



Customer Story

IPng Networks AS8298

Pim van Pelt <pim@ipng.ch> • 2022-06-10 • Baur au Lac, Zurich



Introduction



Pim van Pelt

Pim van Pelt (PBVP1-RIPE)

- Member of the RIPE community since 1999 (RIPE #34)
 - Has used pim@ipng.nl for 22 years
 - And also pim@ipng.ch for 15 years
 - Incorporated ipng.ch in Switzerland in 2021





Introduction

SixXS

IPv6 Deployment & Tunnel Broker



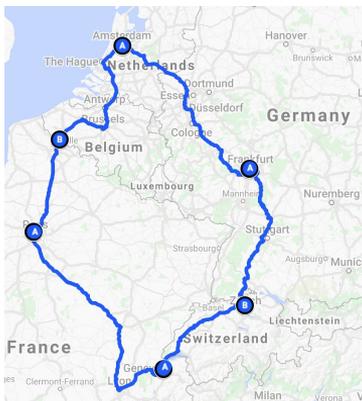
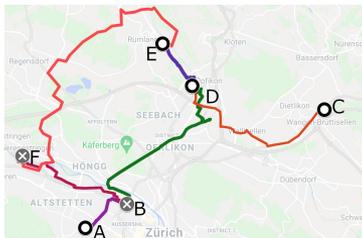
[Enter the SixXS website](#)

SixXS - a global IPv6 Tunnelbroker

- Member of the RIPE community since 1999 (RIPE #34)
 - Founded SixXS w/ Jeroen Massar [[ref](#)]
 - Operated SixXS from ca. 1999 - 2017 [[sunset](#)]
 - Used **AS8298** for Ghost Route Hunter (2003) [[RIPE #44](#)]
-



Introduction



IPng Networks GmbH

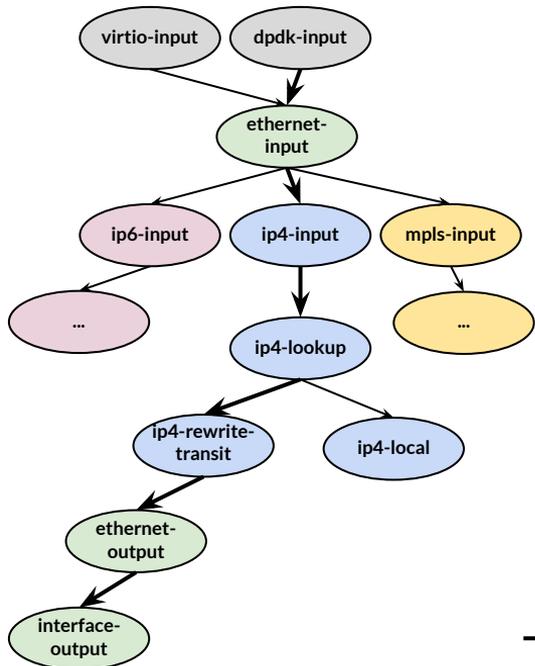
- Developer of Software Routers - VPP and DPDK [[ref](#)]
- Tiny operator from Brüttisellen (ZH), Switzerland [[ref](#)]
- Twelve VPP/Bird2 routers [[ref](#)] (UN/LOCODE names)
- European ring: *peering on the FLAP** [[ref](#)] ~1850 adjacencies
- Acquired AS8298 from SixXS [[ref](#)]



Vector Packet Processing

VPP reads a **vector** of up to 256 packets from its interfaces:

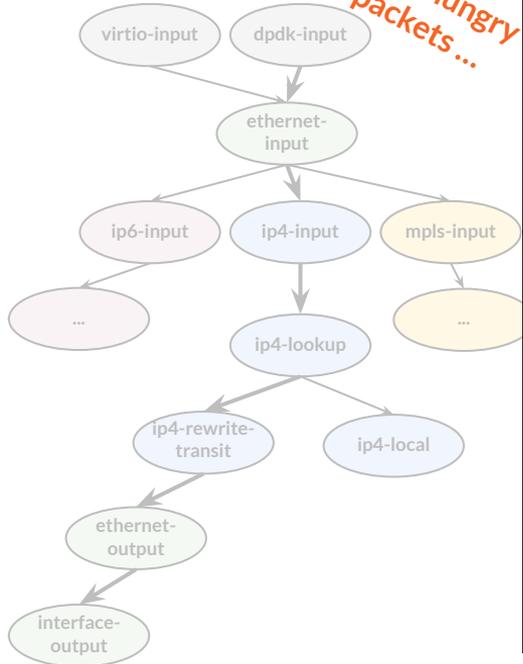
1. Packets are prefetched (or directly written) into CPU **d-cache**
2. All packets go through a directed graph
 - a. First packet: graph node's code loaded into CPU **i-cache**
 - b. All additional packets: fully in d/i-cache: 7-20x faster
3. Packets then traverse *as a vector* into the next node(s)
 - a. Optimized with SIMD (SSE, AVX, AVX512, ...)
 - b. No context switches, good TLB hit rate due to hugepages
 - c. Lockless: multi-threading gives linear scaling
4. Hardware offload: use silicon if available
5. Plugins: rearrange the graph nodes and add functionality





