Load Balancing with nftables II

by Laura García (Zen Load Balancer Team)
Netdev 1.2

1. Development Evolution
2. Use Cases Review
3. Benchmarks
4. Work To Do

Development Evolution

Development Evolution

New expressions available

➔ **nft_numgen**: number generator with two modes.

- Incremental
- Random

Ability to scale the values and add an offset

➔ **nft_hash**: Hash any selector concatenation, one mode.

- Jenkins

nft_numgen expression (based on xt_statistics)

➔ Incremental counter for round robin scheduler.

ip daddr <vip> tcp dport <vport> dnat to **numgen inc mod 2** map { 0 : <ipaddr0>, 1 : <ipaddr1> }

meta mark set **numgen inc mod 3 offset 100**

(100, 101, 102, 100, ...)

➔ Random generation for weight scheduler.

ip daddr <vip> tcp dport <vport> dnat to **numgen random mod 2** \

map { 0 : <ipaddr0>, 1 : <ipaddr1> }

meta mark set **numgen random mod 3 offset 100**

(100-102)

nft_hash expression

➔ Hash function for persistence.

ip daddr <vip> tcp dport <vport> dnat to **jhash ip saddr mod 2** \
map { 0: <ipaddr0>, 1: <ipaddr1> }
meta mark set **jhash ip saddr mod 3 seed 0xabcd offset 100**
(100-102)

Requirements:

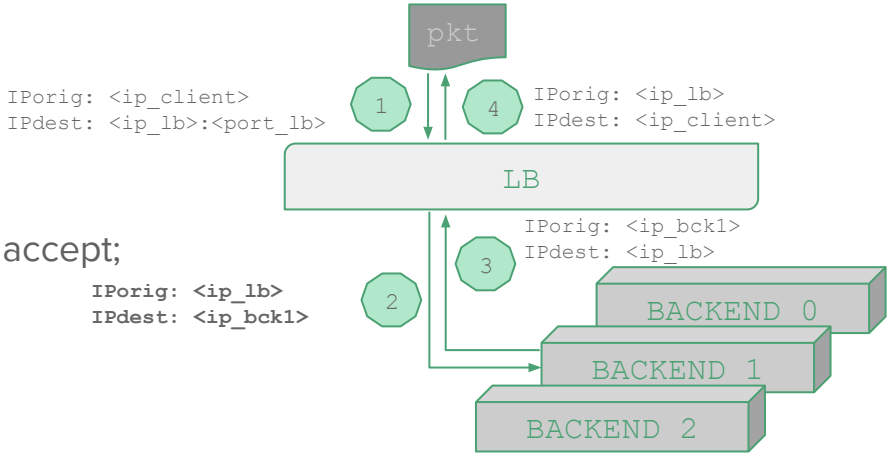
- ★ kernel >= 4.8.0-rc4+ (nf-next branch)
- ★ libnftnl > 1.0.6
- ★ nftables > 0.7 (not yet released)

Use Cases Review

(The definitive syntax)

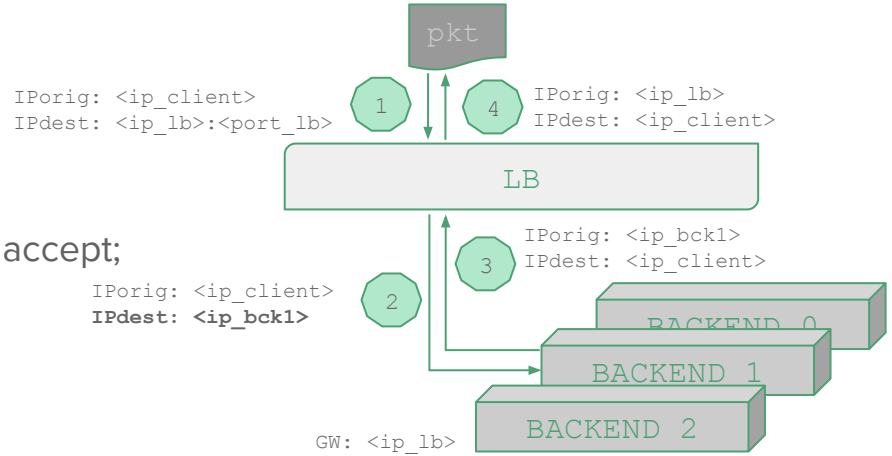
sNAT Topology

```
table ip nat {
        chain prerouting {
                type nat hook prerouting priority 0; policy accept;
                ip daddr <ip_lb> tcp dport <port_lb> dnat to numgen inc mod 3 map { \
                        0 : <ip_bck0>, \
                        1 : <ip_bck1>, \
                        2 : <ip_bck2> }
        }
        chain postrouting {
                type nat hook postrouting priority 100; policy accept;
                masquerade
        }
}
```



