Portal > Knowledgebase > Technical > Networking > Multicast with GRE

Multicast with GRE

Jem Camba - 2019-03-06 - in Networking

What is Multicast?

Multicast (often referred to as multicasting) is where data transmission is addressed to a group of destination computers simultaneously. Multicast can be a one-to-many or a many-to-many distribution. Multicast requires the source to send a packet only once, even if it needs to be delivered to a large number of receivers.

Why is Multicasting disabled on Packet's network?

Packet does not support multi-cast in its default Layer 3 network topology. This is due to performance and security concerns around multi-tenant switch and router scaling issues.

To do this, we suggest leveraging a GRE tunnel.

What is GRE?

Generic Routing Encapsulation (or GRE for short), is a tunneling protocol developed by Cisco Systems that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links over an Internet Protocol network.

How to setup a GRE tunnel between two devices

Setting a GRE tunnel between devices is pretty straight forward. The following is a basic configuration for CentOS devices.

Device 1:

```
DEVICE=gre1
BOOTPROTO=none
ONBOOT=yes
TYPE=GRE
PEER_OUTER_IPADDR=Site2.public.address
```

```
PEER_INNER_IPADDR=Site2.private.address
MY_INNER_IPADDR=Site1.private.address
```

Device 2:

```
DEVICE=gre1
BOOTPROTO=none
ONBOOT=yes
TYPE=GRE
PEER_OUTER_IPADDR=Site1.public.address
PEER_INNER_IPADDR=Site1.private.address
MY_INNER_IPADDR=Site2.private.address
```

On both devices, bring the interface up by running `ifup gre1`

To verify configuration, you can run `ifconfig gre` and sample the output here, on device 1:

```
[root@centos-ewr1 ~]# ifconfig gre1
gre1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1476
        inet 10.100.126.3  netmask 255.255.255.255  destination 10.88.152.3
        unspec 00-00-00-00-00-00-F0-00-00-00-00-00-00-00-00-00  txqueuelen 0
(UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@centos-ewr1 ~]#
```

The same ran on device 2:

```
[root@centos-sjc1 ~]# ifconfig gre1
gre1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1476
        inet 10.88.152.3  netmask 255.255.255.255  destination 10.100.126.3
        unspec 00-00-00-00-00-00-F0-00-00-00-00-00-00-00-00-00  txqueuelen 0
(UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
[root@centos-sjc1 ~]#
```

Ping Device 2 <> Device 1

```
[root@centos-sjc1 ~]# ping -c5 10.100.126.3
PING 10.100.126.3 (10.100.126.3) 56(84) bytes of data.
64 bytes from 10.100.126.3: icmp_seq=1 ttl=64 time=73.5 ms
64 bytes from 10.100.126.3: icmp_seq=2 ttl=64 time=73.5 ms
64 bytes from 10.100.126.3: icmp_seq=3 ttl=64 time=73.5 ms
64 bytes from 10.100.126.3: icmp_seq=4 ttl=64 time=73.5 ms
64 bytes from 10.100.126.3: icmp_seq=5 ttl=64 time=73.4 ms
--- 10.100.126.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 73.416/73.516/73.558/0.179 ms
[root@centos-sjc1 ~]#
```

```
[root@centos-ewr1 ~]# ping -c5 10.88.152.3
PING 10.88.152.3 (10.88.152.3) 56(84) bytes of data.
64 bytes from 10.88.152.3: icmp_seq=1 ttl=64 time=73.4 ms
64 bytes from 10.88.152.3: icmp_seq=2 ttl=64 time=73.1 ms
64 bytes from 10.88.152.3: icmp_seq=3 ttl=64 time=73.4 ms
64 bytes from 10.88.152.3: icmp_seq=4 ttl=64 time=73.4 ms
64 bytes from 10.88.152.3: icmp_seq=5 ttl=64 time=73.4 ms
--- 10.88.152.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 73.183/73.405/73.489/0.360 ms
[root@centos-ewr1 ~]#
```

A quick TCPDump `tcpdump -n -i bond0 proto 47` on `bond0` you can verify the traffic between the two devices are flowing through the GRE tunnel

```
17:36:22.295418 IP 147.75.64.13 > 147.75.69.141: GREv0, length 56: IP
10.100.126.3.48950 > 10.88.152.3.ssh: Flags [.], ack 2897, win 249, options
[nop,nop,TS val 2332411 ecr 6130351], length 0

17:36:25.360737 IP 147.75.64.13 > 147.75.69.141: GREv0, length 124: IP
10.100.126.3.48950 > 10.88.152.3.ssh: Flags [P.], seq
1764:1832, ack 2897, win 249, options [nop,nop,TS val 2335476 ecr 6130351], length
68

17:36:25.360917 IP 147.75.69.141 > 147.75.64.13: GREv0, length 124: IP
```

```
10.88.152.3.ssh > 10.100.126.3.48950: Flags [P.], seq
2897:2965, ack 1832, win 149, options [nop,nop,TS val 6133490 ecr 2335476], length
68
```

Note: the above method sends traffic in plain text. To encrypt traffic it would be wise to setup IPSec, Wireguard, etc.

Enabling Multicast

To ensure `gre1` has multicast enabled in the event of a reboot please check `/usr/sbin/ifup-pre-local` and make sure it looks like this example:

```
#!/bin/bash
set -o errexit -o nounset -o pipefail -o xtrace
iface=${1#*-}
case $iface in
 bond0 | enp0s20f0) ip link set $iface address 0c:c4:7a:81:0a:84;;
        enp0s20f1) ip link set $iface address 0c:c4:7a:81:0a:85 && sleep 4;;
        gre1) ip link set $iface multicast on;;
 *) echo "ignoring unknown interface $iface" && exit 0;;
 esac
```

It should be noted, the interface named `gre1` can of course be renamed to something that matches your deployment & configuration.

External Resources

- GRE Tunneling (how to)
- GRE Tunnel Linux > Cisco
- IPIP and GRE Encapsulation

Tags
GRE
multicast