MPLS in VPP

Using Linux Control Plane

Pim van Pelt <pim@ipng.ch> • 2023-09-26 • NLNOG
Thanks to: Adrian Pistol <vifino@posteo.net>

nlnog

netherlands network operator group

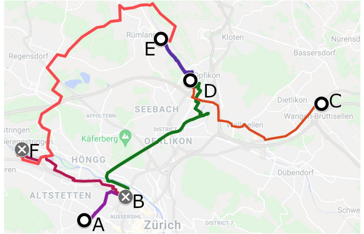
Act 1: Introductions

Intro: Pim van Pelt (PBVP1-RIPE)

Pim van Pelt

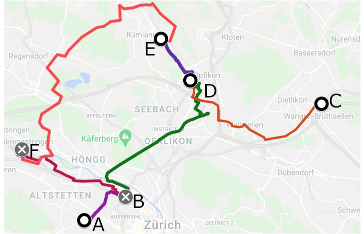
- Member of the RIPE community since 1999 (RIPE #34)
  - Has used pim@ipng.nl for 24 years
  - And also pim@ipng.ch for 17 years
  - Incorporated ipng.ch in Switzerland in 2021

Intro: IPng Networks GmbH

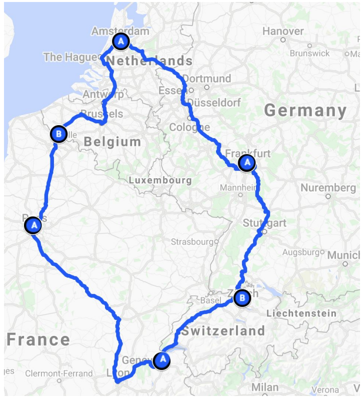


- Developer of Software Routers - VPP and DPDK [ref]
- Tiny operator from Brüttisellen (ZH), Switzerland [ref]

Intro: IPng Networks GmbH

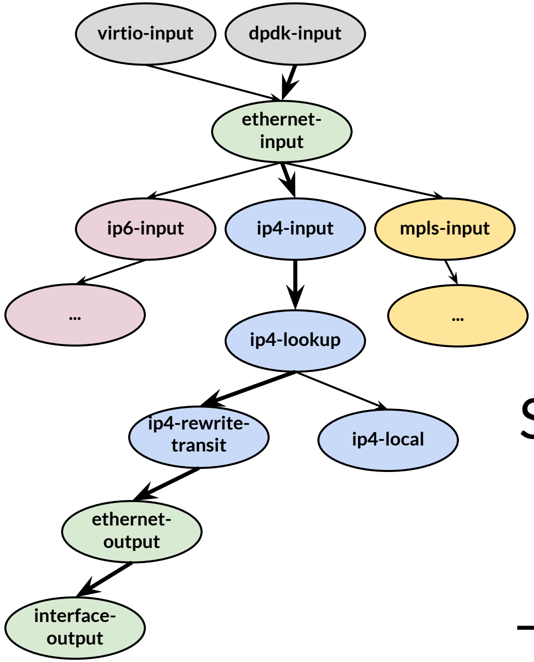


- Developer of Software Routers - VPP and DPDK [ref]
- Tiny operator from Brüttisellen (ZH), Switzerland [ref]



- Twelve VPP/Bird2 routers [ref] (UN/LOCODE names)
- European ring: *peering on the FLAP** [ref] ~1950 adjacencies
- Acquired AS8298 from SixXS [ref]

Intro: Vector Packet Processing



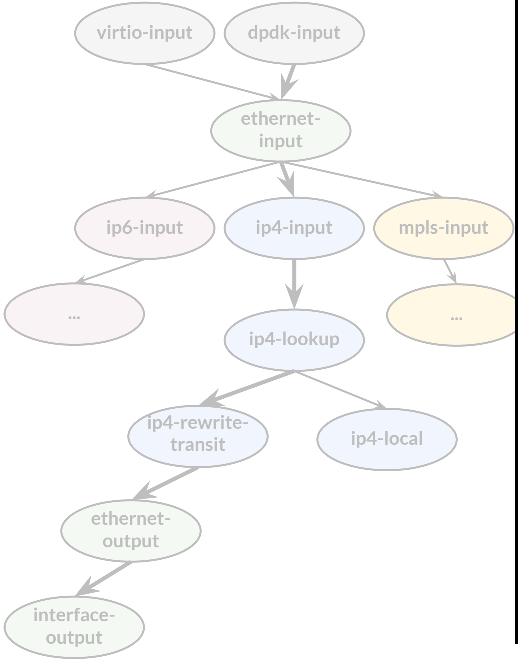
VPP is an open source router that can:

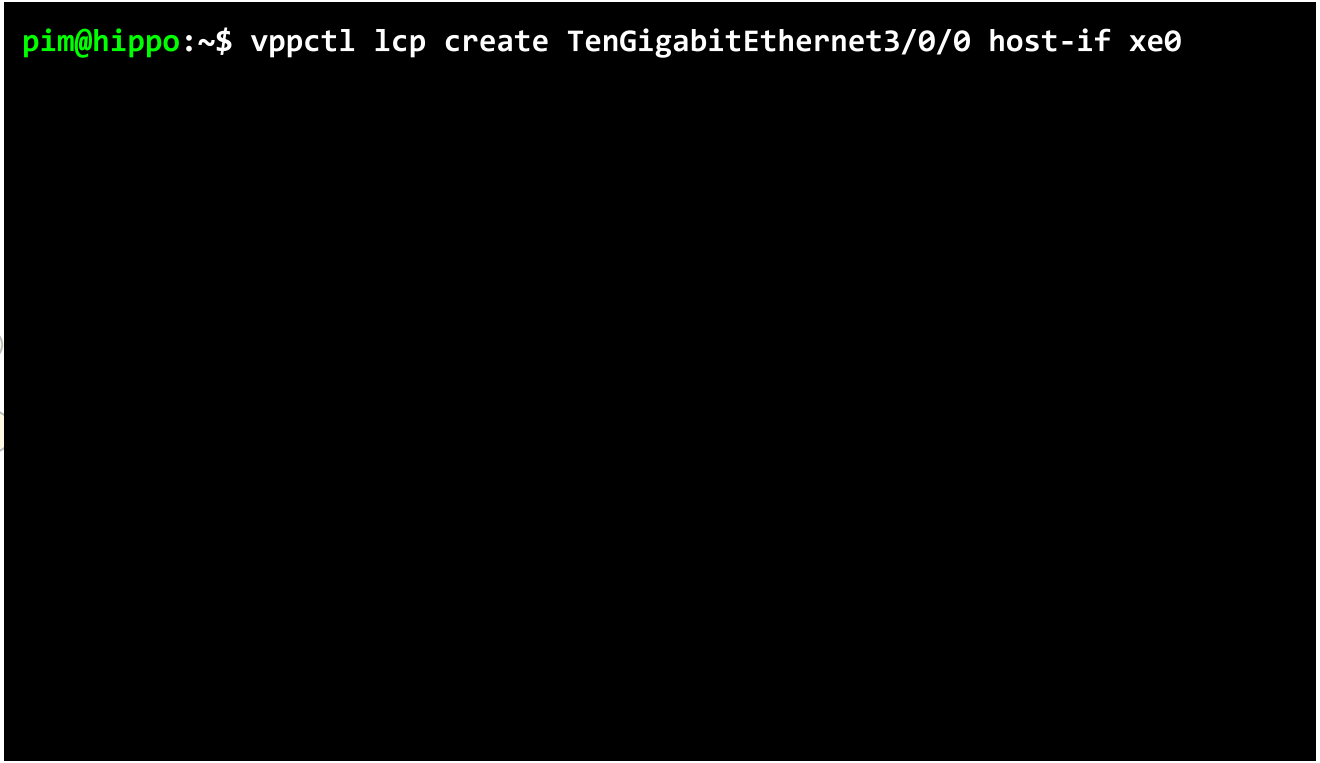
- provide *very* fast networking dataplane
- using DPDK, RDMA, VirtIO, VMXNet3, AVF, ...
- easily exceeds 150Mpps+ and 180Gbps+
- on commodity AMD64 hardware!

See SwiNOG #37 [video] or DENOG #14 [video]

- **Linux Control Plane** plugin [github]
- adds BGP/OSPF/VRRP/etc to VPP

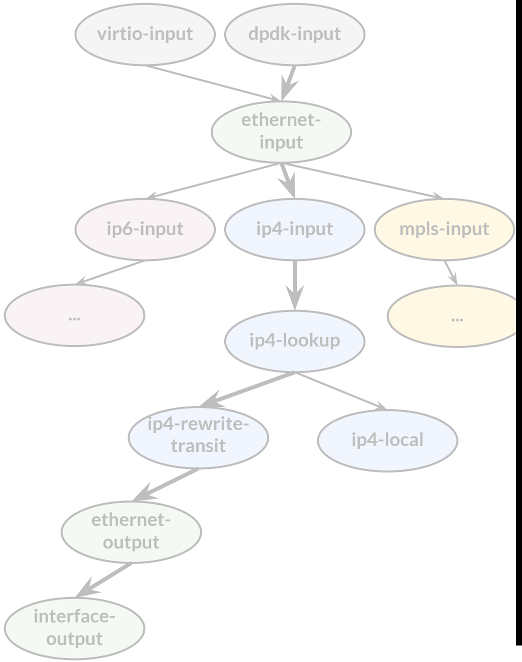
Intro: VPP LinuxCP

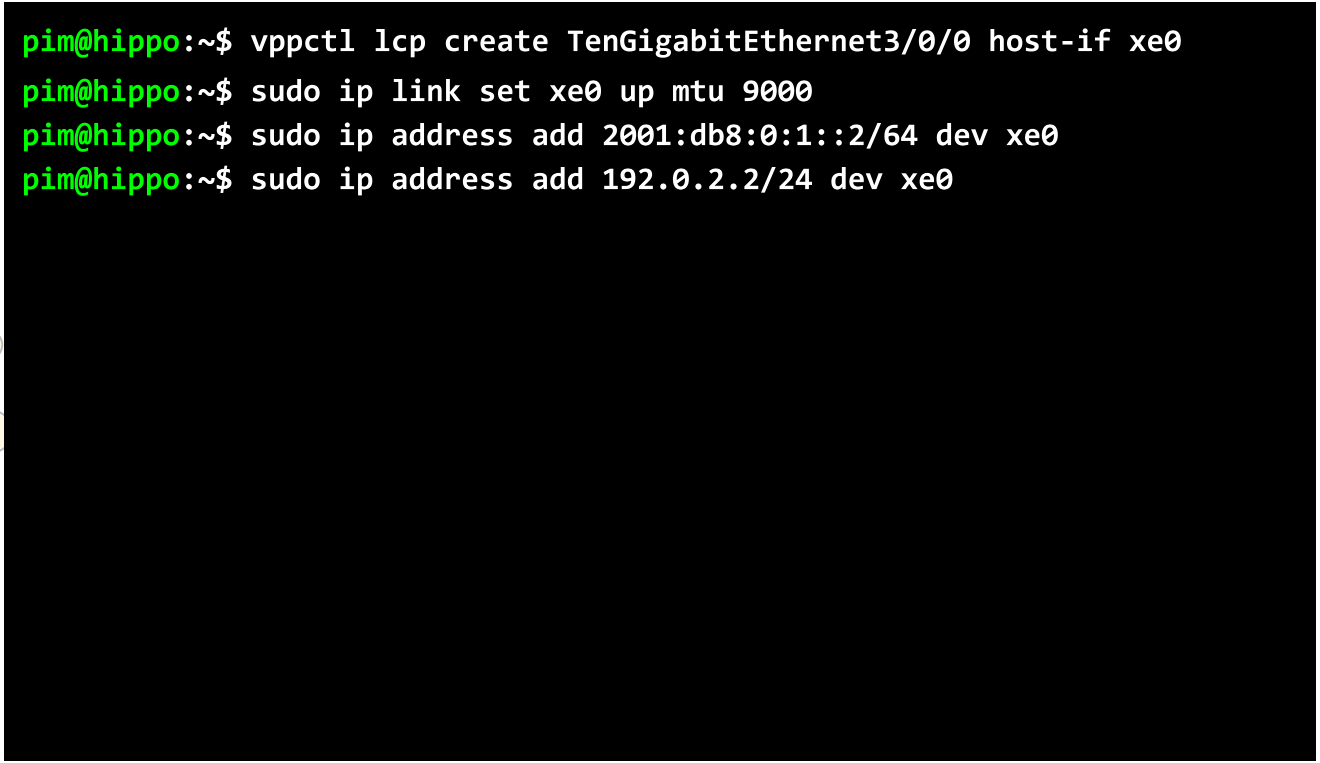




```
pim@hippo:~$ vppctl lcp create TenGigabitEthernet3/0/0 host-if xe0
```

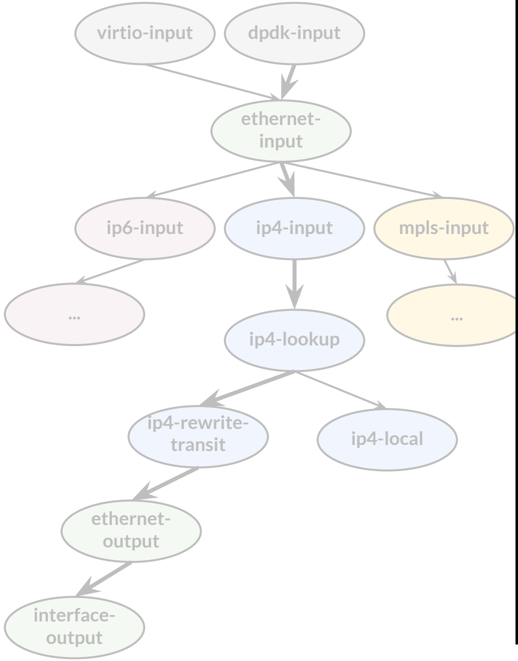
Intro: VPP LinuxCP

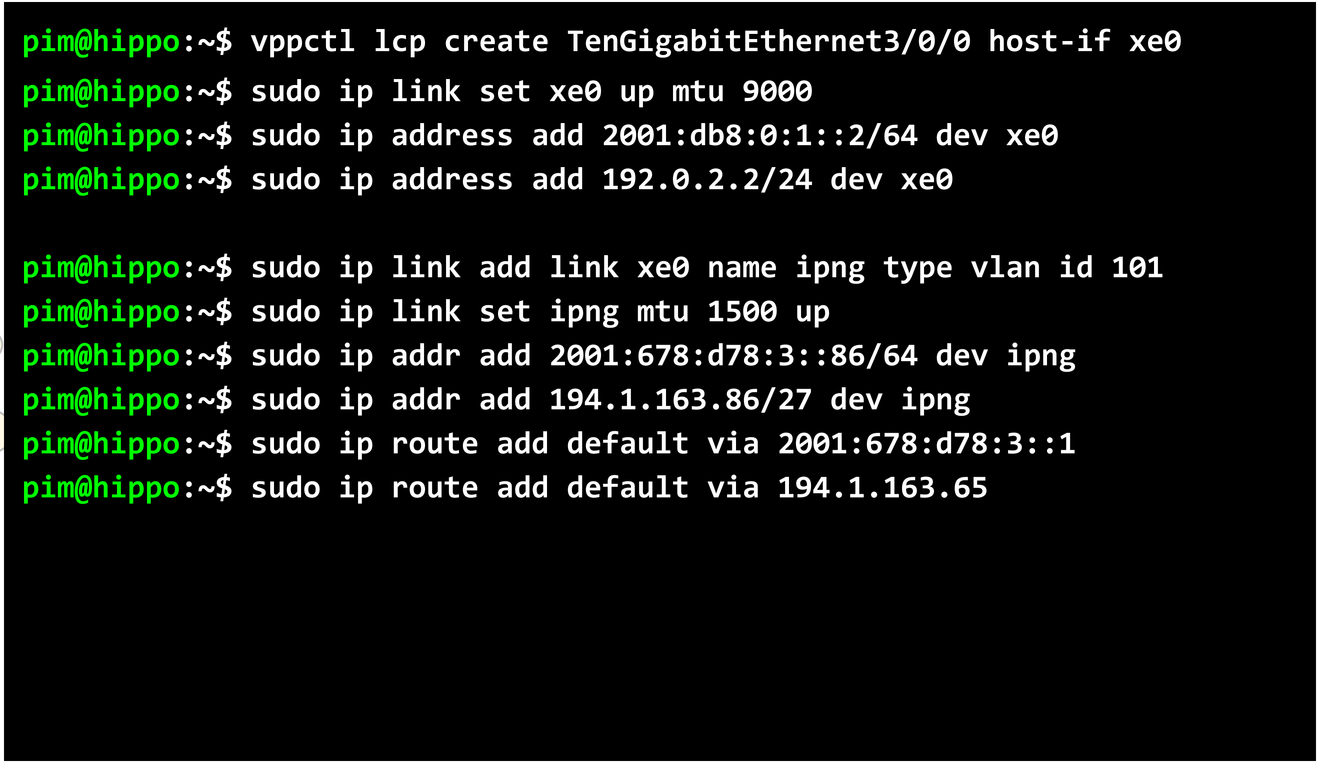




```
pim@hippo:~$ vppctl lcp create TenGigabitEthernet3/0/0 host-if xe0
pim@hippo:~$ sudo ip link set xe0 up mtu 9000
pim@hippo:~$ sudo ip address add 2001:db8:0:1::2/64 dev xe0
pim@hippo:~$ sudo ip address add 192.0.2.2/24 dev xe0
```

