12th ANNUAL WORKSHOP 2016

MULTI-RAIL LNET FOR LUSTRE

Amir Shehata, Lustre Network Engineer, Intel Corp
Olaf Weber, Senior Software Engineer, SGI Storage Software

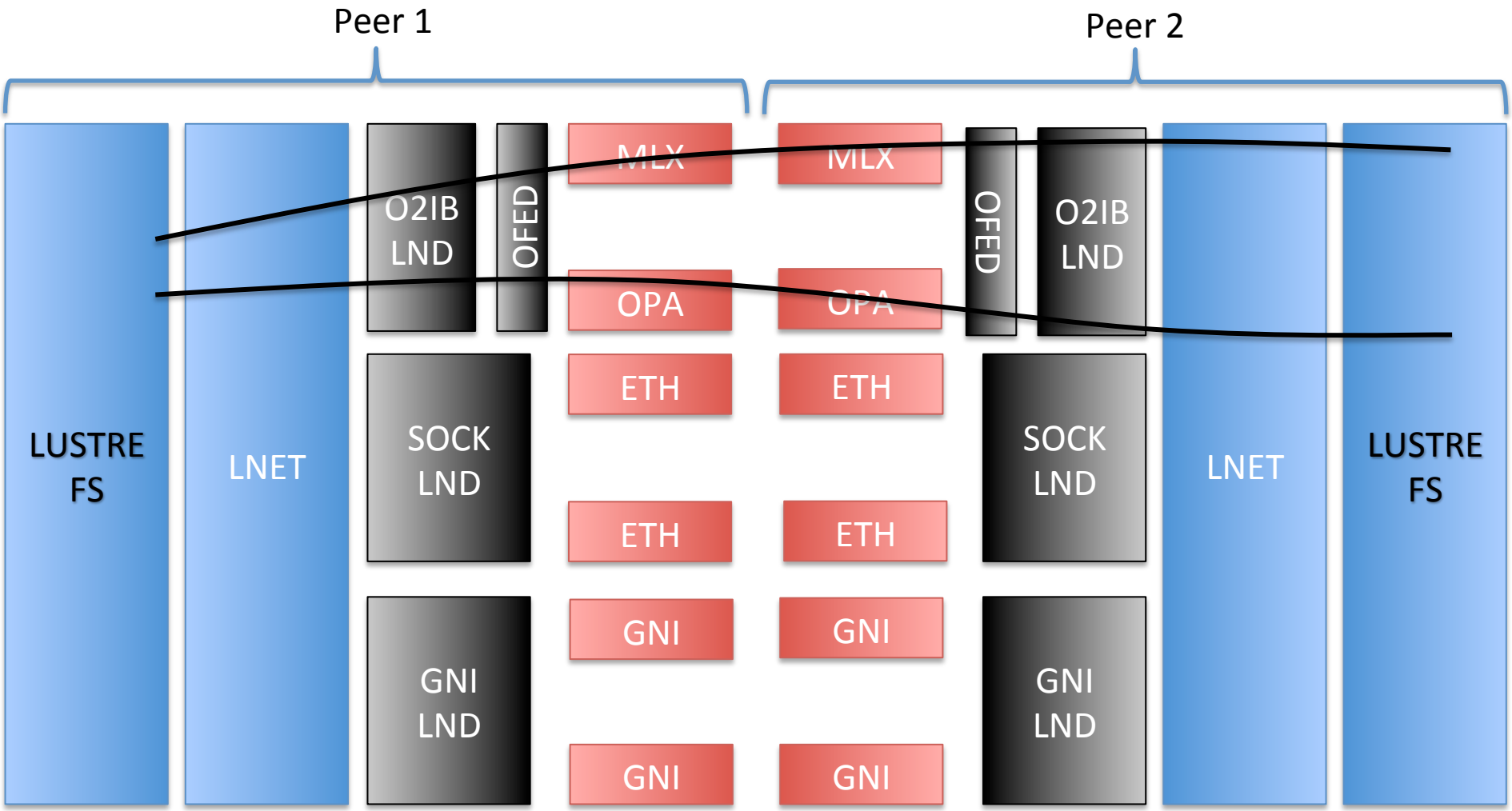
April 6th, 2016

MULTI-RAIL LNET: WHAT AND WHY

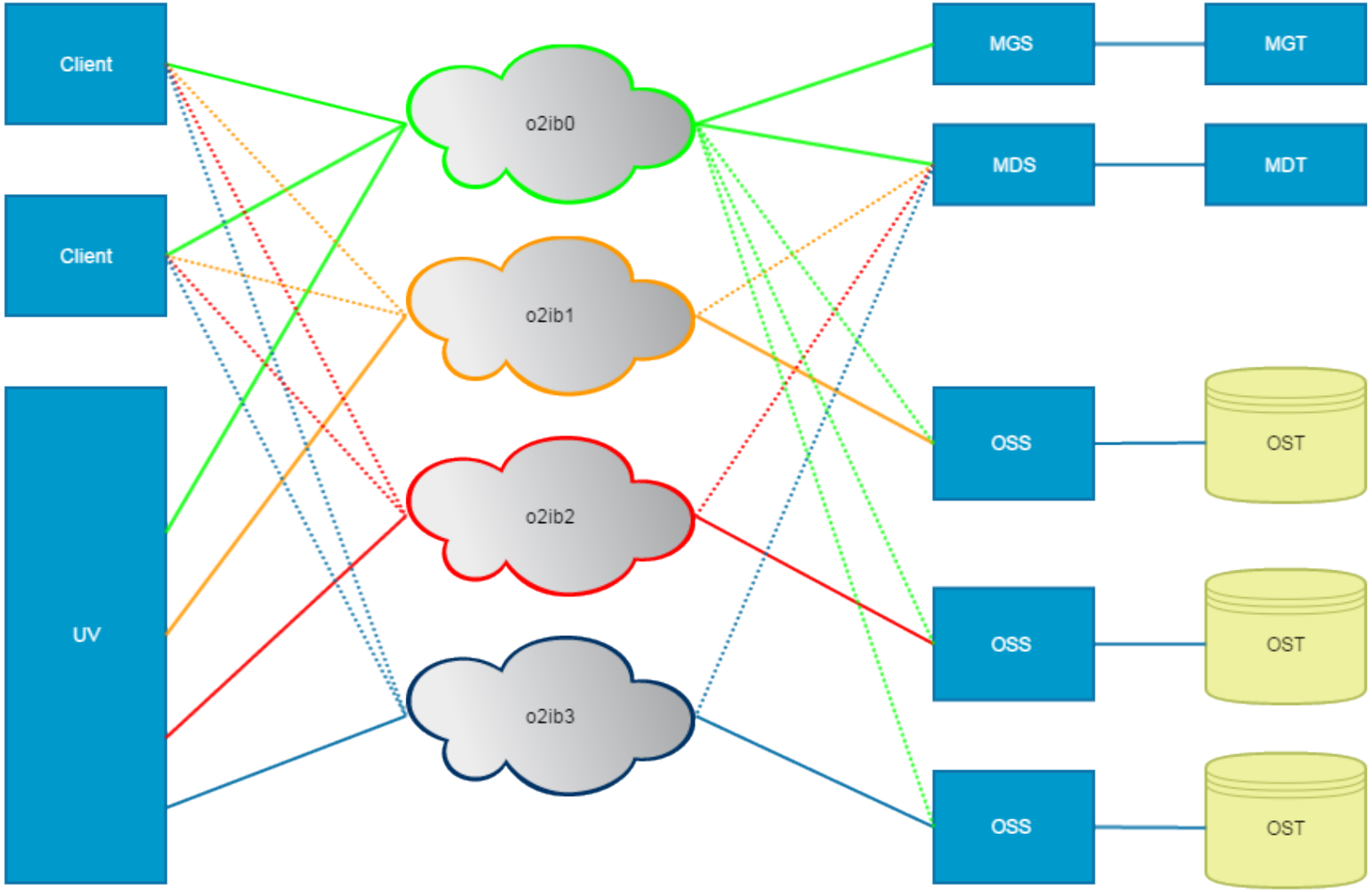
AGENDA

- Overview of Multi-Rail design in LNet
