SOLUTION BRIEF

By: Humza Altaf, SONiC Network Engineer

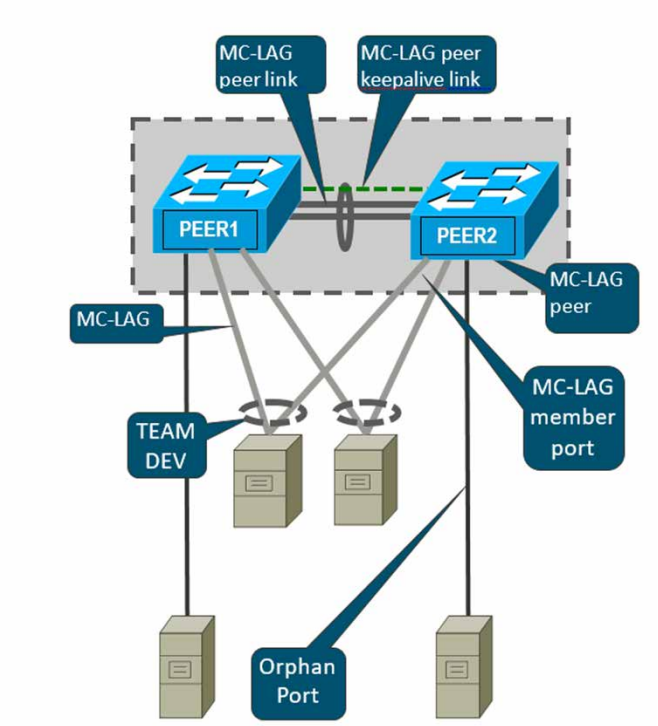
Revision No.	Description	Editor	Date
1.0	Guide for MCLAG in SONiC	Humza Altaf	Oct 27, 2023

Table of Contents

Introduction to MCLAG

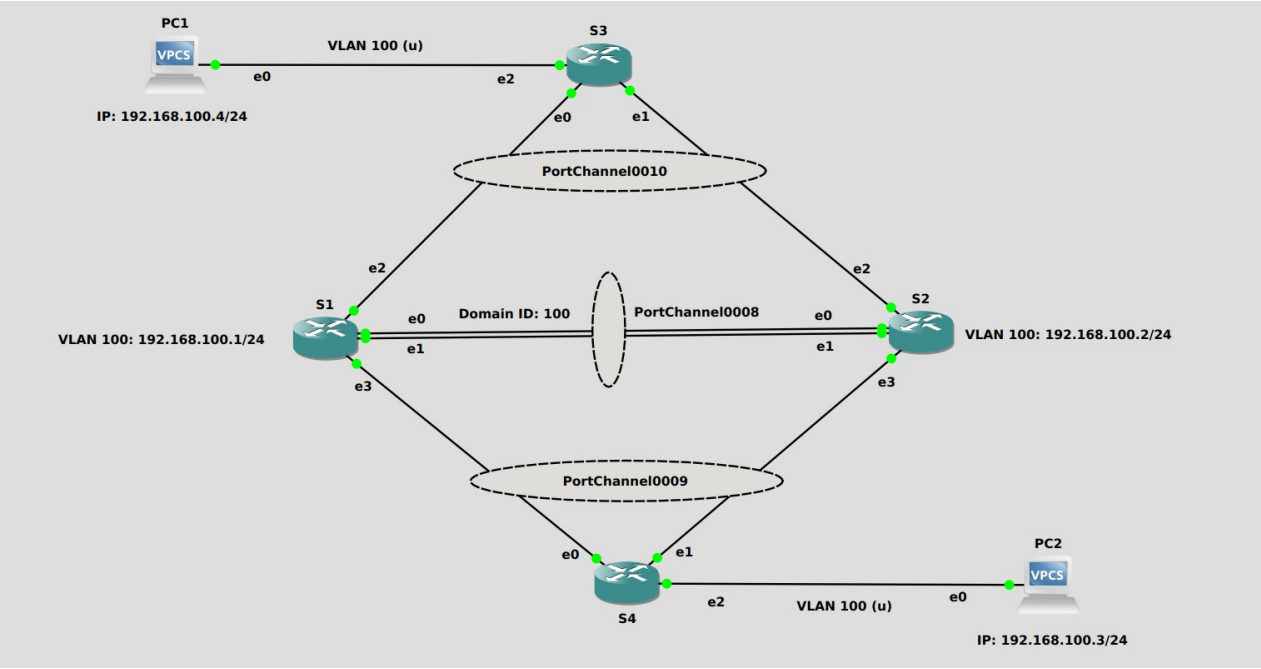
PortChannel, also known as Link Aggregation or EtherChannel, is a network technology used to aggregate multiple physical network links into a single logical link. This approach enhances network performance, redundancy, and fault tolerance by distributing traffic across these combined links. PortChannel allows for the simultaneous use of multiple connections between network devices, such as switches and routers, providing increased bandwidth and network resilience.

