MCLAG (L2 Scenario) Guide for SONiC in GNS3

Revision History

Revision No.	Description	Editor	Date
1.0	MCLAG (L2 Scenario) Guide for SONiC in GNS3	Humza Altaf	Jul 4, 2023

Table of Contents

Introduction 4
MCLAG 4
Benefits of MCLAG 4
Terminologies 5
Testbed 6
Network Topology 6
Configurations 7
Starting ICCPd 7
Step-1 7
PortChannel Configurations 8
Step-2 8
Step-3 9
MCLAG Configurations 10
Step-4 10
Step-5 10
MCLAG Status 11
Step-6 11
Step-7 11
Result 12
References 13

Introduction

This document provides a comprehensive guide to configuring MCLAG using GNS3, a powerful network simulation tool. With the help of GNS3, users can create virtual instances of SONiC, allowing for thorough testing and evaluation of its various features. Through practical demonstrations and detailed instructions, this document aims to equip readers with the knowledge and insights needed to configure MCLAG in their network environments successfully.

The testing guide focuses on deploying the "MCLAG" on a given network topology, and conducting tests by running essential commands in the SONiC CLI. The step-by-step procedure outlined in this guide ensures the proper verification of MCLAG feature.

MCLAG

The increasing scale of Layer 2 networks, driven by technologies like virtualization, necessitates protocols and controls to mitigate the negative effects of network topology loops. The Spanning Tree Protocol (STP) has been the primary solution, providing a loop-free environment, but it only allows one active path between devices, limiting network capacity and causing topology changes. The Rapid Spanning Tree Protocol (RSTP) improves convergence time but still suffers from significant delays

Multi-chassis link aggregation groups (MC-LAGs) enable a client device to form a logical LAG interface between two MC-LAG peers. An MC-LAG provides redundancy and load balancing between the two MC-LAG peers, multihoming support, and a loop-free Layer 2 network without running STP. On one end of an MC-LAG, there is an MC-LAG client device, such as a server, that has one or more physical links in a link aggregation group (LAG). This client device uses the links as a LAG. On the other side of the MC-LAG, there can be a maximum of two MC-LAG peers. Each of the MC-LAG peers has one or more physical links connected to a single client device. The MC-LAG peers use the Inter-Chassis Control Protocol (ICCP) to exchange control information and coordinate with each other to ensure that data traffic is forwarded properly.

Benefits of MCLAG

- Reduces operational expenses by providing active-active links within a Link Aggregation Group (LAG).
- Provides faster layer 2 convergence upon link and device failures.
- Adds node-level redundancy to the normal link-level redundancy that a LAG provides.
- Improves network resiliency, which reduces network downtime as well as expenses.

Terminologies

MC-LAG peer	MC-LAG switch, one of a pair.
MC-LAG member port	One of a set of ports (port channels) that form an MC-LAG.
MC-LAG	Combined port channel between the MC-LAG peers and the downstream device.
MC-LAG peer link	It is the connection as the data backup path between the two peers. The connection can be a physical port, a PortChannel, or a VXLAN tunnel. This peer link is used to carry data traffic when an MC-LAG member port is down
MC-LAG peer keepalive link	It is a Layer 3 link that joins one peer device to the other peer device. The peer-keepalive link carries periodic heartbeat between peer devices, and it is used to synchronize the state between MC-LAG peer devices. It is strongly recommended to configure redundant keepalive links.
Orphan port	Non-MC-LAG member port.