- Why LNet level implementation
- Multi-Rail Use Case scenarios
- How to configure and use Multi-Rail

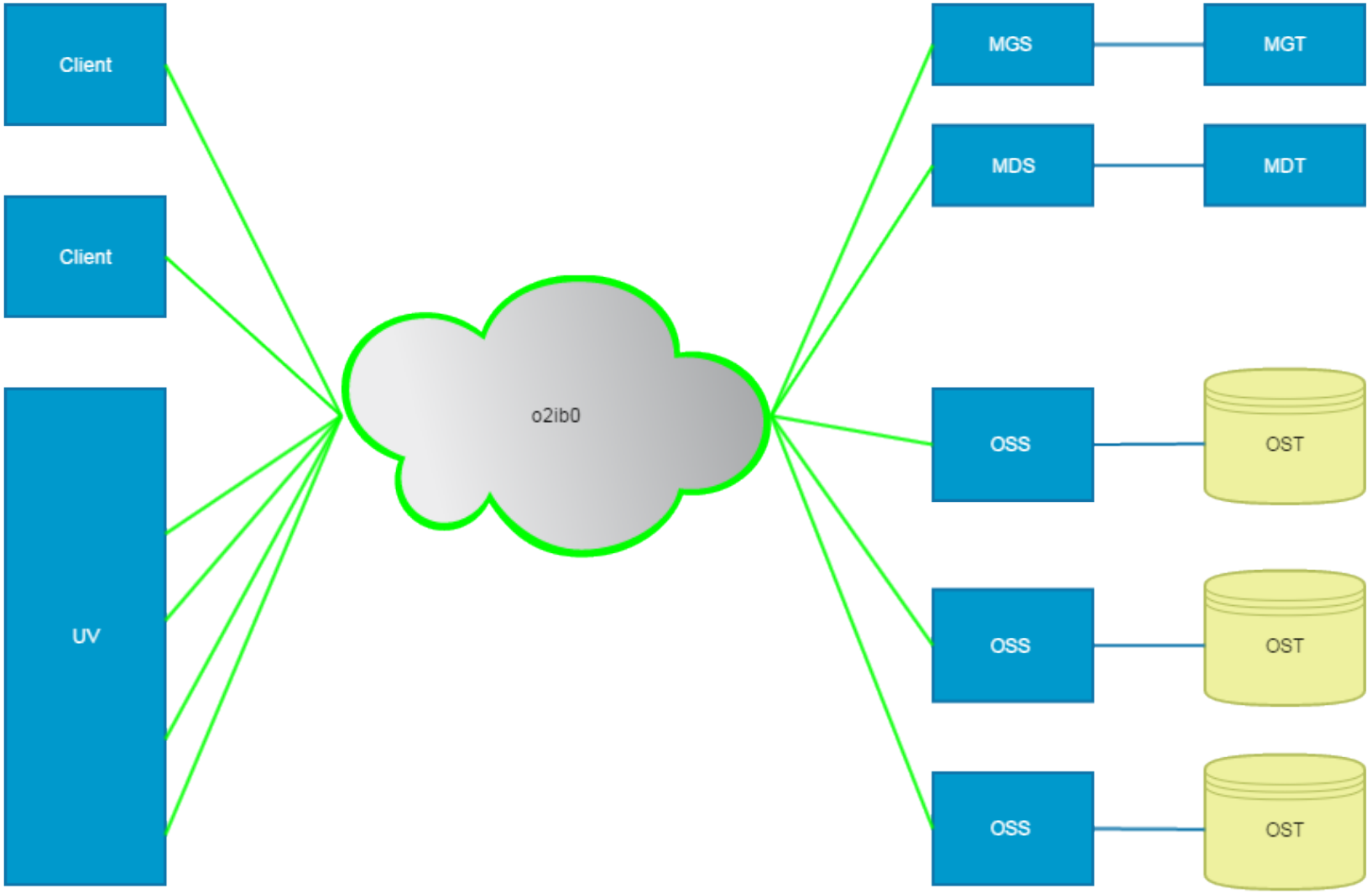
OVERVIEW


Peer 1
Peer 2
LUSTRE FS
LNET
O2IB LND
OFED
SOCK LND
GNI LND
MLX
OPA
ETH
ETH
GNI
GNI
MLX
OPA
ETH
ETH
GNI
GNI
OFED
O2IB LND
SOCK LND
GNI LND
LNET
LUSTRE FS

