

# LNet Health User Documentation

## Overview

LNet Multi-Rail has implemented the ability for multiple interfaces to be used on the same LNet network or across multiple LNet networks. The LNet Health feature adds the ability maintain a health value for each local and remote interface. This allows the Multi-Rail algorithm to consider the health of the interface before selecting it for sending. The feature also adds the ability to resend messages across different interfaces when interface or network failures are detected. This allows LNet to mitigate communication failures before passing the failures to upper layers for further error handling. To accomplish this, LNet Health monitors the status of the send and receive operation and uses this status to increment the interface's health value in case of success and decrement it in case of failure.

## Health Value

The health value of a local or remote interface is set to LNET\_MAX\_HEALTH\_VALUE. This is 1000. The value itself is arbitrary and is meant to allow for health granularity as opposed to having a boolean state, healthy and unhealthy. The granularity allows the Multi-Rail algorithm to select the interface that has the most likelihood of sending or receiving a message.

## Reasons For Failure

LNet health behavior depends on the type of failure detected:

### Local/Resend

A local failure has occurred, such as no route found or an address resolution error. These failures could be temporary, therefore LNet will attempt to resend the message. LNet will decrement the health value of the local interface and will select it less often if there are multiple available interfaces.

### Local/No-Resend

A local non-recoverable error occurred in the system, such as out of memory error. In these cases LNet will not attempt to resend the message. LNet will decrement the health value of the local interface and will select it less often if there are multiple available interfaces.

### Remote/No-Resend

If LNet successfully sends a message, but the message doesn't complete or an expected reply is not received then it's classified as remote error. LNet will not attempt to resend the message to avoid duplicate messages on the remote end. LNet will decrement the health value of the remote interface and will select it less often if there are multiple available interfaces.

### Remote/Resend

There are set of failures where we could be reasonably sure that the message was dropped before getting to the remote end. In this case LNet will attempt to resend the message. LNet will decrement the health value of the remote interface and will select it less often if there are multiple available interfaces.

## User Interface

LNet Health is turned off by default in releases prior to 2.12.3. Starting from 2.12.3 the feature is turned on by default. There are multiple module parameters available to control the LNet Health feature. They are described in the following sections.

All the module parameters are implemented in sysfs and are located in `/sys/module/lnet/parameters/`. They can be set directly by echoing a value into them as well as from `lnetctl`.

### `lnet_health_sensitivity`

When LNet detects a failure on a particular interface it will decrement its Health Value by `lnet_health_sensitivity`. The value defaults to 100. The lesser the value the longer it takes for that interface to become healthy again. For example, if set to 100, it takes 10 seconds to fully recover from 0 to 1000; if set to 50, it takes 20 seconds to fully recover from 0 to 1000.

If set to 0, which means the health value will not be decremented. In essence the health feature is turned off.

When a failure occurs on an interface then its Health Value is decremented and the interface is flagged for recovery. The recovery mechanism is described in section `lnet_recovery_interval`.

```
lnetctl set health_sensitivity: sensitivity to failure
    0 - turn off health evaluation
    >0 - sensitivity value not more than 1000
```

## lnet\_recovery\_interval

When LNet detects a failure on a local or remote interface it will place that interface on a recovery queue. There is a recovery queue for local interfaces and another for remote interfaces.

Prior to 2.15, the interfaces on the recovery queues will be LNet PINGed every `lnet_recovery_interval`. This value defaults to 1 second. On every successful PING the health value of the interface pinged will be incremented by `lnet_health_sensitivity`.

Having this value configurable allows system administrators to control the amount of control traffic on the network.

```
lnetctl set recovery_interval: interval to ping unhealthy interfaces
>0 - timeout in seconds
```

Since 2.15, `lnet_recovery_interval` was deprecated and the fixed interval was replaced with exponential backoff. For more information, see [LU-13569](#).

## lnet\_retry\_count

When LNet detects a failure which it deems appropriate for re-sending a message it will check if a message has passed the maximum `retry_count` specified. After which if a message wasn't sent successfully a failure event will be passed up to the layer which initiated message sending.

```
lnetctl set retry_count: number of retries
0 - turn of retries
>0 - number of retries
```

## lnet\_transaction\_timeout

This timeout is somewhat of an overloaded value. It carries the following functionality:

- A message is abandoned if it is not sent successfully when the `lnet_transaction_timeout` expires and the `retry_count` is not reached.
- A GET or a PUT which expects an ACK expires if a REPLY or an ACK respectively, is not received within the `lnet_transaction_timeout`.

