#### **Guide for MCLAG in SONiC**

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

# Simplify SONiC adoption with Hardware Nation.

Talk with our specialists to learn about our integrated approach that includes guidance, training, professional services, support, and orchestration.

#### **Table of Contents**

Introduction to MCLAG	3
Network Topology	4
Port Mapping	5
Configurations	5
Step 1	5
Step 2	6
Step 3	7
Step 4	9
Step 5	9
Step 6	10
Step 7	11
Step 8	11
Step 9	12
Step 10	12
Result	13
References	15

## 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 linter-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 Ø	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 <u>here</u>.

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					l de la companya de l
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 function-ality 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 i<u>c</u>cpd.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					337237505
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: Interface	~\$ show ip Master	interfaces IPv4 address/mask	Admin/Oper	BGP Neighbor	Neighbor IP
Ethernet0		10.0.0/31	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



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:~$ <u>s</u>udo config portchannel member add PortChannel0009 Ethernet12
```

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

• show interfaces portchannel



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

-	~\$ show vlan b			
•			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>

VLAN ID	IP Address	Ports	Port Tagging	Proxy ARP
=======+ 100   	192.168.100.1/24	-=====================================	tagged	=+========   disabled   

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 <u>PortChannel0010</u>

#### 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

	:~\$ show vlan brie			
	IP Address		Port Tagging	Proxy ARP
   100   	192.168.100.1/24	PortChannel0008   PortChannel0009   PortChannel0010	tagged	disabled

## 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.



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 vl	an brief		
VLAN ID   IP Address	Ports	Port Tagging	Proxy ARP
+=======   100   	===+==================================	untagged   tagged	disabled   

mda	anish@sonic:	~\$ show vlan	brief		
Ì	VLAN ID	IP Address	Ports	Port Tagging	Proxy ARP
+=:   	============ 100   		+=====================================	+=====================================	+===========   disabled   

#### 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
мти	:	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.

Edit View Go Ca		3  ← →		11						
🔳 🖉 💿 🛅 🚺	🗎 🖹 🎑 🔍 🗸 🔪		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~							
oply a display filter «	<ctrl-></ctrl->									
Time	Source	Destination	Protocol	Length Info						