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, load balancing between the two MC-LAG peers 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.



Network Topology

The GNS3 network topology consists of four switches: S1, S2, S3, and S4 with three portchannels "PortChannel0008," "PortChannel0009," and "PortChannel0010". PortChannel0008 is between S1 and S2, while PortChannel0010 links S1, S3 and S2, S3. Likewise, PortChannel0009 establishes a reliable connection between S1, S4 and S2, S4. All portchannels carry tagged VLAN 100 traffic, while PC1 and PC2 are assigned untagged VLAN 100.



Port Mapping

GNS3	SONiC
Ethernet 0	Ethernet 0
Ethernet 1	Ethernet 4
Ethernet 2	Ethernet 8
Ethernet 3	Ethernet 12

Configurations

For the above topology, all hosts and switches are first configured before sending traffic. First, switch S1 is configured and the same steps are repeated for the switch S2. Command Reference guide is also available on GitHub for SONiC, whose link is given here.

Follow these steps to configure S1.

Step 1

In the community SONiC, an ICCPd Docker container is not initiated as part of the default startup process. This behaviour 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
b7be60c883c1   docker-gbsyncd-vs:latest           "/usr/local/bin/supe…"   10 seconds ago   Up 7 seconds              gbsyncd
4aa054965be3   docker-fpm-frr:latest              "/usr/bin/docker_ini…"   11 seconds ago   Up 8 seconds              bgp
77010f3a92d0   docker-router-advertiser:latest    "/usr/bin/docker-ini…"   16 seconds ago   Up 13 seconds             radv
d84bbac26c89   docker-syncd-vs:latest             "/usr/local/bin/supe…"   21 seconds ago   Up 17 seconds             syncd
25b452eb4669   docker-teamd:latest                "/usr/local/bin/supe…"   21 seconds ago   Up 17 seconds             teamd
ada42802d4e8   docker-orchagent:latest            "/usr/bin/docker-ini…"   28 seconds ago   Up 24 seconds             swss
26cdf3877d9e   docker-sonic-restapi:latest        "/usr/local/bin/supe…"   29 seconds ago   Up 25 seconds             restapi
1109f1d019cf   docker-eventd:latest               "/usr/local/bin/supe…"   29 seconds ago   Up 24 seconds             eventd
5ccc1f007bc6   docker-database:latest             "/usr/local/bin/dock…"   41 seconds ago   Up 40 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
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.

Step 1 (Continued)

