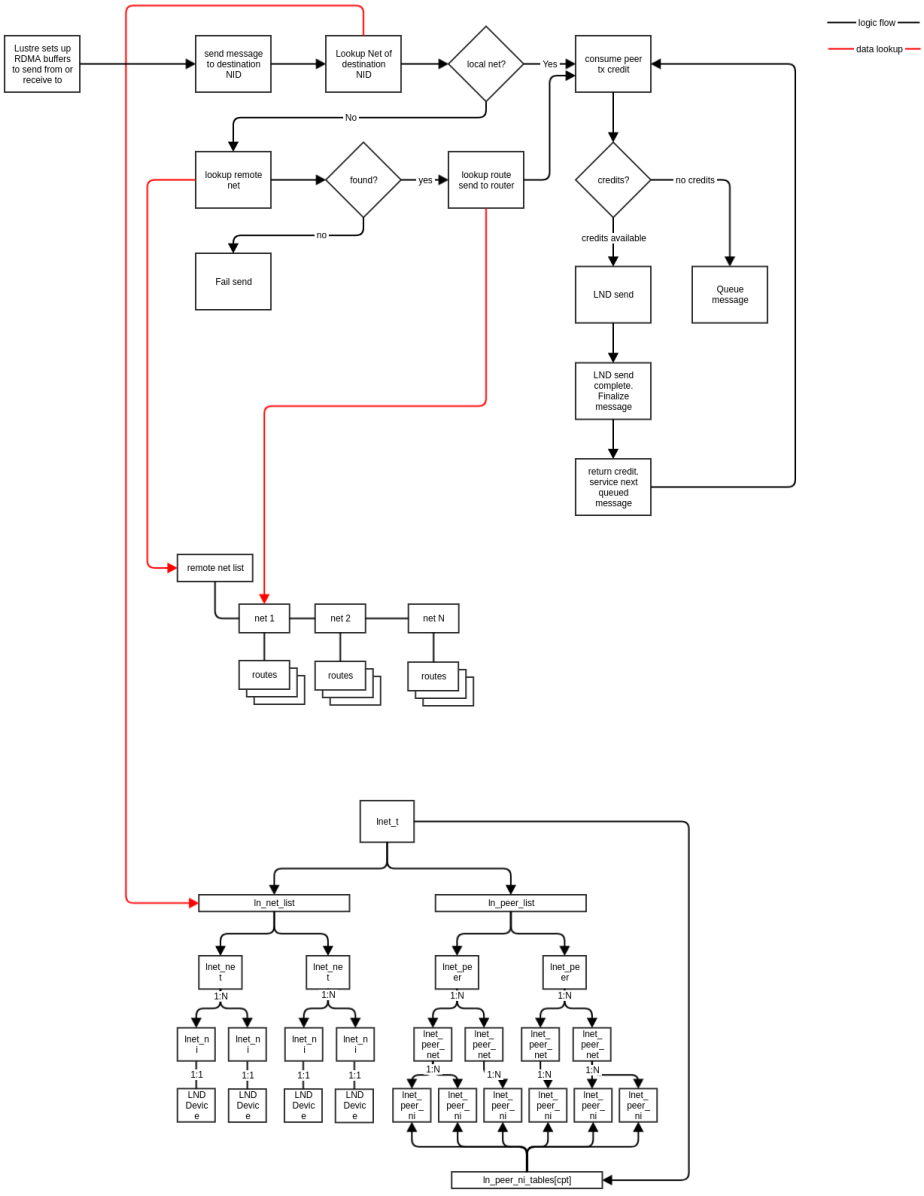
LNet Message Handling Overview

Active Sending

A node which actively sends a message can be a client or a server. The below flow diagram describes the message processing