15 11.941958	192.168.100.1	192.168.100.2	TCP	78 33383 → 8	3888 [SYN] 3	Seq=0 Win=6:	3420 Len=0 /	ISS=9060	SACK_PERM=	1 TSval=34
16 12.725680	192.168.100.4	192.168.100.3	ICMP	102 Echo (pir	ng) request	id=0xefe6	seq=1/256	ttl=64	(reply in :	17)
17 12.731603	192.168.100.3	192.168.100.4	ICMP	102 Echo (pin	ng) reply	id=0xefe6	seq=1/256	ttl=64	(request in	n 16)
18 12.840234	0c:28:94:48:00:00	Slow-Protocols	LACP	124 v1 ACTOR	0c:28:94:41	B:00:00 P: :	1 K: 10010	**DCSG*A	PARTNER OC	
19 12.965415	192.168.100.1	192.168.100.2	TCP	78 39207 → 8	3888 [SYN] :	Seq=0 Win=6	3420 Len=0 1	1SS=9060	SACK_PERM=:	1 TSval=34
20 13.734752	192.168.100.4	192.168.100.3	ICMP	102 Echo (pin	ng) request	id=0xf0e6	seq=2/512	ttl=64	(reply in :	21)
21 13.736406	192.168.100.3	192.168.100.4	ICMP	102 Echo (pin	ng) reply	id=0xf0e6	seq=2/512	ttl=64	(request in	n 20)
22 13.982176	192.168.100.1	192.168.100.2	TCP	78 46379 → 8						
23 14.739199	192.168.100.4	192.168.100.3	ICMP	102 Echo (pir	ng) request	id=0xf1e6	seq=3/768	ttl=64	(reply in :	24)
24 14.743861	192.168.100.3	192.168.100.4	ICMP	102 Echo (pir	ng) reply	id=0xf1e6	seq=3/768	ttl=64	(request i	n 23)
25 14.982470	192.168.100.1	192.168.100.2	TCP	78 37293 → 8	3888 [SYN] 3	Seq=0 Win=6:	3420 Len=0 1	MSS=9060	SACK_PERM=:	1 TSval=34
26 15.746768	192.168.100.4	192.168.100.3	ICMP	102 Echo (nin	tagunar (no	id=0xf2e6	seg=4/102	4 ++1-6/	(renly in	27)
					ig) icquest			+, LLL-04	· (ICPLY III	21)
27 15.748273	192.168.100.3	192.168.100.4	ICMP	102 Echo (pir		id=0xf2e6	seq=4/102			
28 15.991056 29 16 412973 mme 1: 78 bytes of ernet II, Src: 0 2.10 Virtual LAN, ernet Protocol V	192.168.100.1 Ac:1d:57.b1:00:00 Don wire (624 bits), 77 BC:1d:57:b1:00:00 (OC PRI: 0, DEI: 0, ID: Yersion 4, Src: 192.1	192.168.100.2 Slow-Protocols 8 bytes captured (62 :1d:57:b1:00:00), Ds 100 68.100.1, Dst: 192.1	TCP	102 Echo (pir 78 45143 → 8 124 v1 ACTOP	ng) reply 3888 [SYN] : Oc:1d:57:b .3 icmp seq	PC1 =1 ttl=64 t	, seq=4/1024 3420 Len=0 1 1 K 10010	4, ttl=64 MSS=9060 **DCSG*A Q = s	A (request SACK_PERM=	in 26) 1 TSval=34
28 15.991056 29 16 412973 ame 1: 78 bytes of hernet II, Src: 0 2.10 Virtual LAN, ternet Protocol V	192.168.100.1 0c:1d:57:b1:00:00 0n wire (624 bits), 7 0c:1d:57:b1:00:00 (0c PPI: 0 DEI: 0 ID:	192.168.100.2 Slow-Protocols 8 bytes captured (62 :1d:57:b1:00:00), Ds 100 68.100.1, Dst: 192.1	TCP LACP 4 4 5 84 bytes f 84 bytes f 84 bytes f 84 bytes f 84 bytes f	102 Echo (pir 78 45143 - E 124 v1 ACTOP	<pre>ig) reply 3888 [SYN] 3 actidity for the second second</pre>	Seq=0 Win=63 1:00:00 P: 3 PC1 =1 ttl=64 t =2 ttl=64 t =3 ttl=64 t =4 ttl=64 t	, seq=4/1024 3420 Len=0 I 1 K 10010 1 me=2.470 m ime=2.371 m ime=2.858 m ime=2.614 m	4, ttl=64 MSS=9060 **DCSG*A Q = s s s s s	ACK_PERM=	in 26) 1 TSval=34