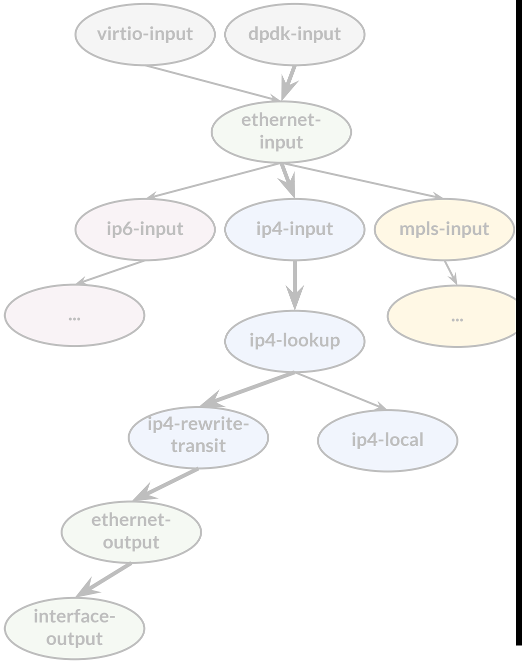
Intro: VPP LinuxCP

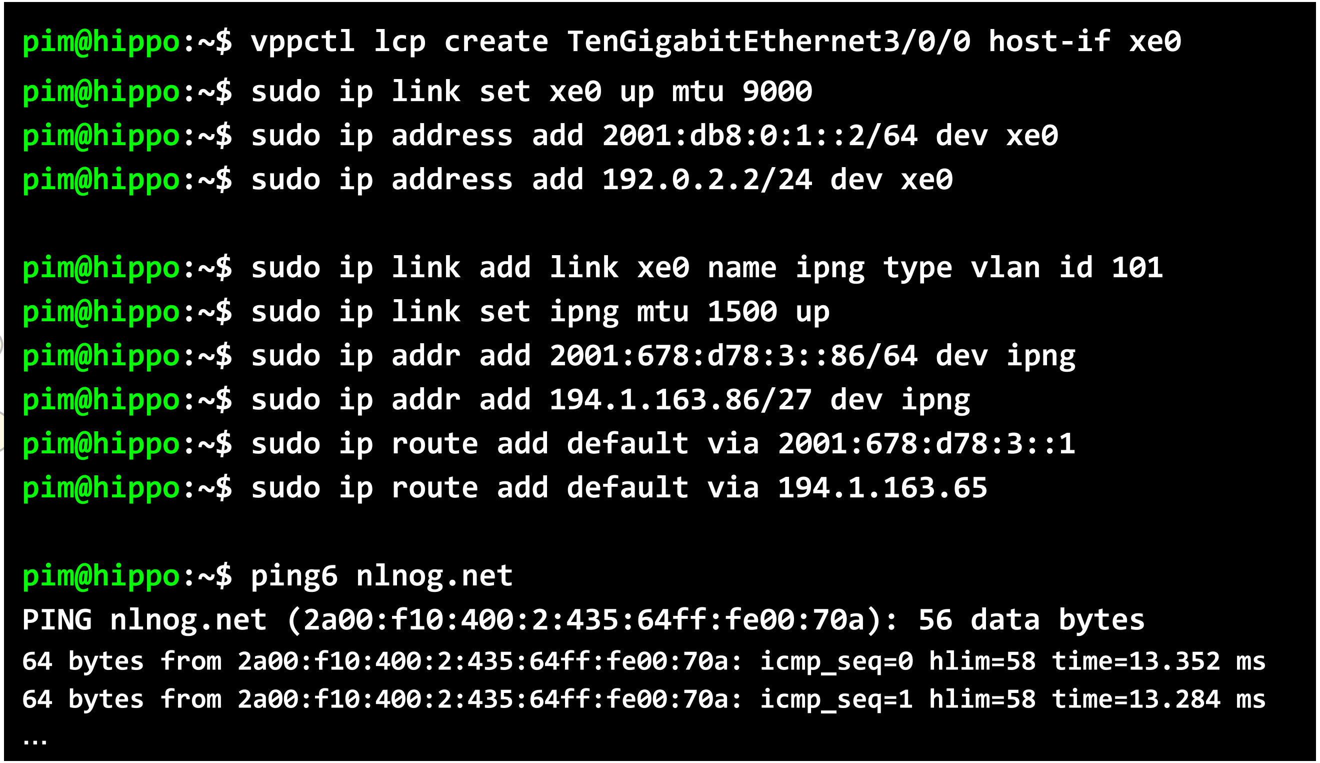




```
pim@hippo:~$ vppctl lcp create TenGigabitEthernet3/0/0 host-if xe0
pim@hippo:~$ sudo ip link set xe0 up mtu 9000
pim@hippo:~$ sudo ip address add 2001:db8:0:1::2/64 dev xe0
pim@hippo:~$ sudo ip address add 192.0.2.2/24 dev xe0

pim@hippo:~$ sudo ip link add link xe0 name ipng type vlan id 101
pim@hippo:~$ sudo ip link set ipng mtu 1500 up
pim@hippo:~$ sudo ip addr add 2001:678:d78:3::86/64 dev ipng
pim@hippo:~$ sudo ip addr add 194.1.163.86/27 dev ipng
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
```

Intro: VPP LinuxCP





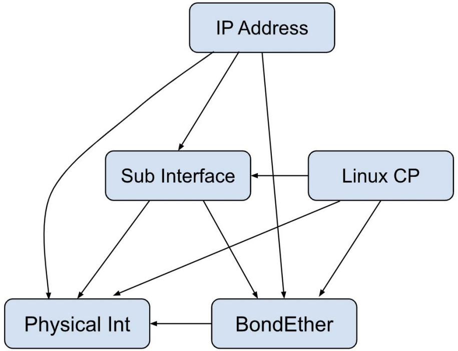
```
pim@hippo:~$ vppctl lcp create TenGigabitEthernet3/0/0 host-if xe0
pim@hippo:~$ sudo ip link set xe0 up mtu 9000
pim@hippo:~$ sudo ip address add 2001:db8:0:1::2/64 dev xe0
pim@hippo:~$ sudo ip address add 192.0.2.2/24 dev xe0

pim@hippo:~$ sudo ip link add link xe0 name ipng type vlan id 101
pim@hippo:~$ sudo ip link set ipng mtu 1500 up
pim@hippo:~$ sudo ip addr add 2001:678:d78:3::86/64 dev ipng
pim@hippo:~$ sudo ip addr add 194.1.163.86/27 dev ipng
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:~$ ping6 nlnog.net
PING nlnog.net (2a00:f10:400:2:435:64ff:fe00:70a): 56 data bytes
64 bytes from 2a00:f10:400:2:435:64ff:fe00:70a: icmp_seq=0 hlim=58 time=13.352 ms
64 bytes from 2a00:f10:400:2:435:64ff:fe00:70a: icmp_seq=1 hlim=58 time=13.284 ms
…
```

Configuration - Dataplane - vppcfg

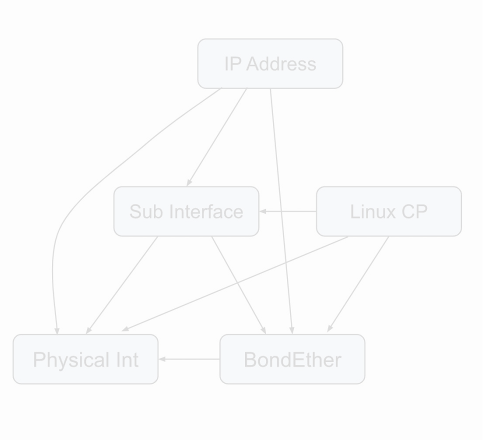
Wrote a vppcfg utility [github] that:

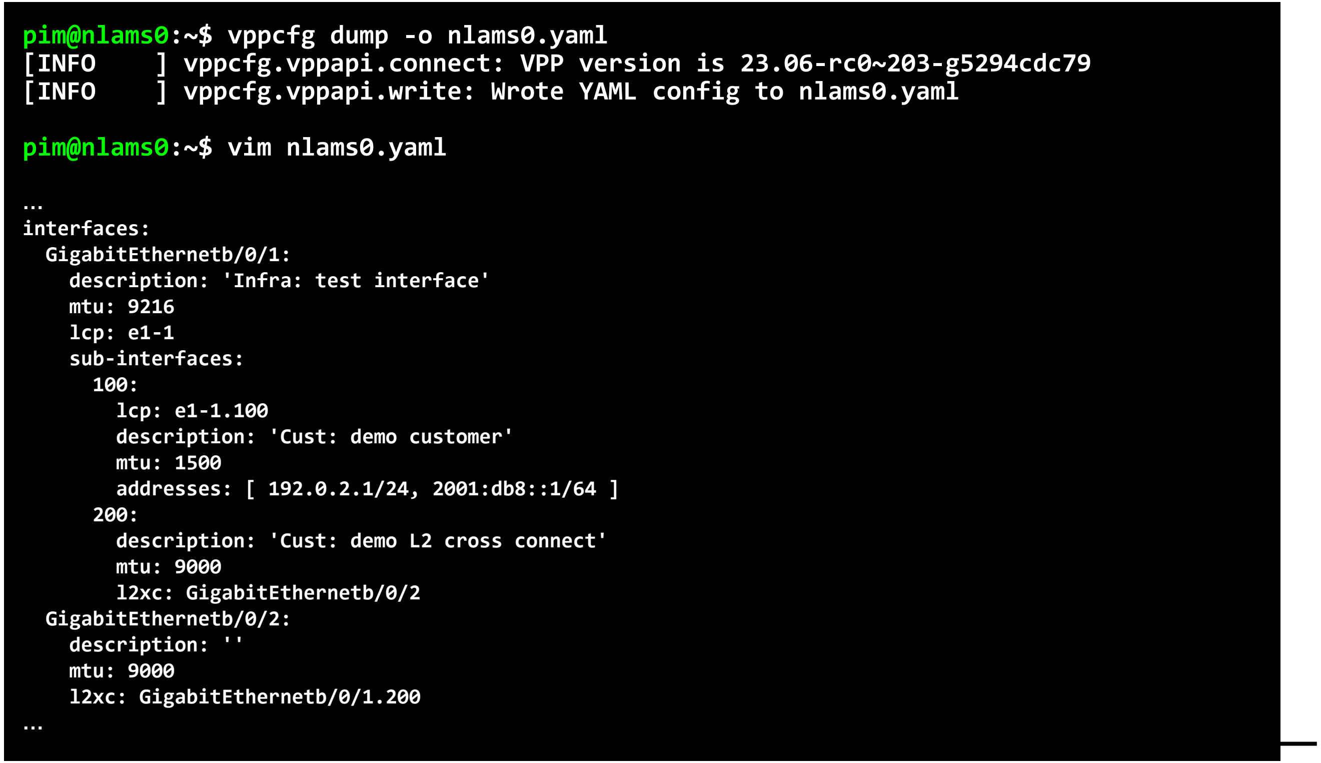
- Reads a YAML configuration file [user guide]
  - Checks it for syntax using Yamale
  - Checks it for semantics using a constraints language
- Dumps running state into a YAML file, using VPP API
- Plans a path from the running state to the required state
  - Uses declarative sequencing with a DAG
- Applies any new configuration to VPP using API or CLI

Targeting inclusion upstream in VPP 23.10



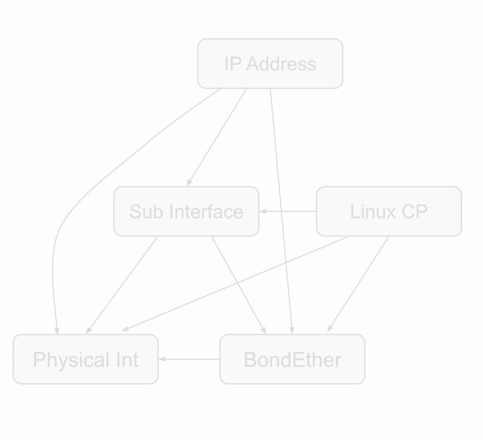
Configuration - Dataplane - vppcfg

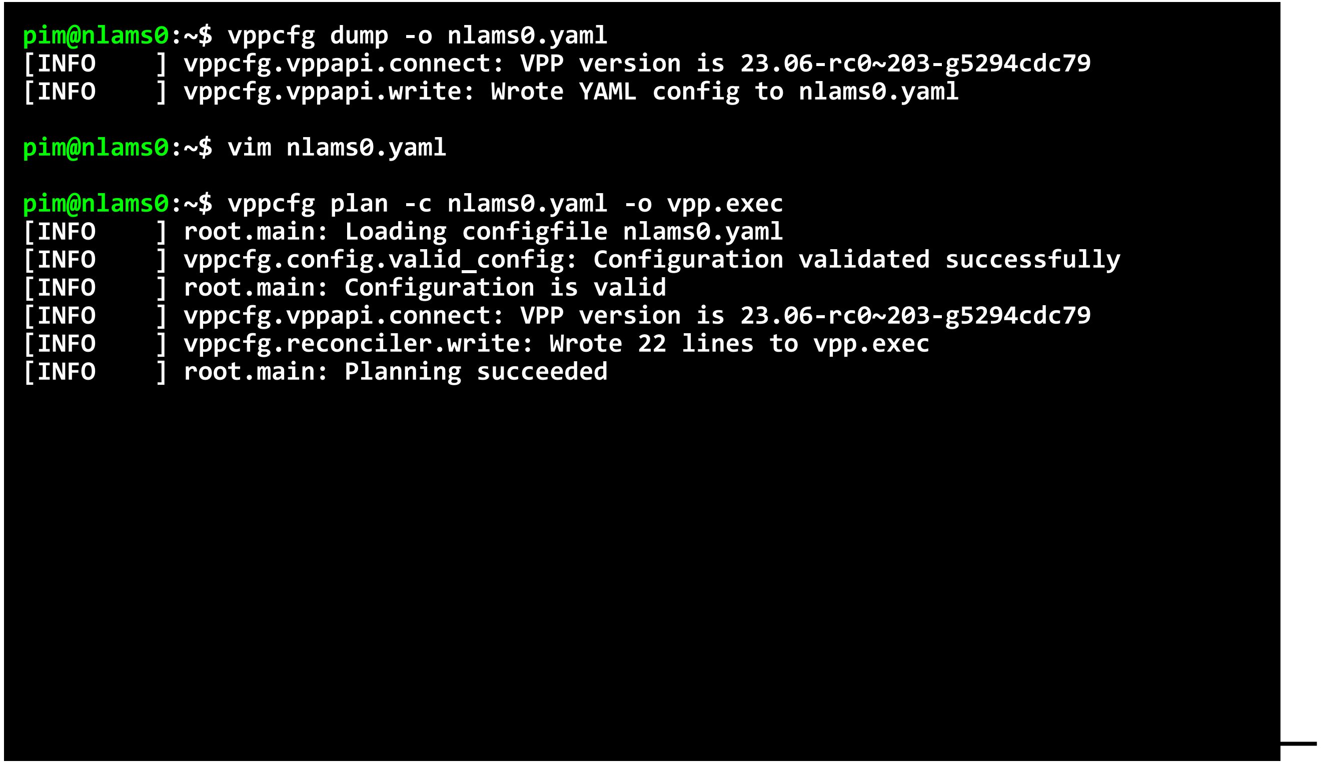




```
pim@nlams0:~$ vppcfg dump -o nlams0.yaml
[INFO    ] vppcfg.vppapi.connect: VPP version is 23.06-rc0~203-g5294cdc79
[INFO    ] vppcfg.vppapi.write: Wrote YAML config to nlams0.yaml

pim@nlams0:~$ vim nlams0.yaml

…
interfaces:
  GigabitEthernetb/0/1:
    description: 'Infra: test interface'
    mtu: 9216
    lcp: e1-1
    sub-interfaces:
      100:
        lcp: e1-1.100
        description: 'Cust: demo customer'
        mtu: 1500
        addresses: [ 192.0.2.1/24, 2001:db8::1/64 ]
      200:
        description: 'Cust: demo L2 cross connect'
        mtu: 9000
        l2xc: GigabitEthernetb/0/2
  GigabitEthernetb/0/2:
    description: ''
    mtu: 9000
    l2xc: GigabitEthernetb/0/1.200
…
```

