



Deployment Guide for SRX Series Services Gateways in Chassis Cluster Configuration

Version 1.3

First release June 2013

Last updated February 2014



Contents

Introduction	3
Chassis Cluster Concepts.....	4
Scenarios for Chassis Cluster and Configuration Steps.....	10
1) Basic Active/Standby scenario	10
2) Active/Standby with multiple RETHs	15
3) Active/Standby with LAG/LACP	16
4) Active/Active.....	17
5) Use of non-reth interfaces with dual ISP	19
6) HA links passing through a switched network.....	21
Chassis Cluster Optional Failover Configuration.....	23
Chassis Cluster Failure Behavior for Control and Fabric Links	32
Conclusion.....	35

Introduction

One of the features that SRX Series Services gateways support is Chassis Cluster, also referred to as High Availability (HA). With Chassis Cluster it is possible to have two identical SRX series devices connected together to form a logical single device, which provides redundancy in case of a failure in one of the two nodes or their surroundings. Chassis cluster can be used in different topologies and together with various other features, to provide the desired failover functionality and redundancy.

This document intends to assist the network administrator in configuring SRX series devices for cluster setup. To be able to do so, first the chassis cluster concepts are outlined. Next it is explained how a basic chassis cluster should be enabled and which optional features can be added for use in various relevant topologies and levels of redundancy. Where necessary, also pitfalls and requirements have been noted.

In this document both SRX High-end and SRX Branch models are covered.¹

¹ SRX Branch models are SRX100, SRX110, SRX210, SRX220, SRX240, SRX550, SRX650.
SRX High-end models are SRX1400, SRX3400, SRX3600, SRX5600, SRX5800.
Note that only SRX110 does not support chassis cluster.

Chassis Cluster Concepts

First we will define what the chassis cluster feature is and how it works. The principles are the same for both SRX High-end and SRX Branch models.

To begin with, it is important to understand that in an SRX device there is a clear separation between the control plane and data plane.

The control plane (or RE, routing engine) has the function of managing the device. The RE functions include tasks like hosting the management sessions (CLI, JWeb, NSM/Space), pushing down configuration elements to the data plane, such as configured security policies and interface properties and pulling statistics/counters from the data plane. The RE is also where the dynamic routing protocols (if used) are run and it will push the resulting forwarding table and updates towards the data plane.

The data plane (or PFE, packet forwarding engine) is the part that is concerned with the actual forwarding and security processing of traffic.

Chassis cluster

A chassis cluster consists of two identical SRX devices that are working together to provide stateful failover of processes, services and traffic flow. In case a failure occurs in one of the nodes, this will be detected and as needed, a failover to the other node will occur, so that the traffic will continue to be forwarded. The cluster nodes are connected together with two links called control link and fabric link.

Cluster-ID

When setting up two devices to form a cluster, both devices are first configured with a cluster-id and a node-id.

A cluster-id identifies a cluster. For two SRX devices to join together to form a cluster, they have to have the same cluster-id configured. Cluster-id numbering can range from 1-15². Configuring a cluster-id of 0 is equivalent to disabling chassis cluster.

The node-id is assigned to each node inside a cluster. The node-id can only have a value of 0 or 1. This means that a cluster of two nodes always consists of a node0 and a node1.

How to exactly configure the cluster-id and node-id is described later in the document.

Note: When using multiple clusters that are connected to the same L2 domain, a unique cluster-id needs to be used for each cluster. Otherwise you may get duplicate mac addresses on the network, because the cluster-id is used to form the virtual interface mac addresses.

² From Junos 12.1X45 major release onwards, the cluster-id range will be enlarged to 1-255.

Interfaces

The goal of a cluster is to be able to move/failover traffic flow from one box to the other when needed. In SRX chassis cluster, for this purpose a special interface type is introduced: redundant ethernet (reth). A reth interface is a virtual interface. It will be active on one of the two nodes only and it has the ability to move/failover to the other node.

A reth interface is configured just like a regular physical ethernet (fe, ge, xe) interface. Like a physical interface, a reth interface can contain multiple logical interfaces (or sub-interfaces), such as reth0.0, reth0.1, reth0.2 and so on. These logical interfaces will be assigned a family, such as inet/inet6/bridge, an IP address, a vlan-id etcetera, just like with regular ethernet interfaces. Also like a regular ethernet interface, a reth interface will have a mac address. The mac address for the reth is a virtual mac address that does not correlate to the mac address of the physical link members. A reth interface virtual mac address starts with "00:10:DB:FF". The octets after that are derived from the cluster-id and the specific reth interface number.

When a reth interface fails over to the other node, all its logical interfaces will also fail-over and become active on the other node.

A reth interface is bound to one physical ethernet interface (fe-, ge- or xe- interface) on each node (or multiple in case of link aggregation, LAG, we will get to that later). This way, the reth can be active on one node or the other through the associated physical interface.

Both the physical interfaces that are used for the reth interface need to be connected to the same L2 broadcast domain(s) or vlans. This will ensure that the L3 topology will stay the same after a failover and L2 switches just need to be updated with respect to the port where the reth resides. How the switches are updated is described in the topic *Failover*.

Data plane redundancy-groups

To control on which node a reth interface will be active, in SRX chassis clusters there is the concept of 'redundancy-groups' (RG). A reth interface is always bound to one redundancy-group. A redundancy-group can contain multiple reth interfaces. A redundancy-group is active on only one of the two nodes (the primary node for that RG), and backup on the other node (the secondary node for that RG).

On the node where the redundancy-group is active, the associated reth interfaces are also active on the associated physical interfaces. This means a reth interface will effectively move from one node to the other when a failover of the parent redundancy-group occurs.

Session sync

In order to keep the traffic flowing uninterrupted after a failover, the sessions that were present on the previous primary node, need to also be present on the new primary node. To achieve this, each session that is created or deleted will be synchronized toward the secondary node (excluding ICMP sessions). This happens over the fabric link, using RTO (real-time objects) packets between the two nodes PFEs. When the session is synced to the backup node, the timeout value on the backup node will in most cases reflect a timeout value of 8 times the initial value on the primary node (i.e. 14400 seconds instead of 1800 seconds for TCP traffic).

```
root> show security flow session
node0:
-----

Session ID: 25, Policy name: allow/6, State: Active, Timeout: 1788, Valid
  In: 10.2.1.2/60187 --> 10.1.1.2/22;tcp, If: reth1.0, Pkts: 28, Bytes: 3268
  Out: 10.1.1.2/22 --> 10.2.1.2/60187;tcp, If: reth0.0, Pkts: 37, Bytes: 5077
Total sessions: 1

node1:
-----

Session ID: 4, Policy name: allow/6, State: Backup, Timeout: 14394, Valid
  In: 10.2.1.2/60187 --> 10.1.1.2/22;tcp, If: reth1.0, Pkts: 0, Bytes: 0
  Out: 10.1.1.2/22 --> 10.2.1.2/60187;tcp, If: reth0.0, Pkts: 0, Bytes: 0
Total sessions: 1
```

Control plane redundancy-group

In a chassis cluster, each node will have one routing-engine (RE). Only on one of the nodes the RE will be 'primary' while the RE in the other node is 'secondary'. Just like with reth interfaces that use redundancy-groups 1 and higher, also the RE is associated to a 'redundancy-group' which determines on which node the RE is primary.

The redundancy-group number reserved for the control plane is "0". The primary RE manages the whole chassis cluster. It will be in contact with the PFE in both nodes and will sync state with the secondary RE in the other chassis.

The RE communication with the remote node, such as communication with PFE, kernel state synchronization with RE, configuration synchronization, and Juniper Stateful Redundancy Protocol (JSRP) heartbeats between the nodes is handled via the control link. More on the HA links usage in topic *HA links*.

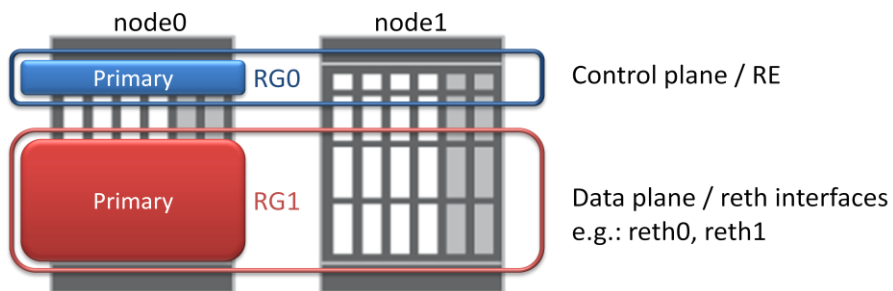
Failover

During a chassis cluster failover the primary ownership of the redundancy groups moves from one node to the other. Each redundancy-group has the ability to failover individually.

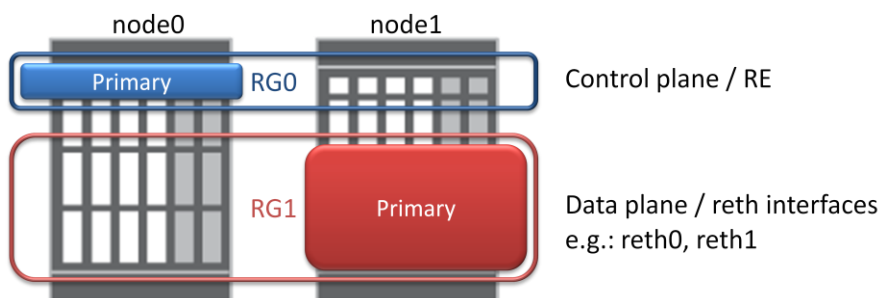
In the below diagram, every redundancy-group can move from left to right and vice versa independently of the other redundancy-groups. This means that the control plane redundancy-

group (RG0) can be active on the same node or on a different node as the data plane redundancy-group (RG1). Also when there are multiple data plane redundancy-groups configured (RG1 and higher), each of them can be active on either node0 or node1, independent of each other.