Vector Packet Processing - example

Always hungry
for packets ...



```
pim@hippo:~$ vppctl
vpp# set interface state TenGigabitEthernet3/0/0 up
vpp# set interface mtu packet 9000 TenGigabitEthernet3/0/0
vpp# set interface ip address TenGigabitEthernet3/0/0 2001:db8:0:1::2/64
vpp# set interface ip address TenGigabitEthernet3/0/0 192.0.2.2/24
vpp# ip route add 2000::/3 via 2001:db8:0:1::1
vpp# ip route add 0.0.0.0/0 via 192.0.2.1
```

```
pim@hippo:~$ vppctl show interface TenGigabitEthernet3/0/0
```

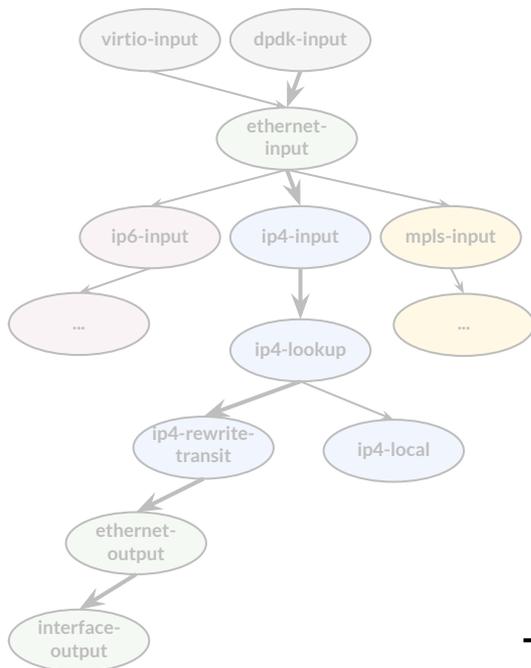
TenGigabitEthernet3/0/0	up	9000/0/0/0	rx packets	5969930253
			rx bytes	2139798549228
			tx packets	14517083897
			tx bytes	6864831067486
			drops	945
			ip4	3862409855
			ip6	2107502378



VPP: Linux Controlplane

Wrote a VPP Plugin [[github](#)] that:

1. Creates tun/tap interface in Linux for a given VPP interface
 - a. VPP->Linux: Traffic to **ip4-local** and **ip6-local** is punted into TAP
 - b. Linux->VPP: packets into TAP are inserted into **virtio-input**
2. Syncs interface changes in VPP into Linux
3. Listens to **Netlink messages** and syncs Linux changes into VPP
4. Allows operators to use VPP almost exactly as if it were Linux
 - Configure interfaces, addresses, routes by hand, or ...
 - ...using common tools like ip(1), FRR, or BIRD/BIRD2