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`

```
mdanish@sonic:~$ sudo systemctl start iccpd
mdanish@sonic:~$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED             STATUS                       PORTS     NAMES
d647e2e077bb   docker-iccpd:latest                  "/usr/local/bin/supe…"   26 seconds ago      Up 15 seconds                          iccpd
a3bbeda20773   docker-sonic-telemetry:latest        "/usr/local/bin/supe…"   35 seconds ago      Exited (0) 16 seconds ago              telemetry
736d7d166867   docker-snmp:latest                   "/usr/local/bin/supe…"   46 seconds ago      Up 43 seconds                          snmp
ae0ee4b6ee4f   docker-platform-monitor:latest       "/usr/bin/docker_ini…"   56 seconds ago      Up 53 seconds                          pmon
819d34c1c266   docker-sonic-mgmt-framework:latest   "/usr/local/bin/supe…"   About a minute ago  Up About a minute                      mgmt-framework
4d7a192b413e   docker-lldp:latest                   "/usr/bin/docker-lld…"   About a minute ago  Up About a minute                      lldp
b7be60c883c1   docker-gbsyncd-vs:latest             "/usr/local/bin/supe…"   2 minutes ago       Up 2 minutes                           gbsyncd
4aa054965be3   docker-fpm-frr:latest                "/usr/bin/docker_ini…"   2 minutes ago       Up 2 minutes                           bgp
77010f3a92d0   docker-router-advertiser:latest      "/usr/bin/docker-ini…"   2 minutes ago       Up 2 minutes                           radv
d84bbac26c89   docker-syncd-vs:latest               "/usr/local/bin/supe…"   2 minutes ago       Up 2 minutes                           syncd
25b452eb4669   docker-teamd:latest                  "/usr/local/bin/supe…"   2 minutes ago       Up 2 minutes                           teamd
ada42802d4e8   docker-orchagent:latest              "/usr/bin/docker-ini…"   2 minutes ago       Up 2 minutes                           swss
26cdf3877d9e   docker-sonic-restapi:latest          "/usr/local/bin/supe…"   2 minutes ago       Up 2 minutes                           restapi
1109f1d019cf   docker-eventd:latest                 "/usr/local/bin/supe…"   2 minutes ago       Up 2 minutes                           eventd
5ccc1f007bc6   docker-database:latest               "/usr/local/bin/dock…"   2 minutes ago       Up 2 minutes                           database
```

In the above figure, it can be seen that iccpd container is up and running.

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

Step 2

By default, all interfaces are routed (L3) and IP is assigned to them. To check the status of IP addresses, use the following command given below:

- `show ip interfaces`

```
admin@sonic:~$ show ip interfaces
Interface    Master    IPv4 address/mask    Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -------------------  ------------  --------------  -------------
Ethernet0              10.0.0.0/31          up/up         ARISTA01T2      10.0.0.1
Ethernet4              10.0.0.2/31          up/up         ARISTA02T2      10.0.0.3
```

Step 2 (Continued)

Remove the IP addresses to make that interface a switch port (L2). For this, the command is given below:

- `sudo config interface ip remove/add <interface_name> <ip_addr>`

```
admin@sonic:~$ sudo config interface ip remove Ethernet0 10.0.0.0/31
admin@sonic:~$ sudo config interface ip remove Ethernet4 10.0.0.2/31
admin@sonic:~$ sudo config interface ip remove Ethernet8 10.0.0.4/31
```

Note: It is better practice to save configurations after executing two or three commands by using “sudo config save -y” command.

Step 3

Now create Portchannels between switches. Before creating Portchannels, check the status by using the following command given below:

- `show interfaces portchannel`

```
admin@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
-----  ----------  ----------  -------
```

In the above table, no Portchannel is created.

Step 3 (Continued)

To establish connectivity between the "S1" and "S2" switches, it is necessary to create three portchannels named "PortChannel0008," "PortChannel0009," and "PortChannel0010." 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 PortChannel0008
mdanish@sonic:~$ sudo config portchannel add PortChannel0009
mdanish@sonic:~$ sudo config portchannel add PortChannel0010
```

The table below demonstrates the mapping of ports with PortChannels.

PortChannel0008	Ethernet0, Ethernet4
PortChannel0010	Ethernet8
PortChannel0009	Ethernet12

Step 4

Now make ports be a member of the portchannels by using the following command given below:

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