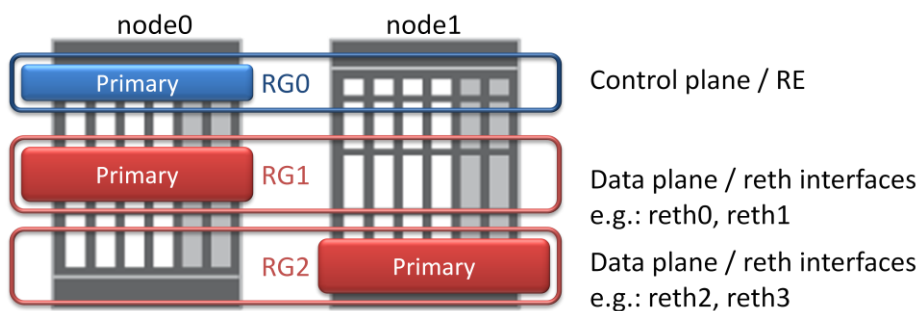
Both RG0 and RG1 can be active on the same node



RG0 and RG1 can be active on different nodes



A/A setup: multiple RG1+ configured, and each can be active on a different node



When the data plane redundancy-groups (RG1 or higher, also referred to as RG1+) failover, the reth interfaces belonging to the redundancy-group move towards the physical interfaces on the new primary node.

Note: If Redundancy group 0 fails over, dynamic routing processes such as ISIS, OSPF, BGP will restart causing possible loss of routes on peer devices. Use of 'graceful-restart' is recommended to avoid loss of neighbor routing during a RG0 failover event. Graceful-restart can be enabled with the command 'set routing-options graceful-restart'.

In order to update the mac address tables on connected switches after a failover, G-ARPs will be broadcast from the new physical interface in use (for each logical interface). The G-ARP packets use the reth interface virtual mac address as the source mac address and hence the connected switches will learn that this mac address has moved to the new port on the switched network. Layer3 devices (routers, hosts) do not need any manual intervention after a failover, because the reth interface will keep using the same mac address and the same IP address. For transparent mode setups, upon failover the previous physical interface is taken down for 2 seconds to update the CAM/ARP tables on the connected devices.

A failover event can be triggered manually or automatically, based on system monitored events as well as configurable options. For more details on that, see *Chassis Cluster Failover Behaviors & Optional Failover Configuration*.

HA links

In a cluster there needs to be a lot of communication between the two cluster nodes. This communication is mainly required to synchronize states and send keepalives to detect that the other node is still present.

This communication is going over two different links: a 'control' link and a 'fabric' link.

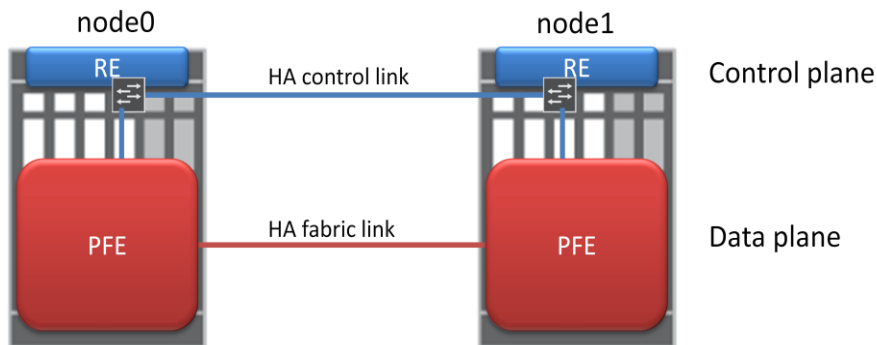
The control link is used to send control traffic between both the REs and between the Primary RE (RG0 primary) and the remote PFE.

The fabric link connects both PFE's together. This link is utilized for two main functions.

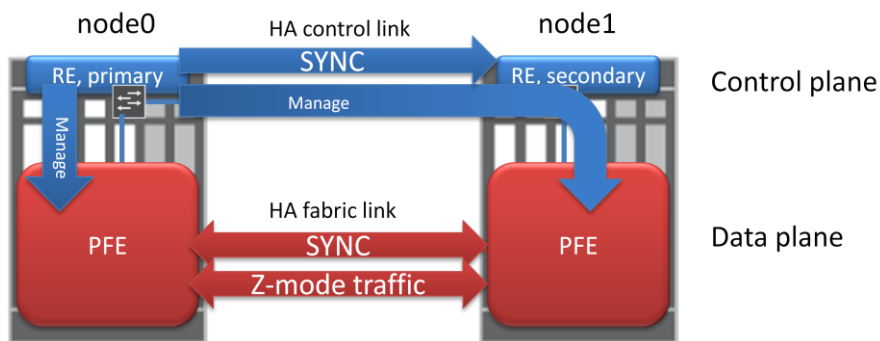
The first function is to synchronize the session states between the two nodes. This is done via RTO (real-time objects) packets going over the fabric link. There are many types of RTO messages, but some of the most important ones are the 'session create' and 'session delete' messages.

The second function of the fabric link is to pass traffic that needs to cross both nodes. This can only occur in A/A scenarios where traffic might enter an interface on one cluster node and needs to exit out of an interface on the other cluster node. Such traffic is also called Z-mode traffic.

Chassis cluster HA links concept



Chassis cluster HA links usage



If a control or fabric link fails, it will not result in an outage. However the redundancy of the cluster will be impacted (see for details chapter *Chassis Cluster Failure Behavior for Control and Fabric Links*). Only if both control and fabric links fail at the same time, and hence both nodes see the other node as 'lost', it would result in both nodes becoming Primary for all redundancy-groups, which will impact traffic due to the duplicate IP addresses (all reth active on both nodes) and lack of session sync.

To avoid the control or fabric link functionality from failing, there is the option to add a second control link (High-end SRX models) and/or a second fabric link (all SRX models). This will provide redundancy in case of a control or fabric link failure.

For more details on control or fabric link failure modes and requirements for using dual control or fabric links, see chapter *Chassis Cluster Failure Behavior for Control and Fabric Links*.

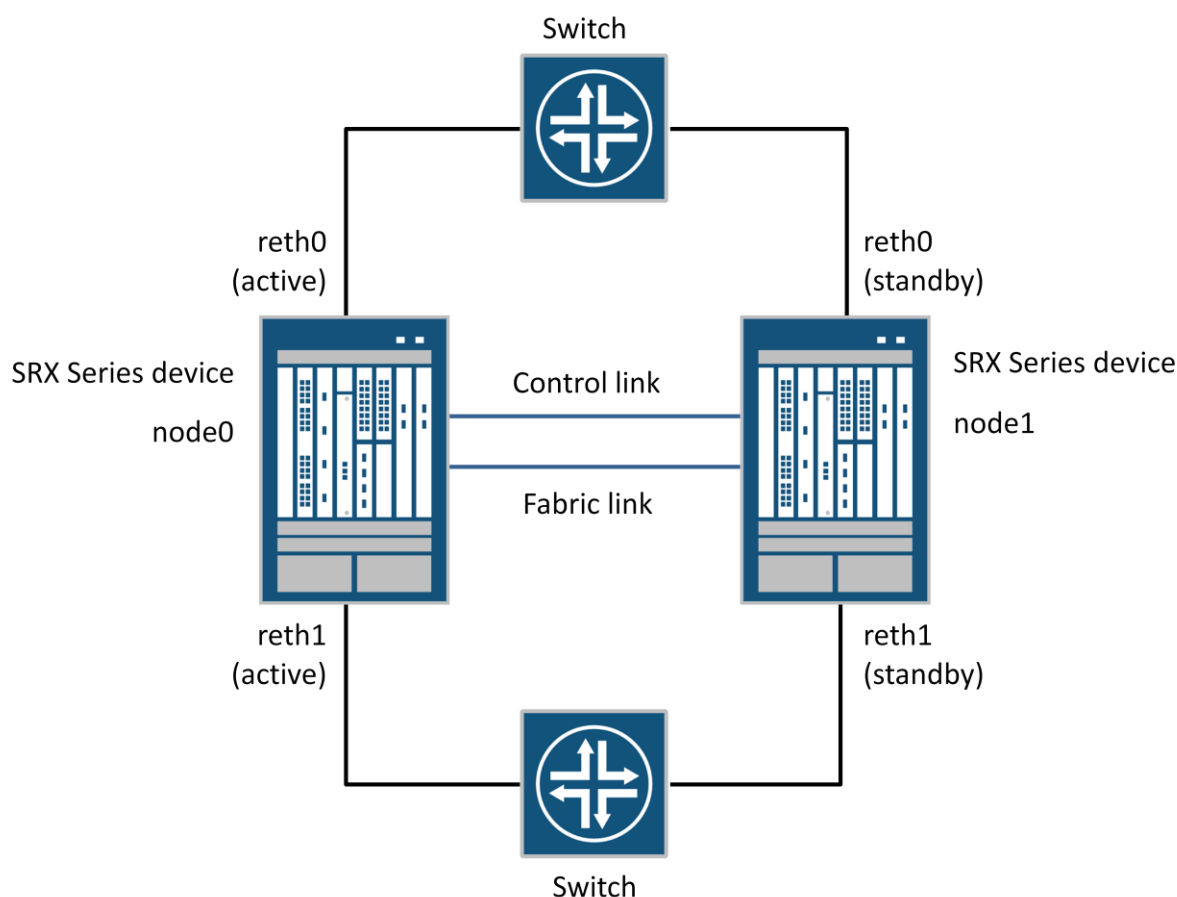
Scenarios for Chassis Cluster and Configuration Steps

1) Basic Active/Standby scenario

The basic chassis cluster scenario consists of only the most basic features of chassis cluster in SRX series devices. The other scenarios will build on this one.

