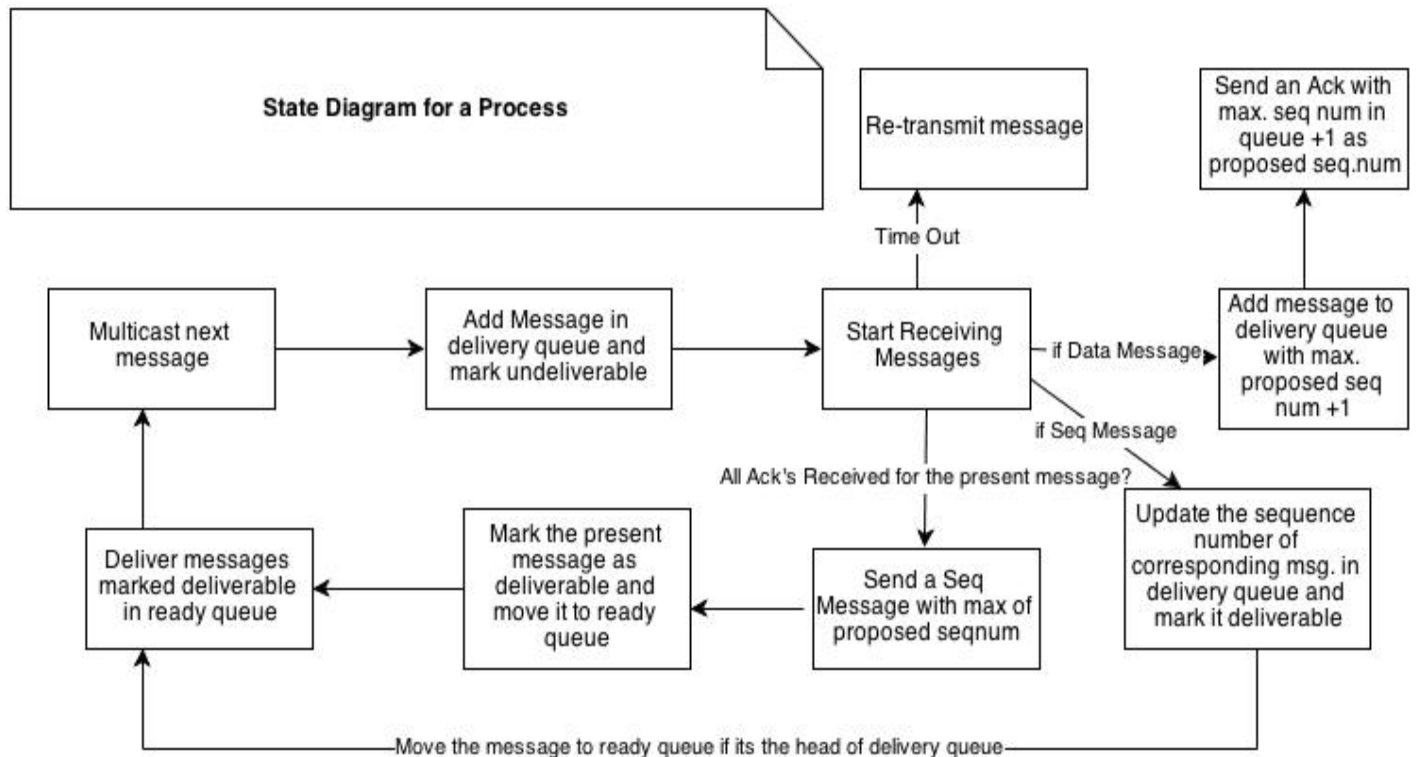


CS505 - Project 2 - Reliable Total Order Multicast Protocol

State Diagram:



System Architecture:

The system solves the problem of reliable total order delivery of messages in a network of processes each sending a given set of messages iteratively. Total order delivery essentially means the delivery of the same messages in the same order in all the set of processes in the system. Each message has a message identification number (id) and a sender identification number with which we identify the messages delivered. Each process sends messages reliably (retransmits after a given time out if it does not receive an acknowledgement). Each process maintains two queues - a delivery queue and a delivered queue. Once a process delivers message it also notes down the message id and the sender id in the delivered queue so that it won't accept duplicate messages. The delivery queue is a C++ hash map. The Key being the "sequence number, process id" pair. The value is the data message. C++ hash map stores all its <Key,Value> pairs in sorted order using a BST kind of a data structure.

Once a process gets all the acks, which is checked through a C++ map of <hostname,ack

flag> it sends a sequence message with the maximum of all the proposed sequence numbers in the acks to all the processes. The processes on receiving this sequence number, mark the corresponding message in their delivery queue as deliverable, reorder the queue and start delivering messages.

To keep track of sequence number, we update a variable of sequence in two cases:

1. Sending a message.
2. Receiving a sequence message.

Design Decisions:

1. All the nodes may not start at the same instant so used a retransmit timer to make all nodes receive messages.
2. Used `gettimeofday()` to both check the ack timer. The "Server.h" file contains the functions "get_now" to get the current time in milli seconds and "time_to_seconds" finds the difference between two given times. These can be changed in order to look for other scenarios.
3. A map of hostnames to IP addresses is also being used to track the hosts to which the message is to be sent or resent.
4. An ack will always be sent to the sender if the message is valid and not a duplicate which is already delivered.
5. A map is used to maintain a queue to make it sortable according to sequence number and process id.

Implementation Issues:

1. Delay guarantees may vary and retransmissions will happen accordingly.
2. The processes are assumed to be able to communicate with each other. If they can't open a socket or bind fails, the program will exit.
3. Some of the output messages revealing all events occurring may be suppressed. Please change the debug flag in "Server.h" to 1 to print all messages.