```
mdanish@sonic:~$ sudo config portchannel member add PortChannel0008 Ethernet0
mdanish@sonic:~$ sudo config portchannel member add PortChannel0008 Ethernet4
mdanish@sonic:~$ sudo config portchannel member add PortChannel0010 Ethernet8
mdanish@sonic:~$ sudo config portchannel member add PortChannel0009 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
-----  -------------  -----------  -------------------------
 0008  PortChannel0008  LACP(A)(Up)  Ethernet4(S) Ethernet0(S)
 0009  PortChannel0009  LACP(A)(Up)  Ethernet12(S)
 0010  PortChannel0010  LACP(A)(Up)  Ethernet8(S)
```

Note: In the above figure, the status of the ports is (S) "selected." This status will be displayed when a Portchannel is configured on all the switches, and the ports are members of it.

Step 5

Now create VLAN for topology. Before creating VLAN, check the VLAN table by using the following command given below:

- `show vlan brief`

```
admin@sonic:~$ show vlan brief
+-----------+--------------+---------+----------------+-------------+
| VLAN ID   | IP Address   | Ports   | Port Tagging   | Proxy ARP   |
+===========+==============+=========+================+=============+
+-----------+--------------+---------+----------------+-------------+
```

In the above table, no VLAN is created. Now create VLAN 100 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>`

```
mdanish@sonic:~$ show vlan brief
+-----------+-----------------+-----------------+----------------+-------------+
|   VLAN ID | IP Address      | Ports           | Port Tagging   | Proxy ARP   |
+===========+=================+=================+================+=============+
|       100 | 192.168.100.1/24 | PortChannel0008 | tagged         | disabled    |
|           |                 | PortChannel0009 | tagged         |             |
|           |                 | PortChannel0010 | tagged         |             |
+-----------+-----------------+-----------------+----------------+-------------+
```

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

```
mdanish@sonic:~$ sudo config vlan member add 100 PortChannel0008
mdanish@sonic:~$ sudo config vlan member add 100 PortChannel0009
mdanish@sonic:~$ sudo config vlan member add 100 PortChannel0010
```

Step 6

To configure MCLAG on “S1”, use the following commands given below:

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

- `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 100 192.168.100.1 192.168.100.2 PortChannel0008
mdanish@sonic:~$ sudo config mclag unique-ip add Vlan100
mdanish@sonic:~$ sudo config mclag member add 100 PortChannel0009
mdanish@sonic:~$ sudo config mclag member add 100 PortChannel0010
```

Assign the IP address on VLAN 100 by using the following command given below:

- `sudo config interface ip add Vlan100 192.168.100.1/24`

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   |
+===========+=================+=================+================+=============+
|       100 | 192.168.100.1/24 | PortChannel0008 | tagged         | disabled    |
|           |                 | PortChannel0009 | tagged         |             |
|           |                 | PortChannel0010 | tagged         |             |
+-----------+-----------------+-----------------+----------------+-------------+
```

Step 7

Repeat steps 1-6 for the switch S2.

Step 8

After configuring S1 and S2, create portchannels on S3 and S4. Below is the displayed status of portchannels on S3 and S4 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
-----  -------------  -----------  -------------------------
 0010  PortChannel0010  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
-----  -------------  -----------  -------------------------
 0009  PortChannel0009  LACP(A)(Up)  Ethernet4(S) Ethernet0(S)
```

After creating Portchannels, create VLAN 100 on S3, S4 and make “Ethernet 8” untagged and portchannels “PortChannel0010, PortChannel0009” as tagged. Below is the displayed status of VLAN on S3 and S4 respectively.

```
mdanish@sonic:~$ show vlan brief
+-----------+--------------+-----------------+----------------+-------------+
|   VLAN ID | IP Address   | Ports           | Port Tagging   | Proxy ARP   |
+===========+==============+=================+================+=============+
|       100 |              | Ethernet8       | untagged       | disabled    |
|           |              | PortChannel0010 | tagged         |             |
+-----------+--------------+-----------------+----------------+-------------+
```

```
mdanish@sonic:~$ show vlan brief
+-----------+--------------+-----------------+----------------+-------------+
|   VLAN ID | IP Address   | Ports           | Port Tagging   | Proxy ARP   |
+===========+==============+=================+================+=============+
|       100 |              | Ethernet8       | untagged       | disabled    |
|           |              | PortChannel0009 | tagged         |             |
+-----------+--------------+-----------------+----------------+-------------+
```