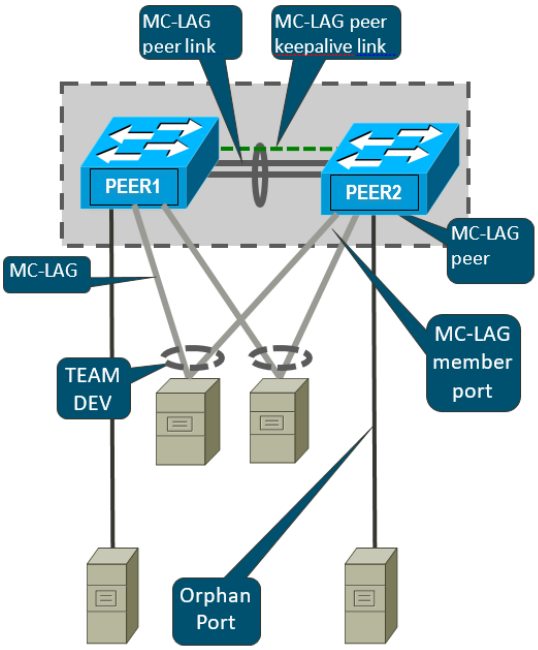


Figure: Terminologies for MCLAG

Testbed

To set up the testbed for MCLAG configuration please refer to the document [Installation of GNS3 and vTestbed setup for SONiC](#).

Network Topology

The GNS3 network topology consists of four switches: MCLAG-1, MCLAG-2, Access, and Core with three portchannels "PortChannel0001," "PortChannel0002," and "PortChannel0003". PortChannel0001 connects MCLAG-1 and MCLAG-2, while PortChannel0002 links MCLAG-1 with Access. Likewise, PortChannel0003 establishes a reliable connection between MCLAG-1 and Core. All portchannels carry tagged Vlan 10 traffic, while PC1 and PC2 are assigned untagged Vlan10.