This value defaults to 5 seconds.

```
lnetctl set transaction_timeout: Message/Response timeout
>0 - timeout in seconds
```

## Important Note

It is important to note that prior to LNet Health the LND timeout defaulted to 50 seconds. With LNet health this is no longer the case. The LND timeout will now be a fraction of the `lnet_transaction_timeout` as described in the next section.

This means that in networks where very large delays are expected then it will be necessary to increase this value accordingly.

## lnet\_lnd\_timeout

This is not a configurable parameter. But it is derived from two configurable parameters: `lnet_transaction_timeout` and `retry_count`

```
lnet_lnd_timeout = lnet_transaction_timeout / retry_count
```

As such there is a restriction that `lnet_transaction_timeout >= retry_count`

The core assumption here is that in a healthy network, sending and receiving LNet messages should not have large delays. There could be large delays with RPC messages and their responses, but that's handled at the PTLRPC layer.

## Showing LNet Health Configuration Settings

`lnetctl` can be used to show all the LNet health configuration settings using the `lnetctl global show` command

```
#> lnetctl global show
global:
  numa_range: 0
  max_intf: 200
  discovery: 1
  retry_count: 0
  transaction_timeout: 5
  health_sensitivity: 0
  recovery_interval: 1
```

## Showing LNet Health Statistics

LNet Health statistics are shown under a higher verbosity settings. To show the local interface health statistics:

```
lnetctl net show -v 3
```

to show the remote interface health statistics:

```
lnetctl peer show -v 3
```

A sample output

```
#> lnetctl net show -v 3
net:
- net type: tcp
  local NI(s):
  - nid: 192.168.122.100@tcp
    status: up
    interfaces:
      0: eth0
    statistics:
      send_count: 0
      rcv_count: 0
      drop_count: 0
    sent_stats:
      put: 0
      get: 0
      reply: 0
      ack: 0
      hello: 0
    received_stats:
      put: 0
      get: 0
      reply: 0
      ack: 0
      hello: 0
    dropped_stats:
      put: 0
      get: 0
      reply: 0
      ack: 0
      hello: 0
    health stats:
      health value: 1000
      interrupts: 0
      dropped: 0
      aborted: 0
      no route: 0
      timeouts: 0
      error: 0
    tunables:
      peer_timeout: 180
      peer_credits: 8
      peer_buffer_credits: 0
      credits: 256
    dev cpt: -1
    tcp bonding: 0
    CPT: "[0]"
```

There is a new YAML block, `health stats`, which displays the health statistics for each local or remote network interface.

Global statistics also dump the global health statistics as below:

```
#> lnetctl stats show
statistics:
  msgs_alloc: 0
  msgs_max: 0
  rst_alloc: 0
  errors: 0
  send_count: 0
  resend_count: 0
  response_timeout_count: 0
  local_interrupt_count: 0
  local_dropped_count: 0
  local_aborted_count: 0
  local_no_route_count: 0
  local_timeout_count: 0
  local_error_count: 0
  remote_dropped_count: 0
  remote_error_count: 0
  remote_timeout_count: 0
  network_timeout_count: 0
  rcv_count: 0
  route_count: 0
  drop_count: 0
  send_length: 0
  rcv_length: 0
  route_length: 0
  drop_length: 0
```

## Initial Settings Recommendations (prior to 2.13)

LNet Health is off by default. This means that `lnet_health_sensitivity` and `lnet_retry_count` are set to 0.

Setting `lnet_health_sensitivity` to 0 will not decrement the health of the interface on failure and will not change the interface selection behavior. Furthermore the failed interfaces will not be placed on the recovery queues. In essence, turning off the LNet Health feature.

The LNet Health settings will need to be tuned for each cluster. But the base configuration would be as follows:

```
#> lnetctl global show
global:
  numa_range: 0
  max_intf: 200
  discovery: 1
  retry_count: 2
  transaction_timeout: 5
  health_sensitivity: 1
  recovery_interval: 1
```

This setting will allow a maximum of two retries for failed messages within the 5 second transaction timeout.

If there is a failure on the interface the health value will be decremented by 1 and the interface will be LNet PINGed every 1 second.