⇒ VPP is Linux's *software equivalent* of an ASIC dataplane ⇐

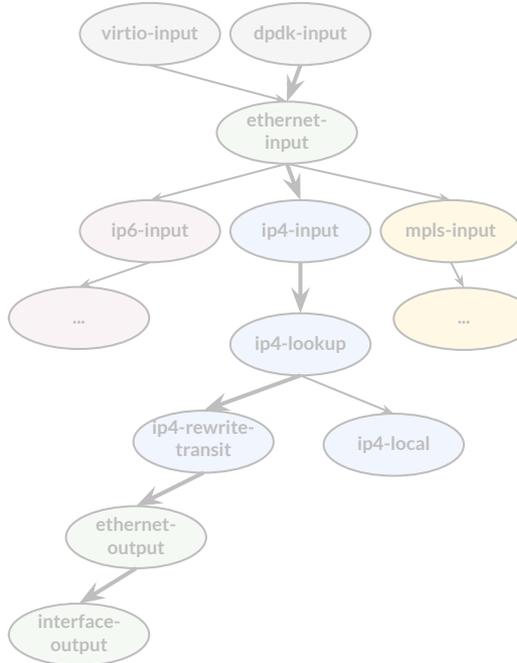


VPP: Linux Controlplane - ip

```
pim@hippo:~$ vppctl lcp create TenGigabitEthernet3/0/0 host-if xe0
pim@hippo:~$ sudo ip link set xe0 up mtu 9000
pim@hippo:~$ sudo ip address add 2001:db8:0:1::2/64 dev xe0
pim@hippo:~$ sudo ip address add 192.0.2.2/24 dev xe0

pim@hippo:~$ sudo ip link add link xe0 name servers type vlan id 101
pim@hippo:~$ sudo ip link set servers mtu 1500 up
pim@hippo:~$ sudo ip addr add 2001:678:d78:3::86/64 dev servers
pim@hippo:~$ sudo ip addr add 194.1.163.86/27 dev servers
pim@hippo:~$ sudo ip route add default via 2001:678:d78:3::1
pim@hippo:~$ sudo ip route add default via 194.1.163.65

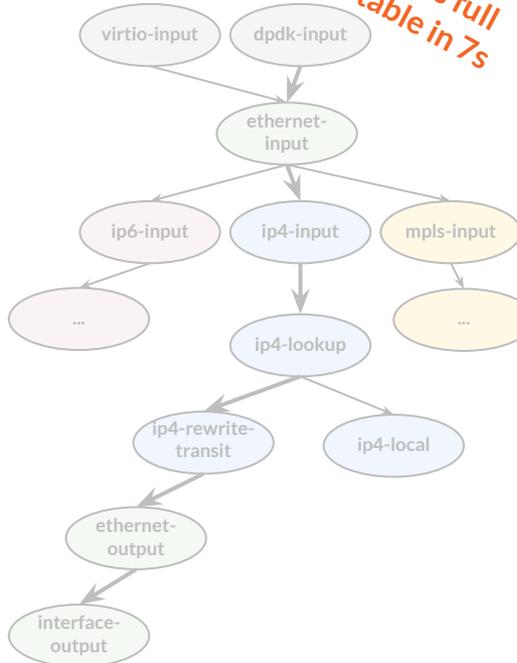
pim@hippo:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=121 time=1.348 ms
...
```





VPP: Linux Controlplane - Bird2

Converges full BGP table in 7s



```
pin@defra0:~$ birdc show route count
```

```
BIRD 2.0.7 ready.
```

```
8018726 of 8018726 routes for 907513 networks in table master4
```

```
1467252 of 1467252 routes for 144687 networks in table master6
```

```
523590 of 523590 routes for 261795 networks in table t_roa4
```

```
105464 of 105464 routes for 52732 networks in table t_roa6
```

```
Total: 10115032 of 10115032 routes for 1366727 networks in 4 tables
```

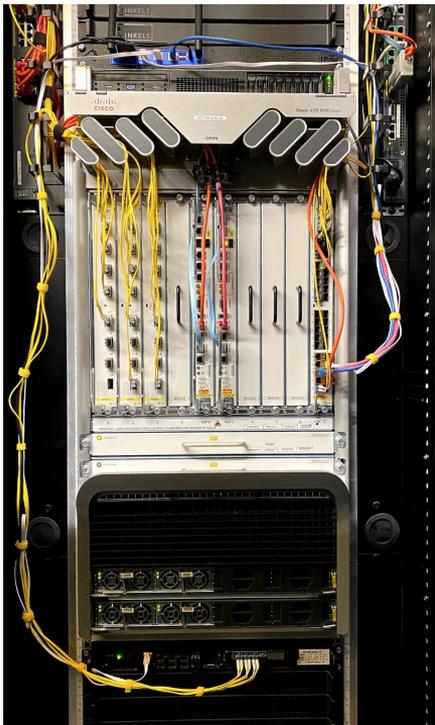
```
pin@defra0:~$ birdc show ospf neighbor ospf6
```

```
BIRD 2.0.7 ready.
```

```
ospf6:
```