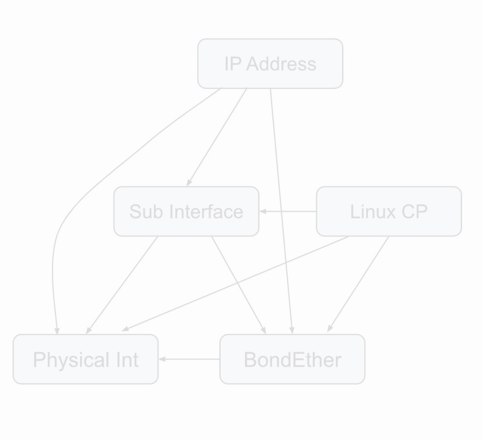
Configuration - Dataplane - vppcfg

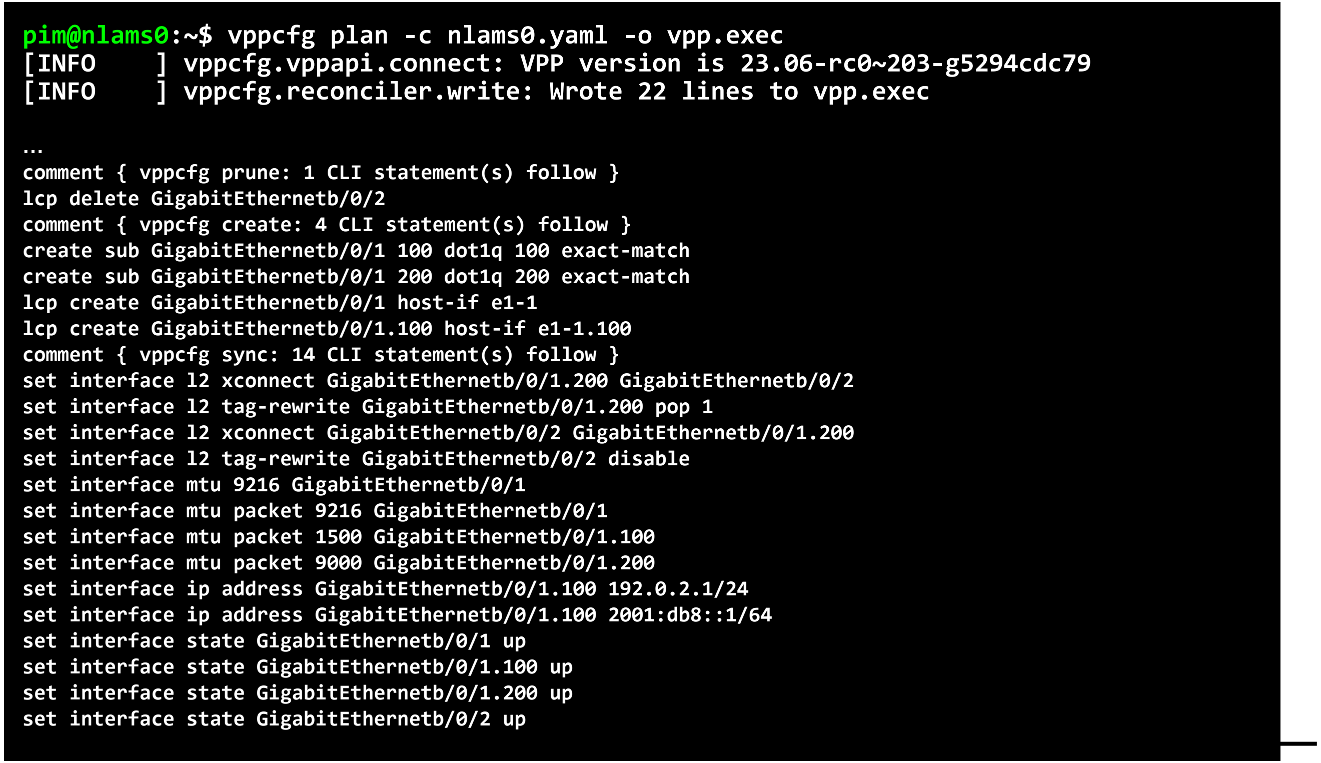




```
pim@nlams0:~$ vppcfg dump -o nlams0.yaml
[INFO    ] vppcfg.vppapi.connect: VPP version is 23.06-rc0~203-g5294cdc79
[INFO    ] vppcfg.vppapi.write: Wrote YAML config to nlams0.yaml

pim@nlams0:~$ vim nlams0.yaml

pim@nlams0:~$ vppcfg plan -c nlams0.yaml -o vpp.exec
[INFO    ] root.main: Loading configfile nlams0.yaml
[INFO    ] vppcfg.config.valid_config: Configuration validated successfully
[INFO    ] root.main: Configuration is valid
[INFO    ] vppcfg.vppapi.connect: VPP version is 23.06-rc0~203-g5294cdc79
[INFO    ] vppcfg.reconciler.write: Wrote 22 lines to vpp.exec
[INFO    ] root.main: Planning succeeded
```

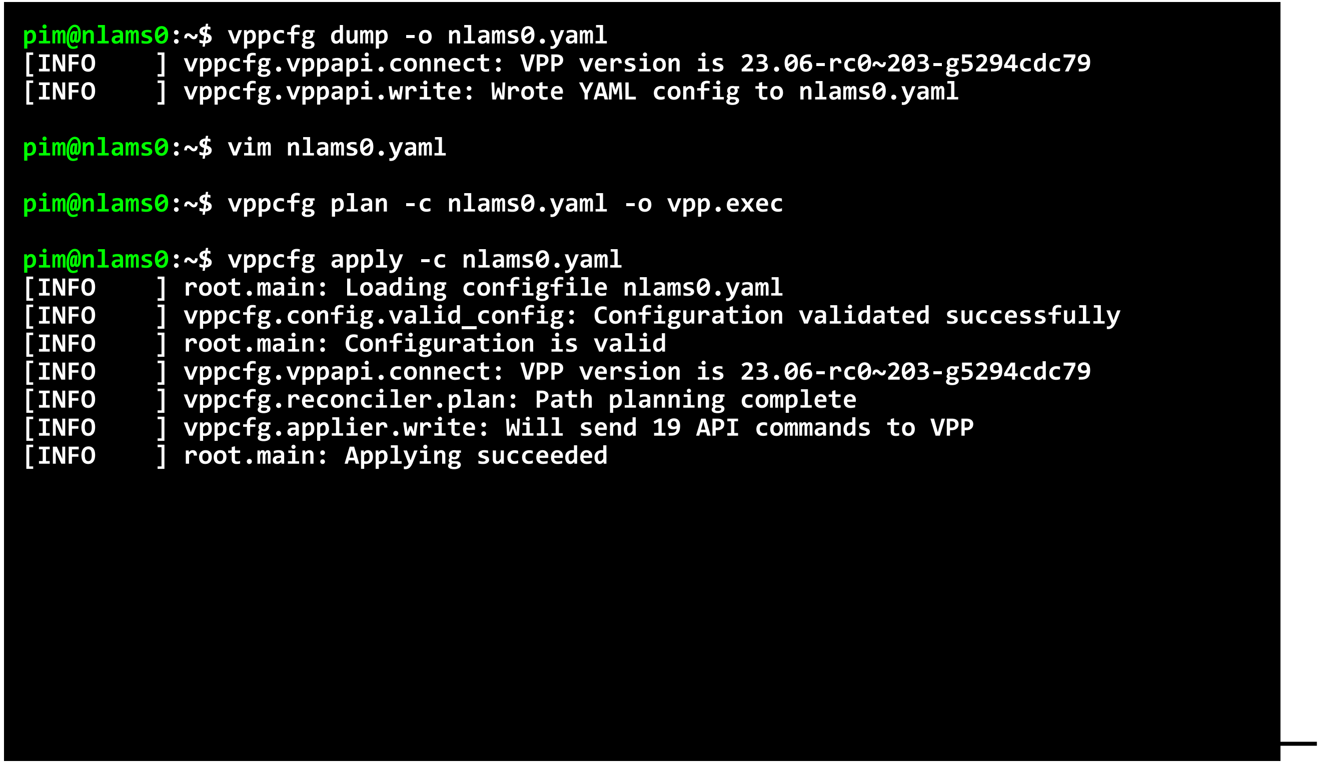
Configuration - Dataplane - vppcfg



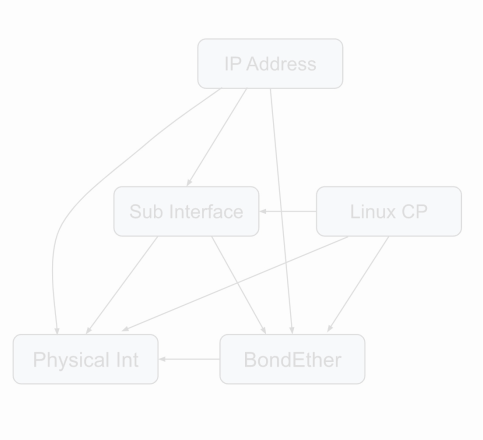


```
pim@nlams0:~$ vppcfg plan -c nlams0.yaml -o vpp.exec
[INFO    ] vppcfg.vppapi.connect: VPP version is 23.06-rc0~203-g5294cdc79
[INFO    ] vppcfg.reconciler.write: Wrote 22 lines to vpp.exec

…
comment { vppcfg prune: 1 CLI statement(s) follow }
lcp delete GigabitEthernetb/0/2
comment { vppcfg create: 4 CLI statement(s) follow }
create sub GigabitEthernetb/0/1 100 dot1q 100 exact-match
create sub GigabitEthernetb/0/1 200 dot1q 200 exact-match
lcp create GigabitEthernetb/0/1 host-if e1-1
lcp create GigabitEthernetb/0/1.100 host-if e1-1.100
comment { vppcfg sync: 14 CLI statement(s) follow }
set interface l2 xconnect GigabitEthernetb/0/1.200 GigabitEthernetb/0/2
set interface l2 tag-rewrite GigabitEthernetb/0/1.200 pop 1
set interface l2 xconnect GigabitEthernetb/0/2 GigabitEthernetb/0/1.200
set interface l2 tag-rewrite GigabitEthernetb/0/2 disable
set interface mtu 9216 GigabitEthernetb/0/1
set interface mtu packet 9216 GigabitEthernetb/0/1
set interface mtu packet 1500 GigabitEthernetb/0/1.100
set interface mtu packet 9000 GigabitEthernetb/0/1.200
set interface ip address GigabitEthernetb/0/1.100 192.0.2.1/24
set interface ip address GigabitEthernetb/0/1.100 2001:db8::1/64
set interface state GigabitEthernetb/0/1 up
set interface state GigabitEthernetb/0/1.100 up
set interface state GigabitEthernetb/0/1.200 up
set interface state GigabitEthernetb/0/2 up
```

Configuration - Dataplane - vppcfg



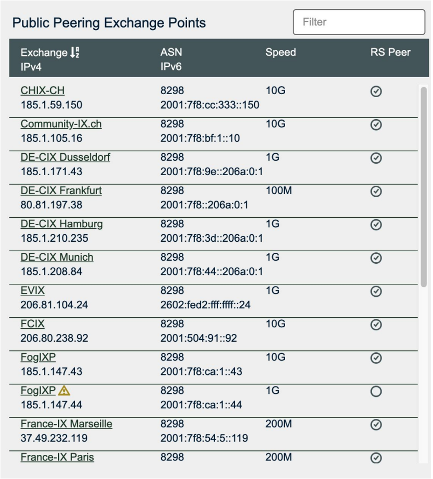
```
pim@nlams0:~$ vppcfg dump -o nlams0.yaml
[INFO    ] vppcfg.vppapi.connect: VPP version is 23.06-rc0~203-g5294cdc79
[INFO    ] vppcfg.vppapi.write: Wrote YAML config to nlams0.yaml

pim@nlams0:~$ vim nlams0.yaml

pim@nlams0:~$ vppcfg plan -c nlams0.yaml -o vpp.exec

pim@nlams0:~$ vppcfg apply -c nlams0.yaml
[INFO    ] root.main: Loading configfile nlams0.yaml
[INFO    ] vppcfg.config.valid_config: Configuration validated successfully
[INFO    ] root.main: Configuration is valid
[INFO    ] vppcfg.vppapi.connect: VPP version is 23.06-rc0~203-g5294cdc79
[INFO    ] vppcfg.reconciler.plan: Path planning complete
[INFO    ] vppcfg.applier.write: Will send 19 API commands to VPP
[INFO    ] root.main: Applying succeeded
```



Configuration - Controlplane - Kees

Rewrote a kees utility* [github] that:

- Reads set of YAML configuration file
  - Augments with PeeringDB and IRRDBs (bgpq4)
- Constructs per-router configuration
- Uses Jinja2 (Ansible) to emit configuration files
- rsync to router, safe reloads of controlplane
  - Unbound, Firewall, Bird2, Borgmatic, SNMP Agent, VRRP
  - And of course: VPP configs

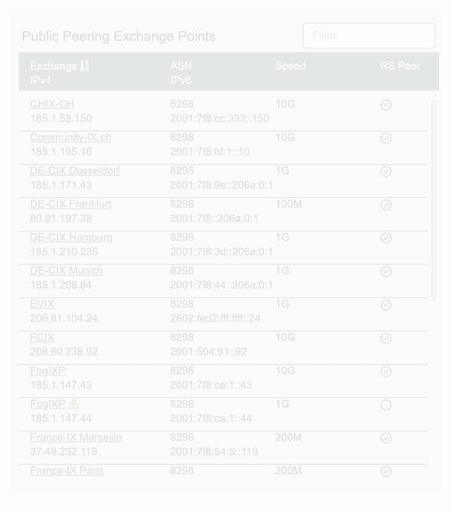


Public Peering Exchange Points

Exchange IPv4	ASN IPv6	Speed	RS Peer
CHIX-CH 185.1.59.150	8298 2001:7f8:cc:333::150	10G	✓
Community-IX.ch 185.1.105.16	8298 2001:7f8:bf:1::10	10G	✓
DE-CIX Dusseldorf 185.1.171.43	8298 2001:7f8:9e::206a:0:1	1G	✓
DE-CIX Frankfurt 80.81.197.38	8298 2001:7f8::206a:0:1	100M	✓
DE-CIX Hamburg 185.1.210.235	8298 2001:7f8:3d::206a:0:1	1G	✓
DE-CIX Munich 185.1.208.84	8298 2001:7f8:44::206a:0:1	1G	✓
EVIX 206.81.104.24	8298 2602:fed2:fff:ffff::24	1G	✓
FCIX 206.80.238.92	8298 2001:504:91::92	10G	✓
FogIXP 185.1.147.43	8298 2001:7f8:ca:1::43	10G	✓
FogIXP ⚠ 185.1.147.44	8298 2001:7f8:ca:1::44	1G	○
France-IX Marseille 37.49.232.119	8298 2001:7f8:54:5::119	200M	✓
France-IX Paris	8298	200M	✓

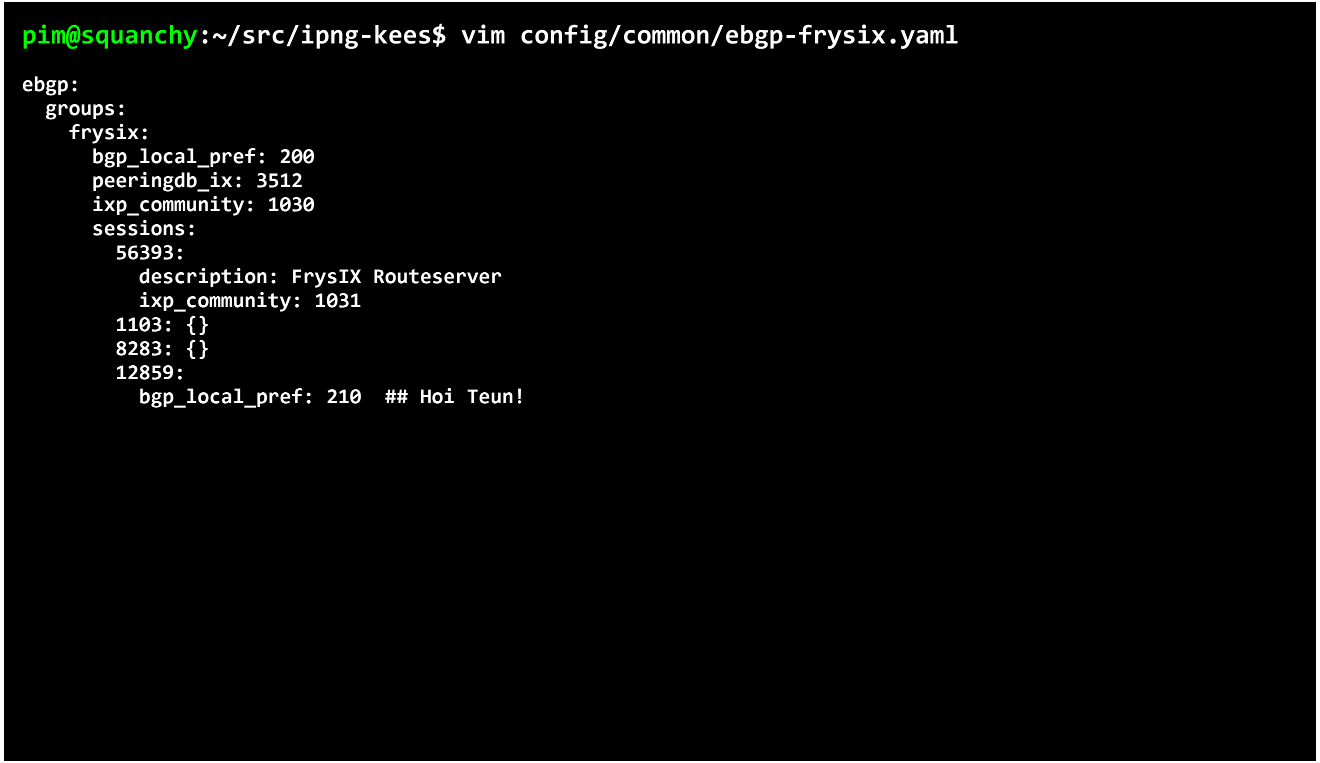
***) Originally written by Coloclue AS8283**

