Research and Development

A book of interesting technical things that RackCorp works on from time to time

Performance Measurements of Linux, DanOS, VYOS, VPP, and Linux XDP at 100GE

> Tests still being performed - VPP Still as yet untested

Results:

Table below represent Millions of Packets Per Second (MPPS) send for forwarding via the router software vs packetloss of the end destination of expected packets.
Note:
- droptest % represents the % of loss of the legitimate packets
- some higher rates not tested once significant loss was demonstrated at lower levels
- no optimisations performed on these routers unless otherwise noted below

Single traffic flow Test, 50 firewall policies
- Single destination IP, single protocol, same src/dst port
- 50 Firewall Policies
- % indicates forwarding packetloss
- VYOSXDP is not running any firewall policies as firewall not supported
- LinXDP running custom firewall + 802.1Q

MPPS	0.45	0.75	1.5	3	4.5	6	7.5	9	10	12	15	18
Danos	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	13.7%	31.1%	40.3%			
LinXDP	0.0%	0.0%	0.0%	0.5%	28.2%	46.2%	57.3%	65.1%	70.1%	74.4%	80.2%	
VYOS	21.7%	53.1%	63.9%	88.3%	92.2%	94.1%						
VYOS XDP *	0.0%	0.0%	0.0%	2.7%	35.5%	51.7%	61.7%	68.1%	73.0%	76.3%	81.5%	

GRE Test, multiple flows, GRE encapsulations received on other end, fragmentation off
- Random Destination IPs within single /24, multiple src/dst ports, all packets GRE encap

- 50 Firewall policies
- % indicates forwarding packetloss
- LinXDP running custom firewall / GRE tunnel code + 802.1Q

MPPS	0.45	0.75	1.5	3	4.5	6	7.5	9	10	12	15	18
Danos	0.0%	0.0%	0.0%	0.0%	10.8%	34.3%	47.8%	56.7%	63.1%			
Linux	0.0%	0.0%	0.0%	34.7%	55.7%	66.5%	73.5%	77.7%				
LinXDP	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.2%	10.1%	30.9%	42.3%
VYOS	0.0%	0.0%	0.0%	7.2%	36.7%	51.3%						

900K Route Test, multiple flows
- Random Destination IPs within single /24, multiple src/dst ports
- 50 Firewall Policies
- % indicates forwarding packetloss
- LinXDP running custom firewall + 802.1Q
- VYOSXDP is not running any firewall policies as firewall not supported
- 900k routes loaded into routing table

MPPS	3	4.5	6	7.5	9	10	12	15	18	20	30
Danos	0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	15.1%	27.0%	49.2%
LinXDP	0%	0.0%	0.0%	0.0%	0.1%	12.7%	24.9%	30.0%	43.4%	59.1%	65.1%
VYOS	0%	9.2%	28.9%	43.2%							
VYOSXDP**	0%	0.0%	0.0%	0.0%	0.0%	0.1%	21.0%	28.9%	42.6%	48.8%	64.5%

DDoS Drop Test (50% traffic dropped)
- Random Destination IPs within single /24, UDP traffic, multiple src/dst ports
- 50 Firewall Policies
- % indicates forwarding packetloss of packets that were not supposed to be dropped
- LinuxXDP running custom firewall + 802.1Q
- VYOSXDP not tested because it has no firewall capability
- 900k routes loaded into routing table

MPPS	3	4.5	6	7.5	9	10	12	15	18	20	30
Danos	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.6%	22.5%	47.2%
LinuxXDP	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	30.4%	37.2%	45.1%	65.6%
VYOS	0.0%	8.7%	26.7%	39.5%							

Test Environment & pktgen tool

Network Card:

mlx5_core 0000:3b:00.1: firmware version: 16.27.6120
mlx5_core 0000:3b:00.1: 126.016 Gb/s available PCIe bandwidth, limited by 8.0 GT/s PCIe x16 link at 0000:3a:00.0 (capable of 252.048 Gb/s with 16.0 GT/s PCIe x16 link)

We wont go into building pktgen as there's plenty of doco out there on this. Just for reference purposes on how we ran pktgen:

```
LD_LIBRARY_PATH=/usr/local/lib64/ /root/pktgen-dpdk/usr/local/bin/pktgen -l 2,4,6 -n 2 -a
3b:00.1 -d librte_net_mlx5.so  -- -p 0x1 -P -m "[4:6].0"
```

Traffic generated:
Static destination MAC (The test Target)
pktgen:

```
set 0 rate 10   (This is % of 100GE, adjusted accordingly at 1% = 1.5MPPS)
set 0 size 64
set 0 count 50000000
set 0 proto udp
set 0 dst ip 10.22.23.102
set 0 src ip 10.22.22.101/24
set 0 dst mac XX:XX:XX:XX:d1:7b
set 0 src mac XX:XX:XX:XX:36:75
set 0 type ipv4
```

Single flow:
Single target IP address
UDP traffic, 64 bytes per packet, same src/dst ports
No firewall policies

Single flow, 5 firewall policies:
Single target IP address
UDP traffic, 64 bytes per packet, same src/dst ports
5 firewall policies