In this basic scenario there are two SRX devices (node0 and node1), connected by two HA links (control and fabric). Reth interfaces are configured for each node and they are connected together by one or more switches.

This basic chassis cluster can be represented as follows.



In this case there are two reth interfaces (reth0 and reth1). A reth interface bundles the two physical interfaces (one from each node) together. A reth interface is part of a redundancy-group. Only the member that is on the node which is 'primary' (active) for that redundancy-group, will be active. The member on the 'secondary' (passive) node will be inactive, i.e. not send/receive any traffic³.

³ The secondary interface itself will actually still pass traffic, but there will be no flow processing for it. I.e., it is flow processing that stops the traffic. One example of traffic that can still go through the interface is ARP requests received on the secondary node interface, which will be sent to the active RE. The response to the

Each reth interface can have one or more logical/subinterfaces under it (e.g. reth0.0, reth0.1, reth0.2), each using a different vlan tag.

In a chassis cluster, both nodes together will form one logical chassis. This is reflected in the slot numbering, which continues over the two nodes as if they were one device. For example, in an SRX210 cluster, on node1 the port that was originally ge-0/0/0, will be referred to as ge-2/0/0 in the configuration. SRX210 node1 has a slot numbering offset of 2, meaning that 2 needs to be added to the slot designation of interfaces in node1. Table 1 indicates the slot numbering offset for all SRX platforms.

Table 1 also indicates for each SRX platform which ports are used for control, fabric and fxp0 ports.

SRX model	Control ports	Fabric ports	Management (fxp0)	node1 slot numbering offset
SRX100	fe-0/0/7	any interface (must be configured)	fe-0/0/6	1
SRX210	fe-0/0/7	any interface (must be configured)	fe-0/0/6	2
SRX220	ge-0/0/7	any interface (must be configured)	ge-0/0/6	3
SRX240	ge-0/0/1	any interface (must be configured)	ge-0/0/0	5
SRX550	ge-0/0/1	any interface (must be configured)	ge-0/0/0	9
SRX650	ge-0/0/1	any interface (must be configured)	ge-0/0/0	9
SRX1400	ge-0/0/10 and ge-0/0/11	any interface (must be configured)	built-in fxp0 port on RE	4
SRX3400	built-in control ports on front panel	any interface (must be configured)	built-in fxp0 port on RE	8
SRX3600	built-in control ports on front panel	any interface (must be configured)	built-in fxp0 port on RE	13
SRX5600	built-in control ports (port 0) on SPC card (must be configured)	any interface (must be configured)	built-in fxp0 port on RE	6
SRX5800	built-in control ports (port 0) on SPC card (must be configured)	any interface (must be configured)	built-in fxp0 port on RE	12
Note: SRX110 does not support chassis cluster				

Table 1. Chassis cluster interfaces to be used for each SRX model

Configuring the basic chassis cluster is done through the following steps.

1. Make sure that both devices that will be clustered together are running the exact same Junos software release.
2. Connect both the control link and fabric link between the two devices before configuring chassis cluster. Which ports can be used for the control and fabric link, and optional fxp0 for out of band management, depends on the model of SRX and is specified below in Table 1.
3. From the device configuration remove all references to the interfaces that will be used as control link, fabric link or fxp0. The easiest and cleanest way to do this is to simply delete the current configuration, as in this example:

```
[edit]
```

```
root@barbrady# delete
```

ARP requests will be sent out on the primary node interface. Another example is IP Monitoring, which can send pings from the secondary node as well, to test connectivity.

```

This will delete the entire configuration

Delete everything under this level? [yes,no] (no) yes

root@barbrady# set system root-authentication plain-text-password

New password:

Retype new password:

[edit]

root@barbrady# commit

commit complete

```

Alternatively, in case you want to keep the rest of the configuration, just delete the interfaces that will become control link, fxpo and fabric link in cluster mode. Also delete all references to those interfaces (e.g. under security zones). Then commit the configuration.

4. In operational mode, configure chassis cluster-id (non-zero value) and node-id on each node separately. Then reboot for the changes to take effect and change from standalone mode to cluster mode. In this example the cluster-id is selected to be 1.

Execute the following operational command in node 0

```
set chassis cluster cluster-id 1 node 0 reboot
```

Execute the following operational command in node 1

```
set chassis cluster cluster-id 1 node 1 reboot
```

5. Only for SRX5600/5800 devices, in configuration mode on both cluster nodes configure the connected control link on both nodes. Once committed, both nodes should see each other to allow synchronization of configuration for the following configuration steps, which are done on only one cluster node.

Execute the following configuration commands on each node separately (the FPC numbers here are example).

```

set chassis cluster control-ports fpc 0 port 0
set chassis cluster control-ports fpc 12 port 0
commit

```

6. In configuration mode on either node0 or node1 (configuration will be synced), configure fabric link interfaces (interface numbers here are example).

```

set interface fab0 fabric-options member-interfaces xe-1/0/0
set interface fab1 fabric-options member-interfaces xe-13/0/0

```

7. Configure node specific configuration

```
set groups node0 system host-name SRX1
set groups node0 system backup-router <gateway-ip-address> destination <subnet to be reached>
set groups node0 system interface fxp0 unit 0 family inet address 172.16.1.1/24

set groups node1 system host-name SRX2
set groups node1 system backup-router <gateway-ip-address> destination <subnet to be reached>
set groups node1 system interface fxp0 unit 0 family inet address 172.16.1.2/24

set apply-groups "${node}"
```

8. Configure RG1+ at chassis cluster and define the amount of reth interfaces to be used

The following commands will create RG1 for node0 and node 1.

```
set chassis cluster redundancy-group 1 node 0
set chassis cluster redundancy-group 1 node 1
```

The priorities for the RG on each node are defined as follows.

```
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1 # standby (default priority is 100, the
higher number has higher priority for active/standby decision)

set chassis cluster reth-count 2 # it will allow to create two reth interfaces (e.g., reth0 and reth1)

set interfaces <interface name> gigether-options redundant-parent reth0
set interfaces <interface name + N> gigether-options redundant-parent reth0

# N means the total number of FPC for each SRX platform (i.e. slot numbering offset in Table 1).

set interfaces <interface name> gigether-options redundant-parent reth1
set interfaces <interface name + N> gigether-options redundant-parent reth1

# N means the total number of FPC for each SRX platform (i.e. slot numbering offset in Table 1).
```

9. Configure the reth interfaces and apply them to the desired redundancy-group

```
set interface reth0 redundant-ether-options redundancy-group 1
set interface reth1 redundant-ether-options redundancy-group 1
set interface reth0 unit 0 family inet address 10.1.1.1/24
set interface reth1 unit 0 family inet address 10.2.1.1/24

set security zones security-zone untrust interface reth0.0
set security zones security-zone trust interface reth1.0
```

An example of basic chassis cluster configuration is as follows (example from SRX240, with node1 slot number offset of 5):

```
set groups node0 system host-name SRX-A
set groups node0 interfaces fxp0 unit 0 family inet address 172.30.72.207/23
set groups node1 system host-name SRX-B
set groups node1 interfaces fxp0 unit 0 family inet address 172.30.72.213/23
set apply-groups "${node}"
set system root-authentication encrypted-password #####
set chassis cluster reth-count 2
set chassis cluster redundancy-group 1 node 0 priority 200
```

