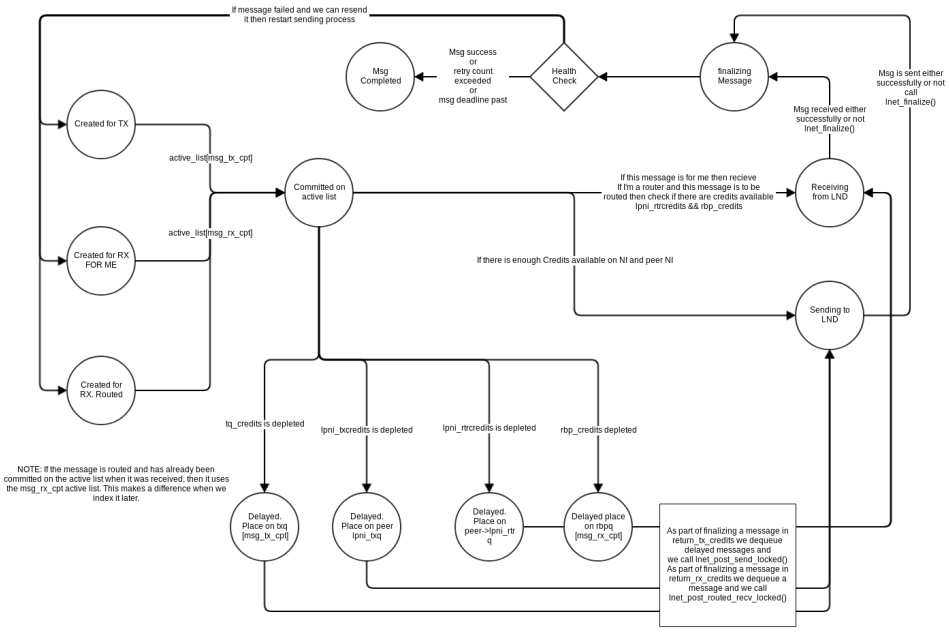
Message Life Cycle FSM



Overview

The states described in the diagram above are not exclusive state. For example a message can be both committed on the list active list and queued on a delayed queue. The way this is implemented in the code is by the use of bit flags to indicate what state the message is in at any given point.

Expiring Active Messages

We attempted to implement a mechanism by which we expire active messages which have been delayed past the message deadline. This seemed like a useful feature on the router specifically, since we can flush the queues on busy systems. This allows routers to continue servicing other requests. However, the implementation proved difficult due to many race conditions in the code. Therefore we decided to implement this as a separate patch.

The main issue with the implementation is on the routers and when expiring a REPLY. Before we go further the reason I lump the REPLY case with routing is because a REPLY is a special case. When a GET is parsed in lnet_parse_get() the message block used to receive the GET is also used to send the REPLY. One option is to free the GET message and create a new message for the REPLY message, I believe we do the same for the ACK message.

When a message is received the msg_rx_cpt is set by the use of lnet_cpt_of_nid() function. This function takes a look at the NI and passed to it, and finds a CPT which is associated with the NI. There is a feature where an NI can be restricted to a subset of the available CPTs on the system. This will restrict all processing and memory allocation related to the NI to his subset of CPTs, this includes message processing being sent or received on that NI.

When a message is sent the same logic is taken into account and msg_tx_cpt is set. In all circumstances except for router and REPLY cases only one of the msg_tx_cpt or msg_rx_cpt are valid.

In these two cases the msg_rx_cpt is set first in lnet_parse() upon receipt of the message and then when the message is posted the msg_tx_cpt is set.

When a message is received, in the router case, an appropriate router buffer is selected and if there are no credits available on that router pool, then the message is queued on the rbp_msgs queue. This message queue is indexed by the msg_rx_cpt. The message can alos be queued on the lpni_rtrq if there are no credits available there.

By the same token when sending a message the message can be queued if there are no credits either on the on the NI's transmit queue, ni_tx_queues, index by the msg_tx_cpt, or on the peer NI's lpni_txq.

To be able to detect if a message has been delayed past its deadline we will need to traverse the active message list, which itself is indexed by the cpt. However, depending whether the message was received first or whether it was sent first, the cpt will be the msg_rx_cpt in the former case while will be the msg_tx_cpt in the latter case.

To traverse the active message list and expire delayed messages we will need to:

1. Know which cpt to lock to traverse the active message list. Locking the EX lock is too heavy weight for the fast path.
2. When we encounter a message that has past its deadline we need to remove it off the delayed list, which means we need to lock the appropriate cpt lock, which could be different from the one we're currently holding.
    a. The cpt lock can be the msg_tx_delayed if the message is queued on one of the ni_tx_queues

    b. The cpt lock can be the msg_rx_delayed if the message is queued on one of the rpb_queues.
3. If we drop and relock the correct cpt, we need to deal with cases where the message might have been finalized and freed while we have the lock dropped.

We considered resolving this issue by having only one cpt. Calculate the cpt using the lnet_cpt_of_nid() if it has not already been set for messages which are routed or for a REPLY message. However, this solution breaks the feature where we need to restrict operations on the CPTs the NI to be used. This is of particular issue in these cases as the NI the message received on can be different from the NI a message is sent on. This is absolutely true for routed messages, since routed messages by definition need to go on a different network. If we use the same CPT, then credit calculations will be messed up.

At the end we decided that we do not see enough issues due to delayed messages to justify making risky changes under a tight timeline. It will be better to think through all the cases before making this change.