msocket.h

Constants:

- `T`: Timeout value.
- `probability_of_drop`: Probability of dropping a message.
- `MAX_BUFF_SIZE`: Maximum buffer size.
- `SOCK_MTP`: Maximum transmission attempts.
- `MAX_SOCKETS`: Maximum number of sockets.

Data Structures:

`SOCK_INFO`

- Fields:
  - `int sock_id`: Socket ID.
  - `char IP[20]`: IP address.
  - `int port`: Port number.
  - `int errno_val`: Error number.
- Purpose: Represents information about a socket.

`sender_window`

- Fields:
  - `int wsize`: Window size.
  - `int nack[5]`: Negative acknowledgment array.
- Purpose: Represents the sender's window for reliable data transmission.

`receiver_window`

- Fields:
  - `int wsize`: Window size.
  - `int nack[5]`: Negative acknowledgment array.
- Purpose: Represents the receiver's window for reliable data reception.

`sockets`

- Fields:
  - `int status`: Socket status

    -2: marked to close as soon as transmission is complete.

    -1:closed

    0:socket created but not yet bound

1: created and bound

- `int pid`: Process ID.
- `int sockid`: Socket ID.
- `int port`: Port number.
- `char ip[20]`: IP address.
- `char sbuff[10][MAX_BUFF_SIZE+1]`: Sender buffer.
- `char rbuff[5][MAX_BUFF_SIZE+1]`: Receiver buffer.
- `struct sender_window swnd`: Sender window.
- `struct receiver_window rwnd`: Receiver window.

- Purpose: Represents socket information and buffers.

Functions:

`int m_socket(int family, int type, int protocol)`

- Parameters: `family` - Address family, `type` - Socket type, `protocol` - Protocol.
- Returns: Socket file descriptor.
- Description: Creates a socket.

`int m_bind(int sockfd, const struct sockaddr_in *src_addr, socklen_t src_len, const struct sockaddr_in *dest_addr, socklen_t dest_len)`

- Parameters: `sockfd` - Socket file descriptor, `src_addr` - Source address, `src_len` - Length of source address, `dest_addr` - Destination address, `dest_len` - Length of destination address.
- Returns: 0 on success, -1 on failure.
- Description: Binds a socket to a specific address.

`int m_sendto(int sockfd, const void *buf, size_t len, int flags, const struct sockaddr_in *dest_addr, socklen_t addrlen)`

- Parameters: `sockfd` - Socket file descriptor, `buf` - Buffer containing the message, `len` - Length of the message, `flags` - Flags, `dest_addr` - Destination address, `addrlen` - Length of destination address.
- Returns: Number of bytes sent.
- Description: Sends a message to a specific address.

`int m_recvfrom(int sockfd, void *buf, size_t len, int flags, struct sockaddr_in *src_addr, socklen_t *addrlen)`

- Parameters: `sockfd` - Socket file descriptor, `buf` - Buffer to store the received message, `len` - Maximum length of the message, `flags` - Flags, `src_addr` - Source address, `addrlen` - Length of source address.
- Returns: Number of bytes received.
- Description: Receives a message from a specific address.

`int m_close(int sockfd)`

- Parameters: `sockfd` - Socket file descriptor.
- Returns: 0 on success, -1 on failure.
- Description: Closes a socket.

`int dropMessage(float p)`

- Parameters: `p` - Probability of dropping the message.
- Returns: 1 if the message should be dropped, 0 otherwise.
- Description: Determines whether to drop a message based on the probability.

msocket.c

Data Structures:

1. `struct SOCK_INFO`

- Fields:
  - `int sock_id`: Socket ID.
  - `char IP[16]`: IP address.
  - `int port`: Port number.
  - `int errno_val`: Error number.
- Purpose: Stores information about a socket.

2. `struct sockets`

- Fields:
  - `int sockid`: Socket ID.
  - `int status`: Socket status (-1: Free, 0: Bound, 1: Connected, -2: Closed).
  - `int pid`: Process ID.
  - `char ip[16]`: IP address.
  - `int port`: Port number.
  - `struct wnd rwnd`: Receiver window.
  - `struct wnd swnd`: Sender window.
  - `char rbuff[5][512]`: Receive buffer.
  - `char sbuff[10][512]`: Send buffer.
- Purpose: Represents a socket in the socket table along with its associated data.

3. `struct wnd`

- Fields:
  - `int wsize`: Window size.
  - `int nack[5]`: NACK array.
- Purpose: Defines a window structure for sender and receiver windows.

Functions:

1. `void attach_shared_memory()`

- Description: Attaches shared memory segments for the `SM` and `sock_info` structures.

