

GUIDE

By: Humza Altaf, SONiC Network Engineer

Revision No.	Description	Editor	Date
1.0	Guide for BGP Unnumbered between SONiC and FRR VM	Humza Altaf	Mar 16, 2024

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 FRR	3
System Architecture	3
Network Topology	4
Port Mapping	5
Configurations	5
Step 1	6
Step 2	7
Step 3	8
Step 4	9
Step 5	9
Step 6	10
Step 7	11
Result	12
References	12

Introduction to FRR

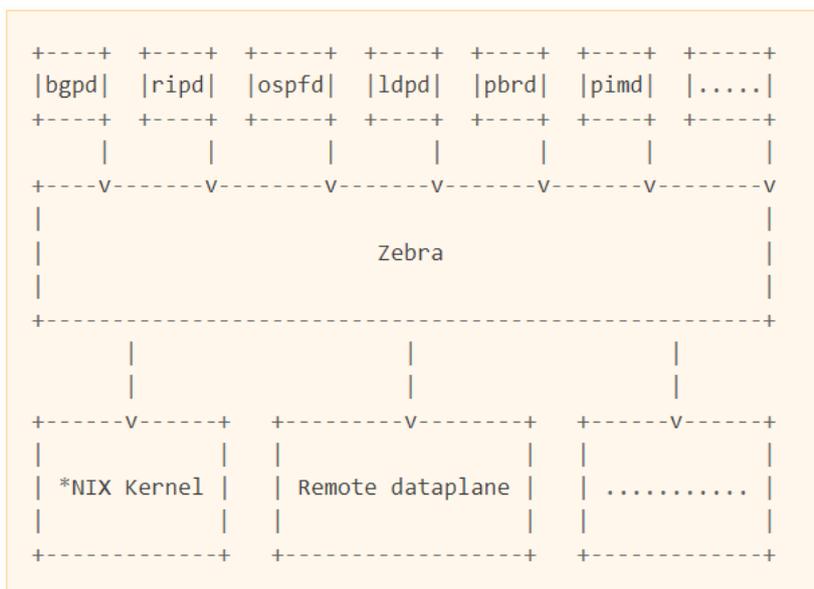
FRRouting (FRR) is a free and open-source Internet routing protocol suite for Linux and Unix platforms. It implements BGP, OSPF, RIP, IS-IS, PIM, LDP, BFD, Babel, PBR, OpenFabric and VRRP, with alpha support for EIGRP and NHRP.

FRR's seamless integration with native Linux/Unix IP networking stacks makes it a general-purpose routing stack applicable to a wide variety of use cases including connecting hosts/VMs/containers to the network, advertising network services, LAN switching and routing, Internet access routers, and Internet peering.

FRR has its roots in the Quagga project. In fact, it was started by many long-time Quagga developers who combined their efforts to improve on Quagga's well-established foundation in order to create the best routing protocol stack available. We invite you to participate in the FRRouting community and help shape the future of networking.

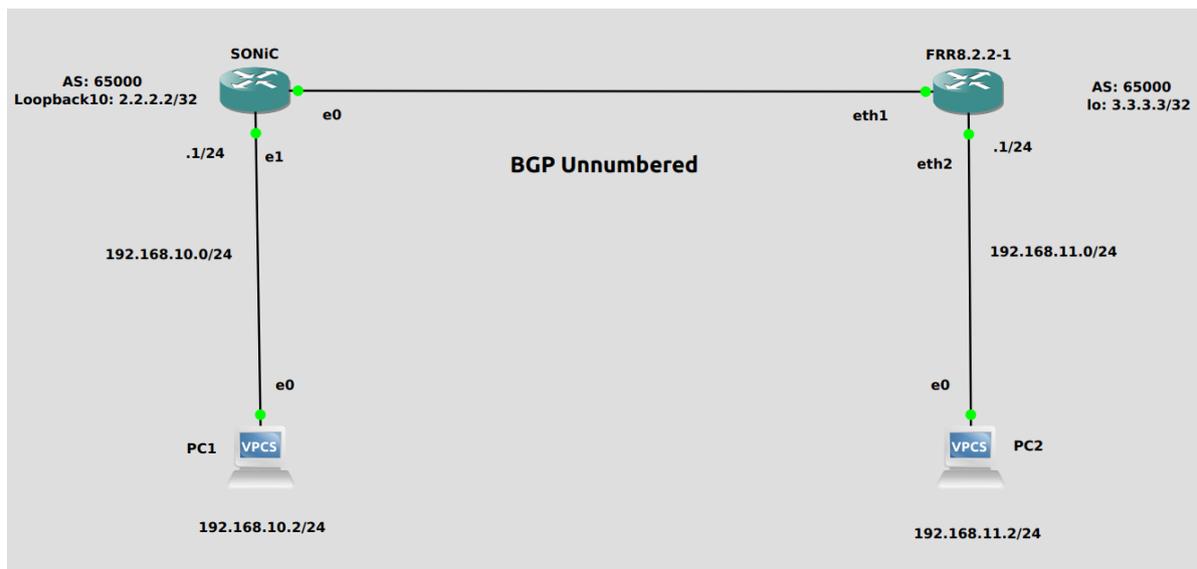
System Architecture

Traditional routing software is made as a one-process program which provides all of the routing protocol functionalities. FRR takes a different approach. FRR is a suite of daemons that work together to build the routing table. Each major protocol is implemented in its own daemon, and these daemons talk to a middleman daemon (zebra), which is responsible for coordinating routing decisions and talking to the data plane.