Step 9

Check the MCLAG status on “S2” by using the following command given below:

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

This command retrieves and displays the current state of the specified MCLAG instance on the S2 switch.

```
mdanish@sonic:~$ mclagdctl -i 100 dump state
The MCLAG's keepalive is: OK
MCLAG info sync is: completed
Domain id: 100
Local Ip: 192.168.100.2
Peer Ip: 192.168.100.1
Peer Link Interface: PortChannel0008
Keepalive time: 1
sesssion Timeout : 15
Peer Link Mac: 0c:27:3a:e4:00:00
Role: Standby
MCLAG Interface: PortChannel0009,PortChannel0010
Loglevel: NOTICE
```

Step 10

Assign IP addresses to hosts PC1 and PC2 by using command given below:

- `ip <ip_addr> <subnet mask>`

```
PC1> ip 192.168.100.4/24 255.255.255.0
Checking for duplicate address...
PC1 : 192.168.100.4 255.255.255.0
```

After assigning IP addresses, check the status of IP address using command given below:

- `show ip`

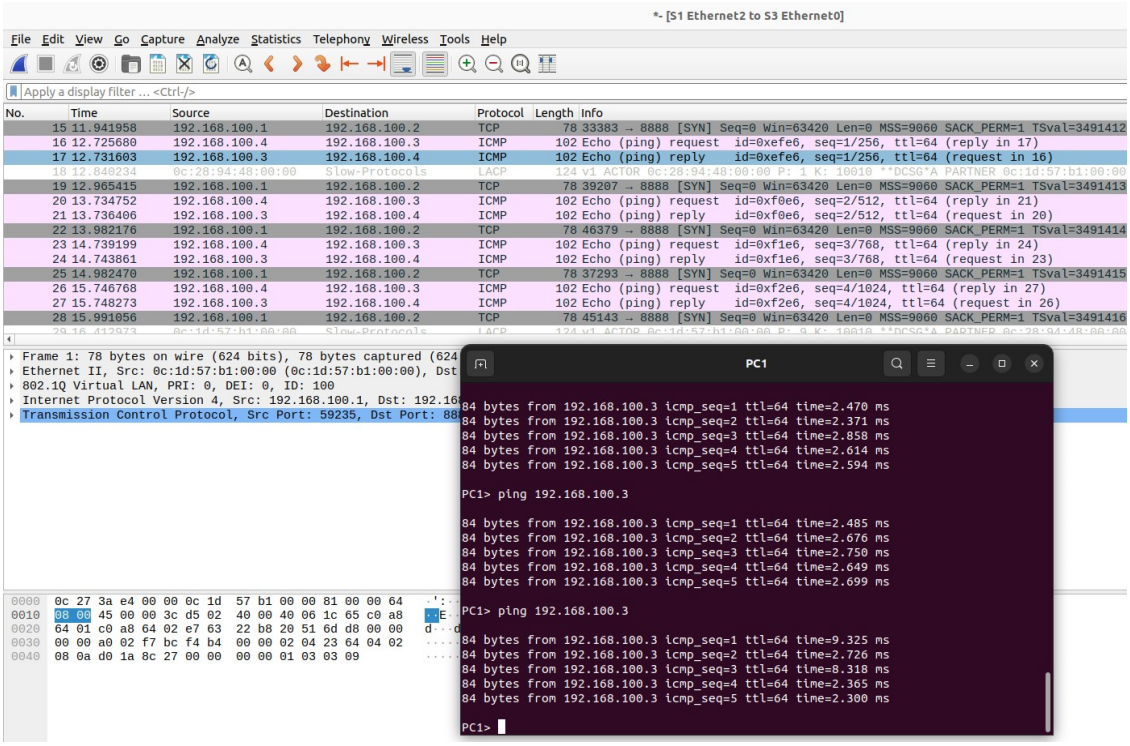
```
PC1> sh ip

NAME        : PC1[1]
IP/MASK     : 192.168.100.4/24
GATEWAY     : 255.255.255.0
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10000
RHOST:PORT  : 127.0.0.1:10001
MTU         : 1500
```