WITHOUT MULTI-RAIL



WITH MULTI-RAIL



MULTI-RAIL OBJECTIVES

- **Multi-Rail allows nodes to communicate across multiple interfaces**
  - Using multiple interfaces connected to one Luster Network
  - Using multiple interfaces connected to several Lustre Networks
  - Use different Network Interface types
  - These interfaces are used simultaneously (active-active)

WHY IN LNET

- **Two possible solutions**
  - Implement Multi-Rail in LNet
    - Utilize the same or different network interface types.
      - Ex: try to send over OPA/IB network, if that fails send over TCP network
  - Implement Multi-Rail in the LND
    - It will have to be implemented for every LND
    - Only bonds interfaces of the same type
- **This Multi-Rail design and implementation is done in LNet**
- **This is a collaboration between Intel and SGI**

WHAT DOES MULTI-RAIL GIVE US

- **From LNet perspective, there are two major advantages**
  - Increasing LNet performance by aggregating bandwidth of multiple interfaces
  - Increasing network resiliency by trying all possible interfaces before a message is declared not deliverable

INCREASING BANDWIDTH - CLIENTS

- **SGI Big Clients**
  - SGI UV 300: 32 socket NUMA system
  - SGI UV 3000: 256 socket NUMA system
  - Systems with multiple TB of memory need a lot of bandwidth
  - Increase bandwidth by adding more interfaces.

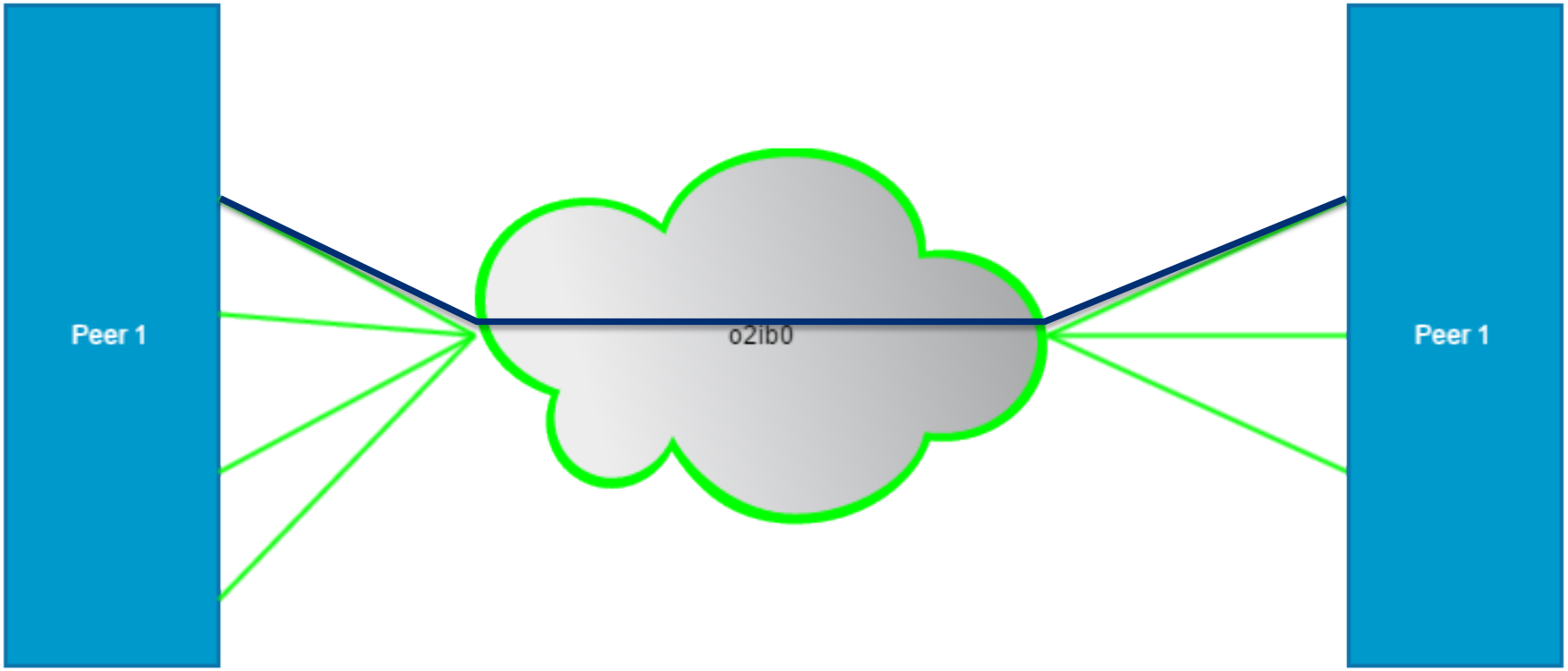
INCREASING BANDWIDTH - SERVERS

- **Big clusters**
  - Bandwidth to the server becomes a bottleneck
  - Add more interfaces to the servers and configure LNet to use them.
  - Messages can also be multiplexed over the interfaces of the remote peer, which can be configured statically or discovered dynamically.

