Ubiquiti

EdgeOS IPv6 on Single WAN / IPv4 LB FO Working Config

From my article here: https://community.ui.com/questions/IPv6-on-Single-WAN-IPv4-LB-FO-Working-Config/2ed8f836-e02e-4b65-aba1-8a0ebff1e16c

I have residential Spectrum service with native IPv6 via DHCPv6-PD via eth0 and AT&T via eth1 which I only care to use for IPv4. After a lot of trial and error, I came up with the following config and transition script. When a transition is detected it runs a simple script which does several things:

- Checks eth0 Spectrum connectivity by pinging Google DNS 10 times. If it's up we do the following:
- Resets local traffic by disabling/enabling lb-local-metric-change(This is the only way I could ever get UNMS and local DNS to communicate using the primary WAN after a failback event without a reboot, there's a post somewhere in here of mine that talks more about this)
- On a failback to Spectrum event the IPv6 address is again released, we kill dhcp6c and then renew.
- Reset Conntrack to get all clients on the right circuit.

On a failover to AT&T event, the Spectrum IPv6 address is released and I kill the dhcp6c daemon for good measure, which is the "else" condition of this script.

For whatever reason, I could NOT get IPv6 to start back up correctly without killing dhcp6c before a renew. I tried and tried, but this is the only configuration that works all the time without reboots in between. IPv6 would typically work for a minute or two and then stop.

Net result: In the event Spectrum dies, AT&T takes over and I lose IPv6, but my computers don't have bad IPv6 addresses they are trying to route to. On a failback event, traffic moves back over to Spectrum and we get our IPv6 connectivity back.

So there it is. I read elsewhere that "not a good idea' or "not possible"... but with a little creativity you can certainly have it both ways.

You'll want to add the following configuration entry to your load balancing configuration, to call a script on transition.

```
set load-balance group G transition-script /config/scripts/transition
```

Use vi to create /config/scripts/transition and mark it executable with chmod /config/scripts/transition +x

```
#!/bin/vbash
run=/opt/vyatta/sbin/vyatta-cfg-cmd-wrapper
op=/opt/vyatta/bin/vyatta-op-cmd-wrapper
if sudo ping -c 10 -I eth0 8.8.8.8 &> /dev/null
then
 echo Reset Local and UNMS Traffic
 $run begin
 echo Disable LB
 $run set load-balance group G lb-local-metric-change disable
 $run commit sleep 5
 echo Enable LB
 $run set load-balance group G lb-local-metric-change enable
 $run commit
 $run end
 echo Spectrum Up, Renew DHCPV6
 $op release dhcpv6-pd interface eth0 sudo killall -9 dhcp6c
```

```
 $op renew dhcpv6-pd interface eth0
else
 echo Spectrum Outage, Pulling DHCPV6
 $op release dhcpv6-pd interface eth0 sudo killall -9 dhcp6c
fi
echo Flush Conntrack sudo conntrack -F
exit 0
```

Linked here is a video of this in action.
https://drive.google.com/file/d/1_hpMCMDTN5-W5nRZeqU8FOfjIJ2a4DX2/view

Started with normal conditions - both circuits operational. Spectrum primary, AT&T failover.

Left window, IPv6 ping to Google.com
Right window IPv4 ping to Google DNS @ 8.8.8.8

0:10 reset cable modem, forcing loss of Internet on the primary and only IPv6 circuit
0:23 cable connectivity lost
0:49 IPv4 traffic moves to AT&T circuit
1:09 IPv6 connectivity pulled via transition script, shutting down IPv6 connectivity/RA.
4:13 cable connectivity restored, IPv4 traffic starts flowing to Spectrum (note the response time change from 22ms to 10ish)
4:17 IPv6 connectivity restored via transition script.

Unique solution ID: #1092
Author: n/a
Last update: 2021-06-01 08:41