Configuration - Controlplane - Kees



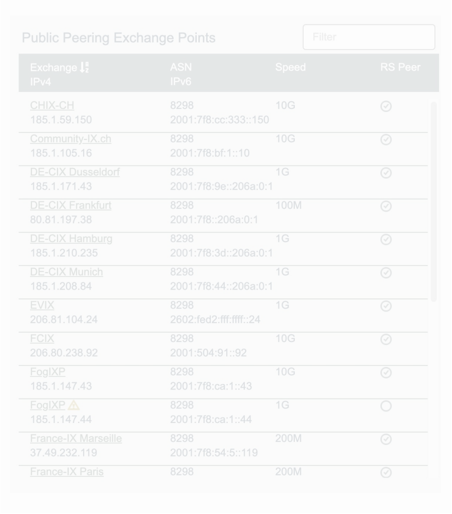
Public Peering Exchange Points

Exchange IPv4	ASN IPv6	Speed	RS Peer
CHIX-CH 185.1.59.150	8298 2001:7f8:cc:333::150	10G	✓
Community-IX.ch 185.1.105.16	8298 2001:7f8:bf:1::10	10G	✓
DE-CIX Dusseldorf 185.1.171.43	8298 2001:7f8:9e::206a:0:1	1G	✓
DE-CIX Frankfurt 80.81.197.38	8298 2001:7f8::206a:0:1	100M	✓
DE-CIX Hamburg 185.1.210.235	8298 2001:7f8:3d::206a:0:1	1G	✓
DE-CIX Munich 185.1.208.84	8298 2001:7f8:44::206a:0:1	1G	✓
EVIX 206.81.104.24	8298 2602:fed2:fff:ffff::24	1G	✓
FCIX 206.80.238.92	8298 2001:504:91::92	10G	✓
FogIXP 185.1.147.43	8298 2001:7f8:ca:1::43	10G	✓
FogIXP ⚠ 185.1.147.44	8298 2001:7f8:ca:1::44	1G	○
France-IX Marseille 37.49.232.119	8298 2001:7f8:54:5::119	200M	✓
France-IX Paris	8298	200M	✓



```
pim@squanchy:~/src/ipng-kees$ vim config/common/ebgp-frysix.yaml

ebgp:
  groups:
    frysix:
      bgp_local_pref: 200
      peeringdb_ix: 3512
      ixp_community: 1030
      sessions:
        56393:
          description: FrysIX Routeserver
          ixp_community: 1031
        1103: {}
        8283: {}
        12859:
          bgp_local_pref: 210  ## Hoi Teun!
```

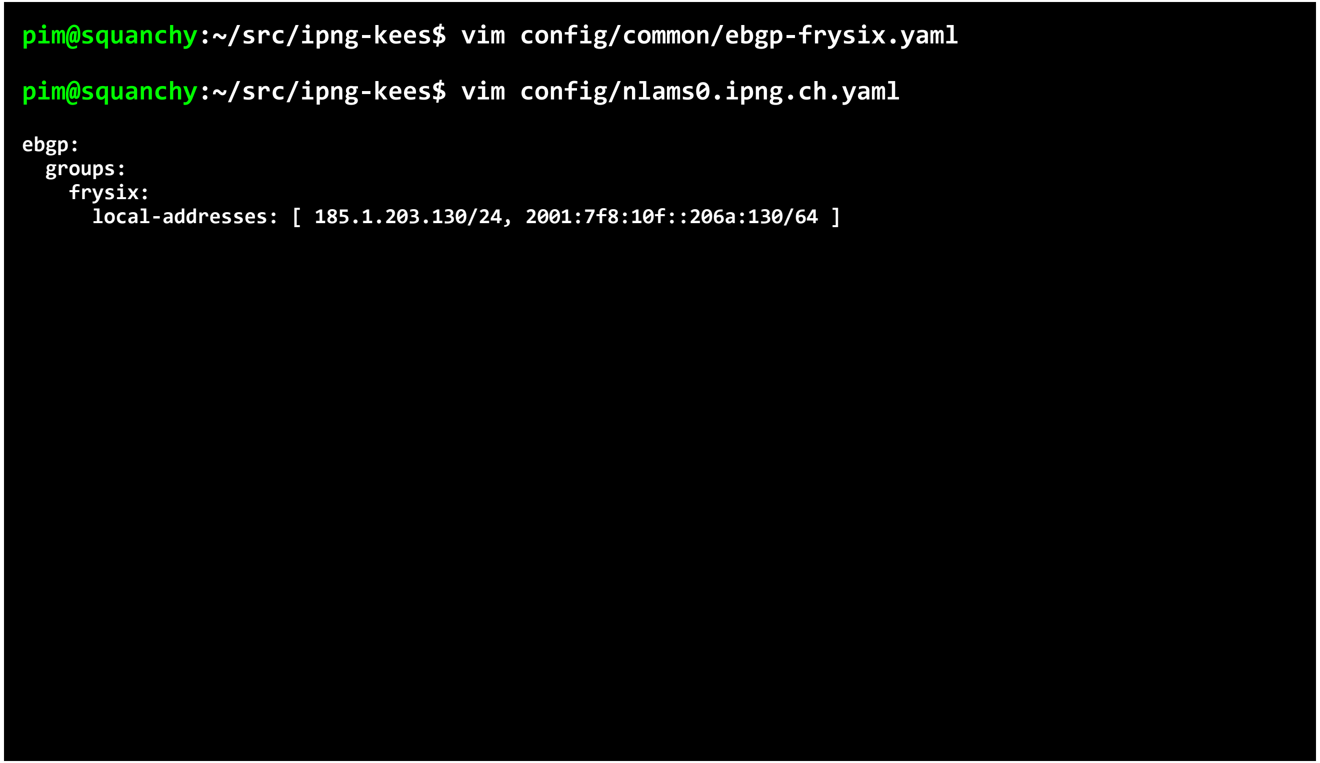
Configuration - Controlplane - Kees



Public Peering Exchange Points

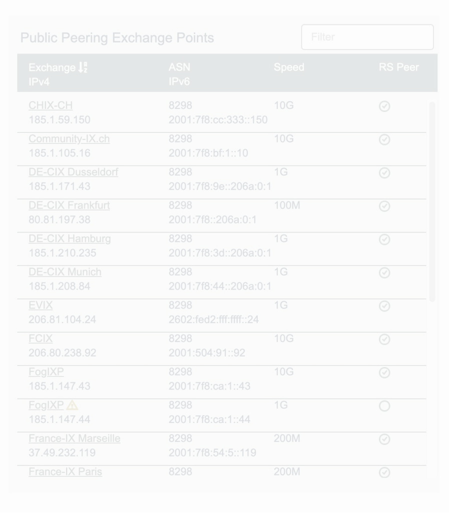
Filter

Exchange IPv4	ASN IPv6	Speed	RS Peer
CHIX-CH 185.1.59.150	8298 2001:7f8:cc:333::150	10G	✓
Community-IX.ch 185.1.105.16	8298 2001:7f8:bf:1::10	10G	✓
DE-CIX Dusseldorf 185.1.171.43	8298 2001:7f8:9e::206a:0:1	1G	✓
DE-CIX Frankfurt 80.81.197.38	8298 2001:7f8::206a:0:1	100M	✓
DE-CIX Hamburg 185.1.210.235	8298 2001:7f8:3d::206a:0:1	1G	✓
DE-CIX Munich 185.1.208.84	8298 2001:7f8:44::206a:0:1	1G	✓
EVIX 206.81.104.24	8298 2602:fed2:fff:ffff::24	1G	✓
FCIX 206.80.238.92	8298 2001:504:91::92	10G	✓
FogIXP 185.1.147.43	8298 2001:7f8:ca:1::43	10G	✓
FogIXP ⚠ 185.1.147.44	8298 2001:7f8:ca:1::44	1G	○
France-IX Marseille 37.49.232.119	8298 2001:7f8:54:5::119	200M	✓
France-IX Paris	8298	200M	✓



```
pim@squanchy:~/src/ipng-kees$ vim config/common/ebgp-frysix.yaml

pim@squanchy:~/src/ipng-kees$ vim config/nlams0.ipng.ch.yaml

ebgp:
  groups:
    frysix:
      local-addresses: [ 185.1.203.130/24, 2001:7f8:10f::206a:130/64 ]
```

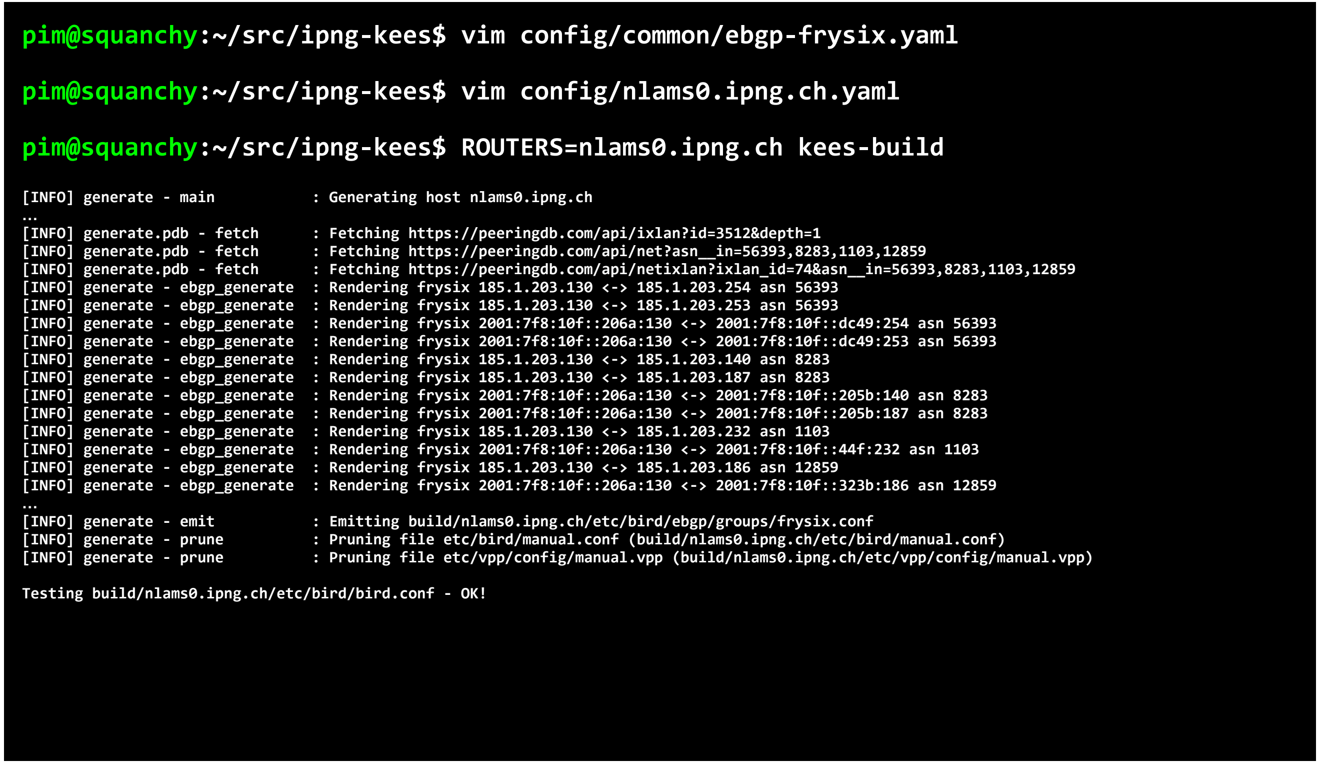
Configuration - Controlplane - Kees



Public Peering Exchange Points

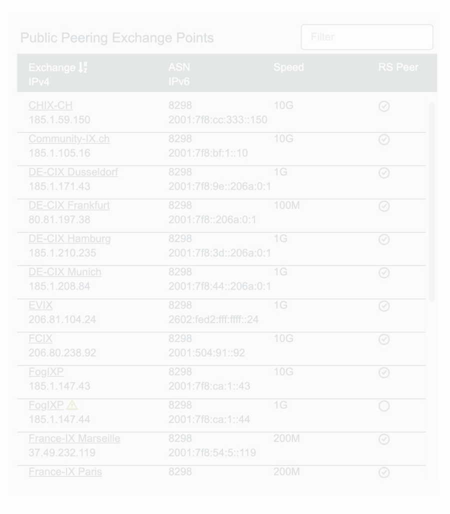
Filter

Exchange IPv4	ASN IPv6	Speed	RS Peer
CHIX-CH 185.1.59.150	8298 2001:7f8:cc:333::150	10G	✓
Community-IX.ch 185.1.105.16	8298 2001:7f8:bf:1::10	10G	✓
DE-CIX Dusseldorf 185.1.171.43	8298 2001:7f8:9e::206a:0:1	1G	✓
DE-CIX Frankfurt 80.81.197.38	8298 2001:7f8::206a:0:1	100M	✓
DE-CIX Hamburg 185.1.210.235	8298 2001:7f8:3d::206a:0:1	1G	✓
DE-CIX Munich 185.1.208.84	8298 2001:7f8:44::206a:0:1	1G	✓
EVIX 206.81.104.24	8298 2602:fed2:fff:ffff::24	1G	✓
FCIX 206.80.238.92	8298 2001:504:91::92	10G	✓
FogIXP 185.1.147.43	8298 2001:7f8:ca:1::43	10G	✓
FogIXP ⚠ 185.1.147.44	8298 2001:7f8:ca:1::44	1G	○
France-IX Marseille 37.49.232.119	8298 2001:7f8:54:5::119	200M	✓
France-IX Paris	8298	200M	✓



```
pim@squanchy:~/src/ipng-kees$ vim config/common/ebgp-frysix.yaml

pim@squanchy:~/src/ipng-kees$ vim config/nlams0.ipng.ch.yaml

pim@squanchy:~/src/ipng-kees$ ROUTERS=nlams0.ipng.ch kees-build

[INFO] generate - main           : Generating host nlams0.ipng.ch
…
[INFO] generate.pdb - fetch      : Fetching https://peeringdb.com/api/ixlan?id=3512&depth=1
[INFO] generate.pdb - fetch      : Fetching https://peeringdb.com/api/net?asn__in=56393,8283,1103,12859
[INFO] generate.pdb - fetch      : Fetching https://peeringdb.com/api/netixlan?ixlan_id=74&asn__in=56393,8283,1103,12859
[INFO] generate - ebgp_generate  : Rendering frysix 185.1.203.130 <-> 185.1.203.254 asn 56393
[INFO] generate - ebgp_generate  : Rendering frysix 185.1.203.130 <-> 185.1.203.253 asn 56393
[INFO] generate - ebgp_generate  : Rendering frysix 2001:7f8:10f::206a:130 <-> 2001:7f8:10f::dc49:254 asn 56393
[INFO] generate - ebgp_generate  : Rendering frysix 2001:7f8:10f::206a:130 <-> 2001:7f8:10f::dc49:253 asn 56393
[INFO] generate - ebgp_generate  : Rendering frysix 185.1.203.130 <-> 185.1.203.140 asn 8283
[INFO] generate - ebgp_generate  : Rendering frysix 185.1.203.130 <-> 185.1.203.187 asn 8283
[INFO] generate - ebgp_generate  : Rendering frysix 2001:7f8:10f::206a:130 <-> 2001:7f8:10f::205b:140 asn 8283
[INFO] generate - ebgp_generate  : Rendering frysix 2001:7f8:10f::206a:130 <-> 2001:7f8:10f::205b:187 asn 8283
[INFO] generate - ebgp_generate  : Rendering frysix 185.1.203.130 <-> 185.1.203.232 asn 1103
[INFO] generate - ebgp_generate  : Rendering frysix 2001:7f8:10f::206a:130 <-> 2001:7f8:10f::44f:232 asn 1103
[INFO] generate - ebgp_generate  : Rendering frysix 185.1.203.130 <-> 185.1.203.186 asn 12859
[INFO] generate - ebgp_generate  : Rendering frysix 2001:7f8:10f::206a:130 <-> 2001:7f8:10f::323b:186 asn 12859
…
[INFO] generate - emit           : Emitting build/nlams0.ipng.ch/etc/bird/ebgp/groups/frysix.conf
[INFO] generate - prune          : Pruning file etc/bird/manual.conf (build/nlams0.ipng.ch/etc/bird/manual.conf)
[INFO] generate - prune          : Pruning file etc/vpp/config/manual.vpp (build/nlams0.ipng.ch/etc/vpp/config/manual.vpp)

Testing build/nlams0.ipng.ch/etc/bird/bird.conf - OK!
```

