ECMP Convergence Acceleration in sonic for i/f down events

Table of Contents

[1 Document History](#_Toc499739852)

[2 Abbreviations](#_Toc499739853)

[3 References](#_Toc499739854)

[4 Problem Overview](#_Toc499739855)

[5 Proposed Solutions](#_Toc499739856)

[5.1 BGP next hop tracking](#_Toc499739857)

[5.2 Removing the ecmp path](#_Toc499739858)

[6 Dependencies](#_Toc499739859)

[7 CLI](#_Toc499739860)

[8 Unit Test Plan](#_Toc499739861)

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| A | 06/06/2017 | Nikos | Initial version |
| B | 09/17/2017 | Nikos | Update design, SAI APIs and Unit Test Plan |
|  |  |  |  |

# Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| BGP | Border Gateway Protocol |
| ECMP | Equal Cost Multi Path |
| IGP | Interior Gateway Protocol |
| RIB | Routing Information Base |

# References

# Problem Overview

It is standard practice in today’s networks to distribute traffic across multiple paths. This is achieved either statically or dynamically with protocols such as BGP, or IGPs. By default, BGP will choose only one path during its bestpath selection but in today’s data centers, network nodes are connected to multiple other nodes with the intend to provide increased bandwidth as well as multiple equal cost paths for redundancy purposes and load balancing. As such, BGP is configured to take advantage of the multiple available paths and selects them, if available, for a prefix during bestpath calculation for installation into the RIB and forwarding.

When one of those paths becomes unavailable due to an interface going down or administratively shut, BGP will either detect that or be notified of changes in its next hops. It will then determine which paths are affected by this change, usually via a walk of the prefix table, and for each prefix it will rerun its bestpath selection algorithm and modify the ecmp paths calculated for that prefix accordingly. It will then proceed downloading all the affected prefixes into RIB which in turn it will download into sonic.

This process is not instantaneous and until BGP converges from such an event and the hardware is updated, forwarding is still using the unavailable path which results in black holing traffic partially for any prefix that makes use of that path.

It is therefore desirable to reduce significantly the amount of time traffic is being dropped over the unavailable path and accelerate at a lower level the ecmp groups convergence by removing from forwarding any paths being affected due to the interface state change.

# Proposed Solutions

Before outlining the two proposed solutions, it is important to understand the different ways BGP may track next hop changes and how it reacts to them. This will provide more insight as to what advantages and disadvantages each solution has and decide what is best suited for the current sonic architecture and data center deployments as well as what may be desirable in the future.

## BGP next hop tracking

Different BGP implementations, may have different ways of tracking and validating BGP next hops.

Older BGP implementations are using a polling/periodic mechanism. In those cases, BGP uses a timer and upon timer expiration, the RIB is polled and next hops are validated. In between walks, next hop changes due to an interface flap that doesn’t affect a BGP session, are not detected. Similarly, between scan cycles, network failures are not detected and routing loops or traffic loss may occur.

Newer BGP implementations, are using an event driven mechanism for next hop address tracking. Usually this involves BGP registering with the RIB the prefixes it needs RIB to track. Next hop changes are rapidly reported to BGP for the registered prefixes as they are updated in the RIB. This event driven optimization improves overall BGP convergence by reducing the response time to next hop changes for routes installed in the RIB.

## Removing the ecmp path

The simplest solution to solve the problem from implementation’s perspective, is to perform the following steps **inline** in the orachagent when the i/f down state change is detected:

* Obtain the alias of the interface.
* Walk all nexthops to see which ones go over the affected interface.
* Mark any affected nexthops.
* For each affected nexthop go through the ecmp groups and see if there is an affected member.
* If there is, call the remove\_next\_hop\_group\_member SAI API to remove the path from the ecmp group in the SAI db.

The steps are similar on i/f state up:

* Obtain the alias of the interface.
* Walk all nexthops to see which ones go over the affected interface.
* Clear flag of any affected nexthops.
* For each affected nexthop go through the ecmp groups and see if there is an affected member.
* If there is, call the create\_next\_hop\_group\_member SAI API to remove the path from the ecmp group in the SAI db.

This approach is preferable than the one removing the ecmp member from the group at the orchagent level.

# Dependencies

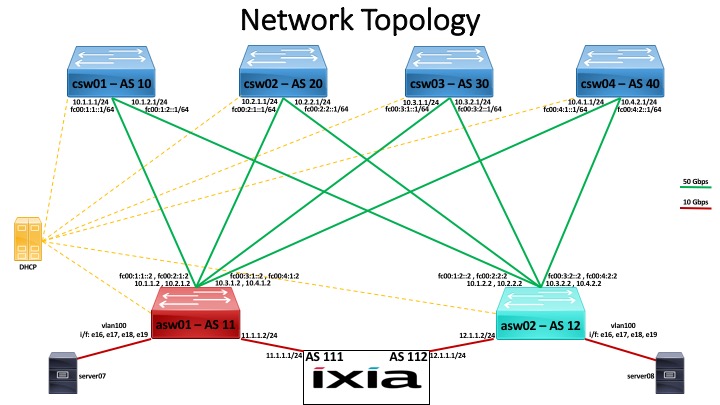
There are only software dependencies in delivering either solution. They both depend on the implementation of the create/remove\_next\_hop\_group\_member SAI APIs.

# CLI

It is desirable to have this behavior by default since it improves convergence and not controlled via a CLI option. This is also in line with existing vendor implementations.

# Unit Test Plan

The following topology has been used to unit test and validate the new functionality:



Prefixes are advertised from IXIA towards asw02 and traffic is generated to those prefixes from IXIA towards asw01 (DUT).

Test #1: Verify upon BGP prefix reception, that all prefixes are received as expected and ecmp group and membership is correct. Result: PASS

Test #2: Bring down the link on csw01 connecting csw01 and asw01. Verify that traffic loss is 1sec or less (result was 90ms or less) . Verify that ecmp group membership changes accordingly in saidb and in hw. Verify BGP converges prefixes to new ecmp group and original one is removed. Result: PASS

Test #3: Bring link up on csw01 connecting csw01 and asw01. Verify there is no traffic loss. Verify ecmp group membership doesn’t change. Verify BGP converges prefixes to new ecmp group and original one is removed. Result: PASS

Test #4: Bring all csw links down connecting to asw01 one by one. Verify ecmp group membership is as expected and eventually the ecmp group has no members and SAI API doesn’t generate any exceptions. Result: PASS

Test #5: Bring down the link on csw01 connecting csw01 and asw01. Verify ecmp group membership changes to reflect the removal of one member. Bring link back up. Verify that ecmp group membership changes again and the marked member is added back. Result: PASS