Result

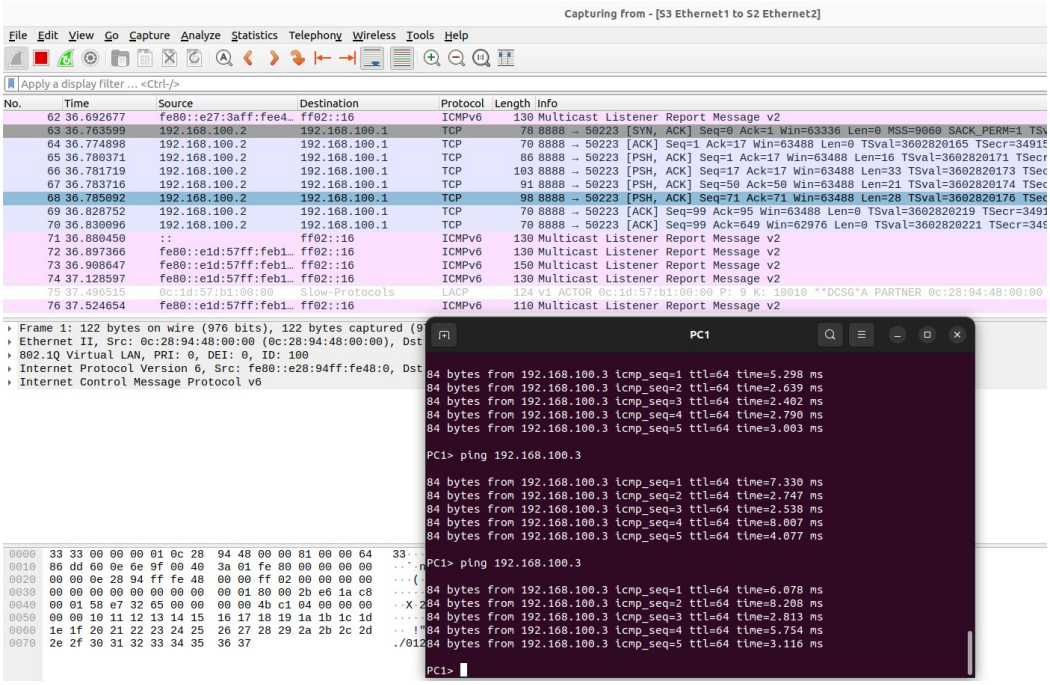
PC1 to PC2

Once the switches and hosts are configured, communication becomes possible among hosts in the same VLAN. As is evident from the provided figure below, PC1 is receiving a response from PC2, as both of them belong to the same VLAN. As per configurations, the role of S1 is “active. When Wireshark is started on the link between S1 and S3, the result shows that packets are being sent and received on this link because S1 is acting as 'active'." Furthermore, the TTL (Time-to-Live) value stays at 64 and remains unchanged because no routing is involved. Therefore, the MCLAG has been successfully configured.



