Multi-Rail Selection Algorithm Router Modifications

There is a set of subcases that are not handled properly by the selection algorithm. These all relate to dealing with routers:

1. MR node/MR router/Non-MR peer
2. MR node/Non-MR router/Non-MR peer
3. MR node/Non-MR router/MR peer

There is a rule that when dealing with a non-MR node we should always use the same local interface. However, routers are an exception to that rule. The router doesn't look at the src NID. It simply forwards the message. It's the final destination that cares about the src NID.

MR node/MR router/Non-MR peer

- MR node should use the same local NI because the final destination is Non-MR
- MR node can send to the different NIs of the MR router

MR node/Non-MR router/Non-MR peer

- MR node should use the same local NI because the final destination is Non-MR
- There is only one interface for the router so MR node will send to that by default

MR node/Non-MR router/MR peer (This is considered a new feature and will be implemented separately)

- MR node can use different local NIs even if the router is Non-MR because the router doesn't care.
- MR node can send to different MR peer NIs because that's allowed

Detailed Design

```
peer = lnet_find_or_create_peer_locked(dst_nid, cpt);
if (!peer->lp_multi-rail && peer_not_local) {
        best_gw = lnet_find_route()
        if (best_gw) {
                /* Don't overwrite the final destination peer */
                 gw_peer = best_gw->lpn_peer;
        }
}

/*
 * at this point we can be keeping track of two peers
 * 1. The peer for the gateway router (gw_peer)
 * 2. The peer for the final destination (peer)
 */

/* Let's deal with the case where the final destination is Non-MR */
if (!peer->lp_multi_rail) {
        /*
     * if the final destination peer is non-MR then
         * check if we have a preferred local NI associated with the final destination peer
         * if yes then my best_ni will be the preferred NI of the peer.
         * if I don't have a best_ni then I'll need to pick one, and stick with it going forward
         * to pick one we need to look at the gw_peer.
         * if the gw_peer is MR then in order to know that we must've looked up a best_gw
         * set the preferred local NI for the final destination peer based on the net_id of that best_gw NID.
         *                 NOTE: Currently we do not support having different gateways reach the same final
destination network.
         *                 IE: tcp1 <gw1>@tcp
          *                         tcp1 <gw2>@tcp2
         *                That's not supported at this point, although there is a ticket opened for it.
         *                Either way, we'll need to stick with only one preferred NI, so we just pick the first
one we come up with.
         * Use the preferred NI to send the message to the non-MR peer
         * goto the pick_peer tag to select one of the router's NIs.
         *
         * if the gw_peer is Non-MR then it only has one peer NI, so we use that to select the local NI based
on the net_id of the gateway's NID.
         * Store that preferred NI in the peer.
         * Use the preferred NI to send the message to the non-MR peer via the single peer NI of the gateway.
         */
}
```

Potential implementation in psudo code

In master

The idea behind the code is to select the local ni we're going to send from based on the gw_peer, but store that NI as the preferred NI for the non-MR non-local peer.

When we deal with the non-MR non-local peer we're always going to use the same interface to send from and not confuse it.

Afterwards we can just deal with the gw directly, whether it's MR or non-MR, then we can just iterate through its interfaces as we normally would

```
if (msg->msg_type == LNET_MSG_REPLY ||
    msg->msg_type == LNET_MSG_ACK ||
    !lnet_peer_is_multi_rail(peer) ||
    best_ni) {
        ...
        if (best_lpni && !lnet_get_net_locked(LNET_NIDNET(dst_nid))) {
                ...
                gw_peer = best_gw->lpni_peer_net->lpn_peer;
                /* use the gw_peer to deal with the router for now */
                ...
        }
        ...

        if (!lnet_peer_is_multi_rail(peer)) {
                __u32 net_id;
                /* if gw_peer is not NULL, then we must have a best_gw */
                if (gw_peer) (
                        net_id = LNET_NIDNET(best_gw->lpni_nid);
                else
                        net_id = LNET_NIDNET(best_lpni->lpni_nid);
                ...
                if (!best_ni) {
                        ...
                        if (final_dst_lpni->lpni_pref_nnids == 1) {
                                best_ni = lnet_nid2ni_locked(final_dst_lpni->lpni_pref.nid, cpt);
                        } else
                                peer_net = lnet_peer_get_net_locked((gw_peer) ? gw_peer : peer, net);
                                ...
                }
                if (!best_ni) {
                        /* if we're routing then we want to look at the gw */
                        struct lnet_lpni lpni_use = (gw_peer) ? best_gw : best_lpni;
                        best_ni = lnet_net2ni_locked(net_id, cpt);
                        /* update the code to use lpni_use instead of best_lpni */
                        ...
                    /* now we need to pick the lpni for which we will update it's preferred local_ni */
                        if (!gw_peer)
                                /* continue doing what the code does now */
                }
                /* Set preferred NI if necessary. */
                if (gw_peer)
                        if (final_dst_lpni->lpni_pref_nnids == 0)
                                /* set preferred ni */
                else
                        if (lpni->lpni_pref_nnids == 0)
                                /* set preferred ni */
        }

        /* deal with the gw directly */
        if (gw_peer)
                peer = gw_peer
}
```

Caveat

One thing to be aware of is that if the entire system is MR, but the config doesn't tell the node and the peer that the other is MR, then that will cause the node to only use one of its interfaces when talking to that peer. Even though it'll use all the interfaces of the MR router and the router will use all of its interfaces and the final destination interfaces, there would still be a performance drop, because the local node is not using the entire bandwidth available to it.

With the Dynamic Discovery this will not be a problem since DD will discover the final destination MR status and will deal with it as MR.

However, in 2.10 the configuration described above will have a performance impact.