Router ID	Pri	State	DTime	Interface	Router IP
194.1.163.7	1	Full/PtP	30.715	xe1-3.100	fe80::6a05:caff:fe32:3e49
194.1.163.34	1	Full/PtP	35.056	xe1-2.200	fe80::6a05:caff:fe32:45ae
194.1.163.140	1	Full/DR	37.944	xe1-1.2006	fe80::5054:ff:feb0:442c

Deploying: Partnership with IP-Max



IP-Max:

- 30G DWDM Zurich-Frankfurt: three separate carriers/paths
- 10G DWDM Frankfurt-Amsterdam: NIKHEF
- 30G of LAG capacity towards DE-CIX
 - Reseller of all DE-CIX exchange points

IPng Networks:

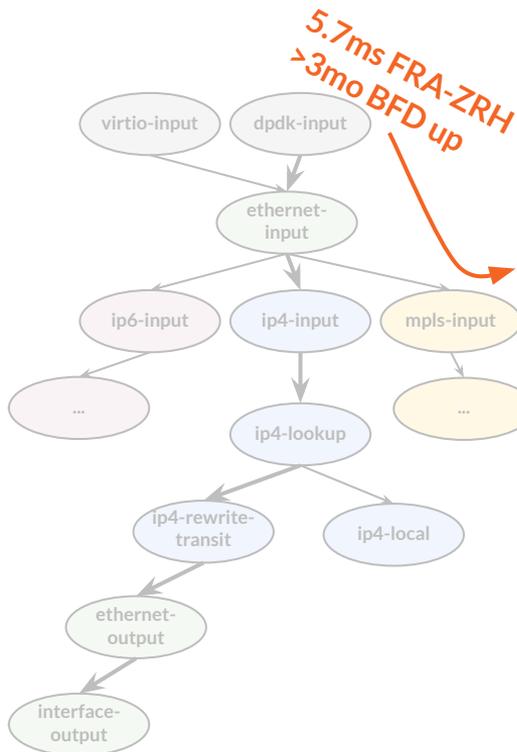
- Te0/0/0/4 – EoMPLS to NTT/eShelter Rümlang (chrma0.ipng.ch)
- Te0/1/0/4 – EoMPLS to Interxion Glattbrugg (chgtg0.ipng.ch)
- Te0/2/0/4 – EoMPLS to NIKHEF Amsterdam (nlams0.ipng.ch)

DE-CIX Peering: 972 ASNs, 238K IPv4, 44K IPv6

- DE-CIX Frankfurt (970 ASNs, peer with **810**) ⇒ 156K IPv4 37K IPv6
- DE-CIX Munich (176 ASNs present, peer with **154**) ⇒ 62K IPv4 3K IPv6
- DE-CIX Dusseldorf (208 ASNs present, peer with **183**) ⇒ 19K IPv4 3K IPv6
- DE-CIX Hamburg (170 ASNs present, peer with **152**) ⇒ 1450 IPv4 215 IPv6



IP-Max: Stable L2 backhaul to DE-CIX



```
pim@chrma0:~$ date
Thu Jun 9 23:08:50 CEST 2022
```

```
pim@chrma0:~$ birdc show bfd sessions
```

IP address	Interface	State	Since	Interval	Timeout
194.1.163.9	xe0-0	Up	2022-04-20 12:35:57	0.100	3.000
fe80::3eec:efff:fe6a:8052	xe0-0	Up	2022-04-20 12:35:57	0.100	3.000
194.1.163.25	xe1-0.812	Up	2022-03-02 01:35:54	0.100	3.000
fe80::6a05:caff:fe32:3e48	xe1-0.812	Up	2022-03-02 01:35:54	0.100	3.000
194.1.163.16	xe1-0.400	Up	2022-04-21 02:36:44	0.100	3.000
fe80::3eec:efff:fe46:69b4	xe1-0.400	Up	2022-04-21 02:36:44	0.100	3.000

```
pim@chrma0:~$ ping defra0
```

```
PING defra0(defra0.ipng.ch (2001:678:d78::7)) 56 data bytes
```

```
...
```

```
--- defra0 ping statistics ---
```

```
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 5.702/5.744/5.786/0.026 ms
```



LibreNMS

VPP: SNMP and NMS

1. Wrote an SNMP Agent [[github](#)]
2. Added *logo* to LibreNMS [[ref](#)]
3. Added *distro* to LibreNMS Agent [[ref](#)]