Network Topology

The GNS3 network topology consists of a SONiC router and an FRR VM (version 8.2.2). The FRR VM acts as a router in this setup. Both routers are in the same ASN, which is 65000. A loopback interface (Loopback10) is created on SONiC with the IP address 2.2.2.2/32, while a loopback interface (lo) is created on FRR with the IP address 3.3.3.3/32. No IP address is assigned to the interfaces between SONiC and FRR. PC1 is in the network 192.168.10.0/24, while PC2 is in the network 192.168.11.0/24.



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, SONiC is configured and the same steps are repeated for the FRR VM. Command Reference guide is also available on GitHub for SONiC, whose link is given [here](#).

Follow these steps to configure SONiC.

Step 1

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  Ethernet0  10.0.0.0/31        up/up       ARISTA01T2    10.0.0.1
Ethernet4  Ethernet4  10.0.0.2/31        up/up       ARISTA02T2    10.0.0.3
```

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.

Now Assign IP addresses as per topology and then check the status of interfaces.

```
admin@sonic:~$ sudo config interface ip add Ethernet4 192.168.10.1/24
admin@sonic:~$ sudo config interface ip add Loopback10 2.2.2.2/32
```

```
admin@sonic:~$ show ip interfaces
/usr/local/lib/python3.9/dist-packages/sonic_platform_base/sonic_xcvr/fields/xcvr_field
assert (bitpos >= 0 and (bitpos+self.size-1 < 8), "bitpos must be within one byte")
Interface  Master  IPv4 address/mask  Admin/Oper  BGP Neighbor  Neighbor IP
-----
Ethernet4  Ethernet4  192.168.10.1/24    up/up       N/A           N/A
```

Step 2

In SONiC, by default BGP is running with ASN 65100. First remove that instance and then configure BGP with unnumbered and announce the network by using the following commands given below in the screenshot.

```
admin@sonic:~$ vtysh
Hello, this is FRRouting (version 8.5.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

sonic# configure
sonic(config)# router bgp 65000
BGP is already running; AS is 65100
sonic(config)# no router bgp 65100
sonic(config)# router bgp 65000
sonic(config-router)# bgp router-id 2.2.2.2
sonic(config-router)# neighbor Ethernet0 interface remote 65000
sonic(config-router)# neighbor Ethernet0 interface remote-as internal
sonic(config-router)# address-family ipv4
sonic(config-router-af)# network 192.168.10.0/24
sonic(config-router-af)# end
```

Configurations can be saved using the “write” or “wr” command. Configurations are saved in “/etc/frr/bgpd.-conf” file.

```
sonic# wr
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Configuration saved to /etc/frr/zebra.conf
Configuration saved to /etc/frr/bgpd.conf
Configuration saved to /etc/frr/staticd.conf
```

Follow these steps to configure FRR router.

Step 3

Assign IP addresses to interfaces by using the following commands given below in the screenshot.

```
Hello, this is FRRouting (version 8.2.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr# configure
frr(config)# interface eth2
frr(config-if)# ip address 192.168.11.1/24
frr(config-if)# no shutdown
frr(config-if)# exit
frr(config)# interface eth1
frr(config-if)# no shutdown
frr(config-if)# exit
frr(config)# interface lo
frr(config-if)# ip address 3.3.3.3/32
frr(config-if)# no shutdown
frr(config-if)# exit
```

The status of interfaces can be checked by using the following commands given below in the screenshot.

```
frr# show interface eth2
Interface eth2 is up, line protocol is up
Link ups:      1      last: 2024/03/16 11:09:58.46
Link downs:    1      last: 2024/03/16 11:09:58.46
vrf: default
index 4 metric 0 mtu 1500 speed 1000
flags: <UP,BROADCAST,RUNNING,MULTICAST>
Type: Ethernet
HWaddr: 0c:ea:28:1f:00:02
inet 192.168.11.1/24
inet6 fe80::eea:28ff:fe1f:2/64
Interface Type Other
Interface Slave Type None
protodown: off
frr# show interface lo
Interface lo is up, line protocol is up
Link ups:      0      last: (never)
Link downs:    0      last: (never)
vrf: default
index 1 metric 0 mtu 65536 speed 0
flags: <UP,LOOPBACK,RUNNING>
Type: Loopback
inet 3.3.3.3/32
Interface Type Other
Interface Slave Type None
protodown: off
```

Step 4

Now configure BGP unnumbered with ASN 65000 and announce the network by using the following commands given below in the screenshot.