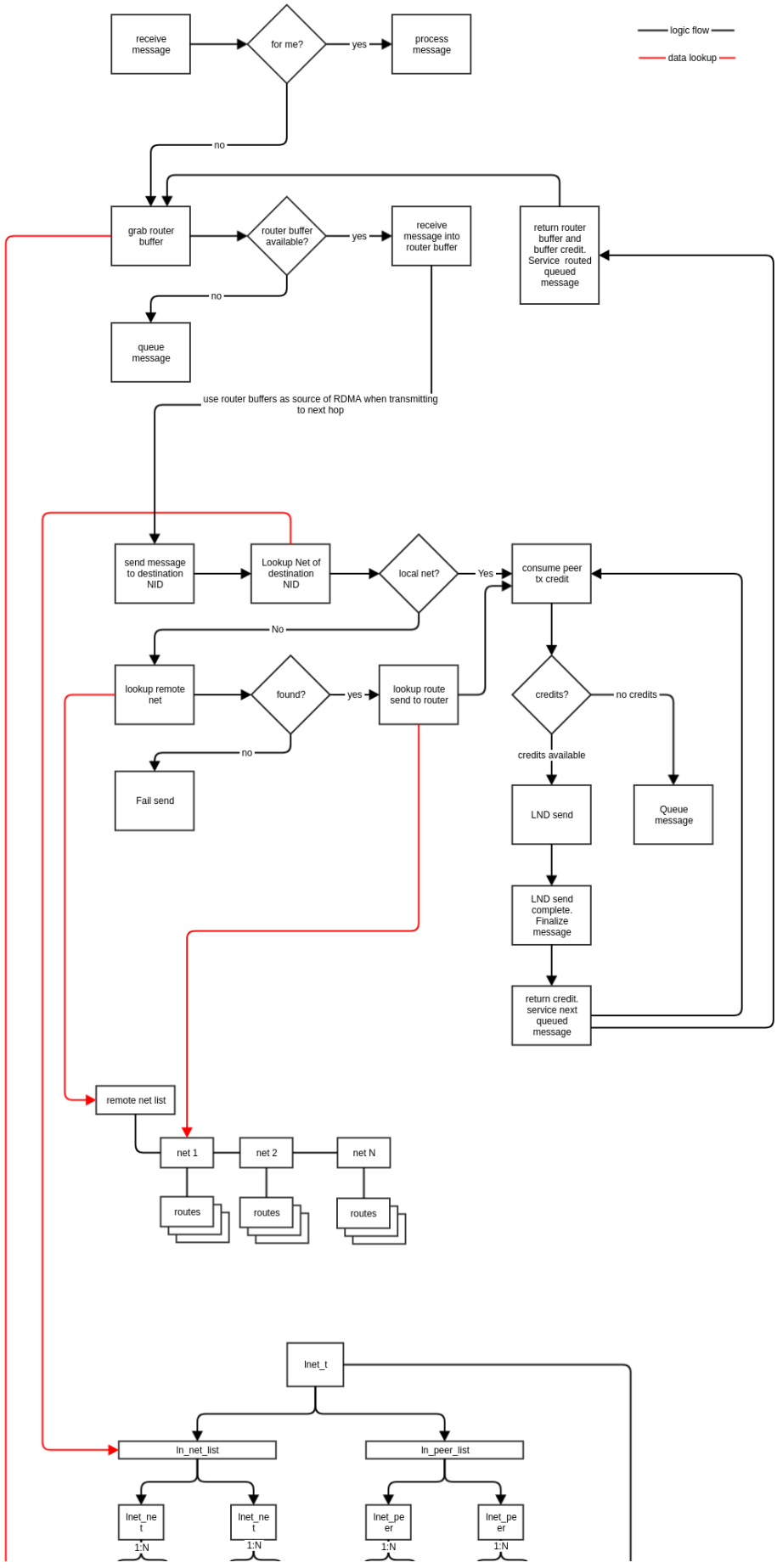


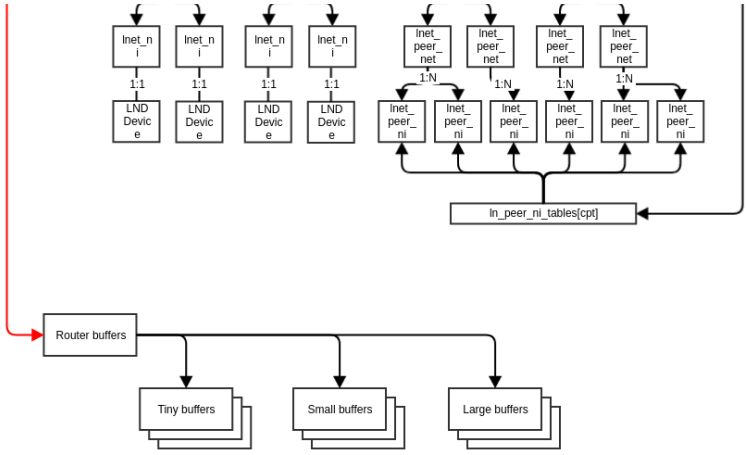
Textual Description

- Lustre requests a message send to a destination NID
- Node looks at the net of the NID and discovers that it is a remote net
- Node looks up the remote net and finds all the list of routes to that remote net
  - If remote net is not found then destination NID is not reachable
  - If there are multiple routes then one is selected based on priority, number of hops or in round robin
- Node starts the message sending process

- Node looks at the number of credits available for the gateway peer
  - If there a enough credits available (determined by peer_credits) then the message is sent to the LND for processing
  - If there aren't enough credits available, then the message is queued internally until another active send message is finalized and a credit is returned. Messages are then popped out in FIFO order
- The LND processing is specific to the LND type. For the purpose of this write up we'll talk about o2iblnd
  - o2iblnd will deal with each of a PUT/GET/ACK/REPLY in slightly different way. For example an LNet PUT can be broken down into PUT_REQ/PUT_ACK|NACK/PUT_DONE. For more details you can take a look at: O2IBLND Detailed Discussion#LNDTXTimeout
  - o2iblnd has its own set of connection credits. If the credits on the connection are depleted then the message is queued until there are credits available, then the message is posted
  - o2iblnd will then place the transmit in the active transmit state and wait for the transmit to complete as reported by the OFED layer.
    - Each message has a deadline, based on the `lnet_transaction_timeout` or the `lnd timeout` configuration parameters. If the deadline is reached while the message is queued or the in the transmit state, then the transmit expires and the message is finalized.
    - Assuming everything works then the message is sent to the router

Router Message Processing


receive message
for me?
yes
process message
logic flow
data lookup
no
grab router buffer
router buffer available?