2. `void detach_shared_memory()`

- Description: Detaches shared memory segments for the `SM` and `sock_info` structures.

3. `void reset_sock_info(struct SOCK_INFO* sock_info)`

- Parameters: `sock_info` - Pointer to a structure containing socket information.
- Description: Resets the values in the `sock_info` structure.

4. `int m_socket(int family, int type, int protocol)`

- Parameters: `family` - Address family, `type` - Socket type, `protocol` - Protocol type.
- Returns: Socket file descriptor on success, -1 on failure.
- Description: Creates a socket and initializes the socket structure.

5. `int m_bind(int sockfd, const struct sockaddr_in *src_addr, socklen_t src_len, const struct sockaddr_in *dest_addr, socklen_t dest_len)`

- Parameters: `sockfd` - Socket file descriptor, `src_addr` - Source address, `src_len` - Length of source address, `dest_addr` - Destination address, `dest_len` - Length of destination address.
- Returns: 0 on success, -1 on failure.
- Description: Binds a socket to a specific source and destination address.

6. `int m_sendto(int sockfd, const void *buf, size_t len, int flags, const struct sockaddr_in *dest_addr, socklen_t addrlen)`

- Parameters: `sockfd` - Socket file descriptor, `buf` - Pointer to the data to send, `len` - Length of the data, `flags` - Flags, `dest_addr` - Destination address, `addrlen` - Length of destination address.
- Returns: Number of bytes sent on success, -1 on failure.
- Description: Sends data through a socket to a specific destination.

7. `int m_recvfrom(int sockfd, void *buf, size_t len, int flags, struct sockaddr_in *src_addr, socklen_t *addrlen)`

- Parameters: `sockfd` - Socket file descriptor, `buf` - Pointer to store received data, `len` - Maximum length of data to receive, `flags` - Flags, `src_addr` - Source address, `addrlen` - Length of source address.
- Returns: Number of bytes received on success, -1 on failure.
- Description: Receives data from a socket along with the source address.

8. `int m_close(int sockfd)`

- Parameters: `sockfd` - Socket file descriptor.
- Returns: 0 on success, -1 on failure.
- Description: Closes a socket and marks its entry in the socket table as free.

initmsocket.c

Constants:

- `T`: Timeout value.

Variables:

- `struct SOCK_INFO *si`: Pointer to a structure containing socket information.

- `time_t stime[25][10]`: Array to store the time of sending for each socket and each message.
- `time_t atime[25]`: Array to store the latest time at which ack size was sent for each socket.
- `int last_ack[25]`: Array to store the index of the last acknowledgment for each socket.
- `int last[25]`: Array to store the index of the last index on which the last in-order message was stored for each socket.
- `int eflag`: Flag to indicate whether the program should terminate.
- `int last_ack_sno[25]`: Array to store the sequence number of the last acknowledged serial number for each socket.
- `int semwrite, semread, SemClose`: Semaphore IDs for controlling access to shared resources.
- `pthread_t pd[3]`: Array of pthread identifiers.

Functions:

`int max(int a,int b)`
- Parameters: `a`, `b` - Integers.
- Returns: Maximum of the two integers.
- Description: Returns the maximum of two integers.

`int M_bind(int sockfd, const struct sockaddr *source_addr, socklen_t source_len)`
- Parameters: `sockfd` - Socket file descriptor, `source_addr` - Source address, `source_len` - Length of source address.
- Returns: 1 on success, -1 on failure.
- Description: Binds a socket to a specific address.

`int M_socket()`
- Returns: 1 on success, -1 on failure.
- Description: Creates a socket.

`void *R(void* SM)`
- Parameters: `SM` - Pointer to a structure containing socket information.
- Returns: NULL.
- Description: Receives data from sockets and processes it.

`void *S(void* SM)`
- Parameters: `SM` - Pointer to a structure containing socket information.
- Returns: NULL.
- Description: Sends data through sockets.

`void *G(void* SM)`
- Parameters: `SM` - Pointer to a structure containing socket information.
- Returns: NULL.
- Description: Handles socket closure.

`void handler(int sig)`

- Parameters: `sig` - Signal number.
- Returns: Void.
- Description: Signal handler for SIGINT.

`int main()`

- Returns: 0 on successful execution.
- Description: Main function of the program, initializes sockets and handles threads for receiving, sending, and socket closure

Probability of drop	Average number of transmissions made to send each message
0.05	1.083333
0.1	1.104167
0.15	1.229167
0.2	1.33333
0.25	1.312500
0.3	2.02083
0.35	2.104167
0.4	2.208333
0.45	2.770833
0.5	2.895833

.