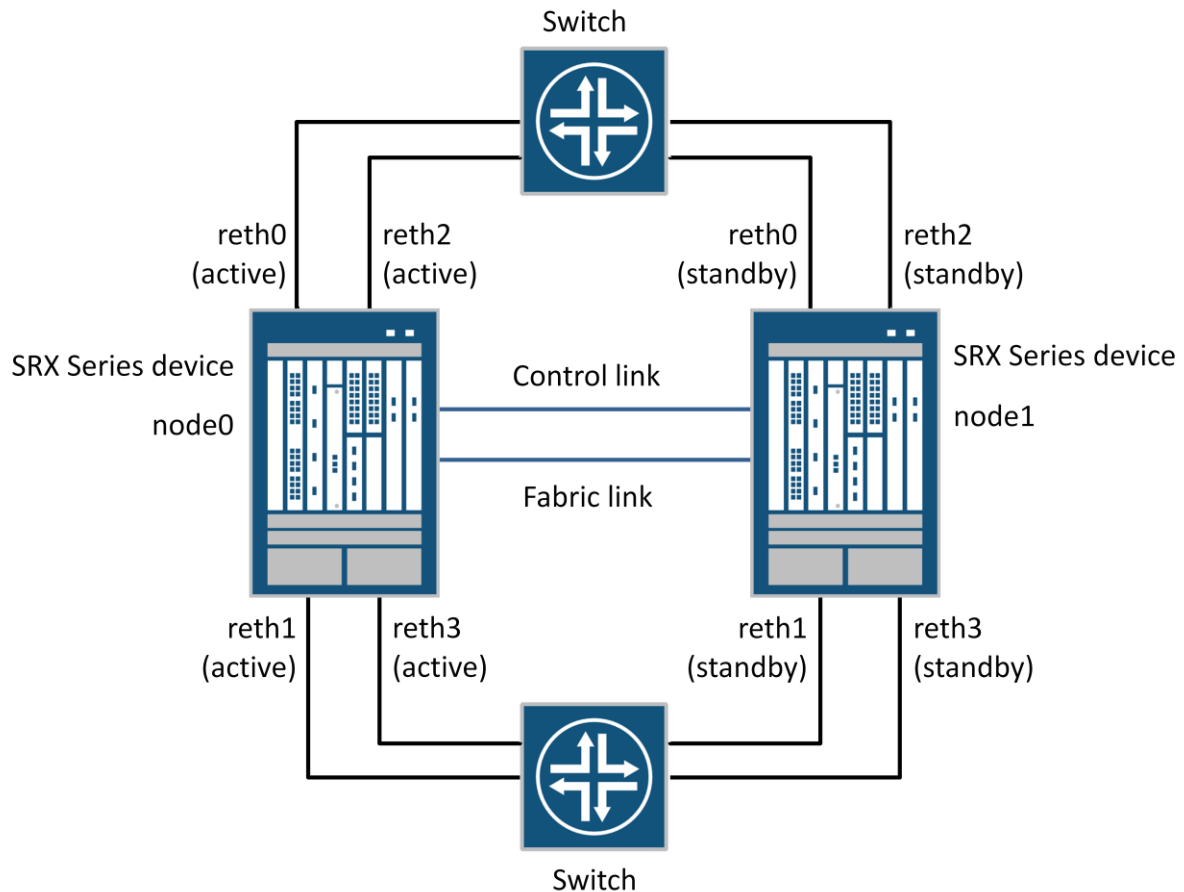
```

set chassis cluster redundancy-group 1 node 1 priority 100
set interfaces fab0 fabric-options member-interfaces ge-0/0/2
set interfaces fab1 fabric-options member-interfaces ge-5/0/2
set interfaces ge-0/0/14 gigether-options redundant-parent reth0
set interfaces ge-0/0/15 gigether-options redundant-parent reth1
set interfaces ge-5/0/14 gigether-options redundant-parent reth0
set interfaces ge-5/0/15 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.1.1.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.2.1.1/24
set security policies from-zone trust to-zone untrust policy P1 match source-address any
set security policies from-zone trust to-zone untrust policy P1 match destination-address any
set security policies from-zone trust to-zone untrust policy P1 match application any
set security policies from-zone trust to-zone untrust policy P1 then permit
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces reth0.0
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces reth1.0

```

2) Active/Standby with multiple RETHs

In this example additional reth interfaces for a total of 4 reths are added to allow for more physical interface connections with different subnets.



To add more reth interfaces, first make sure that the reth-count is configured to a high enough number to allow for expanded reth interface use. Add the new reth interfaces to RG1; then assign physical interfaces to the new reth interfaces. Finally, as with any other interface type, add IP addresses to the reth logical interfaces and associate the interfaces to security zones.

```
set chassis cluster reth-count 4
set interfaces ge-0/0/16 gigether-options redundant-parent reth2
set interfaces ge-0/0/17 gigether-options redundant-parent reth3
set interfaces ge-5/0/16 gigether-options redundant-parent reth2
set interfaces ge-5/0/17 gigether-options redundant-parent reth3

set interfaces reth2 unit 0 family inet address 10.1.2.1/24
set interfaces reth2 redundant-ether-options redundancy-group 1
set interfaces reth3 unit 0 family inet address 10.2.2.1/24
set interfaces reth3 redundant-ether-options redundancy-group 1

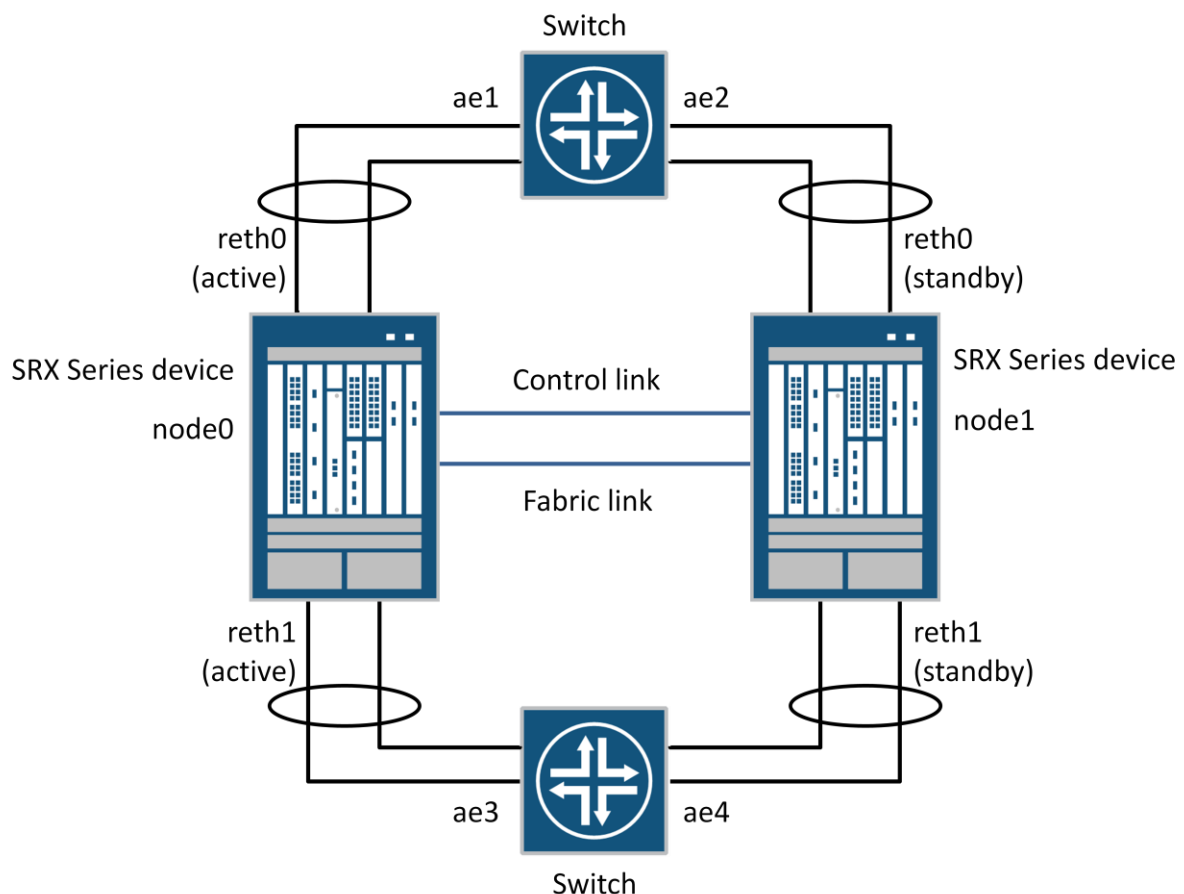
set security zones security-zone untrust2 interfaces reth2.0
set security zones security-zone trust2 interfaces reth3.0
```

Again, each reth interface can contain multiple logical sub-interfaces (reth0.1, reth0.2, reth1.3 etc), using multiple vlans on the switch. The main benefit of adding more reth interfaces is that there will also be more physical interfaces used. This increases the total physical interface bandwidth available.

Another option that can be used with multiple reth interfaces is A/A cluster which will be discussed in topology example 4.

3) Active/Standby with LAG/LACP

Another addition that can be done to a chassis cluster is using link aggregation (LAG) to bundle multiple physical interfaces together. This can increase redundancy as well as available bandwidth. Optionally a LAG can be managed by LACP (link aggregation control protocol).



Multiple interfaces can be bundled in a reth by simply adding another physical interface to the same reth interface. There is no need to configure an 'ae' interface (which is not supported in a cluster).

```
set interfaces ge-0/0/14 gigether-options redundant-parent reth0 ##existing config
set interfaces ge-0/0/15 gigether-options redundant-parent reth1 ##existing config
set interfaces ge-5/0/14 gigether-options redundant-parent reth0 ##existing config
set interfaces ge-5/0/15 gigether-options redundant-parent reth1 ##existing config

set interfaces ge-0/0/16 gigether-options redundant-parent reth0
set interfaces ge-0/0/17 gigether-options redundant-parent reth1
set interfaces ge-5/0/16 gigether-options redundant-parent reth0
set interfaces ge-5/0/17 gigether-options redundant-parent reth1
```

This will create the following bundles:

on node0, ge-0/0/14 and ge-0/0/16 will be bundled in a LAG in reth0

on node0, ge-0/0/15 and ge-0/0/17 will be bundled in a LAG in reth1

on node1, ge-5/0/14 and ge-5/0/16 will be bundled in a LAG in reth0

on node1, ge-5/0/15 and ge-5/0/17 will be bundled in a LAG in reth1

On the connected switch, these four bundles each must be connected to a separate LAG bundle ('ae' interface in case of Juniper EX switch). This means for example that the two reth0 links from node0 need to be connected to a different ae interface on the switch than the two reth0 links from node1.