Is forwarding
18Gbps



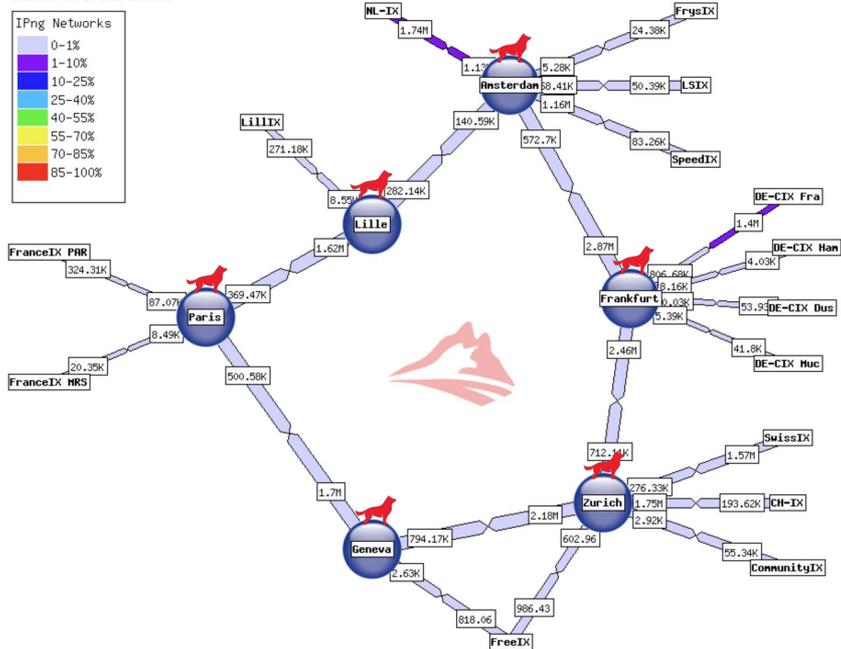
The screenshot displays the LibreNMS interface for a device named 'frggh0'. The top navigation bar includes links for Overview, Devices, Services, Ports, Health, Routing, and Alerts. The device information section shows the system name 'frggh0', IP address '194.1.163.34', and location 'Rue des Saules, 59262 Sainghin en Melantois, France'. Below this, there are three progress bars for Storage Usage, Memory Usage, and Processor Usage. The main content area is divided into several sections: 'Linux frggh0 5.4.0-81-generic #91-Ubuntu SMP Thu Jul 15 19:09:17 UTC 2021 x86_64' with a table of system details; 'Device Group Membership' showing the device is part of 'Routers (BGP)' and 'Routers (OSPF)'; 'Overall Traffic' with a line chart showing network traffic over time, with a peak of 18 Gbps; 'Processors' with a bar chart showing CPU usage for Intel Xeon D-1518 @ 2.20GHz x8, currently at 39%; 'Memory' with a bar chart showing memory usage and a table of memory types: Physical memory (24% / 31%), Virtual memory (25%), Memory buffers (1%), Cached memory (6%), Shared memory (0%), and Swap space (0%); and 'Storage' at the bottom.

System Name	frggh0
Resolved IP	194.1.163.34
Hardware	Supermicro SYS-5018D-FN8T
Operating System	Linux 5.4.0-81-generic (VPP Ubuntu 20.04)
Serial	WM208S007439_E222322X1502206
Object ID	.1.3.6.1.4.1.8072.3.2.10
Contact	noc@ipng.ch
Device Added	107 days 4 hours 34 minutes 17 seconds ago
Last Discovered	2 hours 6 minutes 55 seconds ago
Uptime	8 days 4 hours 38 minutes 22 seconds
Location	Rue des Saules, 59262 Sainghin en Melantois, France
Lat / Lng	N/A

Memory Type	Usage	Limit
Physical memory	24%	31%
Virtual memory	25%	
Memory buffers	1%	
Cached memory	6%	
Shared memory	0%	
Swap space	0%	

Questions, Discussion

Created: Oct 16 2021 21:30:02



If you peer with IPng Networks, thanks!
If you don't: please peer with AS8298
<peering@ipng.ch>

Useful Resources

- VPP:
- VPP Linux CP:
- Articles:
- Twitter:

fd.io

[Github](https://github.com)

ipng.ch

[@IPngNetworks](https://twitter.com/IPngNetworks)

Also: thanks for listening!