```
frr# configure
frr(config)# router bgp 65000
frr(config-router)# bgp router-id 3.3.3.3
frr(config-router)# neighbor eth1 interface remote 65000
frr(config-router)# neighbor eth1 interface remote-as internal
frr(config-router)# address-family ipv4
frr(config-router-af)# network 192.168.11.0/24
frr(config-router-af)# end
frr# wr
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
```

Step 5

Now check the status of BGP Unnumbered on SONiC.

```
sonic# show bgp summary

IPv4 Unicast Summary (VRF default):
BGP router identifier 2.2.2.2, local AS number 65000 vrf-id 0
BGP table version 2
RIB entries 3, using 576 bytes of memory
Peers 1, using 725 KiB of memory

Neighbor      V      AS  MsgRcvd  MsgSent  TblVer  InQ  OutQ
Ethernet0     4      65000    15      15        0     0

Total number of neighbors 1
```

```
sonic# show bgp neighbors
BGP neighbor on Ethernet0: fe80::eea:28ff:fe1f:1, remote AS 65000, local AS 65000, internal link
  Local Role: undefined
  Remote Role: undefined
Hostname: frr
  BGP version 4, remote router ID 3.3.3.3, local router ID 2.2.2.2
  BGP state = Established, up for 00:07:11
  Last read 00:00:11, Last write 00:00:11
  Hold time is 180 seconds, keepalive interval is 60 seconds
  Configured hold time is 180 seconds, keepalive interval is 60 seconds
  Configured conditional advertisements interval is 60 seconds
Neighbor capabilities:
  4 Byte AS: advertised and received
  Extended Message: advertised and received
  AddPath:
    IPv4 Unicast: RX advertised and received
  Extended nexthop: advertised and received
  Address families by peer:
```

Step 6

Now check the status of BGP Unnumbered on FRR router.

```
frr# show bgp summary

IPv4 Unicast Summary (VRF default):
BGP router identifier 3.3.3.3, local AS number 65000 vrf-id 0
BGP table version 4
RIB entries 3, using 552 bytes of memory
Peers 1, using 716 KiB of memory

Neighbor          V      AS  MsgRcvd  MsgSent  TblVer  InQ
eth1              4     65000     16      17        0    0

Total number of neighbors 1
```

```
frr# show bgp neighbors
BGP neighbor on eth1: fe80::ec9:e3ff:fe50:0, remote AS 65000, local AS 65000, internal link
Hostname: sonic
  BGP version 4, remote router ID 2.2.2.2, local router ID 3.3.3.3
  BGP state = Established, up for 00:09:04
  Last read 00:00:04, Last write 00:00:04
  Hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    4 Byte AS: advertised and received
    Extended Message: advertised and received
  AddPath:
    IPv4 Unicast: RX advertised and received
  Extended nexthop: advertised and received
  Address families by peer:
    IPv4 Unicast
```

```
frr# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

C>* 3.3.3.3/32 is directly connected, lo, 00:11:36
B>* 192.168.10.0/24 [200/0] via fe80::ec9:e3ff:fe50:0, eth1, weight 1, 00:10:08
C>* 192.168.11.0/24 is directly connected, eth2, 00:12:07
```

Step 7

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

- `ip <ip_addr> <subnet mask>`

```
PC1> ip 192.168.10.2/24 192.168.10.1
Checking for duplicate address...
PC1 : 192.168.10.2 255.255.255.0 gateway 192.168.10.1
```

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

- `show ip`

```
PC1> sh ip
NAME       : PC1[1]
IP/MASK    : 192.168.10.2/24
GATEWAY    : 192.168.10.1
DNS        :
MAC        : 00:50:79:66:68:00
LPORT     : 10042
RHOST:PORT : 127.0.0.1:10043
MTU       : 1500
```

Result

PC1 to PC2

In conclusion, the implementation of BGP Unnumbered between SONiC and FRR demonstrates its effectiveness in networking. Once switches and hosts are configured, seamless communication occurs, as seen with PC1 receiving a response from PC2. The TTL decrement to 62 confirms successful configuration, highlighting the robustness of BGP Unnumbered in modern networking environments.

```
PC1> ping 192.168.11.2
84 bytes from 192.168.11.2 icmp_seq=1 ttl=62 time=7.559 ms
84 bytes from 192.168.11.2 icmp_seq=2 ttl=62 time=4.240 ms
84 bytes from 192.168.11.2 icmp_seq=3 ttl=62 time=4.424 ms
84 bytes from 192.168.11.2 icmp_seq=4 ttl=62 time=5.138 ms
84 bytes from 192.168.11.2 icmp_seq=5 ttl=62 time=4.323 ms
```

References

<https://github.com/sonic-net/sonic-utilities/blob/master/doc/Command-Reference.md>

<https://frrouting.org/>

**We connect ideas, people,
and technology.**