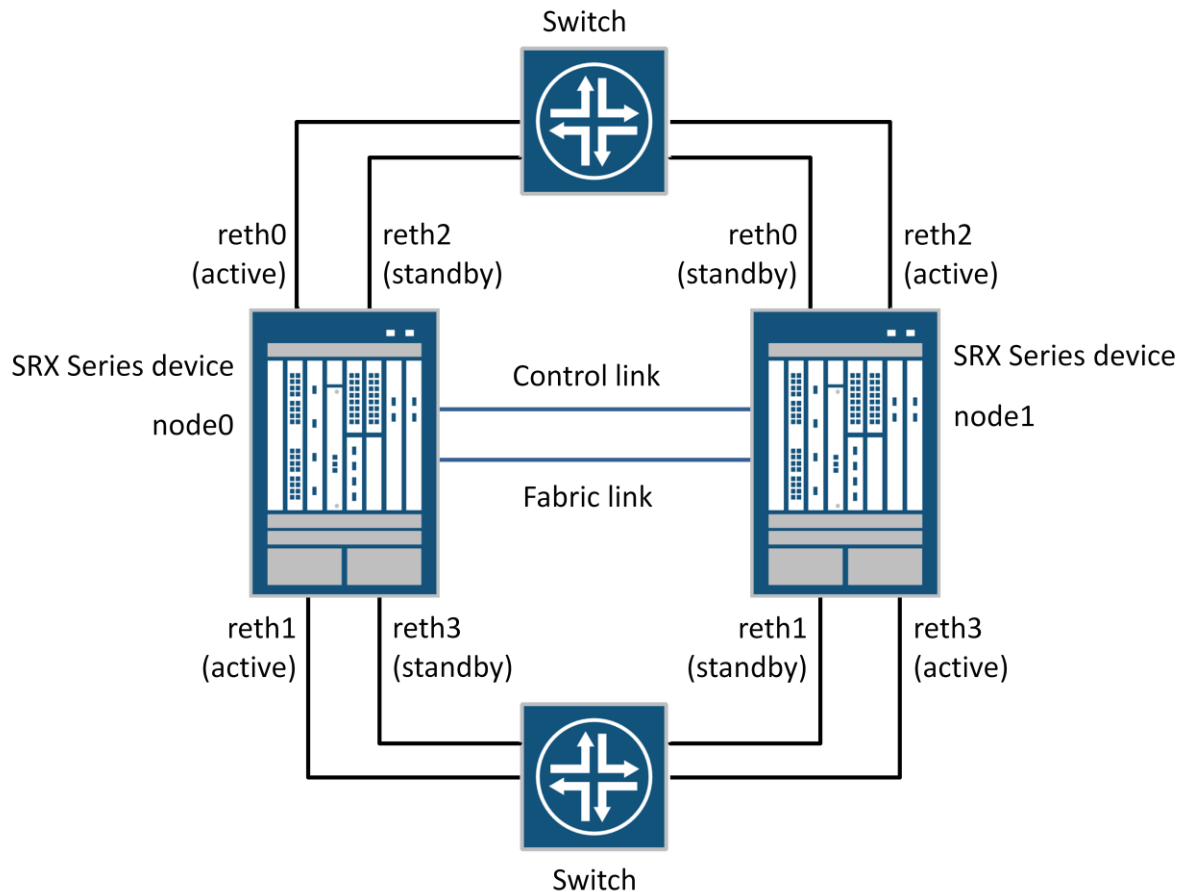
LACP can optionally be added to manage the bundle:

```
set interfaces reth0 redundant-ether-options lacp active
set interfaces reth1 redundant-ether-options lacp active
```

4) Active/Active

In stead of using just one redundancy-group for the data plane (RG1), it is possible to use multiple redundancy-groups for the data plane. When there are multiple RG1+ redundancy-groups, both the nodes can be the primary node for one or more redundancy-groups in the data plane. This means that some reth interfaces can be active on one node, while other reth interfaces are active on the other node. This way an Active/Active cluster can be created, where both nodes carry traffic at the same time.

This example is based on Scenario 1 and it is similar to Scenario 2 "Active/Standby with multiple RETHs". In Scenario 2, all the reth interfaces, reth1-4 , were members of the same redundancy-group (RG1). The difference now is that we create at least one additional RG (RG2) with corresponding reth interfaces (reth2 and reth3 in this case). Each RG can be active on a different node.



Configuration to add to the Scenario 1 configuration:

```
set chassis cluster reth-count 4
set interfaces ge-0/0/16 gigether-options redundant-parent reth2
set interfaces ge-0/0/17 gigether-options redundant-parent reth3
set interfaces ge-5/0/16 gigether-options redundant-parent reth2
set interfaces ge-5/0/17 gigether-options redundant-parent reth3

set interfaces reth2 unit 0 family inet address 10.1.2.1/24
set interfaces reth2 redundant-ether-options redundancy-group 2
set interfaces reth3 unit 0 family inet address 10.2.2.1/24
set interfaces reth3 redundant-ether-options redundancy-group 2
set security zones security-zone untrust2 interfaces reth2.0
set security zones security-zone trust2 interfaces reth3.0
```

In order to make one RG active on each node, priorities for the RG need to be set:

```
set chassis cluster redundancy-group 1 node 0 priority 100  ## Higher priority
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 2 node 0 priority 1
set chassis cluster redundancy-group 2 node 1 priority 100  ## Higher priority
```

Z-mode traffic

Since both nodes now have active interfaces, it can happen that traffic needs to be routed between two interfaces that are active on different nodes. This is called Z-mode traffic. The traffic that needs to travel between the nodes will be forwarded over the fabric link.

With Z-mode processing, the first packet of a session is received on one cluster node (the ingress node). When flow determines that the egress interface is located on the second node, the packet is forwarded over the fabric link with a forward session setup on the ingress node. The packet is then processed by the second node upon which an Active session is installed and the packet is forwarded out the egress link. Finally a backup session is created for the Active session in the initial ingress node.

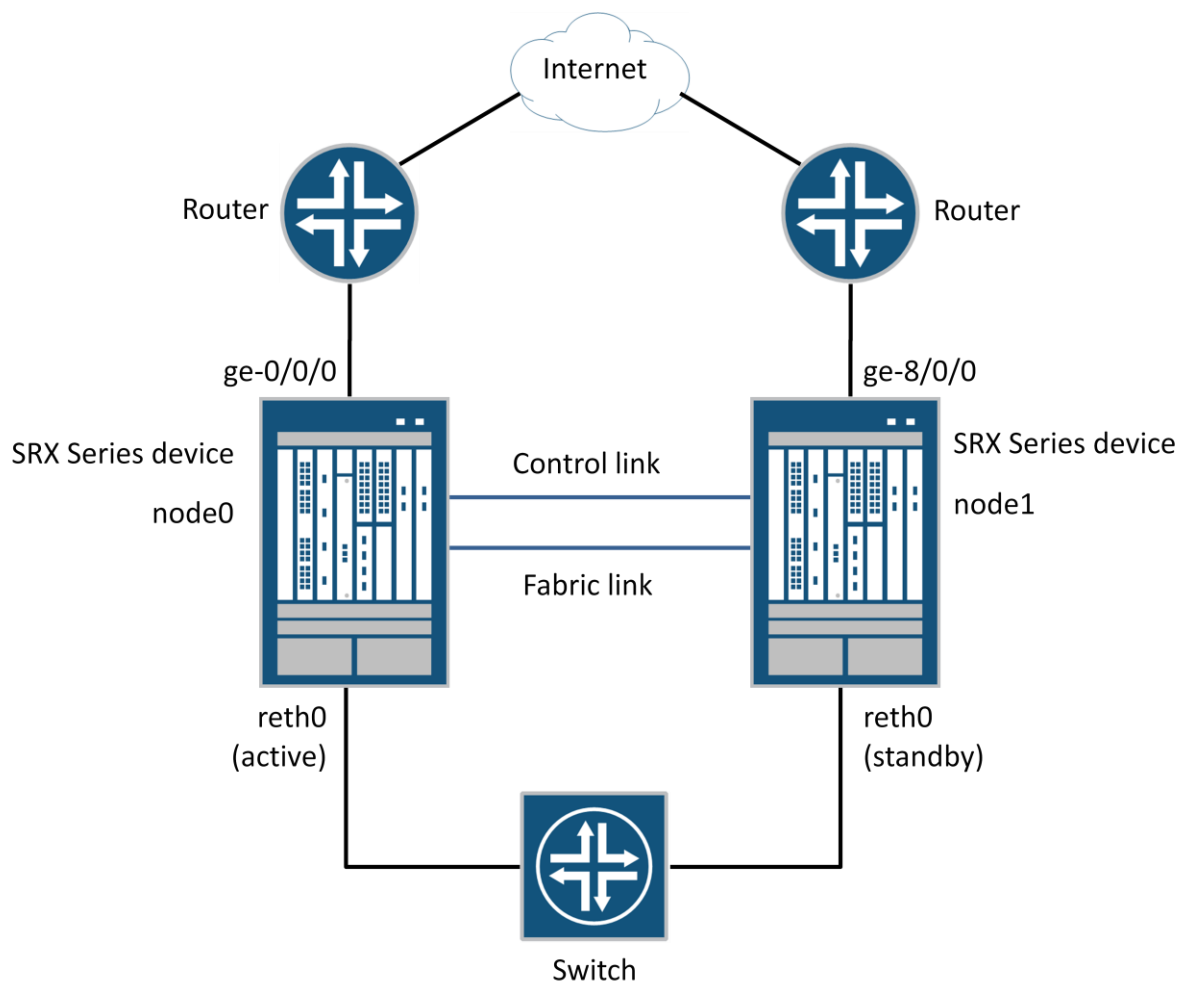
In general it is best to avoid the occurrence of Z-mode traffic as much as possible. This is due to the possible bandwidth limitations of the fabric link (only one physical link is being used) and also some latency will be added.