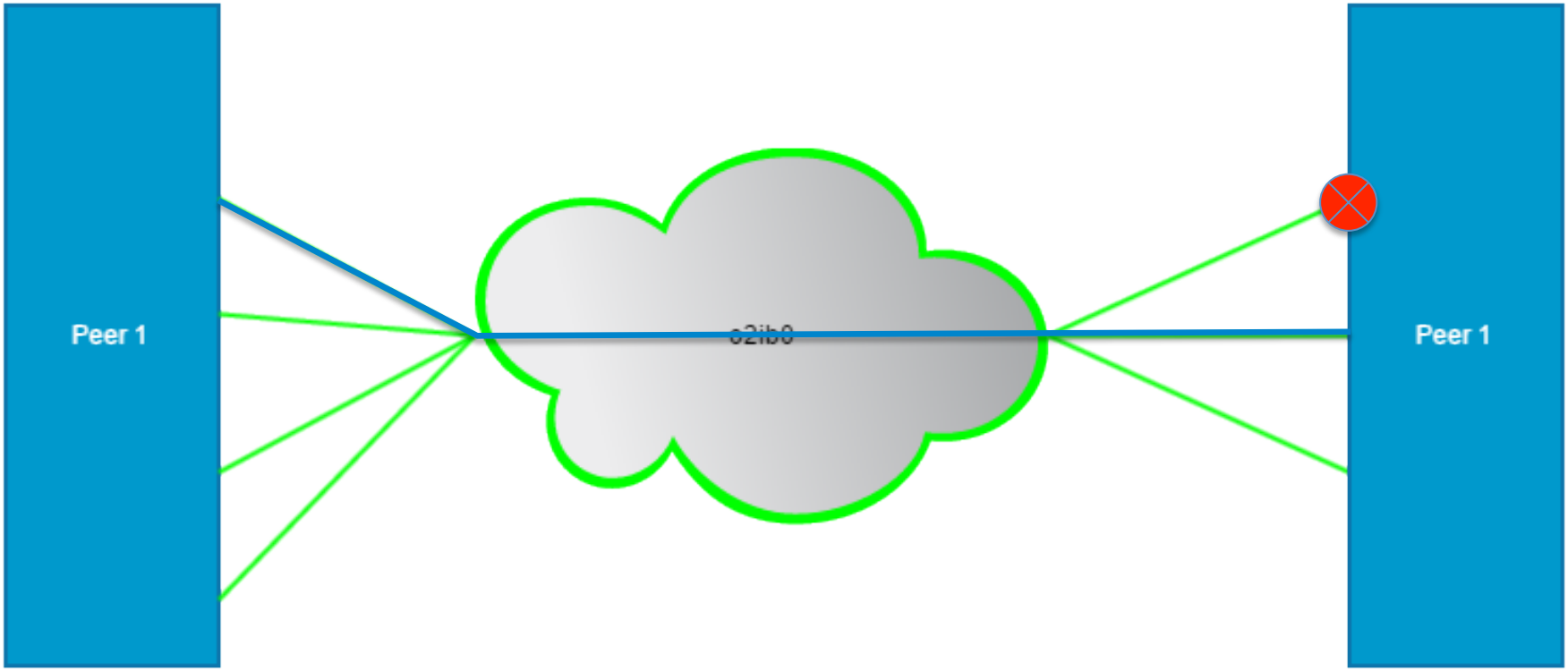
INCREASING BANDWIDTH

- **Multi-Rail**
  - Multiplexes across interfaces
    - Each LNet message is sent over a different local interface
    - Local interfaces are selected depending on several criteria:
      - the NUMA distance between the NI and the message memory.
        - » In large systems (SGI UV) this can be a significant performance advantage.
      - How busy an NI is, determined by a set of credits associated with the NI