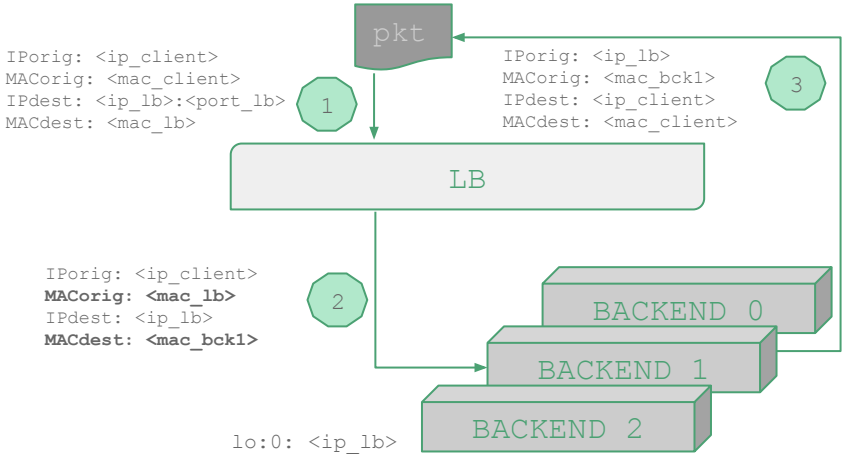
dNAT Topology

```
table ip nat {
        chain prerouting {
                type nat hook prerouting priority 0; policy accept;
                ip daddr <ip_lb> tcp dport <port_lb> dnat to numgen random mod 3 map { \
                        0 : <ip_bck0>, \
                        1 : <ip_bck1>, \
                        2 : <ip_bck2> }
        }
        chain postrouting {
                type nat hook postrouting priority 100; policy accept;
        }
}
```



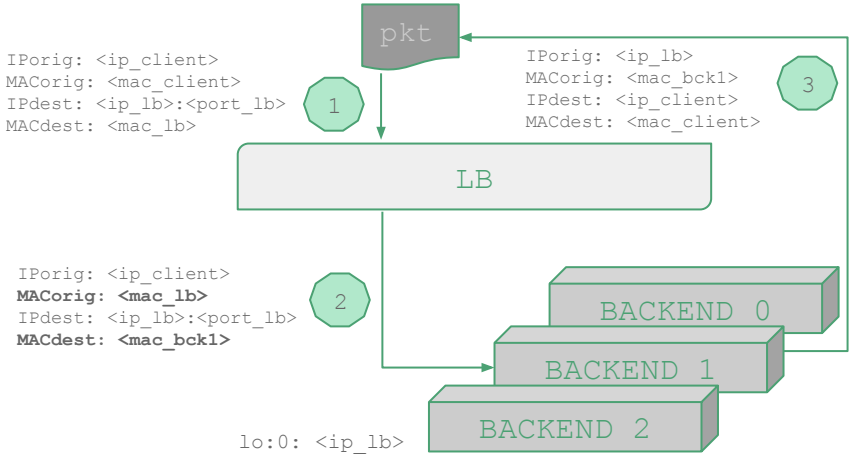
DSR Topology (non connection oriented)

```
table netdev filter {
        chain ingress {
                type filter hook ingress device <if_lb> priority 0; policy accept;
                ip daddr <ip_lb> udp dport <port_lb> ether saddr set <mac_lb> \
                        ether daddr set numgen inc mod 3 \
                        map { \
                                0: <mac_bck0>, \
                                1: <mac_bck1>, \
                                2: <mac_bck2> } \
                        fwd to <if_lb>
        }
}
```



DSR Topology (connection oriented)

```
table netdev filter {
        chain ingress {
                type filter hook ingress device <if_lb> priority 0; policy accept;
                ip daddr <ip_lb> tcp dport <port_lb> ether saddr set <mac_lb> \
                        ether daddr set jhash ip saddr . tcp sport mod 3 seed 0xabcd \
                        map { \
                                0: <mac_bck0>, \
                                1: <mac_bck1>, \
                                2: <mac_bck2> } \
                        fwd to <if_lb>
        }
}
```



Benchmarks

Lab Environment

Kernel version 4.8.0-rc4+ branch nf-next

2 clients, 3 backends & 1 LB

2 cores (3.33 GHz each) i5 660 with 2 threads/core, 4GB RAM @1333 MHz

2 Intel Gigabit Network 82578DM & 82574L per machine

System tuning considerations from József paper

HTTP protocol transferring 229 bytes per connection (client wrk/server nginx)