5) Use of non-reth interfaces with dual ISP

In some cases it can be useful to use the physical interfaces instead of reth interfaces in a cluster. It is possible to combine reth and physical (also called *local*) interfaces in a cluster. It is even possible to not use any reth interfaces at all, but this is not recommended due to functionality that may be missed when not using the reth interfaces (e.g. ipsec VPN can only be terminated through reth interfaces; also without reth, existing sessions may be cleared in case one node suffers from a power failure and hence interfaces used in the session are disappearing).

The combination of reth and physical interfaces is a possible design for example in case there are two links to ISPs while each node connects to one ISP router.

In the following diagram both physical interfaces and reth interfaces are combined, where the physical interfaces would be used to connect to the ISPs, while the reth interface(s) face the internal network.



Routing may be arranged by using dynamic routing protocols. If traffic that comes in on one node has to exit out of an interface on the other node, then it will be transferred to the other node over the HA Fabric link.

If one half of the cluster (one node) fails, then the routing protocols will have to make sure that the traffic will continue over the other node.

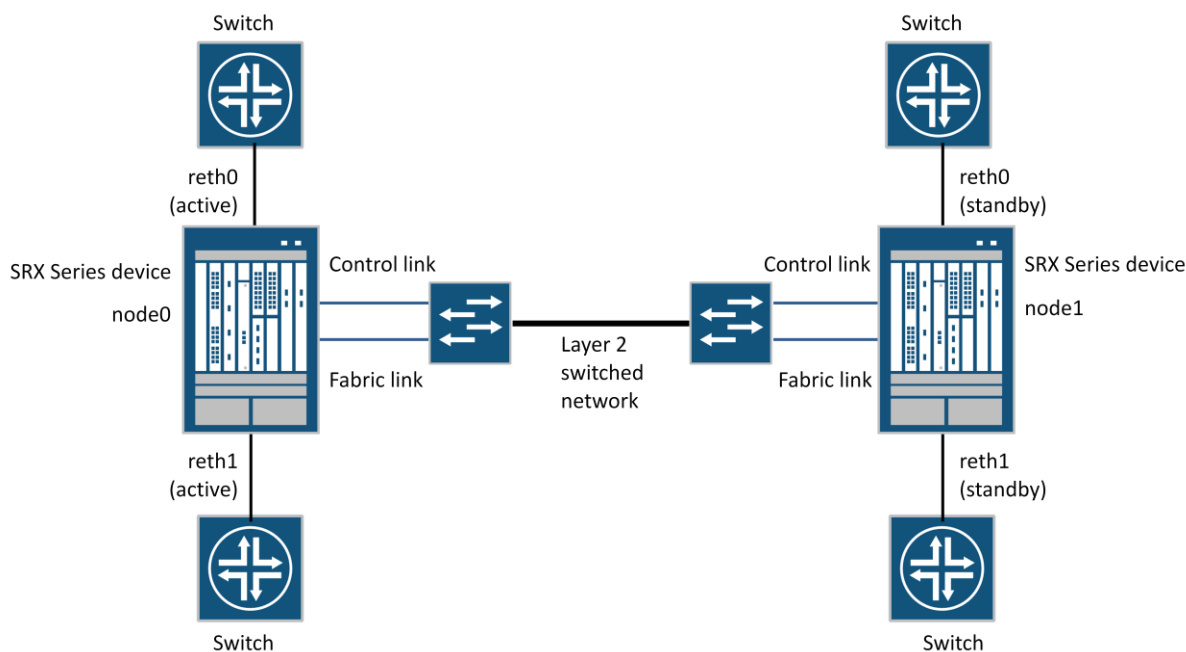
When using physical interfaces in stead of reth interfaces, the level of redundancy is less than when only reth interfaces are used. In the above scenario, when one node would suffer a total power loss, the result would be that the physical interfaces on that node disappear. Flow sessions that were using such disappeared interface cannot be rerouted, but will be deleted. However sessions will be rerouted when an interface goes down or if there there is otherwise a route change. A reroute can only be successful when both the old and new interface used are in the same security zone.

To make sure that the return traffic for sessions initiated from a physical interface, also returns to the same interface, to avoid asymmetric routing, source NAT can be used for sessions initiated from the internal network and going towards the internet.

Note that with this design there may be a lot of Z-mode traffic going over the fabric link. As mentioned earlier this can bring a bandwidth constraint that has to be kept into account.

6) HA links passing through a switched network

Cluster nodes are connected together by the control and fabric links. These links are used for communication inside the cluster and for passing traffic between the nodes. These links are normally directly connected from node to node. However in some cases it may be desirable to be able to pass the HA link connections through a pair of switches or a switched network. That is possible.



In such a case the switched connections have to be setup in a way that from the perspective of the cluster nodes there is no difference with using directly connected HA links to the other node. This means that all the frames that are sent out by one node onto a control or fabric link have to arrive on the other node's corresponding interface, unchanged, in the same order and with a low latency (<100ms).

Each HA link needs to be in a unique vlan on the switched network with no other traffic used for these vlans. These vlans simulate the physical HA links that would be between the nodes when connected directly.

Also when using dual control or fabric links, it is important that each individual link will be using a unique vlan on the switch. This will make sure that packets do not arrive on the wrong port at the other side, which could lead to issues such as flapping ctrl/fab links.

Inside the switched L2 network, between the switches, the vlans used for HA can run over trunks, as long as there is at least 1Gbps bandwidth (10Gbps if using XE fabric link) available for the HA traffic.

The requirements for the HA links when using switches are as follows:

- latency: max 100 ms
- bandwidth: min 1Gbps

- MTU: Fabric links require jumbo frames support, at least 9014 bytes (except for SRX100 which requires at least 1632 bytes MTU).
- When using Z-mode traffic in an A/A cluster, the local/revenue interfaces can have a maximum MTU of 8900 (except for SRX100, which can have 1500 max MTU). Otherwise the packets with some overhead will not be able to fit in the fabric link MTU which is used for the Z-mode forwarding.
- frames need to remain unchanged
- no reordering
- no dropping of frames (depending on the switch, may need to disable IP checksum / length check, igmp snoop and other features that might drop some of the frames).
- switch ports connected to the SRX HA ports are untagged (access port)

NOTE: On SRX-Branch devices (SRX100, 210, 240, 650) prior to Junos 10.2 the control link traffic was tagged with vlan-id 4094. Since Junos 10.2 and higher releases, the control port traffic uses experimental Ether type 0x88b5. For these releases, the vlan tagging can be enabled and disabled using an operational mode command. By default, no vlan tagging will be used when clustering is getting enabled in Junos 10.2 and higher. However, if clustering was enabled in an older version, then after upgrade the vlan tagging will still be enabled. This is something to consider when one of the devices in a cluster gets replaced. If one node uses the vlan tagging and the other one does not, then the control link will remain marked down.

Scenarios:

Upgrade clustered device from 10.1 or previous versions: use of 4094 tag will be maintained by default.

Enable chassis cluster while on Junos 10.2 or higher: No tagging will be used by default

To remove tagging use without needing to disable chassis cluster.

```
>set chassis cluster control-link-vlan enable/disable (A reboot is required)
```

To view status of control port tagging

```
root> show chassis cluster information detail | find tagging
Control port tagging:
Disabled
```

Chassis Cluster Optional Failover Configuration

In a chassis cluster, redundancy-groups can perform failover from the primary node to the secondary node based on system events as well as configurable options. Such failover capability allows for minimal traffic flow interruption caused by network, software or hardware related issues.

Failover Trigger Events

Redundancy Group 0 (Control Plane)

- Manually
- Automatic
 - RE hardware failure (full chassis failover)
 - RE software failure, vmcore coredump
 - SFB Failure (SRX5k only) (full chassis failover)
 - Full power loss on primary node (full chassis failover)

Redundancy Groups 1+ (Data Plane)

- Manually
- Automatic
 - SPC Failure (SRX-HE)
 - SPC Memory leak detection (SRX-HE)
 - Core dumps on PFE plane (includes CP on SRX-HE)
 - Central Point SPU (CP) Failure (SRX-HE)
 - Failure of RE (full chassis failover)
 - SFB Failure (SRX-HE full chassis failover)
 - Full power loss on primary node
- Configurable options
 - Interface Monitoring
 - Preempt
 - IP Monitoring

Manual Failover

A manual failover allows for a controlled failover of an RG group for testing or device management needs. Upon manually failing a redundancy-group the RG group priority on the new primary node will be adjusted from the configured value to 255 and be marked as a manual failover. Further failover events may still occur based on system events or monitored items. To allow normal operation of 'Preempt' or to allow further manual failovers, the manual failover state must be cleared.

