for 1 second to simulate the delay.

Receiver:

On the Receiver side, you need to create a program to receive data from the sender side. In the main program, the receiver acts as a server for the socket channel. It needs to creates a socket first, and then call the bind () function so that the receiver can listen on the socket for incoming packets.

Next, Reciever called the rdt_recv() function to receive the UDP packet through socket and print out the received data. There are two functions called in rdt_recv(): extract() and deliver_data()

The extract () function extracts the packet data from the UDP socket packet.

The deliver_data() is the function to print out the receive data.

Testing

You will need to implement all the functions described above, and compile both the sender and the receiver program. They can be run on the same machine. The receiver needs to print the received data whenever one packet is received.

Submission

- You can use one of the following programming languages for the project: Python, C++, Java. You need to provide a detailed README file telling users how to compile the code.
- You need to provode a PDF file project report with screenshots to show you compile the files, and run your program on the sender and receiver, with screenshots to show the program printout.
- Zip the report, source code, and README file, and submit it on the Blackboard.