28 15.991056 29 16 412973 ame 1: 78 bytes of hernet II, Src: 0 2.10 Virtual LAN, ternet Protocol V	192.168.100.1 Ac:1d:57.b1:00:00 Don wire (624 bits), 77 BC:1d:57:b1:00:00 (0C PRI: 0, DEI: 0, ID: Version 4, Src: 192.1	192.168.100.2 Slow-Protocols 8 bytes captured (62 :1d:57:b1:00:00), Ds 100 68.100.1, Dst: 192.1	TCP LACP 4 4 6 84 bytes f 84 bytes f 84 bytes f 84 bytes f 84 bytes f	102 Echo (pir 78 45143 → 8 124 v1 ACTOP From 192.168.100 From 192.168.100	<pre>ig) reply 3888 [SYN] 3 actidity for the second second</pre>	Seq=0 Win=63 1:00:00 P: 3 PC1 =1 ttl=64 t =2 ttl=64 t =3 ttl=64 t =4 ttl=64 t	, seq=4/1024 3420 Len=0 I 1 K 10010 1 me=2.470 m ime=2.371 m ime=2.858 m ime=2.614 m	4, ttl=64 MSS=9060 **DCSG*A Q = s s s s s	ACK_PERM=	in 26) 1 TSval=34
28 15.991056 29 16 412973 mme 1: 78 bytes of ernet II, Src: 0 2.10 Virtual LAN, ernet Protocol V	192.168.100.1 Ac:1d:57.b1:00:00 Don wire (624 bits), 77 BC:1d:57:b1:00:00 (0C PRI: 0, DEI: 0, ID: Version 4, Src: 192.1	192.168.100.2 Slow-Protocols 8 bytes captured (62 :1d:57:b1:00:00), Ds 100 68.100.1, Dst: 192.1	TCP LACP 4 4 6 8 8 4 bytes f 8 4 bytes f 84 bytes f 84 bytes f 84 bytes f 84 bytes f 84 bytes f 84 bytes f	102 Echo (pir 78 45143 - E 124 v1 4 6709 From 192.168.100 From 192.168.100 From 192.168.100 From 192.168.100 From 192.168.100	<pre>id; reply 3888 [SYN] * ae:1d:57:b .3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq</pre>	Seg=0         Win=63           PC1           =1         ttl=64           =2         ttl=64           =3         ttl=64           =4         ttl=64           =5         ttl=64	ime=2.470 m ime=2.470 m ime=2.371 m ime=2.858 m ime=2.594 m	4, ttl=64 MSS=9060 DCSC*A Q = s s s s s s s	ACK_PERM=	in 26) 1 TSval=34
28 15.991056 29 16 412973 ame 1: 78 bytes of hernet II, Src: 0 2.10 Virtual LAN, ternet Protocol V	192.168.100.1 Ac:1d:57.b1:00:00 Don wire (624 bits), 77 BC:1d:57:b1:00:00 (0C PRI: 0, DEI: 0, ID: Version 4, Src: 192.1	192.168.100.2 Slow-Protocols 8 bytes captured (62 :1d:57:b1:00:00), Ds 100 68.100.1, Dst: 192.1	TCP LACP 4 6 84 bytes f 84 bytes f	102 Echo (pir 78 45143 - 8 124 v1 4 100 rom 192.168.100 rom 192.168.100 rom 192.168.100 rom 192.168.100 192.168.100.3 rom 192.168.100	<pre>g) reply 3888 [SYN] ! Ac.1d:57.b 3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq</pre>	Seq=0 Win=63         PC1         =1 ttl=64 tt         =2 ttl=64 tt         =3 ttl=64 tt         =5 ttl=64 tt         =1 ttl=64 tt	ime=2.470 m ime=2.470 m ime=2.371 m ime=2.614 m ime=2.594 m ime=2.485 m	4, ttl=64 MSS=9060 *DCSG*A Q = s s s s s s s	ACK_PERM=	in 26) 1 TSval=34
28 15.991056 29 16 412973 mme 1: 78 bytes of ernet II, Src: 0 2.10 Virtual LAN, ernet Protocol V	192.168.100.1 Ac:1d:57.b1:00:00 Don wire (624 bits), 77 BC:1d:57:b1:00:00 (0C PRI: 0, DEI: 0, ID: Version 4, Src: 192.1	192.168.100.2 Slow-Protocols 8 bytes captured (62 :1d:57:b1:00:00), Ds 100 68.100.1, Dst: 192.1	TCP LACP 4 4 6 84 bytes f 84 bytes f	102 Echo (pir 78 45143 - E 124 v1 4 7709 102 102 102 102 102 102 102 102 102 102	<pre>ig) reply assa [SYN] :</pre>	