Note: RG-0 has a built in hold time of 5 minutes to prevent rapid back-to-back failovers of RG0. During this time frame the status will show as secondary-hold.

```
>request chassis cluster failover redundancy-group [#] node [#]
```

```
root> show chassis cluster status
```

```
Cluster ID: 12
```

Node	Priority	Status	Preempt	Manual failover
------	----------	--------	---------	-----------------

```
Redundancy group: 0 , Failover count: 0
```

node0	200	primary	no	no
node1	100	secondary	no	no

```
Redundancy group: 1 , Failover count: 0
```

node0	200	primary	yes	no
node1	100	secondary	yes	no

Manual failover of RG0 and RG1.

```
root> request chassis cluster failover redundancy-group 0 node 1
node1:
```

```
-----
Initiated manual failover for redundancy group 0
```

```
root> request chassis cluster failover redundancy-group 1 node 1
node1:
```

```
-----
Initiated manual failover for redundancy group 1
```

```
root> show chassis cluster status
```

```
Cluster ID: 12
```

Node	Priority	Status	Preempt	Manual failover
------	----------	--------	---------	-----------------

```
Redundancy group: 0 , Failover count: 1
```

node0	200	secondary	no	yes
node1	255	primary	no	yes

```
Redundancy group: 1 , Failover count: 1
```

node0	200	secondary	yes	yes
node1	255	primary	yes	yes

Clearing of manual failover of RG0 and RG1.

```
root> request chassis cluster failover reset redundancy-group 0
node0:
```

```
-----
No reset required for redundancy group 0.
```

```
node1:
```

```
-----
Successfully reset manual failover for redundancy group 0
```

```
{secondary:node0}
```

```
root> request chassis cluster failover reset redundancy-group 1
node0:
```

No reset required for redundancy group 1.

node1:

Successfully reset manual failover for redundancy group 1

root> show chassis cluster status

Cluster ID: 12

Node	Priority	Status	Preempt	Manual failover
------	----------	--------	---------	-----------------

Redundancy group: 0 , Failover count: 2

node0	200	secondary	no	no
node1	100	primary	no	no

Redundancy group: 1 , Failover count: 2

node0	200	secondary	no	no
node1	100	primary	no	no

Configurable Data Plane Failover options

- Interface Monitoring

Interface Monitoring allows data redundancy groups (RG1 or higher) to failover based upon a monitored link being physically up or down. Each interface is configured a weight value that upon a link loss is deducted from the redundancy group threshold of 255. Once the redundancy group threshold reaches 0 the RG group is failed over to the secondary node. Please note that the threshold of 255 is hard-coded and cannot be changed.

Example Setup Using the Basic Active/Passive Scenario for SRX240

In our example we are going to monitor all 4 physical links of the reth interfaces and assign them a weight value of 150. If a link were to report down the system would subtract the configured monitored weight of that link and subtract it from the redundancy group threshold of 255. If only 1 link were to report down it would result in a RG group weight of 105. As the RG group weight has not reached 0 there would be no failover as seen below. Note that in this case the redundancy-group priority will remain unchanged.

```
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/14 weight 150
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/15 weight 150
set chassis cluster redundancy-group 1 interface-monitor ge-5/0/14 weight 150
set chassis cluster redundancy-group 1 interface-monitor ge-5/0/15 weight 1504
```

```
root> show chassis cluster interfaces | find Monitoring
```

⁴ Note: in this example the physical interfaces are being monitored. Alternatively the reth interface itself can be monitored, and trigger a failover if the reth interface goes down, i.e. when all it's member physical interfaces on the node have gone down:

```
set chassis cluster redundancy-group 1 interface-monitor reth0 weight 255
```

Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-5/0/15	150	Up	1
ge-5/0/14	150	Up	1
ge-0/0/15	150	Up	1
ge-0/0/14	150	Down	1

root> show chassis cluster status

Cluster ID: 12

Node	Priority	Status	Preempt	Manual failover
------	----------	--------	---------	-----------------

Redundancy group: 0 , Failover count: 0

node0	200	primary	no	no
node1	100	secondary	no	no

Redundancy group: 1 , Failover count: 0

node0	200	primary	no	no
node1	100	secondary	no	no

root> show chassis cluster information

node0:

Redundancy mode:

Configured mode: active-active

Operational mode: active-active

Redundancy group: 0, Threshold: 255, Monitoring failures: none

Events:

Redundancy group: 1, Threshold: 105, Monitoring failures: interface-monitoring

Events:

node1:

Redundancy mode:

Configured mode: active-active

Operational mode: active-active

Redundancy group: 0, Threshold: 255, Monitoring failures: none

Events:

Redundancy group: 1, Threshold: 255, Monitoring failures: none

Events:

Upon a second link failing on node0, the redundancy-group will failover to node1. After failover node0 will mark its priority as 0 to prevent failover for the redundancy-group until the initial failure condition has been resolved.

root> show chassis cluster interfaces | find Monitoring

Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-5/0/15	150	Up	1
ge-5/0/14	150	Up	1
ge-0/0/15	150	Down	1
ge-0/0/14	150	Down	1

root> show chassis cluster status

Cluster ID: 12

Node	Priority	Status	Preempt	Manual failover
------	----------	--------	---------	-----------------

Redundancy group: 0 , Failover count: 0

node0	200	primary	no	no
node1	100	secondary	no	no

Redundancy group: 1 , Failover count: 1

node0	0	secondary	no	no
node1	100	primary	no	no

root> show chassis cluster information

node0:

Redundancy mode:

Configured mode: active-active

Operational mode: active-active

Redundancy group: 0, Threshold: 255, Monitoring failures: none

Events:

Redundancy group: 1, Threshold: -45, Monitoring failures: interface-monitoring

Events:

Feb 15 03:27:41.107 : primary->secondary-hold, reason: Monitor failed: IF

Feb 15 03:27:42.111 : secondary-hold->secondary, reason: Back to back failover interval

node1:

Redundancy mode:

Configured mode: active-active

Operational mode: active-active

Redundancy group: 0, Threshold: 255, Monitoring failures: none

Events:

Redundancy group: 1, Threshold: 255, Monitoring failures: none

Events:

Feb 15 03:31:37.345 : secondary->primary, reason: Remote yeild (100/0)

- Preempt

The use of the *preempt* setting on a redundancy group allows for a redundancy group to failover based on the configured priority values. The cluster node with the higher configured priority is preferred for being the primary node when using *preempt*. Without the use of *preempt* a device will only failover based on failure event, and not upon recovery of that event.

NOTE: Preempt is not available for redundancy group 0

Example Setup Using the Basic Active/Passive Scenario for SRX240

In our example we will configure redundancy group 1 with a priority value of 200 for node0 and 100 for node1. We will also enable preempt to allow failback to node0 upon recovery.

```
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 100
```

```
set chassis cluster redundancy-group 1 preempt
```

Behavior:

Link monitoring has caused a failover event thus initiating a failover to node1. Node0 priority will show 0 until the cause for the failover event has recovered.

```
root> show chassis cluster interfaces | find Monitoring
Interface Monitoring:
Interface      Weight  Status  Redundancy-group
ge-5/0/15      255    Up      1
ge-5/0/14      255    Up      1
ge-0/0/15      255    UP      1
ge-0/0/14      255    Down    1

root> show chassis cluster status
Cluster ID: 12
Node           Priority  Status  Preempt  Manual failover

Redundancy group: 0 , Failover count: 0
node0          200     primary no       no
node1          100     secondary no       no

Redundancy group: 1 , Failover count: 1
node0          0     secondary yes     no
node1          100    primary  yes     no
```

Upon ge-0/0/15 link recovery, node0 will adjust the priority value from 0 back to its configured value. As the redundancy-group has been enabled with *preempt*, node0 configured priority of 200 is seen higher than node1 configured priority of 100, allowing fail back to node0.

```
root> show chassis cluster interfaces | find Monitoring
Interface Monitoring:
Interface      Weight  Status  Redundancy-group
ge-5/0/15      255    Up      1
ge-5/0/14      255    Up      1
ge-0/0/15      255    UP      1
ge-0/0/14      255    UP      1

root# run show chassis cluster status
Cluster ID: 12
Node           Priority  Status  Preempt  Manual failover

Redundancy group: 0 , Failover count: 0
node0          200     primary no       no
node1          100     secondary no       no

Redundancy group: 1 , Failover count: 2
node0          200    primary  yes     no
node1          100    secondary yes     no
```

NOTE: If a redundancy-group is configured so that both nodes have the same priority, the node with the lowest node-id number always takes precedence, and the redundancy-group is primary on it. In a two-node cluster, node 0 always takes precedence in a priority tie

- IP Monitoring

Unlike Interface Monitoring which watches for physical link loss, IP monitoring allows for failover based upon end-to-end reachability of a configured monitored IP address. The reachability test is done by sending ping to the monitored IP address and checking if a response comes back. The monitored IP address can be on a directly connected host in the same subnet as the reth interface or a remote device reachable through a next-hop router.

On the primary node, the source IP address used for the pings will be the reth interface IP address. From the secondary node there will only be pings sent if a *secondary-ip-address* in the same subnet has been configured. This *secondary-ip-address* will be used as the source IP address for the pings that are being sent from the secondary node.

If the primary or secondary node meets failover requirements based on IP monitoring, meaning ip monitor failed, it will mark its priority as 0. If this happens on the primary, a failover will occur. On the other hand, if the secondary node lowers its priority to 0, it will prevent a failover from primary node.