yes
receive message into router buffer
return router buffer and buffer credit. Service routed queued message
no
queue message
use router buffers as source of RDMA when transmitting to next hop
send message to destination NID
Lookup Net of destination NID
local net?
Yes
consume peer tx credit
No
lookup remote net
found?
yes
lookup route send to router
credits?
no credits
no
credits available
Fail send
LND send
Queue message
LND send complete. Finalize message
return credit. service next queued message
remote net list
net 1
net 2
net N
routes
routes
routes
lnet_t
ln_net_list
ln_peer_list
lnet_net
1:N
lnet_net
1:N
lnet_peer
1:N
lnet_peer
1:N



Textual Description

For non-gateway nodes (IE clients and servers), usually lustre sets up the RDMA buffer to receive or transmit from. However for gateways, there are no such buffers. In fact lustre doesn't even have to be loaded at all on the gateways. Therefore, a gateway needs to allocate buffers to accept the RDMAed data into and then turn around and forward this data by RDMAing it to the next hop.

There are 3 router buffer pools, tiny, small and large. These are only allocated when you turn on the routing feature. When a message is received by a gateway and it determines it needs to forward this over to the next hop, then it looks at the size of data in this message and pulls out an appropriately sized buffer (max 1MB). It receives the RDMAed data into that buffer, then turns around and forwards it to the next hop.

- The router receives the message and looks at the destination NID. If the NID is not its own, then it knows it needs to route that message.
  - The router pulls a router buffer from the appropriate pool depending on the size of the message. If there are no available buffers then the message is queued until a buffer becomes available.
  - Once a buffer becomes available then the message is received into that buffer
  - Message is then forwarded to the destination NID and the data is RDMAed from the router buffer which has the data.
    - The send processing described above is followed.