Seq=0         Win=61           PC1         PC1           =1         ttl=64         t           =3         ttl=64         t           =3         ttl=64         t           =5         ttl=64         t           =1         ttl=64         t           =1         ttl=64         t           =1         ttl=64         t           =2         ttl=64         t	seq=4/102 3420 Len=0 I k 10410 ime=2.470 m ime=2.371 m ime=2.858 m ime=2.594 m ime=2.614 m ime=2.676 m	4, ttl=64 MSS=9060 *DCSG*A Q = s s s s s s s s s	ACK_PERM=	in 26) 1 TSval=34
28 15.991056 29 16 412973 ame 1: 78 bytes of hernet II, Src: 0 2.10 Virtual LAN, ternet Protocol V	192.168.100.1 Ac:1d:57.b1:00:00 Don wire (624 bits), 77 BC:1d:57:b1:00:00 (0C PRI: 0, DEI: 0, ID: Version 4, Src: 192.1	192.168.100.2 Slow-Protocols 8 bytes captured (62 :1d:57:b1:00:00), Ds 100 68.100.1, Dst: 192.1	TCP LACP 44 (7) 6184 bytes f 84 bytes f	102 Echo (pir 78 45143 - 8 124 v1 2000 rom 192.168.100 rom 192.168.100 rom 192.168.100 rom 192.168.100 192.168.100.3 rom 192.168.100 rom 192.168.100 rom 192.68.100	<pre>ig) reply assa [SYN] :</pre>	Seq=0         Win=6:           PC1         PC1           =1         ttl=64         tt           =3         ttl=64         tt           =4         ttl=64         t           =1         ttl=64         tt           =2         ttl=64         t           =1         ttl=64         t           =2         ttl=64         t           =2         ttl=64         t	seq=4/102; 3420 Len=0 1 k = 18410 ime=2.470 m ime=2.371 m ime=2.674 m ime=2.594 m ime=2.676 m ime=2.750 m	4, ttl=64 MSS=9060 * DCSG*A Q = s s s s s s s s s s s	ACK_PERM=	in 26) 1 TSval=34
28 15.991056 29 16 412973 ame 1: 78 bytes of hernet II, Src: 0 2.10 Virtual LAN, ternet Protocol V	192.168.100.1 Ac:1d:57.b1:00:00 Don wire (624 bits), 77 BC:1d:57:b1:00:00 (0C PRI: 0, DEI: 0, ID: Version 4, Src: 192.1	192.168.100.2 Slow-Protocols 8 bytes captured (62 :1d:57:b1:00:00), Ds 100 68.100.1, Dst: 192.1	TCP 14 1 1 1 1 1 1 1 1 1 1 1 1 1	102 Echo (pir 78 45143 - E 124 v1 & T09 700 192.168.100 700 192.168.100 700 192.168.100 700 192.168.100 192.168.100 700 192.168.100 700 192.168.100 700 192.168.100	<pre>ig) reply issa [SYN] :</pre>	Seq=0         Win=6;           PC1         PC1           =1         ttl=64         tt           =2         ttl=64         t           =4         ttl=64         t           =5         ttl=64         t           =1         ttl=64         t           =2         ttl=64         t           =1         ttl=64         t           =2         ttl=64         t           =3         ttl=64         t           =4         ttl=64         t	seq=4/102 3420 Len=0 T k = 10010 ime=2.470 m ime=2.371 m ime=2.614 m ime=2.614 m ime=2.616 m ime=2.676 m ime=2.676 m	4, ttl=64 HSS=9960 CSG3A Q ≡ S S S S S S S S S S S S S S S S	ACK_PERM=	in 26) 1 TSval=34