Single flow, 50 firewall policies:
Single target IP address
UDP traffic, 64 bytes per packet, same src/dst ports
50 firewall policies

Single flow, GRE tunnel:
Single target IP address
UDP traffic, 64 bytes per packet, same src/dst ports
50 firewall policies
GRE encap traffic and forward to static destination

Multiple flow:
254 destination IP addresses (multi-flow)
UDP traffic, 64 bytes per packet, random src ports (multi-flow)
50 firewall policies
pktgen:

```
range 0 src port 53 53 1000 1
range 0 dst ip 10.22.23.1 10.22.23.1 10.22.23.254 0.0.0.1
range 0 src ip 10.22.22.101 10.22.22.101 10.22.22.101 0.0.0.0
range 0 src mac XX:XX:XX:XX:36:75 XX:XX:XX:XX:36:75 XX:XX:XX:XX:36:75 00:00:00:00:00:00
range 0 dst mac XX:XX:XX:XX:d1:7b XX:XX:XX:XX:d1:7b XX:XX:XX:XX:d1:7b 00:00:00:00:00:00
enable 0 range
```

900K Route Test:
254 destination IP addresses
UDP traffic, 64 bytes per packet, random src/dst ports
50 firewall policies
900K loaded route table

Drop Test:
1 destination IP addresses, multiple ports (multi-flow)
UDP traffic, 64 bytes per packet, random src/dst ports
50 firewall policies, default deny
900K loaded route table
Half test traffic configured to be dropped

DanOS

Version: 2105
Built on: Fri Jun 11 11:58:32 UTC 2021
HW Model: PowerEdge R440
CPU: Intel Xeon Silver 4210R CPU @ 2.4Ghz

Routing Configuration:
set protocols static arp 10.22.22.101 hwaddr 'XX:XX:XX:XX:XX:XX'
set protocols static arp 10.22.22.101 interface dp0p59s0f1
set protocols static route 10.22.23.0/24 next-hop 10.22.22.101

Firewall policies:
set security ip-packet-filter group ipv4 ip-version ipv4
set security ip-packet-filter group ipv4 rule 1 action drop
set security ip-packet-filter group ipv4 rule 1 match source ipv4 host 1.1.1.1
set security ip-packet-filter group ipv4 rule 2 action drop
set security ip-packet-filter group ipv4 rule 2 match source ipv4 host 1.1.2.1
set security ip-packet-filter group ipv4 rule 3 action drop
set security ip-packet-filter group ipv4 rule 3 match source ipv4 host 1.1.3.1
...etc...
set security ip-packet-filter interface dp0p59s0f1 in ipv4

For GRE test:
set interfaces tunnel tun0 address 10.90.4.102/24
set interfaces tunnel tun0 encapsulation gre
set interfaces tunnel tun0 local-ip 10.22.22.102
set interfaces tunnel tun0 remote-ip 10.22.22.101
set protocols static route 10.22.23.0/24 next-hop 10.90.4.101

VyOS

$ show system cpu
CPU Vendor: GenuineIntel
Model: Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz
Total CPUs: 1,3,5,7,9,11,13,15,17,19
Sockets: 2
Cores: 10
Threads: 1
Current MHz: 1000.128

$ show system memory
Total: 62.54 GB
Free: 61.17 GB
Used: 1.38 GB

$ show version

Version: VyOS 1.4-rolling-202203150317
Release train: sagitta

Built by: autobuild@vyos.net
Built on: Tue 15 Mar 2022 03:17 UTC
Build UUID: 9da98191-be0b-42e1-937a-97fb016b22ac
Build commit ID: f2655e2ae72e8c

Architecture: x86_64
Boot via: installed image
System type: bare metal

Hardware vendor: Dell Inc.
Hardware model: PowerEdge R440
Hardware S/N: XXXXXXXXX
Hardware UUID: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Routing Configuration:
set protocols static arp 10.22.22.101 hwaddr 'XX:XX:XX:XX:XX:XX'
set protocols static route 10.22.23.0/24 next-hop 10.22.22.101

Firewall Policies:
set firewall name TEST_IN rule 1 action 'drop'
set firewall name TEST_IN rule 1 destination address 1.1.1.1
set firewall name TEST_IN rule 2 action 'drop'
set firewall name TEST_IN rule 2 destination address 1.1.2.1
set firewall name TEST_IN rule 3 action 'drop'
set firewall name TEST_IN rule 3 destination address 1.1.3.1
set firewall name TEST_IN rule 4 action 'drop'
set firewall name TEST_IN rule 4 destination address 1.1.4.1
set firewall name TEST_IN rule 5 action 'drop'
set firewall name TEST_IN rule 5 destination address 1.1.5.1
..... etc .....
set interfaces ethernet eth2 firewall in name TEST_IN

For GRE test:
set interfaces tunnel tun0 address '10.90.4.102/24'
set interfaces tunnel tun0 encapsulation 'gre'
set interfaces tunnel tun0 remote '10.22.22.101'
set interfaces tunnel tun0 source-address '10.22.22.102'
set protocols static route 10.22.23.0/24 next-hop 10.90.4.101