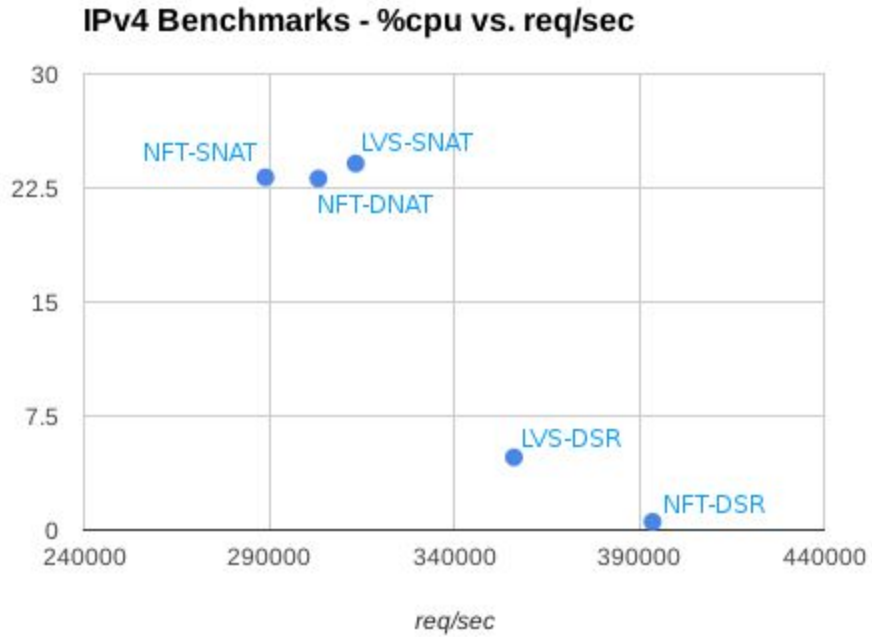
Both IPv4 & IPv6

LB was never saturated during a test of 30 seconds

LVS performance used as a reference

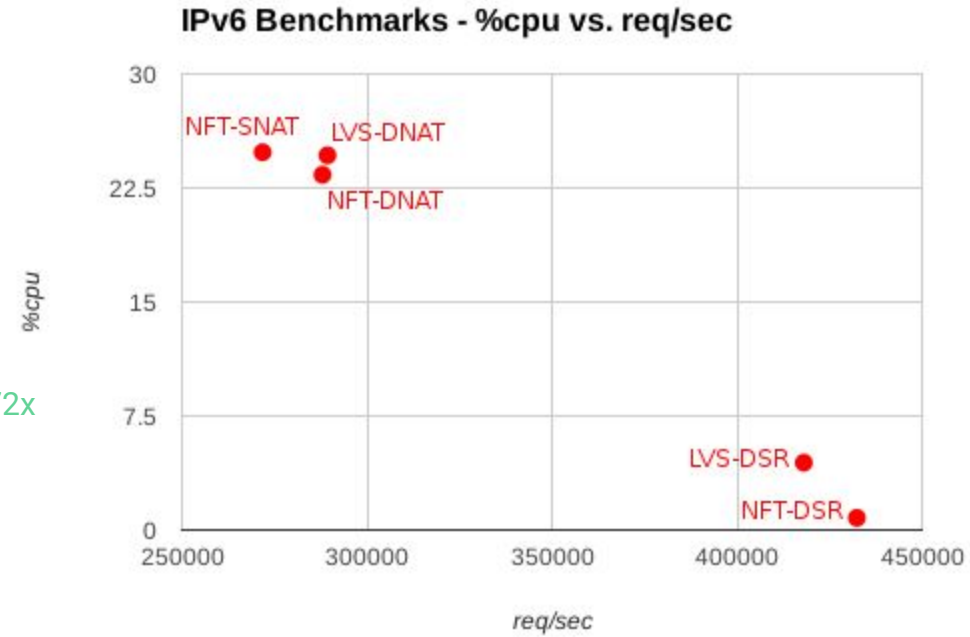
IPv4 Benchmarks

method	req/sec	%cpu	
LVS-SNAT	313427.91	24.11	
NFT-SNAT	289035.54	23.2	
NFT-DNAT	303356.59	23.12	
LVS-DSR	356212.05	4.78	
NFT-DSR	393672.35	0.54	+9.78x



IPv6 Benchmarks

method	req/sec	%cpu	
LVS-DNAT	289320.17	24.65	
NFT-SNAT	271790.98	24.85	
NFT-DNAT	287978.41	23.37	
LVS-DSR	418067.65	4.43	
NFT-DSR	432399.38	0.8	+5.72x



Work To Do

Work To Do

- ★ Lightweight NAT from hook ingress to improve NAT results
- ★ User space nft rules manager:
  - Set basic and complex algorithms
  - Manage different topologies easily
- ★ Health checks monitor
  - Layered support
  - Internal and external monitor

Thank you to:
Outreachy Program & Pablo Neira

Load Balancing with nftables II

Laura García (Zen Load Balancer Team)
lauragl@sofintel.net