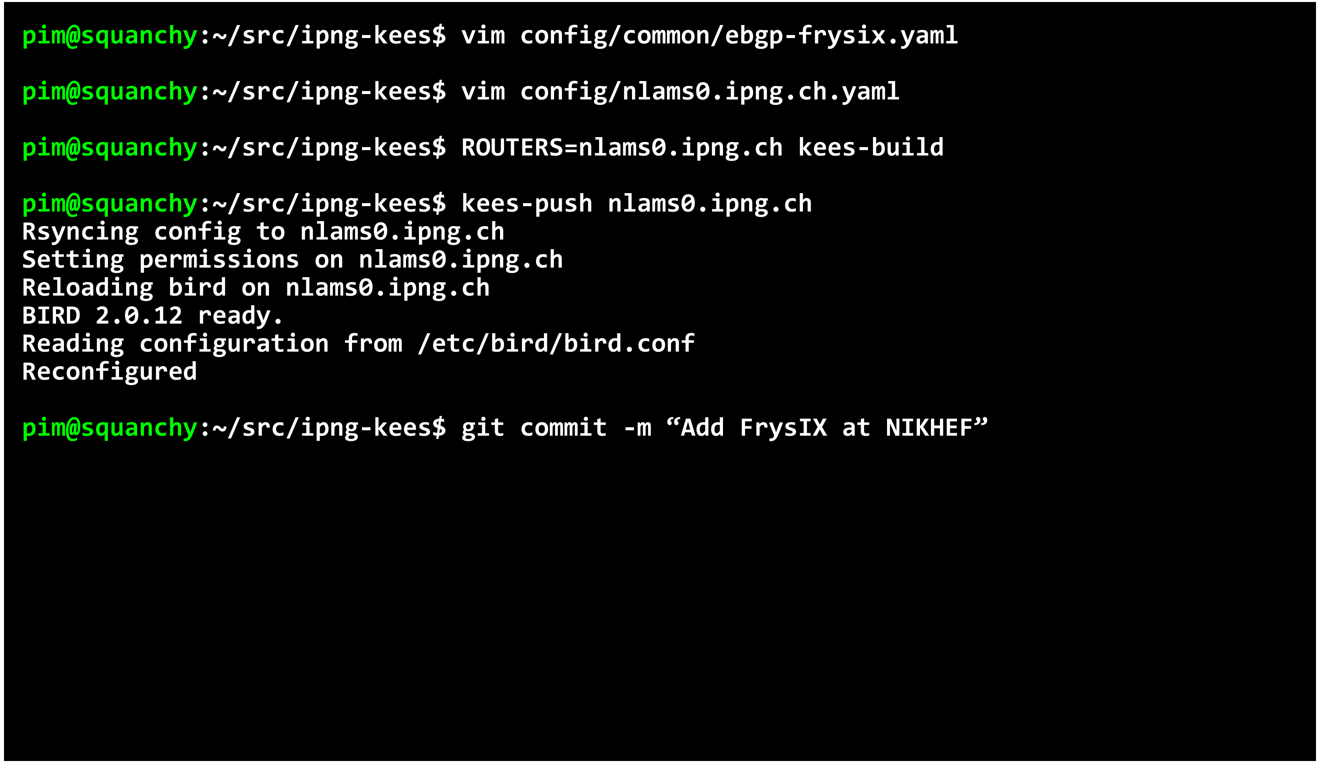
Configuration - Controlplane - Kees



Public Peering Exchange Points

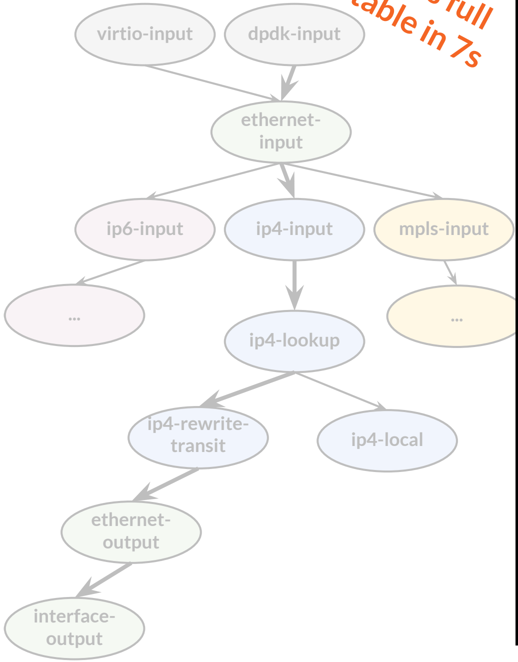
Filter

Exchange IPv4	ASN IPv6	Speed	RS Peer
CHIX-CH 185.1.59.150	8298 2001:7f8:cc:333::150	10G	✓
Community-IX.ch 185.1.105.16	8298 2001:7f8:bf:1::10	10G	✓
DE-CIX Dusseldorf 185.1.171.43	8298 2001:7f8:9e::206a:0:1	1G	✓
DE-CIX Frankfurt 80.81.197.38	8298 2001:7f8::206a:0:1	100M	✓
DE-CIX Hamburg 185.1.210.235	8298 2001:7f8:3d::206a:0:1	1G	✓
DE-CIX Munich 185.1.208.84	8298 2001:7f8:44::206a:0:1	1G	✓
EVIX 206.81.104.24	8298 2602:fed2:fff:ffff::24	1G	✓
FCIX 206.80.238.92	8298 2001:504:91::92	10G	✓
FogIXP 185.1.147.43	8298 2001:7f8:ca:1::43	10G	✓
FogIXP ⚠ 185.1.147.44	8298 2001:7f8:ca:1::44	1G	○
France-IX Marseille 37.49.232.119	8298 2001:7f8:54:5::119	200M	✓
France-IX Paris	8298	200M	✓

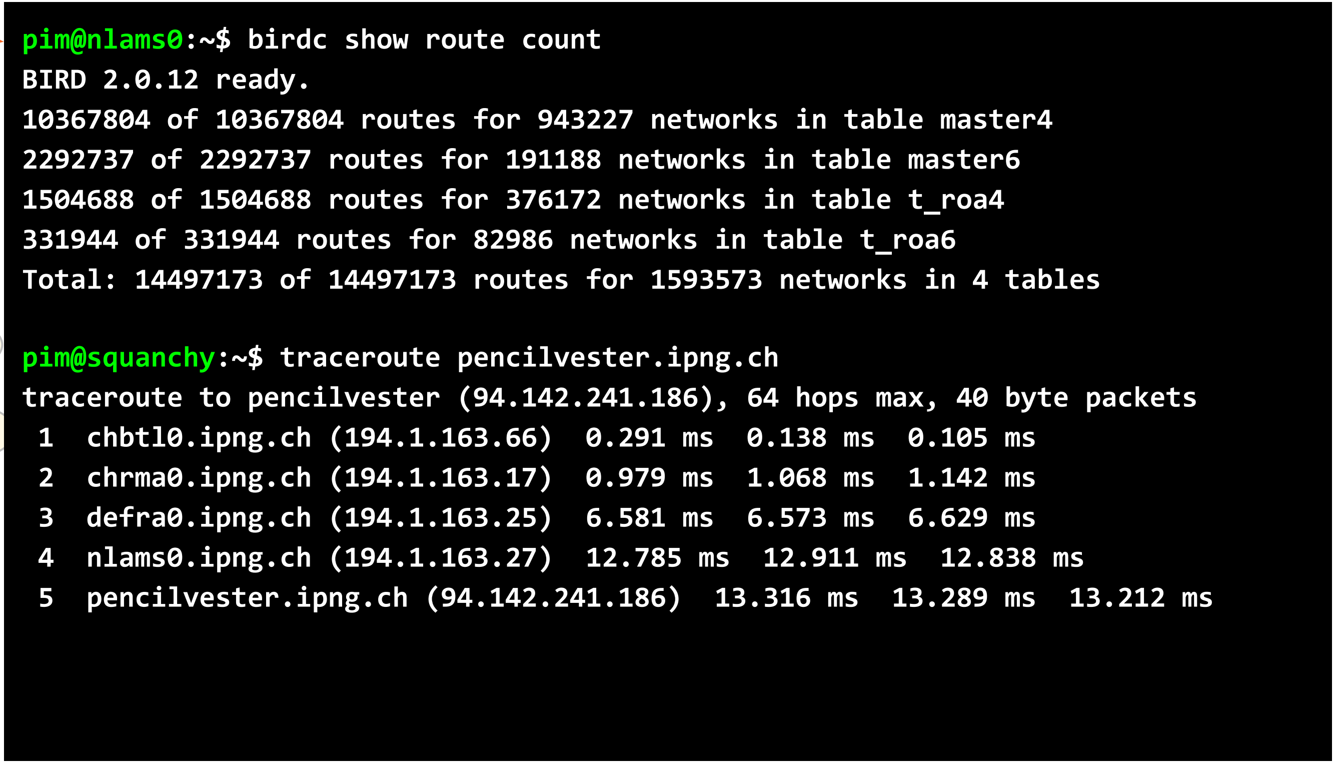


```
pim@squanchy:~/src/ipng-kees$ vim config/common/ebgp-frysix.yaml

pim@squanchy:~/src/ipng-kees$ vim config/nlams0.ipng.ch.yaml

pim@squanchy:~/src/ipng-kees$ ROUTERS=nlams0.ipng.ch kees-build

pim@squanchy:~/src/ipng-kees$ kees-push nlams0.ipng.ch
Rsyncing config to nlams0.ipng.ch
Setting permissions on nlams0.ipng.ch
Reloading bird on nlams0.ipng.ch
BIRD 2.0.12 ready.
Reading configuration from /etc/bird/bird.conf
Reconfigured

pim@squanchy:~/src/ipng-kees$ git commit -m “Add FrysIX at NIKHEF”
```

VPP in Production: tying it all together



Converges full BGP table in 7s



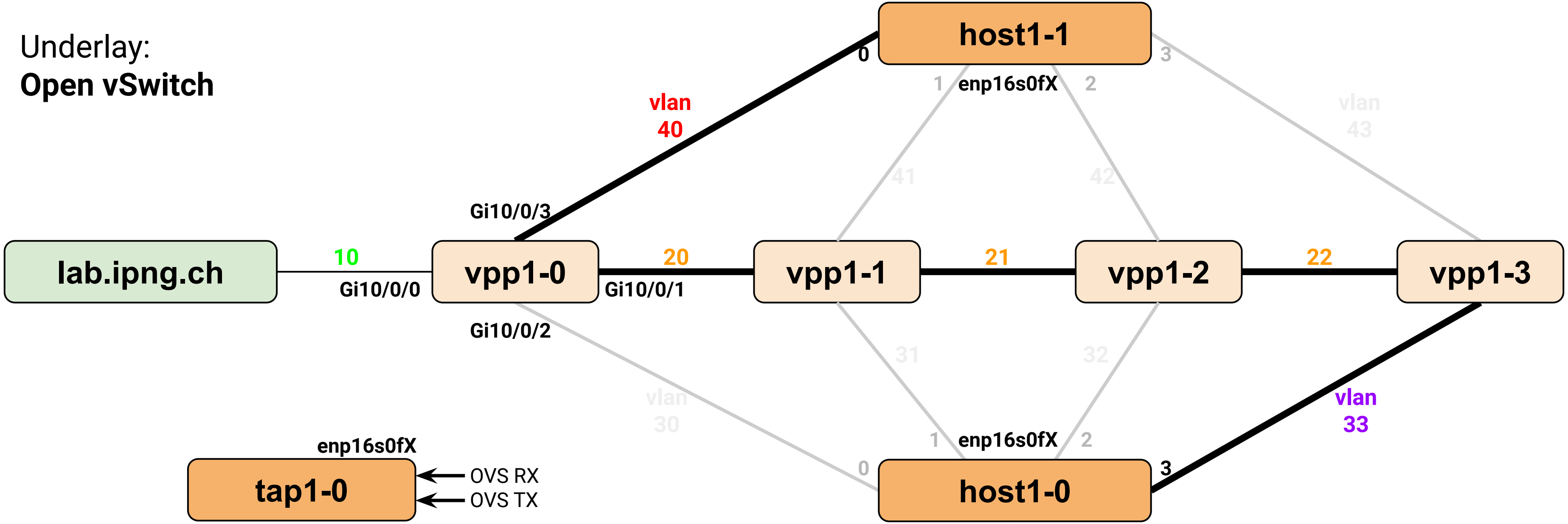
```
pim@nlams0:~$ birdc show route count
BIRD 2.0.12 ready.
10367804 of 10367804 routes for 943227 networks in table master4
2292737 of 2292737 routes for 191188 networks in table master6
1504688 of 1504688 routes for 376172 networks in table t_roa4
331944 of 331944 routes for 82986 networks in table t_roa6
Total: 14497173 of 14497173 routes for 1593573 networks in 4 tables

pim@squanchy:~$ traceroute pencilvester.ipng.ch
traceroute to pencilvester (94.142.241.186), 64 hops max, 40 byte packets
 1  chbtl0.ipng.ch (194.1.163.66)  0.291 ms  0.138 ms  0.105 ms
 2  chrma0.ipng.ch (194.1.163.17)  0.979 ms  1.068 ms  1.142 ms
 3  defra0.ipng.ch (194.1.163.25)  6.581 ms  6.573 ms  6.629 ms
 4  nlams0.ipng.ch (194.1.163.27)  12.785 ms  12.911 ms  12.838 ms
 5  pencilvester.ipng.ch (94.142.241.186)  13.316 ms  13.289 ms  13.212 ms
```