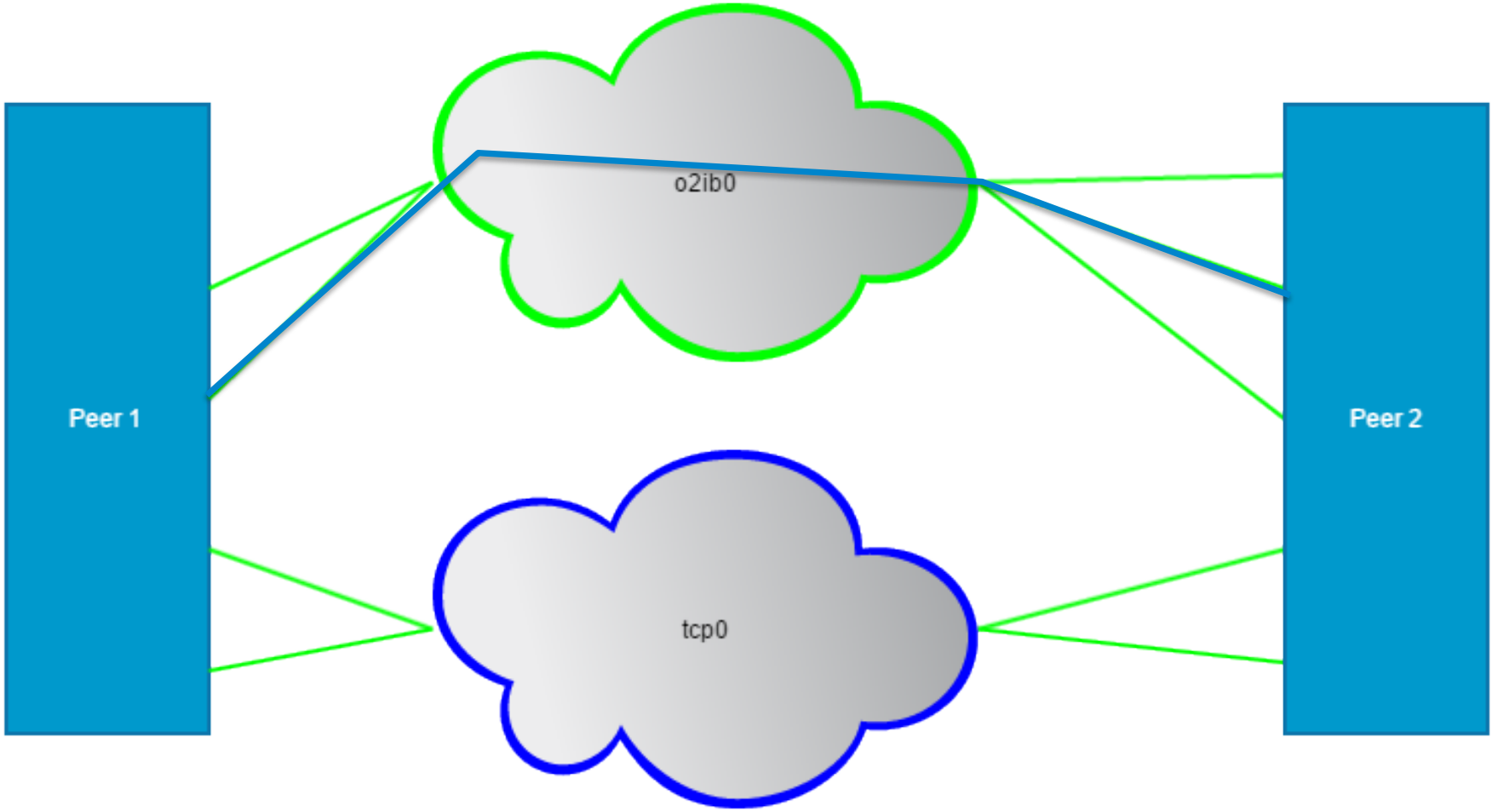
RESILIENCY – MULTIPLE NIS/SAME NETWORK



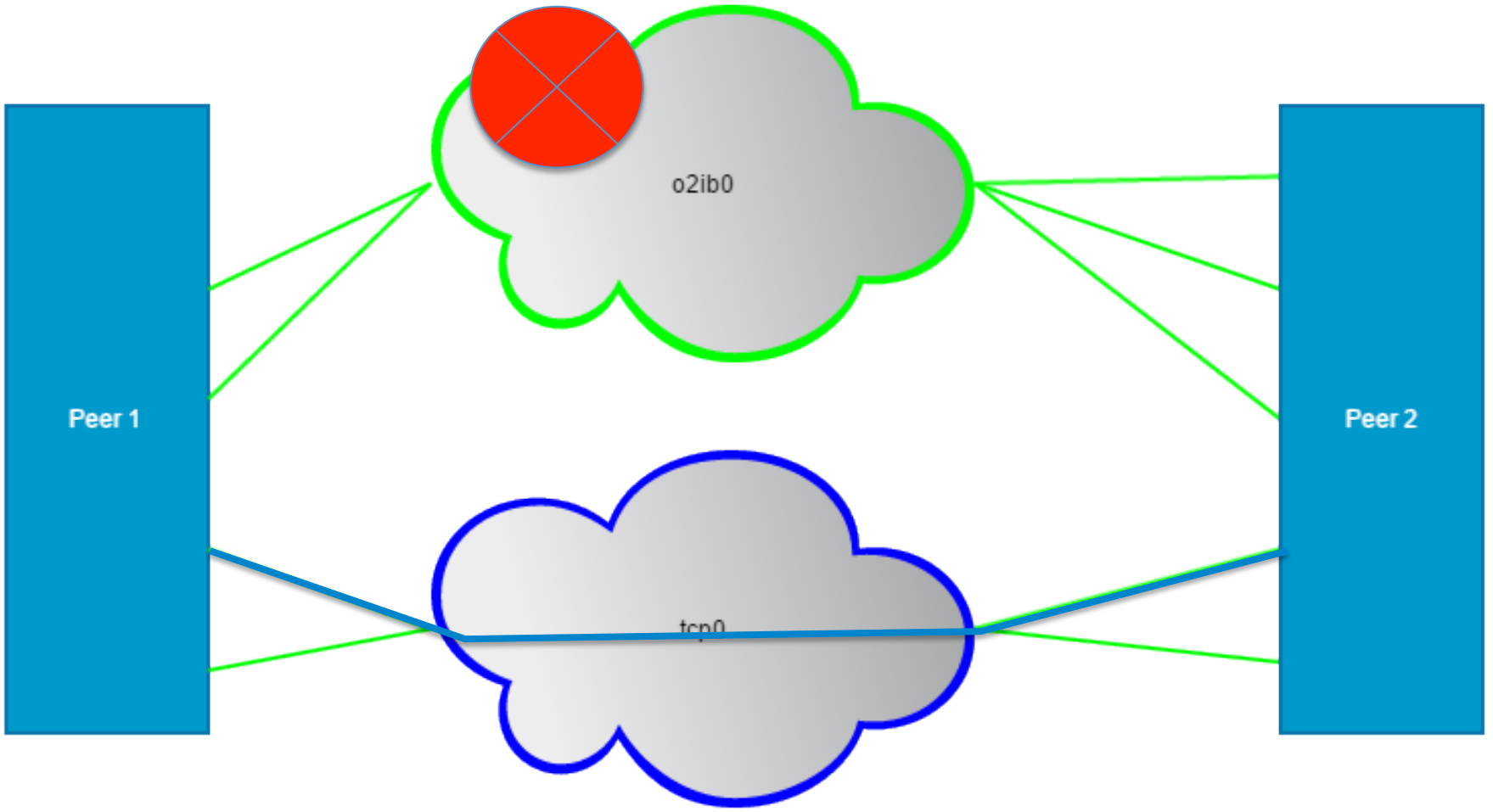
RESILIENCY – MULTIPLE NIS/SAME NETWORK



RESILIENCY – MULTIPLE NETWORKS



RESILIENCY – MULTIPLE NETWORKS



CONFIGURING MULTI-RAIL LNET

USE CASES

- Improved performance
- Improved resiliency
- Allow multiple networks access to the filesystem such that you don’t have to have all clients on the same network.
- Better usage of large clients resource, including NUMA aware clients (IE: SGI UV)
- Fine grained control of traffic

CONFIGURATION SETTINGS

- **The following elements can be configured via User Space utility (Inetctl):**
  - Local Network Interfaces
    - These are the interfaces by which a node sends messages
  - Remote Peer Network Interfaces
    - These are the remote network interfaces to which a node sends messages.
  - Selection Rules
    - These are the set of rules which determine the local network interface/remote peer interface used for communication.

