GF-SLB
A Method for Generalized Fast Server Load Balancing

Anton Kapela
tk@5ninesdata.com
What’s the point?

- Lots of Rack-sized workloads
- Large need for local LB
  - Global LB - solved problem (akdns, cachefly, $cdn_of_the_week)
- Want a simple architecture
  - Fewer moving parts (or gates)
  - Easy to understand solutions
GF-SLB

- **Generalized**
  - Concept applies to any IP datagram
    - TCP and UDP == good
  - Works on routers that support ECMP

- **Fast**
  - Happens in forwarding path
  - Got 10 gig ports?
    - You’ve got a 10 gig server load balancer
GF-SLB Ingredients

- IOS has all the goodies
  - RTR - Response Time Reporter (IP-SLA)
  - Track – Enhanced object tracking
  - ECMP – Equal-Cost Multipath
- Configured, you have GF-SLB
Benefits

- **Hosts do not** need routing protocols
- DSR has desirable side-effects
  - Got a /23 vhost? Alias em’ on lo0!
  - Minimizes Layer-2 Adjacencies
- **No connection tracking**
  - Stateless forwarding of data
  - Statefull end-system tracking
GF-SLB Configuration at a glance

ECMP – Static Routes

New L2 Dest Address

Public Internet

Server Network – 10.0.0.0/29

Direct Server Return Traffic

Eth0 & Lo0
Example Configuration - Recursive Name Server Scenario

- Interface to fictitious resolvers:

```plaintext
interface Vlan100
description resolver-net
ip address 10.0.0.1 255.255.255.248
```
Example Configuration - Recursive Name Server Scenario

- Response time reporter - “Health Check”

```plaintext
rtr 1
type dns target-addr are.you.alive name-server 10.0.0.2
timeout 2000
frequency 9
rtr schedule 1 life forever start-time now

rtr 2
type dns target-addr are.you.alive name-server 10.0.0.3
timeout 2000
frequency 9
rtr schedule 2 life forever start-time now
```
Example Conf - Recursive Name Server Scenario

- Tracking object configured per response time reporter

  track 1 rtr 1

  track 2 rtr 2
Example Conf - Recursive Name Server Scenario

- Static routes controlled by ‘track’
- Multiple routes create ECMP in FIB

```plaintext
ip route 10.10.10.1 255.255.255.255 10.0.0.2 track 1
ip route 10.10.10.2 255.255.255.255 10.0.0.2 track 1

ip route 10.10.10.1 255.255.255.255 10.0.0.3 track 2
ip route 10.10.10.2 255.255.255.255 10.0.0.3 track 2
```
Inspecting CEF Buckets – 2 Routes

output chain:
  loadinfo 021C9EF4, per-session, 2 choices, flags 0003, 5 locks
  flags: Per-session, for-rx-IPv4
  16 hash buckets
  < 0 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
  < 1 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
  < 2 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
  < 3 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
  < 4 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
  < 5 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
  < 6 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
  < 7 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
  < 8 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
  < 9 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
  <10 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
  <11 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
  <12 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
  <13 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
  <14 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
  <15 > IP adj out of Vlan100, addr 10.0.0.3 0248C320

- 8/8 Allocation
Inspecting CEF Buckets – 3 Routes

output chain:
loadinfo 021C9E34, per-session, 3 choices, flags 0003, 5 locks
flags: Per-session, for-rx-IPv4
15 hash buckets
< 0 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
< 1 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
< 2 > IP adj out of Vlan100, addr 10.0.0.4 0248C620
< 3 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
< 4 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
< 5 > IP adj out of Vlan100, addr 10.0.0.4 0248C620
< 6 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
< 7 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
< 8 > IP adj out of Vlan100, addr 10.0.0.4 0248C620
< 9 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
<10 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
<11 > IP adj out of Vlan100, addr 10.0.0.4 0248C620
<12 > IP adj out of Vlan100, addr 10.0.0.2 0248C4A0
<13 > IP adj out of Vlan100, addr 10.0.0.3 0248C320
<14 > IP adj out of Vlan100, addr 10.0.0.4 0248C620

- Auto-smart 5/5/5 Allocation - 12.2(37)SE on 35[56]0
Results
Pseudo-random source IP addr towards ns1 address only
Peak & average sharing within ~10% of balanced

Test Ave: 2.18 kbit
Test Peak: ~46kbit

Test Ave: 1.97 kbit
Test Peak: ~42kbit
Caveats, etc

- 16 max equal paths today
- No connection tracking
  - No statistics available either
  - Netflow? (yuck)
- No target persistence guaranteed
- No least-load consideration
- No L3/L4 NAT/PAT support
- IP src + dst hash by default
  - L4 hash feature in 12.4t and SX/SR
Summary

1. Configure GF-SLB
2. ???
3. #sh ip profit