Rsocket.h

1. This file just contains the definitions of our MRP socket function.
2. r_socket : function contains all the same description as normal socket function.
3. r_bind : bind function which helps in bind
4. r_sendto : transmit a message to another socket
5. r_recvfrom : function shall receive a message from a connection-mode or connectionless-mode socket.
6. r_close : closes a file descriptor, so that it no longer refers to any file and may be reused
7. runnerR : This function is the runner function for thread R which handles all the receiving messages and updates the UnAcknowledged table and received table.
8. runnerS : this is runner function for the thread S which handles all the timeout and the retransmission of the unacknowledged
9. dropMessage : this function drops the message in the recv function when the random value is above a certain threshold

Rsocket.c

1. For the unacknowledged message an array of struct is used where each struct has the following parameters
    a. id : which specifies the id
    b. msg : stores the message
    c. Msg_len : length of the message
    d. Tim : time at which this message
    e. Flags : the set of flags
    f. Destination : the destination port address
    g. Addrlen : the addrlen of the destination port
    h. Lock : a mutex lock to ensure that there are no data race conditions
2. For the received message again the array of structs were declared with following parameters
    a. id : which specifies the id
    b. msg : stores the message
    c. Source : the source port of the message
    d. Addrlen : the addrlen of the destination port
3.
4. Following are the list of extra functions and their brief descriptions
    a. void printUnack(unACK obj) : used of debugging purpose to print the message
    b. void initialize_tables() : function to initialize the unacknowledge table and the received message tables
    c. int getEmptyIndex_unACK() : to find an empty index in the unacknowledged table
    d. int getEmptyIndex_recvMsg() : to find a non empty index in the recvMessage table
    e. size_t produceFinalMessage(int id, char* buf, int len) : to return the combine length of the character array and the int value appended to it.
    f. int decodeRecvMessage(char* msg, int *id) : to decode the received message whether it is Acknowledgment message or data message
    g. int getNumber(char* msg, int from) : to extract the id from the message recieved
    h. void HandleData() : This function is the main function to handle the received data
    i. ssize_t sendACK(int id, struct sockaddr_in addr, socklen_t addr_len) : this function sends the acknowledgement message for id

j. void UpdateACK(int id) : it is the function which updates the unacknowledgment table with ACKnowledgement messages received.
k. int UpdateRecvMsg(int id, char *buf, struct sockaddr_in source_addr, socklen_t addr_len) : This function handles the case when the message received is a data so it updates the received message array accordingly.

p	n_t	len(string)	ratio
0.05	25	23	1.08
0.10	27	23	1.17
0.15	28	23	1.21
0.20	33	23	1.43
0.25	34	23	1.47
0.30	39	23	1.69
0.35	44	23	1.91
0.40	50	23	2.17
0.45	55	23	2.39
0.50	59	23	2.56