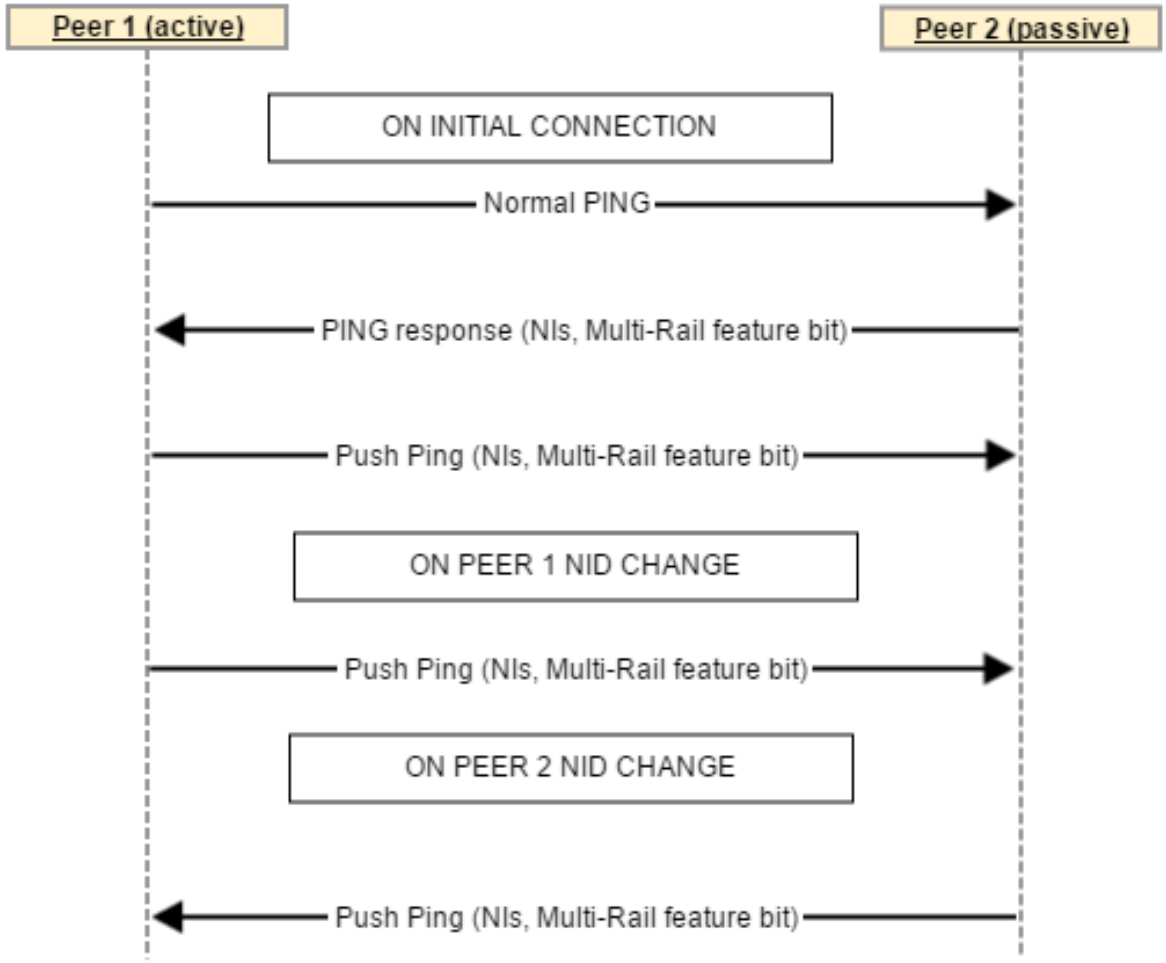
TWO TYPES OF CONFIGURATION METHODS

- **Multi-Rail can be configured statically, via lnetctl (DLC).**
  - The following elements has to be configured statically.
    - Network Interfaces
    - Selection Rules
  - The following elements can be configured statically or discovered dynamically
    - Peer Network Interfaces
  - Configuration can be described via YAML syntax
- **Dynamic discovery can be enabled allowing LNet to discover peers automatically**

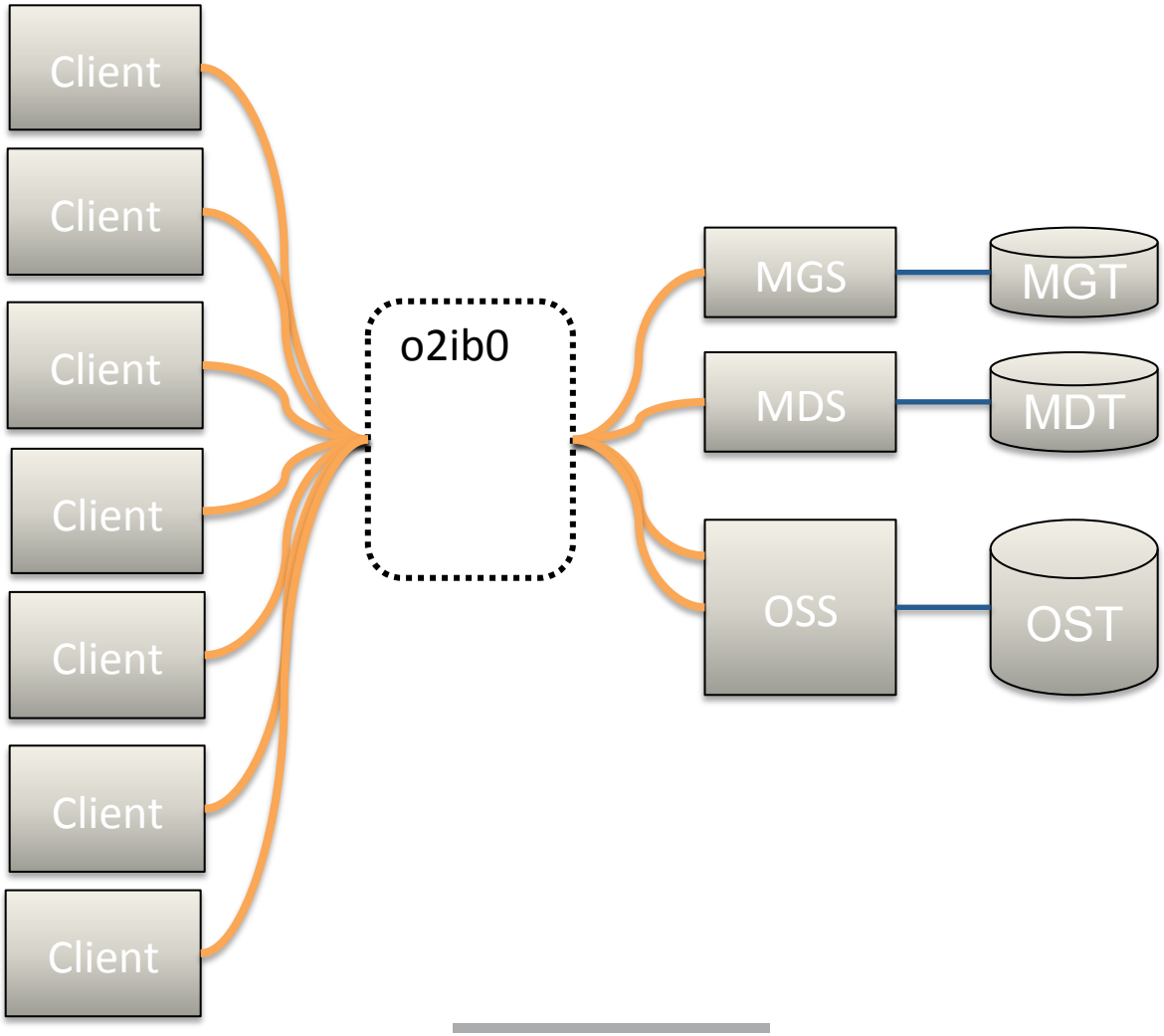
STATIC CONFIGURATION – BASIC CONCEPTS

- **On a node:**
  - Configure Local Networks Interfaces
    - Ex: tcp(eth0, eth1)
      - <eth0 IP>@tcp, <eth1 IP>@tcp
  - Configure Remote Network Interfaces
    - Specify a peers Network Interface IDs (NIDs) which it can be reached on
      - <peerX primary nid>, <peerX nid2>, etc
  - Configure selection Rules
- **The static configuration method is useful**
  - if the cluster is not subject to change and you want to catch any changes to the Network Layout.
  - If Primary NID of a server is not reachable on a specific network