nl nog
netherlands network operator group

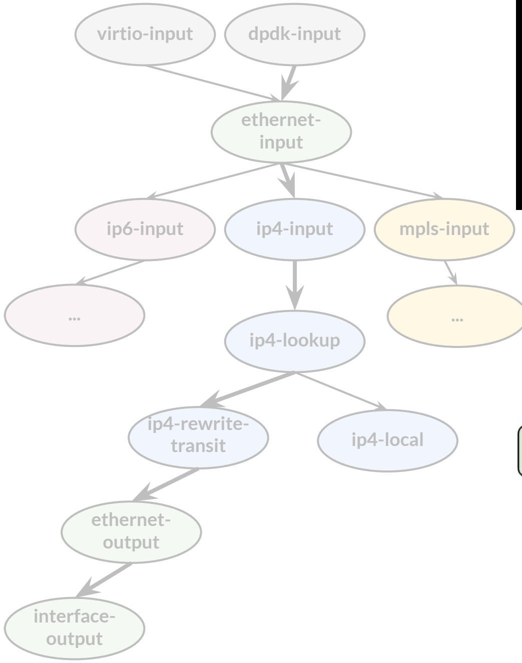
Act 2: MPLS in VPP

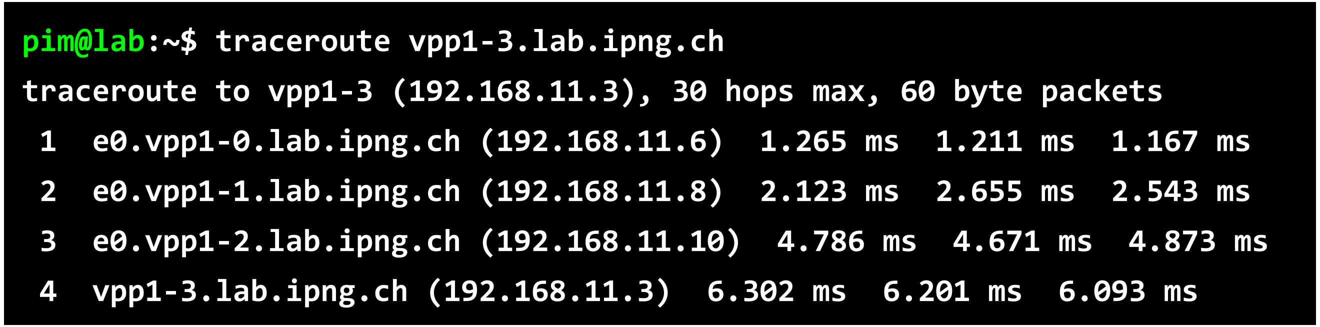
Intro: MPLS in VPP Lab setup

Underlay:
Open vSwitch



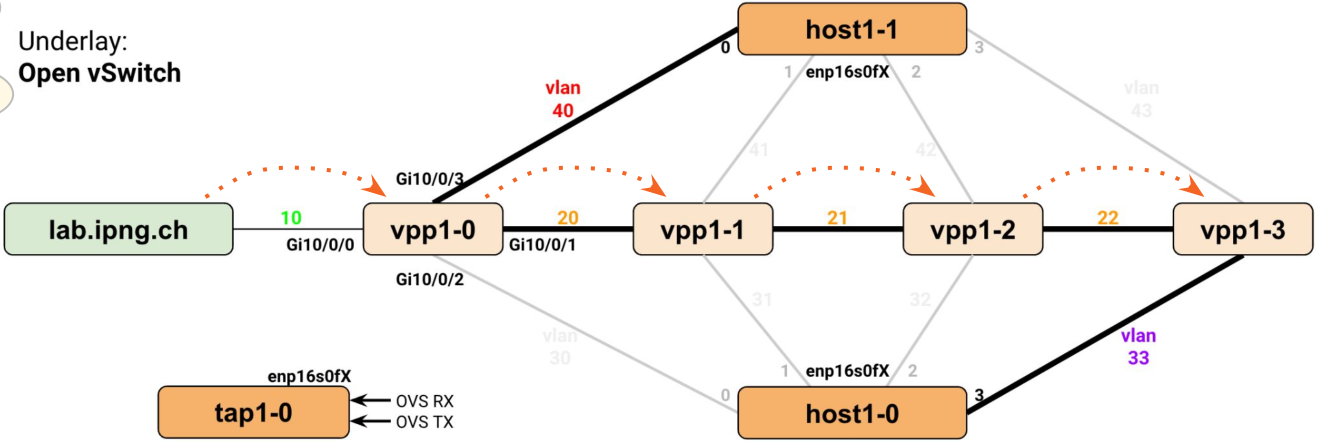
VPP: Without MPLS



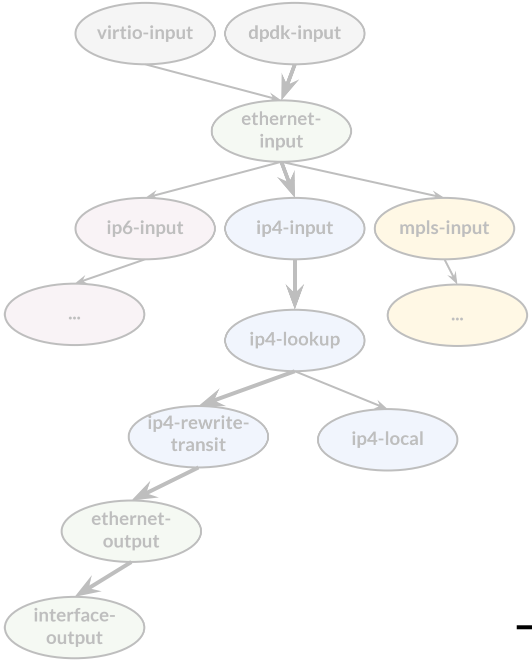


```
pim@lab:~$ traceroute vpp1-3.lab.ipng.ch
traceroute to vpp1-3 (192.168.11.3), 30 hops max, 60 byte packets
 1  e0.vpp1-0.lab.ipng.ch (192.168.11.6)  1.265 ms  1.211 ms  1.167 ms
 2  e0.vpp1-1.lab.ipng.ch (192.168.11.8)  2.123 ms  2.655 ms  2.543 ms
 3  e0.vpp1-2.lab.ipng.ch (192.168.11.10)  4.786 ms  4.671 ms  4.873 ms
 4  vpp1-3.lab.ipng.ch (192.168.11.3)  6.302 ms  6.201 ms  6.093 ms
```

Underlay:
Open vSwitch



VPP: Linux Control Plane and MPLS



Changes to Netlink Listener plugin:

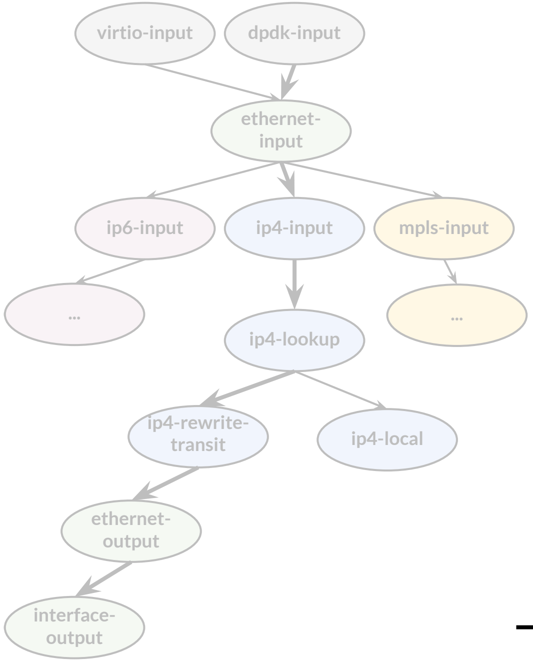
1. Add MPLS encap (PUSH) routes [gerrit]
2. Add MPLS fib (SWAP) routes [gerrit]
3. Add MPLS implicit/explicit-null (POP) [gerrit]

Change to Linux Interface Plugin

1. Add MPLS interface state change callback [gerrit]
2. Forward MPLS traffic from Linux [gerrit]

*) huge thanks to Adrian *vifino* Pistol for all his work

VPP: Linux Control Plane and MPLS (cont.)



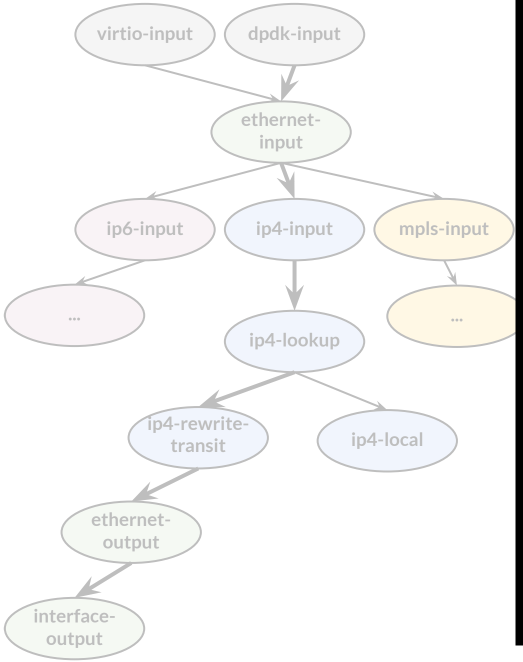
Detailed background and implementation notes:

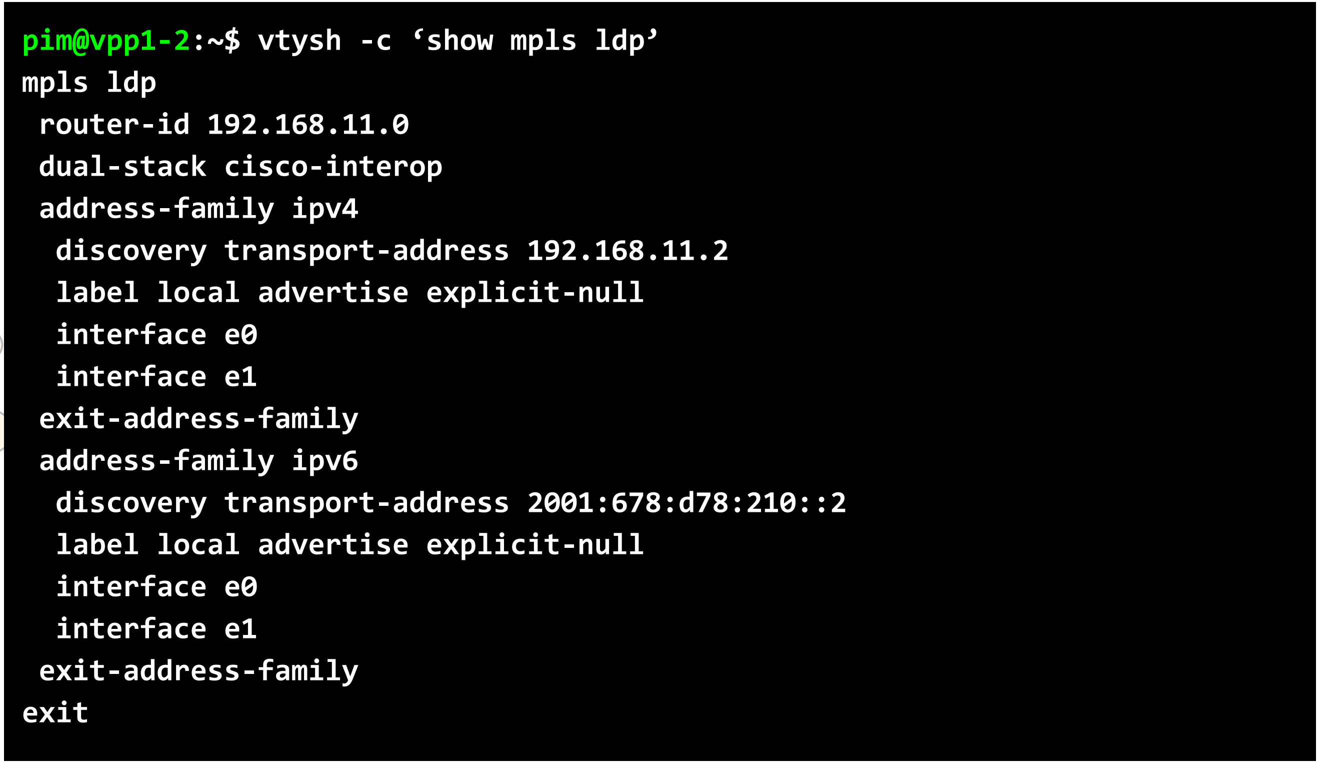
- Part 1 - MPLS anatomy in VPP
- Part 2 - MPLS Performance: LSP, Imp / Exp Null
- Part 3 - Linux CP: POP, SWAP, PUSH
- Part 4 - Linux CP: Cross connecting MPLS

Resulting Code:

- lcpng: Merged in github.com/pimvanpelt/lcpng
- linux-cp: Merged upstream in Gerrit [38702]

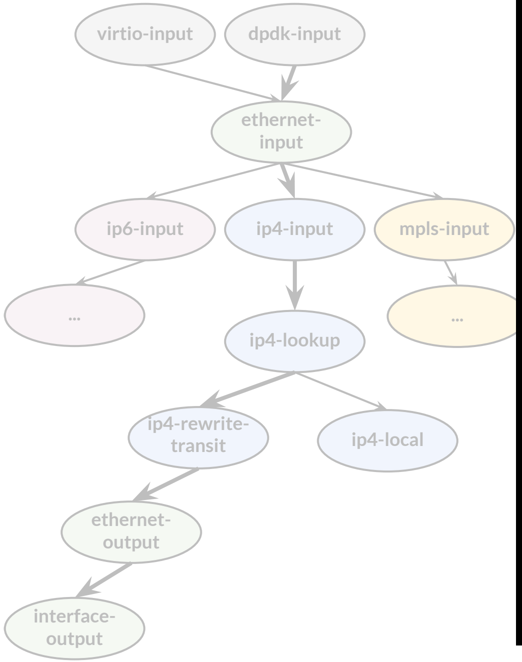
VPP: LinuxCP and MPLS and FRR

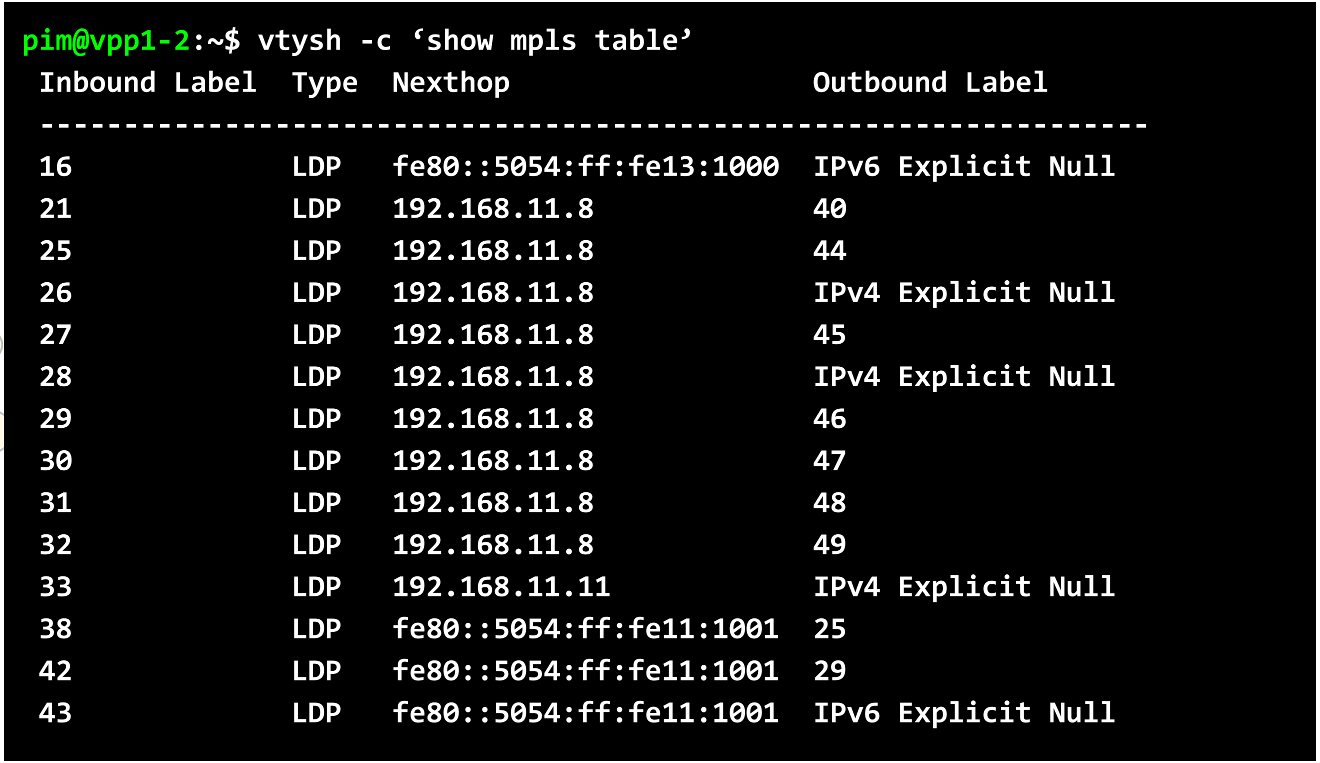




```
pim@vpp1-2:~$ vtysh -c ‘show mpls ldp’
mpls ldp
 router-id 192.168.11.0
 dual-stack cisco-interop
 address-family ipv4
  discovery transport-address 192.168.11.2
  label local advertise explicit-null
  interface e0
  interface e1
 exit-address-family
 address-family ipv6
  discovery transport-address 2001:678:d78:210::2
  label local advertise explicit-null
  interface e0
  interface e1
 exit-address-family
exit
```