Example Setup Using the Basic Active/Passive Scenario for SRX240

In our example we will configure IP Monitoring for two next-hop devices. We will set the system to send pings every second with 5 losses required to declare unreachability to peer. We will also setup a *secondary-ip-address* to allow testing from the secondary node.

```
set chassis cluster redundancy-group 1 ip-monitoring global-weight 255
set chassis cluster redundancy-group 1 ip-monitoring global-threshold 100
set chassis cluster redundancy-group 1 ip-monitoring retry-interval 1
set chassis cluster redundancy-group 1 ip-monitoring retry-count 5
set chassis cluster redundancy-group 1 ip-monitoring family inet 10.1.1.2 weight 50
set chassis cluster redundancy-group 1 ip-monitoring family inet 10.1.1.2 interface
reth0.0 secondary-ip-address 10.1.1.3
set chassis cluster redundancy-group 1 ip-monitoring family inet 10.2.1.2 weight 50
set chassis cluster redundancy-group 1 ip-monitoring family inet 10.2.1.2 interface
reth1.0 secondary-ip-address 10.2.1.3
```

Behavior:

Upon failure of ip-monitoring to a next-hop device the configured monitored weight of 50 will be deducted from the configured global-threshold of 100. Once the configured global-threshold reaches 0 the configured global-weight of 255 will be subtracted from the current redundancy group threshold (default 255). Upon the redundancy-group threshold reaching 0 it will cause a failover event to the secondary node.

```
root> show chassis cluster ip-monitoring status
node0:
```

```
-----
Redundancy group: 1
```

IP address	Status	Failure count	Reason
10.2.1.2	reachable	0	n/a
10.1.1.2	reachable	0	n/a

```

node1:
-----

Redundancy group: 1

IP address      Status      Failure count Reason
10.2.1.2        reachable   0           n/a
10.1.1.2        reachable   0           n/a

```

The next CLI output shows loss of one of the monitored devices on node0. There is no failover event as the global-threshold has not been reached.

```

root> show chassis cluster ip-monitoring status
node0:
-----

Redundancy group: 1

IP address      Status      Failure count Reason
10.2.1.2        reachable   0           n/a
10.1.1.2        unreachable 1           no route to host

node1:
-----

Redundancy group: 1

IP address      Status      Failure count Reason
10.2.1.2        reachable   0           n/a
10.1.1.2        reachable   0           n/a

root> show chassis cluster status
Cluster ID: 12
Node           Priority      Status      Preempt  Manual failover

Redundancy group: 0 , Failover count: 0
  node0         200          primary     no       no
  node1         100          secondary   no       no

Redundancy group: 1 , Failover count: 0
  node0         200          primary     yes      no
  node1         100          secondary   yes      no

```

Loss of second monitored device on node0 causing failover of RG1 to node1.

```

root> show chassis cluster ip-monitoring status
node0:
-----

Redundancy group: 1

IP address      Status      Failure count Reason
10.2.1.2        unreachable 1           unknown
10.1.1.2        unreachable 1           unknown

node1:
-----

Redundancy group: 1

IP address      Status      Failure count Reason
10.2.1.2        reachable   0           n/a
10.1.1.2        reachable   0           n/a

root> show chassis cluster status
Cluster ID: 12
Node           Priority      Status      Preempt  Manual failover

Redundancy group: 0 , Failover count: 0
  node0         200          primary     no       no
  node1         100          secondary   no       no

```

```

Redundancy group: 1 , Failover count: 1
  node0                0          secondary  yes    no
  node1               100          primary    yes    no
  
```

Loss of monitored devices on node1 causing node1 to report a priority of 0 to prevent failover.

```

root> show chassis cluster ip-monitoring status
node0:
  
```

```

Redundancy group: 1
  
```

IP address	Status	Failure count	Reason
10.2.1.2	reachable	0	n/a
10.1.1.2	reachable	0	n/a

```

node1:
  
```

```

Redundancy group: 1
  
```

IP address	Status	Failure count	Reason
10.2.1.2	unreachable	1	no route to host
10.1.1.2	unreachable	1	no route to host

```

root> show chassis cluster status
  
```

```

Cluster ID: 12
  
```

Node	Priority	Status	Preempt	Manual failover
------	----------	--------	---------	-----------------

```

Redundancy group: 0 , Failover count: 0
  
```

node0	200	primary	no	no
node1	100	secondary	no	no

```

Redundancy group: 1 , Failover count: 0
  
```

node0	200	primary	yes	no
node1	0	secondary	yes	no

Chassis Cluster Failure Behavior for Control and Fabric Links

As inter-chassis cluster communication is important for liveness detection as well as state synchronization, a loss of control or fabric link connections impact normal operation of the cluster including possible impact to traffic forwarding. This section describes cluster node behaviour when there is a loss of control or fabric link communication.

Control Link communication loss only (Fabric communication is still successful)

The RG0 secondary node will transition to an Ineligible state then transition to a Disabled state

Fabric Link communication loss only (Control communication is still successful)

SRX-HE

Secondary node will transition to Ineligible then to Disabled state (10.4R3 and lower)

No action taken (10.4R4 and higher)

SRX-Branch

Secondary node will transition to an Ineligible state then to a Disabled state

Control and Fabric communication loss simultaneously

This scenario leads to a split-brain situation, where both devices are unable to detect each other and both believe a peer node has had a complete failure thus both take primary ownership of all RG groups.

Primary node will retain ownership of all RG groups

Secondary node will transition to Primary for all RG groups

When a node has transitioned into a disabled state, the disabled node must be rebooted to clear the disabled state. Before performing a reboot on a disabled node, it is strongly recommended to verify the cause of the disabled state. If the disabled state is caused by control link loss, and the control link has not recovered before the reboot, upon reboot the disabled node will take primary ownership of all redundancy-groups due to lack of communication to the peer. This leads to situation where both nodes are primary nodes, which is also called a “split-brain” scenario.

To automate the recovery for loss of a control link, ‘control-link-recovery’ may be used. When control-link-recovery is used, if a node transitioned into a ‘disabled’ state due to loss of control link, the disabled node will be automatically rebooted upon control link communication being restored to bring the cluster back to normal operating status.

```
#set chassis cluster control-link-recovery
```

Dual Control and Fabric Link

To overcome a single point of failure for the Control and Fabric link communication between the SRX chassis cluster nodes, dual control and fabric links may be used. When using dual fabric links the links are not aggregated but instead one link is used for flow forwarded packets (Z-mode traffic) and the other links is used for heart-beats and RTO objects. In the event that one link fails, the remaining link will handle all the traffic as if the system were configured with only one fabric link.

Dual Control Links (only supported for SRX3k and SRX5k models)

Hardware Requirements

SRX5600 & SRX5800: Secondary RE installed in SCB located in SCB slot 1

NOTE: The secondary RE purpose is only to initialize the switch on the SCB. No additional RE functionality is achieved with use of a secondary RE at time of this writing.

SRX3400 & SRX3600: Chassis Redundancy Module (CRM) installed in RE slot 1

NOTE: The CRM is not a full RE and its purpose is to initialize the second control link. No additional functionality is achieved with use of CRM.

Cabling Setup

SRX5k: Connect a fiber cable using 1Gb SFP transceiver to Control Port 1 of a SPC module (both control links may be on same SPC however it is recommended to have control links on different SPCs)

SRX3k: Connect Fiber using 1Gb SFP transceiver to Control Port 1 on right side of SFB

Configuration Setup

SRX5k: Define location of secondary control link within configuration

```
set chassis cluster control-ports fpc 4 port 1
set chassis cluster control-ports fpc 16 port 1
```

SRX3k: No configuration necessary

Verify behavior of dual control link connectivity

```
root > show chassis cluster statistics
```

Control link statistics:

Control link 0:

Heartbeat packets sent: **5252**
Heartbeat packets received: **5165**
Heartbeat packet errors: 0

Control link 1:

Heartbeat packets sent: **43**
Heartbeat packets received: **42**
Heartbeat packet errors: 0

Dual Fabric Links (supported on all SRX models)

Hardware Requirements

One unused interface port per cluster node

Cabling Setup

Install fiber or ethernet cable between one unused interface port on each node

Configuration Setup

Define location of second fabric link connection.

```
set interfaces fab0 fabric-options member-interfaces ge-0/0/2  ## existing 1st
fabric link.
set interfaces fab1 fabric-options member-interfaces ge-5/0/2  ## existing 1st
fabric link.
set interfaces fab0 fabric-options member-interfaces ge-0/0/35
set interfaces fab1 fabric-options member-interfaces ge-5/0/3
```

Verify behavior of dual fabric link connectivity

```
root> show chassis cluster interfaces
```

Fabric link status: Up

Fabric interfaces:

Name	Child-interface	Status
fab0	ge-0/0/2	Up
fab0	ge-0/0/3	Up
fab1	ge-5/0/2	Up
fab1	ge-5/0/3	Up

⁵ For optimal redundancy, it is recommended to configure each fabric link on a different interface card, where possible.

Conclusion

Chassis cluster is a concept that can be used in Juniper SRX series devices in various topologies to provide redundancy. After reading and by using this document, we hope to have given you the tools and understanding to setup an SRX cluster easily. Also by understanding the concepts and workings, you will be able to control the cluster failover behavior and monitor the cluster status.