28 15.991056 20 18 412073 men 1: 78 bytes of ernet II, Src: ( 2.1Q Virtual LAN, ernet Protocol V ansmission Contro	192,468,100,1 Beridis7,b1:00:00 on wire (624 bits), 7 Scidi57,b1:00:00 (6c PRI: 0, DEI: 0, ID: Fersion 4, Src: 192.1 Sl Protocol, Src Port	192.168.100.2 Slow-Drotocols 8 bytes captured (62 :1d:57:b1:00:00), Ds 100 60:100.1, Dst: 192.1 : 59235, Dst Port: 8	TCP TACO 4 4 6 84 bytes f 84 bytes f	102 Echo (pir 78 45143 - 8 124 v1 2000 rom 192.168.100 rom 192.168.100 rom 192.168.100 rom 192.168.100 192.168.100.3 rom 192.168.100 rom 192.168.100 rom 192.68.100	<pre>ig) reply issa [SYN] :</pre>	Seq=0         Win=6;           PC1         PC1           =1         ttl=64         tt           =2         ttl=64         t           =4         ttl=64         t           =5         ttl=64         t           =1         ttl=64         t           =2         ttl=64         t           =1         ttl=64         t           =2         ttl=64         t           =3         ttl=64         t           =4         ttl=64         t	seq=4/102 3420 Len=0 T k = 10010 ime=2.470 m ime=2.371 m ime=2.614 m ime=2.614 m ime=2.616 m ime=2.676 m ime=2.676 m	4, ttl=64 HSS=9960 CSG3A Q ≡ S S S S S S S S S S S S S S S S	ACK_PERM=	in 26) 1 TSval=34
28 15.991056 29 16 412073 ame 1: 78 bytes o hernet IT, Src: ( 2.10 Virtua LAN, ternet Protocol V ansmission Contro 0 0 0 0 4 5 00 00 64 01 c0 a8 64	192,168,100,1 Bertid-S7:b1100-00 on wire (624 bits), 7 Scid:57:b1:00:00 (0c PRI: 0, DEI: 0, ID: PRI: 0, DEI: 0, ID: Protocol, Src Port	192.168.100.2 Slow_Protocols 8 bytes captured (62 11d:57:b1:00:00), Ds 100 68.100.1, Dst: 192.1 59235, Dst Port: 8	CP LACP 4 4 4 6 84 bytes f 84 bytes f	102 Echo (pir 78 45143 - E 122 v1 4 100 rom 192.168.100 rom 192.168.100 rom 192.168.100 rom 192.168.100 192.168.100.3 rom 192.168.100 rom 192.168.100 rom 192.168.100 rom 192.168.100 rom 192.168.100	<pre>ig) reply 3888 [SYN] i .3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq .3 icmp_seq</pre>	<pre>Seq=0 WAne6 PC1 =1 ttl=64 t =2 ttl=64 t =3 ttl=64 t =5 ttl=64 t =1 ttl=64 t =2 ttl=64 t =3 ttl=64 t =4 ttl=64 t =5 ttl=64 t</pre>	. seq-4/102 3426 Lene® 1 1 k = 46014 ime=2, 470 m ime=2, 470 m ime=2, 470 m ime=2, 584 m ime=2, 584 m ime=2, 564 m ime=2, 569 m ime=2, 699 m	4, ttl=64 HSS=9060 C = S S S S S S S S S S S S S S	ACK_PERM=	in 26) 1 TSval=34
28 15.991056 20 16 412073 me 1: 78 bytes of ernet IT, Src: ( 2.1Q Virtual LAN, ernet Protocol V unsmission Contro 06 06 45 00 00 64 01 c0 88 64 00 00 0 00 27 7	192,468,100,1 BerlitsTiblionead on wire (624 bits), 7: Scidi57:bit00:00 (6c PRI: 0, DEI: 0, ID: Fersion 4, Src: 192.1 Di Protocol, Src Port 0 c d5 02 40 00 40 6 3c d5 02 40 00 40 6 0 2 c 7 63 22 b8 20 5 bc f4 b4 00 00 62 c	192.168.100.2 Slow-Drotocols 8 bytes captured (62 13d:57:b1:00:00), Ds 100 66.100.1, Dst: 192.1 : 59235, Dst Port: 8	CCP Lacov 4 4 4 4 5 84 bytes f 84 bytes f	102 Echo (pir 78 4513 & 104 v1 & TTO 70 192.168.100 from 192.168.100 