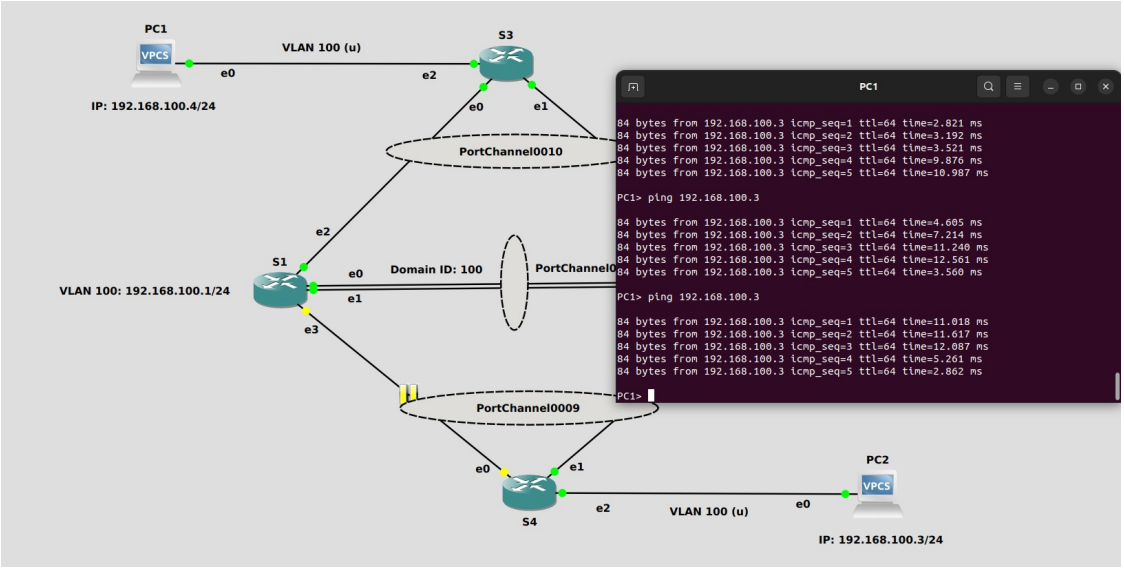
Result (Continued)

The figure below shows that when Wireshark is started on the link between S2 and S3, no packets are sent or received on it because the role of S2 is standby.



One Link Down

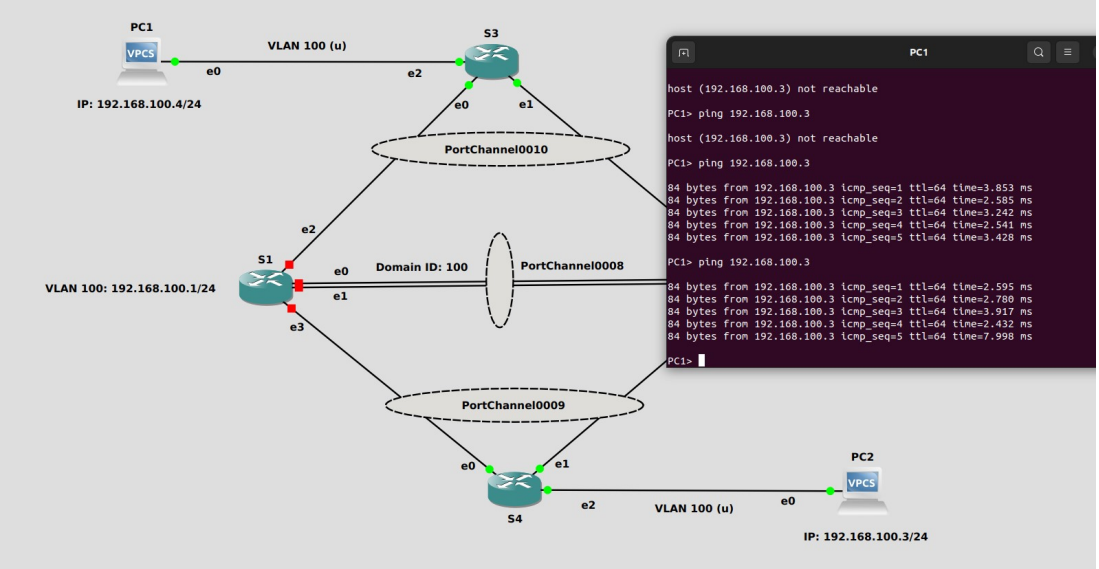
In this scenario, the link between S1 & S4 is paused, but PC1 still receives responses from PC2 because traffic is being sent through the link connecting S2 & S4.



Result (Continued)

S1 is Down

In this scenario, S1 is shut down. The interesting thing is that initially, host PC2 was not reachable because the role of S1 was 'active,' but after a few seconds, traffic was handled by S2.



References

https://github.com/sonic-net/sonic-utilities/blob/master/doc/Command-Reference.md
https://github.com/sonic-net/SONiC/blob/master/doc/mclag/Sonic-mclag-hld.md

We connect ideas, people, and technology.