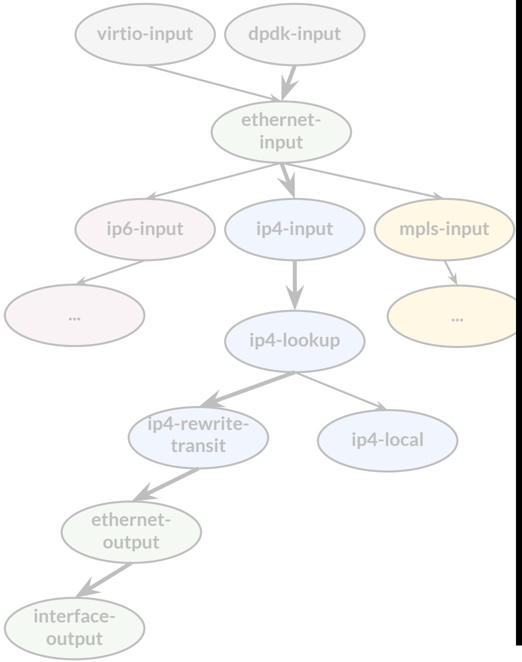
MPLS: FRR View

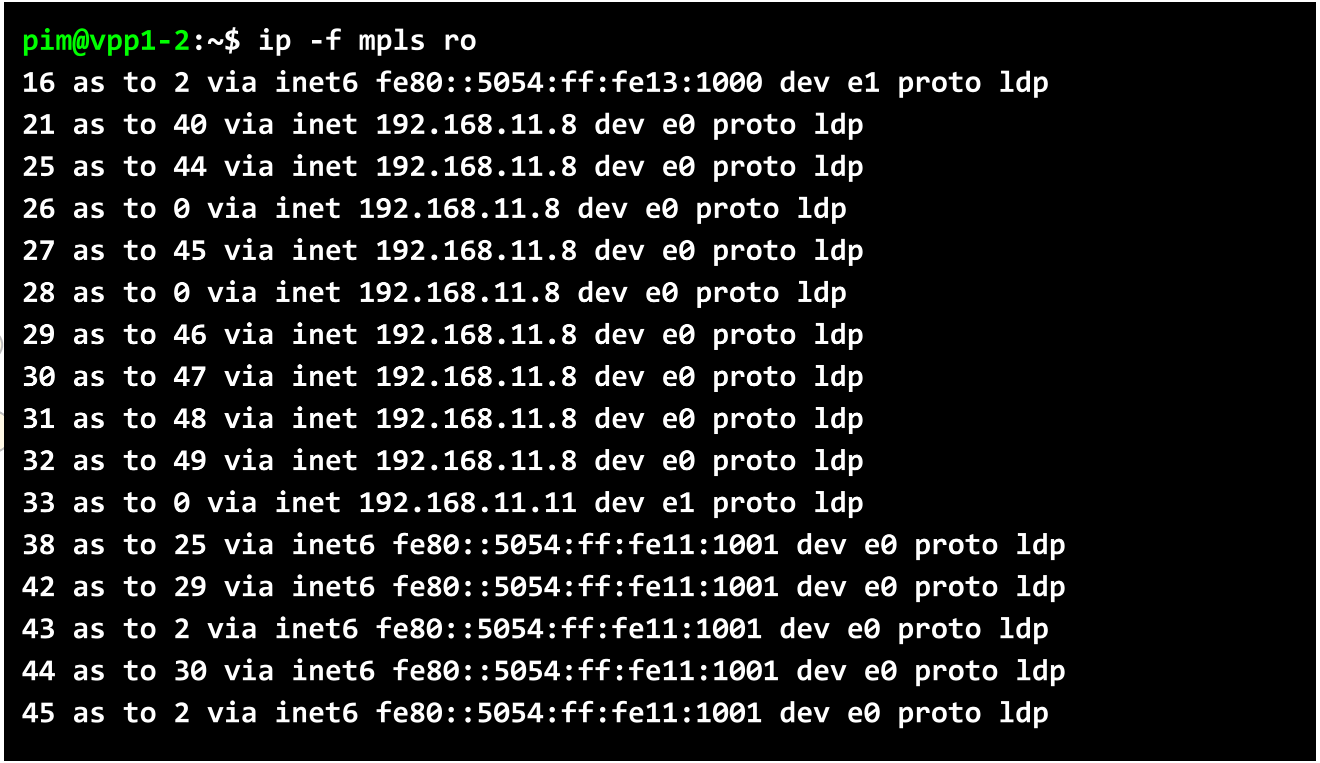




```
pim@vpp1-2:~$ vtysh -c ‘show mpls table’
 Inbound Label  Type  Nexthop                   Outbound Label
 ------------------------------------------------------------------
 16             LDP   fe80::5054:ff:fe13:1000  IPv6 Explicit Null
 21             LDP   192.168.11.8             40
 25             LDP   192.168.11.8             44
 26             LDP   192.168.11.8             IPv4 Explicit Null
 27             LDP   192.168.11.8             45
 28             LDP   192.168.11.8             IPv4 Explicit Null
 29             LDP   192.168.11.8             46
 30             LDP   192.168.11.8             47
 31             LDP   192.168.11.8             48
 32             LDP   192.168.11.8             49
 33             LDP   192.168.11.11            IPv4 Explicit Null
 38             LDP   fe80::5054:ff:fe11:1001  25
 42             LDP   fe80::5054:ff:fe11:1001  29
 43             LDP   fe80::5054:ff:fe11:1001  IPv6 Explicit Null
```

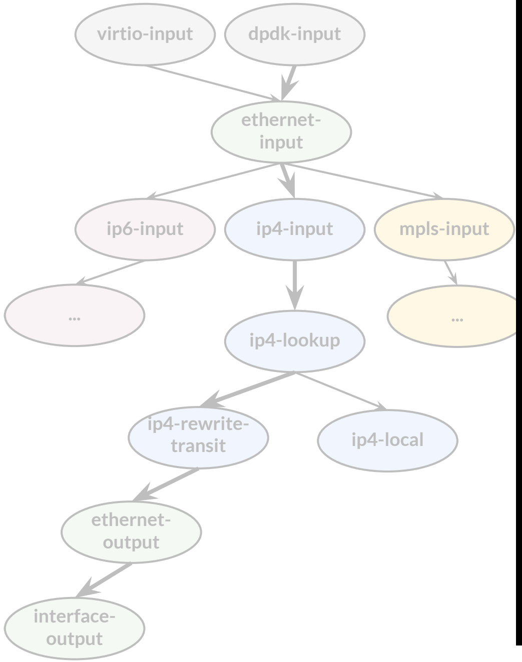
MPLS: Linux view

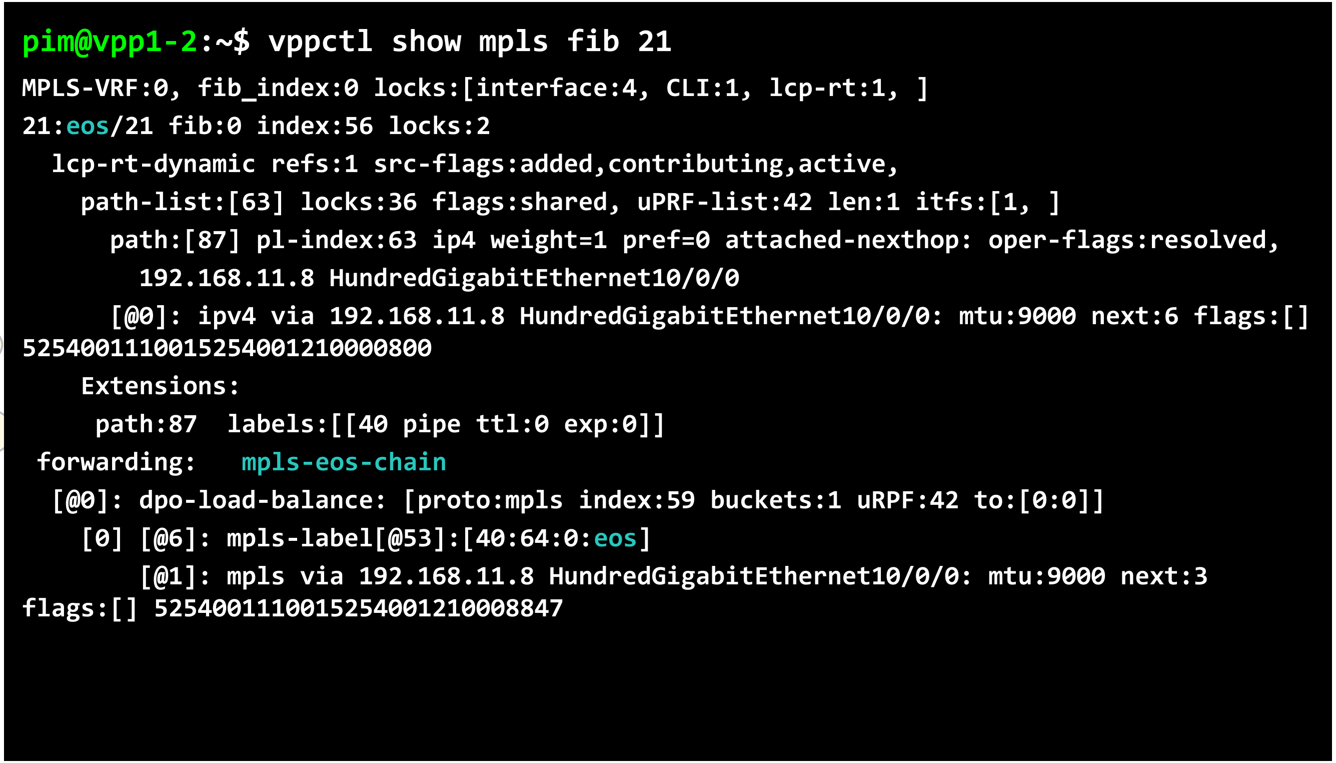




```
pim@vpp1-2:~$ ip -f mpls ro
16 as to 2 via inet6 fe80::5054:ff:fe13:1000 dev e1 proto ldp
21 as to 40 via inet 192.168.11.8 dev e0 proto ldp
25 as to 44 via inet 192.168.11.8 dev e0 proto ldp
26 as to 0 via inet 192.168.11.8 dev e0 proto ldp
27 as to 45 via inet 192.168.11.8 dev e0 proto ldp
28 as to 0 via inet 192.168.11.8 dev e0 proto ldp
29 as to 46 via inet 192.168.11.8 dev e0 proto ldp
30 as to 47 via inet 192.168.11.8 dev e0 proto ldp
31 as to 48 via inet 192.168.11.8 dev e0 proto ldp
32 as to 49 via inet 192.168.11.8 dev e0 proto ldp
33 as to 0 via inet 192.168.11.11 dev e1 proto ldp
38 as to 25 via inet6 fe80::5054:ff:fe11:1001 dev e0 proto ldp
42 as to 29 via inet6 fe80::5054:ff:fe11:1001 dev e0 proto ldp
43 as to 2 via inet6 fe80::5054:ff:fe11:1001 dev e0 proto ldp
44 as to 30 via inet6 fe80::5054:ff:fe11:1001 dev e0 proto ldp
45 as to 2 via inet6 fe80::5054:ff:fe11:1001 dev e0 proto ldp
```

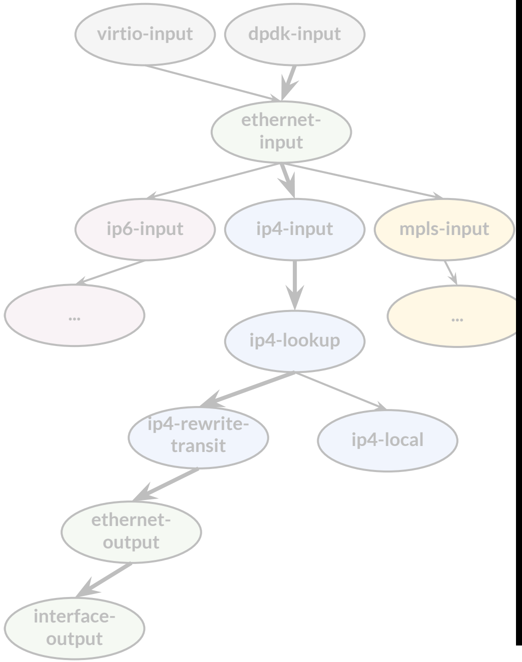
MPLS: VPP view

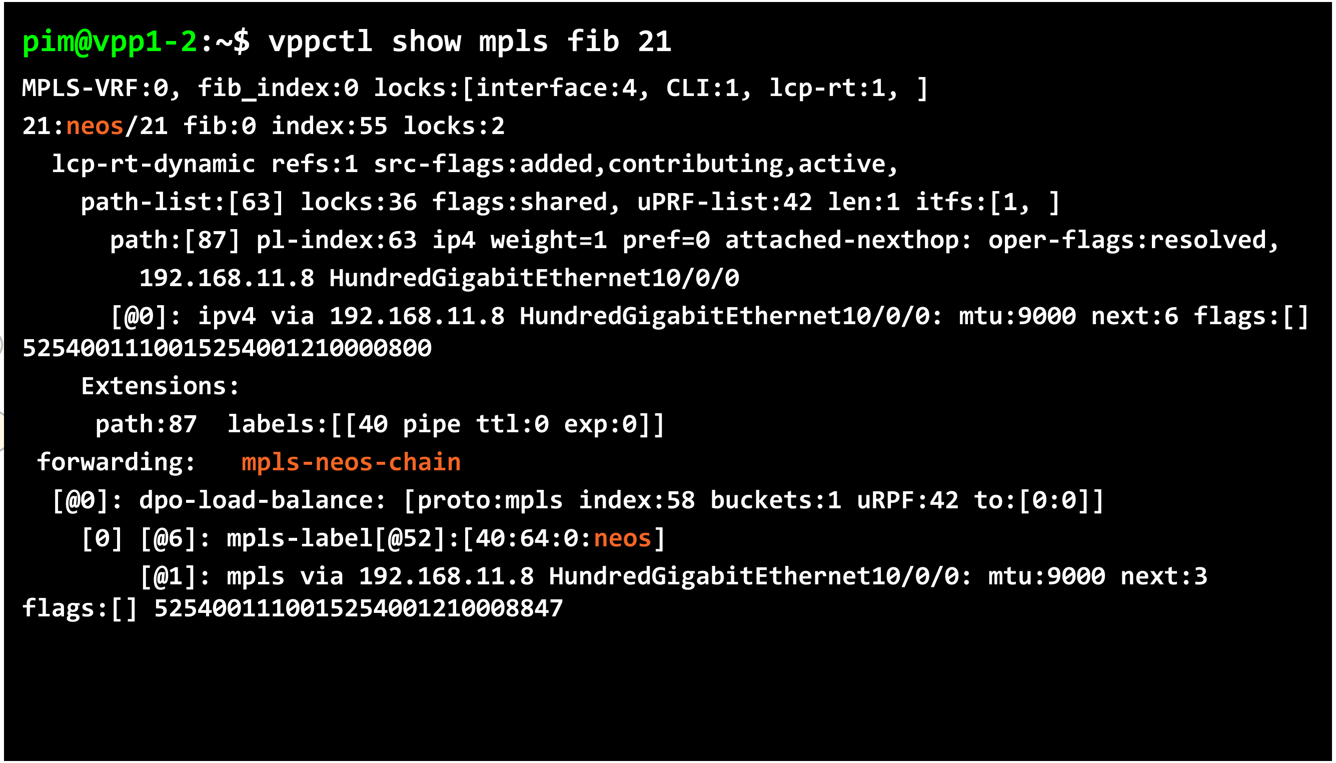




```
pim@vpp1-2:~$ vppctl show mpls fib 21
MPLS-VRF:0, fib_index:0 locks:[interface:4, CLI:1, lcp-rt:1, ]
21:eos/21 fib:0 index:56 locks:2
  lcp-rt-dynamic refs:1 src-flags:added,contributing,active,
    path-list:[63] locks:36 flags:shared, uPRF-list:42 len:1 itfs:[1, ]
      path:[87] pl-index:63 ip4 weight=1 pref=0 attached-nexthop: oper-flags:resolved,
        192.168.11.8 HundredGigabitEthernet10/0/0
      [@0]: ipv4 via 192.168.11.8 HundredGigabitEthernet10/0/0: mtu:9000 next:6 flags:[]
525400111001525400121000800
    Extensions:
     path:87  labels:[[40 pipe ttl:0 exp:0]]
 forwarding:   mpls-eos-chain
  [@0]: dpo-load-balance: [proto:mpls index:59 buckets:1 uRPF:42 to:[0:0]]
    [0] [@6]: mpls-label[@53]:[40:64:0:eos]
        [@1]: mpls via 192.168.11.8 HundredGigabitEthernet10/0/0: mtu:9000 next:3
flags:[] 525400111001525400121000847
```