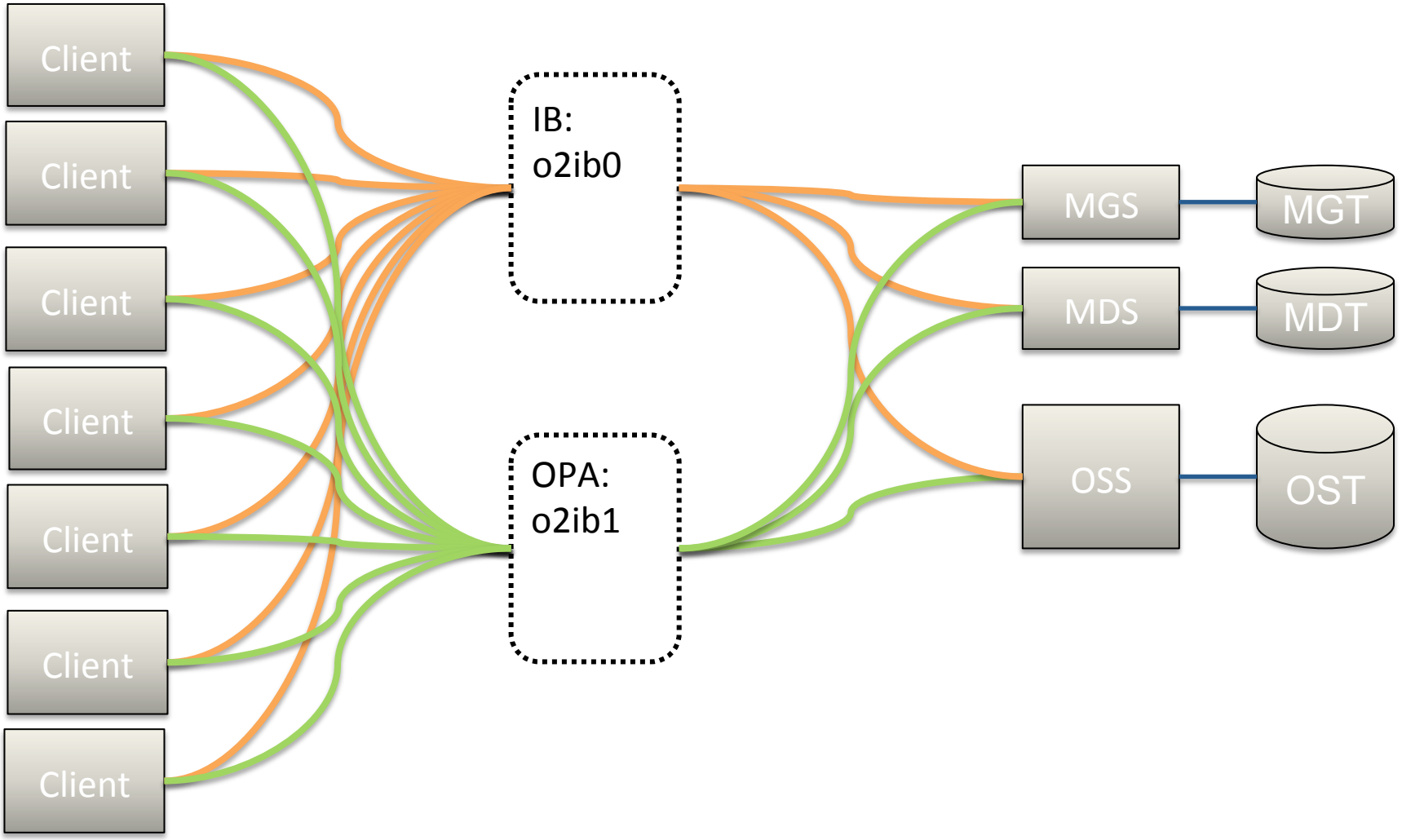
DYNAMIC CONFIGURATION – BASIC CONCEPTS



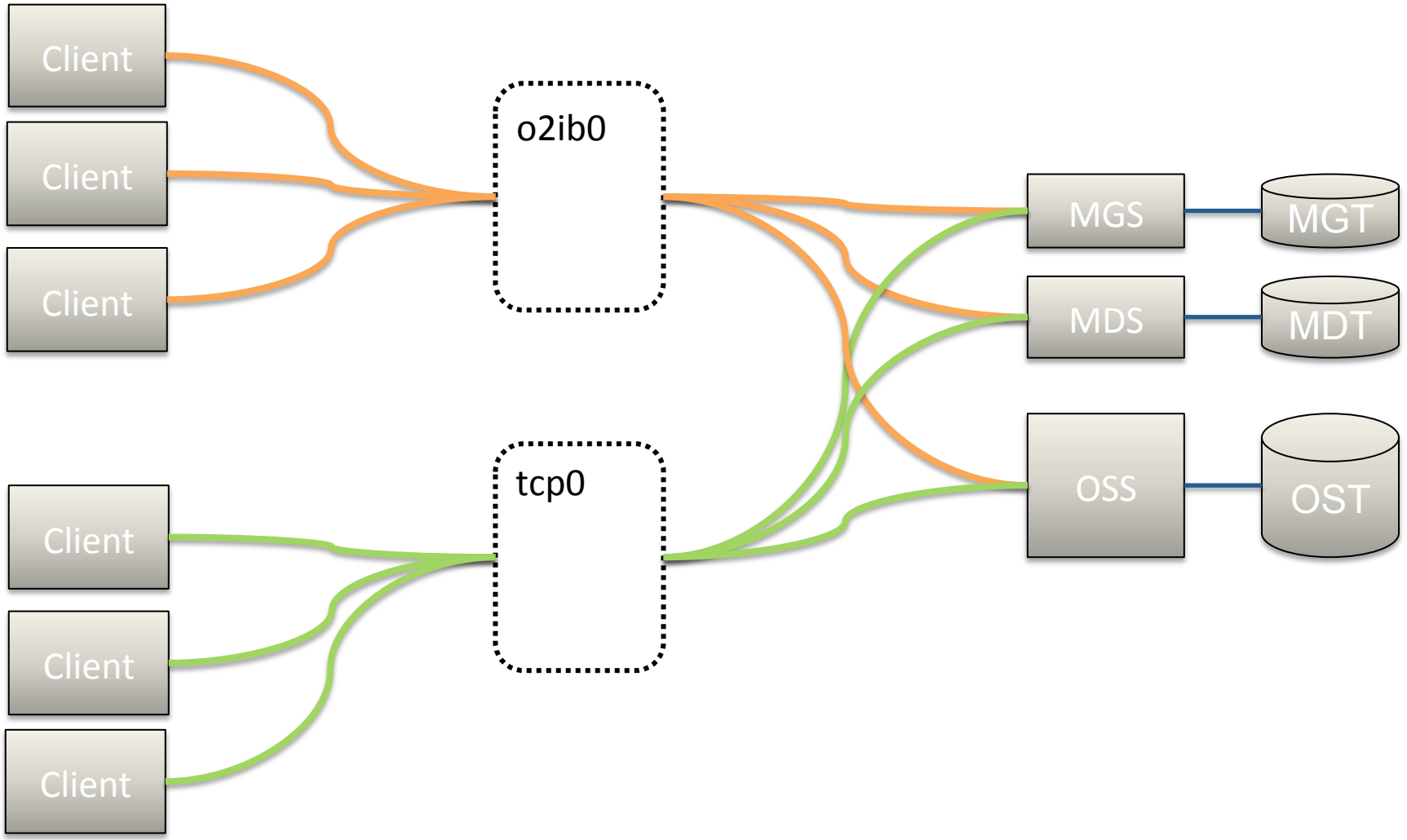
USE CASE 1 – IMPROVED PERFORMANCE



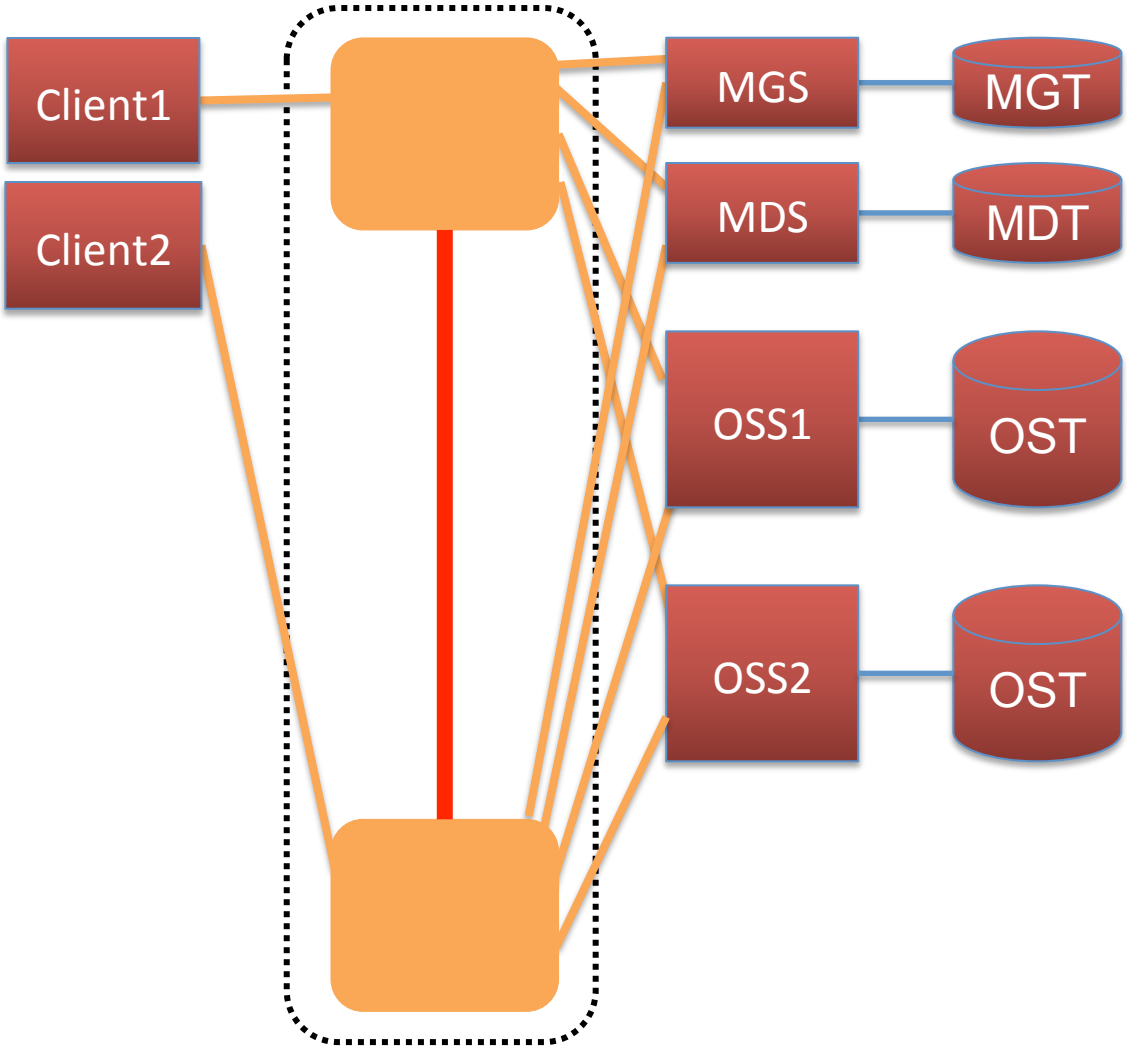
USE CASE 2 – IMPROVED RESILIENCY



USE CASE 3 – MULTI-NETWORK FS ACCESS



USE CASE 4 – TRAFFIC CONTROL



PROJECT STATUS

- **Public project wiki page:**
  - http://wiki.lustre.org/Multi-Rail_LNet
- **Code development is done on the multi-rail branch off the Lustre master repo.**
  - Patches to enable static configuration are under review
  - Unit testing and system testing underway
  - Patches for selection rules are under development
  - Patches for dynamic peer discovery are under development
  - Estimated project completion time: end of this year
  - Master landing date: TBD

OPENFABRICS
ALLIANCE

12th ANNUAL WORKSHOP 2016

THANK YOU

Amir Shehata, Lustre Network Engineer

Intel Corp