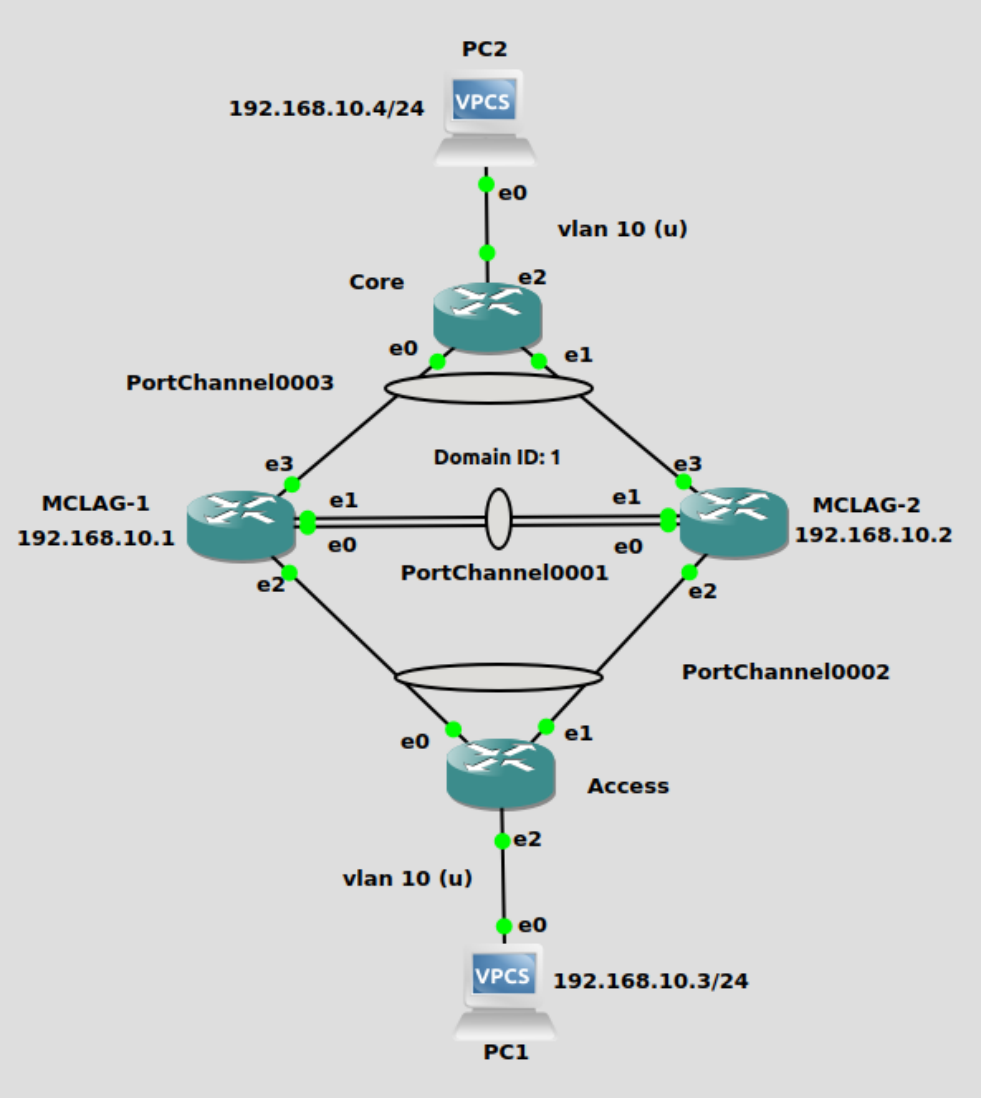


Fig: Network Topology

Configurations

First, the switch “MCLAG-1” is configured, and the same steps are repeated for the “MCLAG-2”. A command reference guide is also available on GitHub for SONiC.

Follow these steps to configure “MCLAG-1” switch.

Starting ICCPd

Step-1

In the community SONiC, there exists an ICCPd Docker container that is not initiated as part of the default startup process. This behavior can be confirmed by executing the specified command provided below:

- **docker ps -a**

```
mdanish@sonic:~$ docker ps -a
CONTAINER ID   IMAGE                              COMMAND                  CREATED          STATUS          PORTS     NAMES
71fb52be91cc   docker-gbsyncd-vs:latest           "/usr/local/bin/supe…"   16 seconds ago   Up 14 seconds             gbsyncd
c8b30ebb3bf9   docker-fpm-frr:latest              "/usr/bin/docker_ini…"   17 seconds ago   Up 15 seconds             bgp
ef877266c35d   docker-router-advertiser:latest    "/usr/bin/docker-ini…"   24 seconds ago   Up 23 seconds             radv
66be0b0ab550   docker-syncd-vs:latest             "/usr/local/bin/supe…"   27 seconds ago   Up 25 seconds             syncd
f746b3cc4fd8   docker-teamd:latest                "/usr/local/bin/supe…"   27 seconds ago   Up 25 seconds             teamd
6abd18d766d4   docker-orchagent:latest            "/usr/bin/docker-ini…"   31 seconds ago   Up 29 seconds             swss
c913ea3d4697   docker-sonic-restapi:latest        "/usr/local/bin/supe…"   32 seconds ago   Up 29 seconds             restapi
08efd772f135   docker-eventd:latest               "/usr/local/bin/supe…"   32 seconds ago   Up 30 seconds             eventd
6374b07ced65   docker-database:latest             "/usr/local/bin/dock…"   44 seconds ago   Up 43 seconds             database
```

The specific service "iccpd.service" refers to a service or daemon running on a Linux-based system. The acronym "iccpd" stands for "Inter-Chassis Communication Protocol Daemon." The iccpd.service is responsible for managing and facilitating the ICCP functionality on the system. It handles the communication and synchronization between the different chassis or devices participating in the ICCP network.

In the default configuration of the community SONiC, the iccpd.service is automatically masked.

```
mdanish@sonic:~$ sudo systemctl start iccpd.service
Failed to start iccpd.service: Unit iccpd.service is masked.
```

The error message "**Failed to start iccpd.service: Unit iccpd.service is masked"** indicates that the iccpd.service unit is currently masked on a system. When a service unit is masked, it means that the system is prevented from starting or stopping the service.