from 192.168.100 from 192.168.100 192.168.100.3 from 192.168.100 from 192.168.100 from 192.168.100 from 192.168.100 192.168.100.3 from 192.168.100.3	<pre>inj reply isses [swn] a trop_seq .3 trop_seq</pre>	<pre>30:20 WARE0 F00:20 PC1 PC1 PC1 PC1 PC1 PC1 PC1 PC1 PC1 PC1</pre>	seq-4/102 320 Lene 0 1 ro 10010 1 ro 10000 1 ro 100000 1 ro 100000 1 ro 100000 1 ro 10000000000000000000000000000000000	4, ttl=64 (ISS=9060 (ISS=9	ACK_PERM=	in 26) 1 TSval=34
28 15.991056 20 16 412073 me 1: 78 bytes of ernet IT, Src: ( 2.1Q Virtual LAN, ernet Protocol V unsmission Contro 06 06 45 00 00 64 01 c0 88 64 00 00 0 00 27 7	192,168,100,1 Bertid-S7:b1100-00 on wire (624 bits), 7 Scid:57:b1:00:00 (0c PRI: 0, DEI: 0, ID: PRI: 0, DEI: 0, ID: Protocol, Src Port	192.168.100.2 Slow-Drotocols 8 bytes captured (62 13d:57:b1:00:00), Ds 100 66.100.1, Dst: 192.1 : 59235, Dst Port: 8	CIP CP CP CP CP CP CP CP CIP CI	102 Echo (pir 78 45143 - 8 122 v1 4 100 rom 192.168.100 rom 192.168.100 rom 192.168.100 rom 192.168.100 192.168.100.3 rom 192.168.100 rom 192.168.100 rom 192.168.100 rom 192.168.100.3 rom 192.168.100.3 rom 192.168.100.3	<pre>ig) reply 3888 [SYN] i 3 icmp_seq 3 icmp_seq</pre>	<pre>3q=3 WARE0 indiana Pict PC1</pre>	.seq-4/102 3420 Lene 1 ime=2,470 m ime=2,371 m ime=2,534 m ime=2,676 m ime=2,676 m ime=2,676 m ime=2,679 m ime=2,699 m ime=2,699 m	4, ttl=64 itss=9060 Q ≡ s s s s s s s s s s s s s	ACK_PERM=	in 26) 1 TSval=34
28 35.991056 20 18 412073 me 1: 78 bytes of hernet II, Src: ( 2.1Q Virtual LAN, ternet Protocol V ansmission Contro 08 00 45 00 00 64 01 co 88 64 00 00 a0 02 f7	192,468,100,1 BerlitsTiblionead on wire (624 bits), 7: Scidi57:bit00:00 (6c PRI: 0, DEI: 0, ID: Fersion 4, Src: 192.1 Di Protocol, Src Port 0 c d5 02 40 00 40 6 3c d5 02 40 00 40 6 0 2 c 7 63 22 b8 20 5 bc f4 b4 00 00 62 c	192.168.100.2 Slow-Drotocols 8 bytes captured (62 13d:57:b1:00:00), Ds 100 66.100.1, Dst: 192.1 : 59235, Dst Port: 8	CP LACP LACP 4 4 4 7 8 8 4 9 4 9 4 9 4 9 4 9 4 9 4 9 4 9 4 9	102 Echo (pir 78 4513 & 104 v1 & TTO 70 192.168.100 from 192.168.100 from 192.168.100 from 192.168.100 192.168.100.3 from 192.168.100 from 192.168.100 from 192.168.100 from 192.168.100 192.168.100.3 from 192.168.100.3	<pre>inj reply ing reply i</pre>	<pre>39:20 WARE0 F00.00 P. PC1 2 ttl=64 t 2 ttl=64 t =3 ttl=64 t =3 ttl=64 t =3 ttl=64 t =3 ttl=64 t =3 ttl=64 t =5 ttl=64 t =1 ttl=64 t =1 ttl=64 t =1 ttl=64 t =3 ttl=64 t</pre>	seq-4/102 320 Lene 0 a ro 10010 ime=2.470 m ime=2.470 m ime=2.531 m ime=2.534 m ime=2.534 m ime=2.534 m ime=2.539 m ime=2.639 m ime=2.726 m ime=3.325 m	4, ttl=64 455=9060 455=9060 455=9060 455=9060 5 5 5 5 5 5 5 5 5 5 5 5 5	ACK_PERM=	in 26) 1 TSval=34

## **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.

Copyright © 2023 Hardware Nation LLC. All Rights Reserved. All other trademarks used in this document are the property of their respective owners.