The above service can be unmasked by using the following command given below:

- **sudo systemctl unmask iccpd**

```
mdanish@sonic:~$ sudo systemctl unmask iccpd
Removed /etc/systemd/system/iccpd.service.
```

ICCPd docker container doesn't start by default, it can be started on demand. To start the ICCPd docker container, the command is given below:

- **sudo systemctl start iccpd**

```
sudo systemctl unmask iccpd.service
docker ps -a
sudo systemctl start iccpd
```

Once the “iccpd.service” has been unmasked, the Docker container associated with it will be initiated, and its status can be observed.

```
mdanish@sonic:~$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS     NAMES
9ecc4c16baef   docker-iccpd:latest                  "/usr/local/bin/supe…"   34 seconds ago   Up 33 seconds             iccpd
a3d324998833   docker-sonic-telemetry:latest        "/usr/local/bin/supe…"   11 minutes ago   Up 10 minutes             telemetry
ba2ff9a1462b   docker-snmp:latest                   "/usr/local/bin/supe…"   11 minutes ago   Up 11 minutes             snmp
ee8adc1d129a   docker-platform-monitor:latest       "/usr/bin/docker_ini…"   11 minutes ago   Up 11 minutes             pmon
51cc1917e1e7   docker-sonic-mgmt-framework:latest   "/usr/local/bin/supe…"   11 minutes ago   Up 11 minutes             mgmt-framework
9f9cae539db3   docker-lldp:latest                   "/usr/bin/docker-lld…"   11 minutes ago   Up 11 minutes             lldp
a567d7dad93e   docker-gbsyncd-vs:latest             "/usr/local/bin/supe…"   12 minutes ago   Up 12 minutes             gbsyncd
d43d26bb57f0   docker-fpm-frr:latest                "/usr/bin/docker_ini…"   12 minutes ago   Up 12 minutes             bgp
```

Note: Whenever the switch restarts, the iccpd Docker container will stop, and it needs to be manually restarted afterward.

PortChannel Configurations

Step-2

To establish connectivity between the "MCLAG-1" and "MCLAG-2" switches, it is necessary to create three portchannels named "PortChannel0001," "PortChannel0002," and "PortChannel0003." This can be accomplished by executing the provided command as follows:

- **sudo config portchannel (add | del) <portchannel_name> [--min-links <num_min_links>] [--fallback (true | false) [--fast-rate (true | false)]**

```
mdanish@sonic:~$ sudo config portchannel add PortChannel0001
mdanish@sonic:~$ sudo config portchannel add PortChannel0002
mdanish@sonic:~$ sudo config portchannel add PortChannel0003
```

The table below demonstrates the mapping of ports with PortChannels.

PortChannel0001	Ethernet0, Ethernet4
PortChannel0002	Ethernet8
PortChannel0003	Ethernet12

This can be accomplished by executing the provided command as follows:

- **config portchannel member (add | del) <portchannel_name><member_portname>**

```
mdanish@sonic:~$ sudo config portchannel member add PortChannel0001 Ethernet0
mdanish@sonic:~$ sudo config portchannel member add PortChannel0001 Ethernet4
mdanish@sonic:~$ sudo config portchannel member add PortChannel0002 Ethernet8
mdanish@sonic:~$ sudo config portchannel member add PortChannel0003 Ethernet12
```

To check the status of portchannels, use the following command given below:

- **show interfaces portchannel**

```
mdanish@sonic:~$ show interfaces portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
       S - selected, D - deselected, * - not synced
  No.  Team Dev         Protocol     Ports
-----  -------------  -----------  -----------------------
 0001  PortChannel0001  LACP(A)(Up)  Ethernet4(S) Ethernet0(S)
 0002  PortChannel0002  LACP(A)(Up)  Ethernet8(S)
 0003  PortChannel0003  LACP(A)(Up)  Ethernet12(S)
```

In the above screenshot, the portchannels on both switches, "MCLAG-1" and "MCLAG-2," exhibit identical statuses.

Below is the displayed status of the portchannels assigned to the Access and Core switches respectively.

```
mdanish@sonic:~$ show interfaces portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
       S - selected, D - deselected, * - not synced
  No.  Team Dev         Protocol     Ports
-----  -------------  -----------  -----------------------
 0002  PortChannel0002  LACP(A)(Up)  Ethernet4(S) Ethernet0(S)
```

```
mdanish@sonic:~$ show interfaces portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
       S - selected, D - deselected, * - not synced
  No.  Team Dev         Protocol     Ports
-----  -------------  -----------  -----------------------
 0003  PortChannel0003  LACP(A)(Up)  Ethernet0(S) Ethernet4(S)
```

Step-3

Once the portchannels have been established, proceed with the creation of VLAN 10 and associate it as a tagged VLAN member across all portchannels, by executing the following set of commands provided below:

- **sudo config vlan (add | del) <vlan_id>**
- **sudo config vlan member add/del [-u|--untagged] <vlan_id> <member_portname>**

```
mdanish@sonic:~$ sudo config vlan add 10
mdanish@sonic:~$ sudo config vlan member add 10 PortChannel0001
mdanish@sonic:~$ sudo config vlan member add 10 PortChannel0002
mdanish@sonic:~$ sudo config vlan member add 10 PortChannel0003
```

MCLAG Configurations

Step-4

To configure MCLAG on both switches use the following commands given below:

- **sudo config mclag {add | del} \<domain-id> \<local-ip-addr> \<peer-ip-addr> \ [\<peer-ifname>]**

domain-id	MCLAG node's unique domain-id
local_ip	MCLAG node's local ipv4 address
peer_ip	MCLAG node's peer ipv4 address
peer_ifname	MCLAG peer interface name; optional for L3 MCLAG, mandatory for L2 MCLAG config

- **sudo config mclag unique-ip {add | del} <Vlan-interface's>**
- **sudo config mclag member {add | del} \<domain-id> <portchannel-names>**

```
mdanish@sonic:~$ sudo config mclag add 1 192.168.10.1 192.168.10.2 PortChannel0001
mdanish@sonic:~$ sudo config mclag unique-ip add Vlan10
mdanish@sonic:~$ sudo config mclag member add 1 PortChannel0002
mdanish@sonic:~$ sudo config mclag member add 1 PortChannel0003
```

Step-5

Add the IP address on Vlan10 by using the following command given below:

- **sudo config interface ip add Vlan10 192.168.10.1/24**

To check the status of the vlan interface, use the following command given below:

- **show vlan brief**

```
mdanish@sonic:~$ show vlan brief
+-----------+----------------+----------------+---------------+-------------+
|   VLAN ID | IP Address     | Ports          | Port Tagging  | Proxy ARP   |
+===========+================+================+===============+=============+
|        10 | 192.168.10.1/24 | PortChannel0001 | tagged        | disabled    |
|           |                | PortChannel0002 | tagged        |             |
|           |                | PortChannel0003 | tagged        |             |
+-----------+----------------+----------------+---------------+-------------+
```

MCLAG Status

Step-6

Check MCLAG status using the command:

- **mclagdctl -i <mclag-id> dump state**

This command is used to retrieve and display the current state of the specified MCLAG instance on MCLAG-2 switch.

```
mdanish@sonic:~$ mclagdctl -i 1 dump state
The MCLAG's keepalive is: OK
MCLAG info sync is: completed
Domain id: 1
Local Ip: 192.168.10.2
Peer Ip: 192.168.10.1
Peer Link Interface: PortChannel0001
Keepalive time: 1
sesssion Timeout : 15
Peer Link Mac: 0c:4b:a4:74:00:00
Role: Standby
MCLAG Interface: PortChannel0003,PortChannel0002
Loglevel: NOTICE
```

Step-7

Assign IP addresses to PC1 and PC2 hosts.

```
PC1> ip 192.168.10.3/24 255.255.255.0
Checking for duplicate address...
PC1 : 192.168.10.3 255.255.255.0

PC1> sh ip

NAME        : PC1[1]
IP/MASK     : 192.168.10.3/24
GATEWAY     : 255.255.255.0
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10020
RHOST:PORT  : 127.0.0.1:10021
MTU         : 1500
```

Result

In the MCLAG setup, two hosts (PC1 & PC2) are engaged in a ping operation shown below. The traffic from PC1 to PC2 and vice versa is being forwarded by the MCLAG-1 as it is in Active mode as depicted in Step-6:

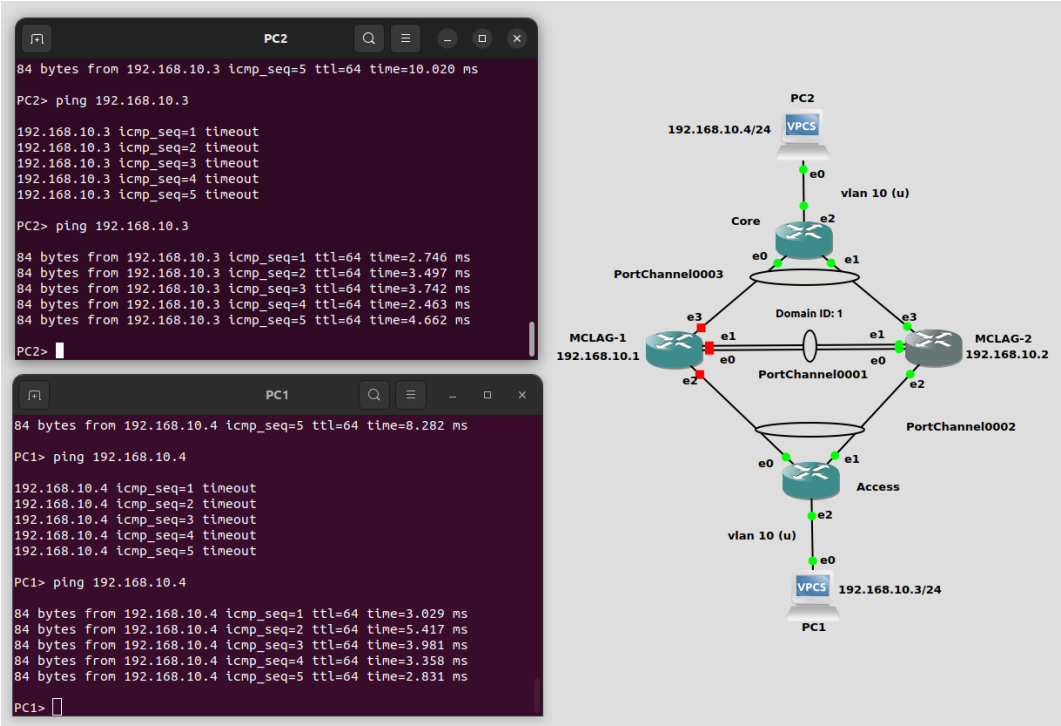
Ping from PC1 to PC2:

```
PC1> ping 192.168.10.4

84 bytes from 192.168.10.4 icmp_seq=1 ttl=64 time=3.053 ms
84 bytes from 192.168.10.4 icmp_seq=2 ttl=64 time=3.212 ms
84 bytes from 192.168.10.4 icmp_seq=3 ttl=64 time=3.479 ms
84 bytes from 192.168.10.4 icmp_seq=4 ttl=64 time=3.446 ms
84 bytes from 192.168.10.4 icmp_seq=5 ttl=64 time=3.416 ms
```

Ping from PC2 to PC1:

```
PC2> ping 192.168.10.3

84 bytes from 192.168.10.3 icmp_seq=1 ttl=64 time=3.168 ms
84 bytes from 192.168.10.3 icmp_seq=2 ttl=64 time=3.236 ms
84 bytes from 192.168.10.3 icmp_seq=3 ttl=64 time=3.513 ms
84 bytes from 192.168.10.3 icmp_seq=4 ttl=64 time=2.211 ms
84 bytes from 192.168.10.3 icmp_seq=5 ttl=64 time=3.274 ms
```

In the following screenshot, when MCLAG-1 encounters a disruption or failure, the network traffic is disrupted for a moment and then resumes. At this moment the traffic is passing through the switch MLAG-2.



References

- *FRRUserManual.* (2022, September Tuesday). Retrieved from https://docs.frrouting.org/_/downloads/en/latest/pdf/
- https://github.com/sonic-net/SONiC/blob/master/doc/mclag/Sonic-mclag-hld.md
- https://github.com/sonic-net/SONiC/blob/master/doc/mclag/MCLAG_Enhancements_HLD.md
- https://www.juniper.net/documentation/us/en/software/junos/mc-lag/topics/concept/mc-lag-feature-summary-best-practices.html