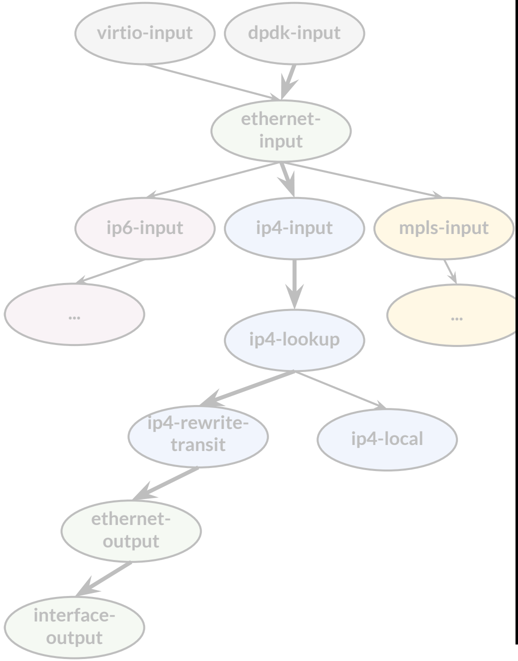
MPLS: VPP view

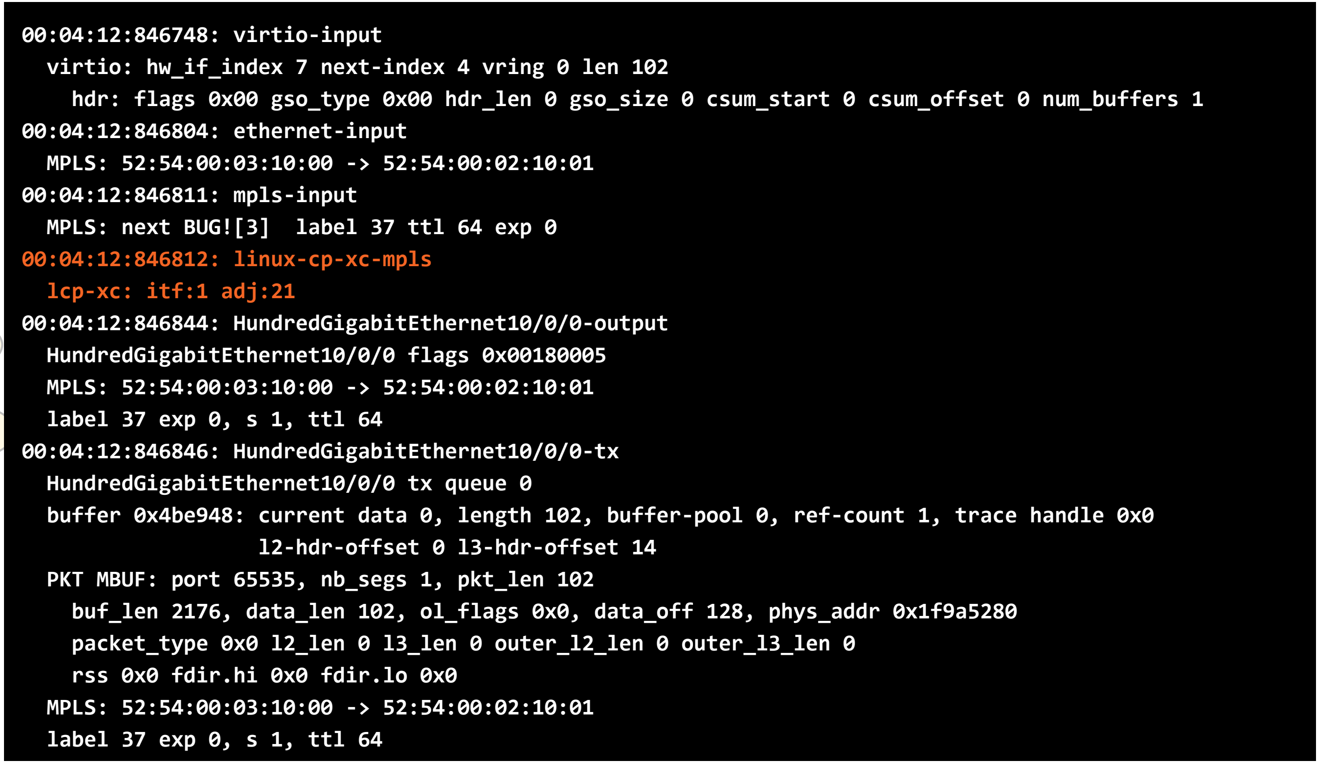




```
pim@vpp1-2:~$ vppctl show mpls fib 21
MPLS-VRF:0, fib_index:0 locks:[interface:4, CLI:1, lcp-rt:1, ]
21:neos/21 fib:0 index:55 locks:2
  lcp-rt-dynamic refs:1 src-flags:added,contributing,active,
    path-list:[63] locks:36 flags:shared, uPRF-list:42 len:1 itfs:[1, ]
      path:[87] pl-index:63 ip4 weight=1 pref=0 attached-nexthop: oper-flags:resolved,
        192.168.11.8 HundredGigabitEthernet10/0/0
      [@0]: ipv4 via 192.168.11.8 HundredGigabitEthernet10/0/0: mtu:9000 next:6 flags:[]
5254001110015254001210000800
    Extensions:
     path:87  labels:[[40 pipe ttl:0 exp:0]]
 forwarding:   mpls-neos-chain
  [@0]: dpo-load-balance: [proto:mpls index:58 buckets:1 uRPF:42 to:[0:0]]
    [0] [@6]: mpls-label[@52]:[40:64:0:neos]
        [@1]: mpls via 192.168.11.8 HundredGigabitEthernet10/0/0: mtu:9000 next:3
flags:[] 5254001110015254001210008847
```

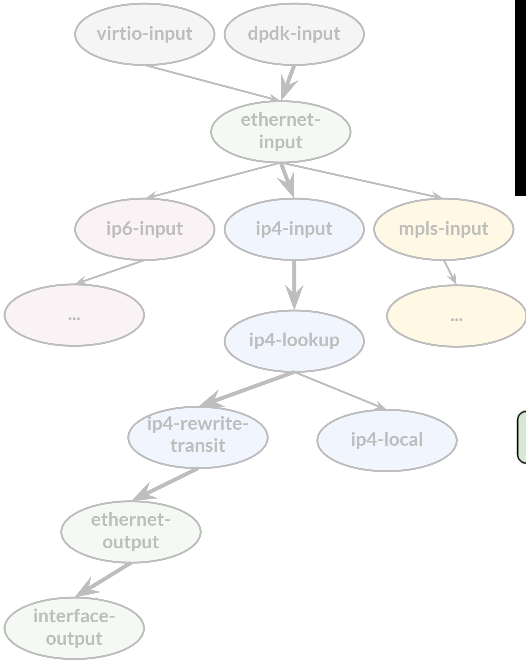
MPLS: VPP linux-cp-xc-mpls

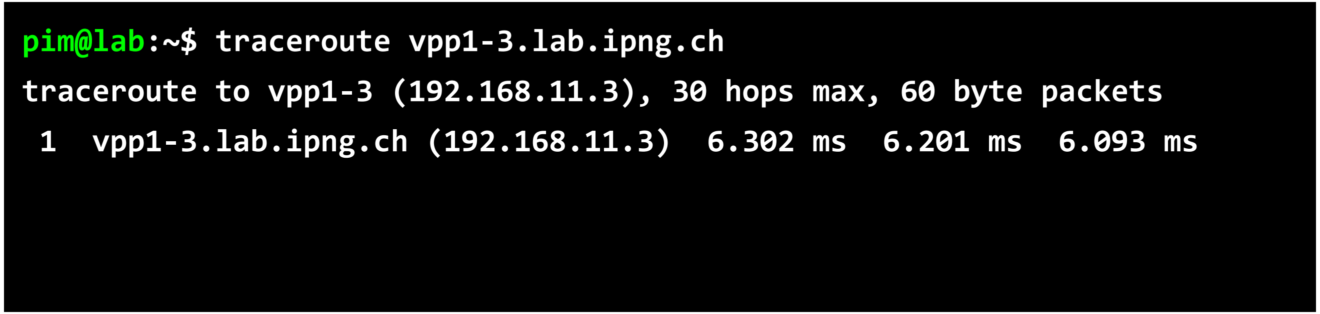




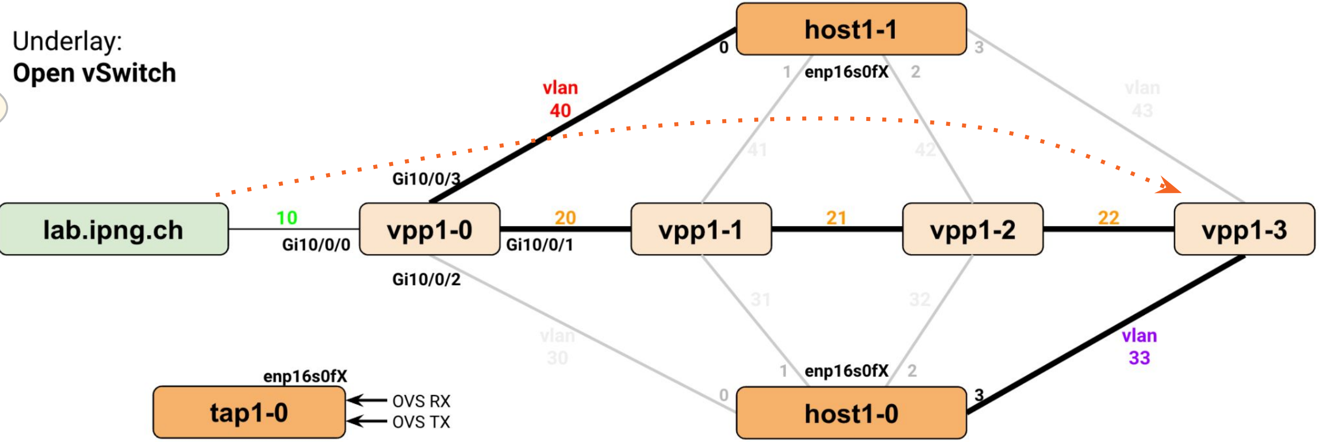
```
00:04:12:846748: virtio-input
  virtio: hw_if_index 7 next-index 4 vring 0 len 102
    hdr: flags 0x00 gso_type 0x00 hdr_len 0 gso_size 0 csum_start 0 csum_offset 0 num_buffers 1
00:04:12:846804: ethernet-input
  MPLS: 52:54:00:03:10:00 -> 52:54:00:02:10:01
00:04:12:846811: mpls-input
  MPLS: next BUG![3]  label 37 ttl 64 exp 0
00:04:12:846812: linux-cp-xc-mpls
  lcp-xc: itf:1 adj:21
00:04:12:846844: HundredGigabitEthernet10/0/0-output
  HundredGigabitEthernet10/0/0 flags 0x00180005
  MPLS: 52:54:00:03:10:00 -> 52:54:00:02:10:01
  label 37 exp 0, s 1, ttl 64
00:04:12:846846: HundredGigabitEthernet10/0/0-tx
  HundredGigabitEthernet10/0/0 tx queue 0
  buffer 0x4be948: current data 0, length 102, buffer-pool 0, ref-count 1, trace handle 0x0
                   l2-hdr-offset 0 l3-hdr-offset 14
  PKT MBUF: port 65535, nb_segs 1, pkt_len 102
    buf_len 2176, data_len 102, ol_flags 0x0, data_off 128, phys_addr 0x1f9a5280
    packet_type 0x0 l2_len 0 l3_len 0 outer_l2_len 0 outer_l3_len 0
    rss 0x0 fdir.hi 0x0 fdir.lo 0x0
  MPLS: 52:54:00:03:10:00 -> 52:54:00:02:10:01
  label 37 exp 0, s 1, ttl 64
```

VPP: With MPLS + Linux CP

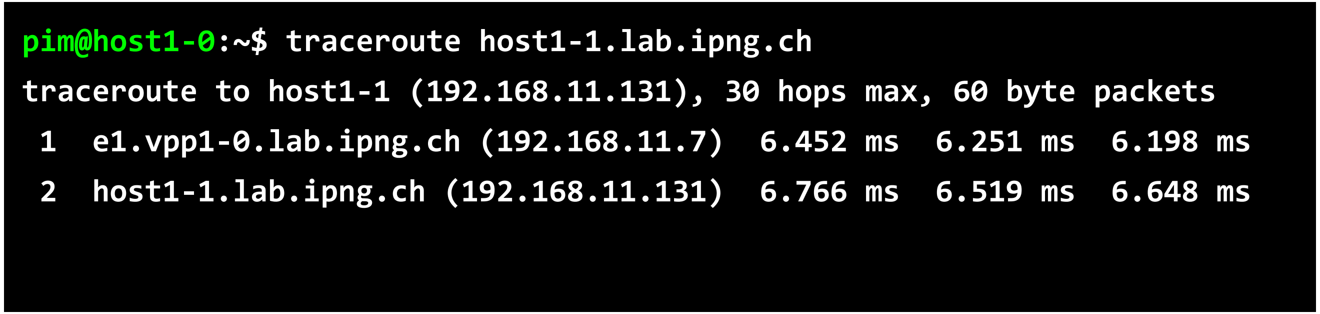




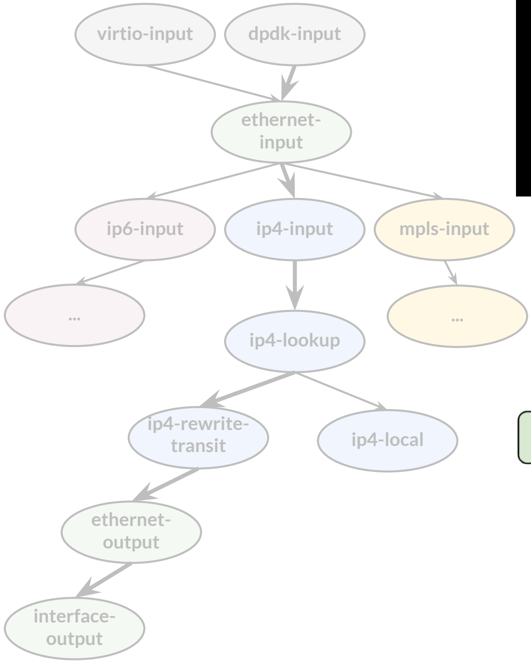
```
pim@lab:~$ traceroute vpp1-3.lab.ipng.ch
traceroute to vpp1-3 (192.168.11.3), 30 hops max, 60 byte packets
 1  vpp1-3.lab.ipng.ch (192.168.11.3)  6.302 ms  6.201 ms  6.093 ms
```



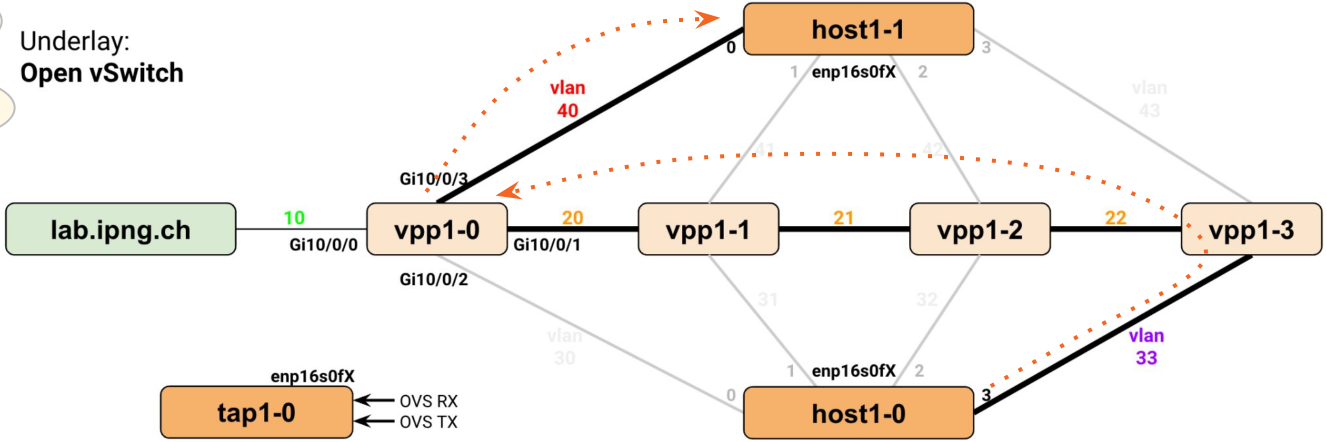
VPP: With MPLS + Linux CP



```
pim@host1-0:~$ traceroute host1-1.lab.ipng.ch
traceroute to host1-1 (192.168.11.131), 30 hops max, 60 byte packets
 1  e1.vpp1-0.lab.ipng.ch (192.168.11.7)  6.452 ms  6.251 ms  6.198 ms
 2  host1-1.lab.ipng.ch (192.168.11.131)  6.766 ms  6.519 ms  6.648 ms
```



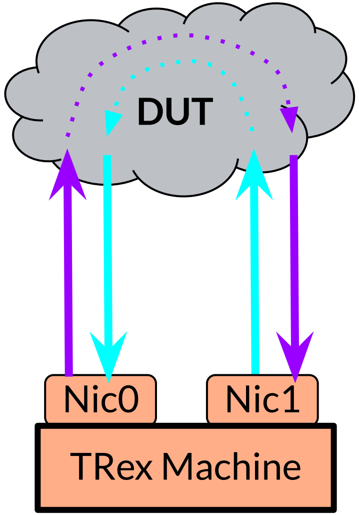
Underlay:
Open vSwitch



Act 3: Performance of VPP

TRex
Realistic Traffic Generator

Config and Startup



Simple configuration:

```
- version: 2
  interfaces: ['5:00.0', '5:00.1']
  port_info:
        - src_mac   : 9c:69:b4:61:ff:40 # T-Rex Nic0
          dest_mac  : 3c:ec:ef:c6:fb:26 # DUT MAC A
        - src_mac   : 9c:69:b4:61:ff:41 # T-Rex Nic1
          dest_mac  : 3c:ec:ef:6a:80:db # DUT MAC B
```

Startup:

```
$ sudo ./t-rex-64 -i -c 6
$ ./trex-console
```

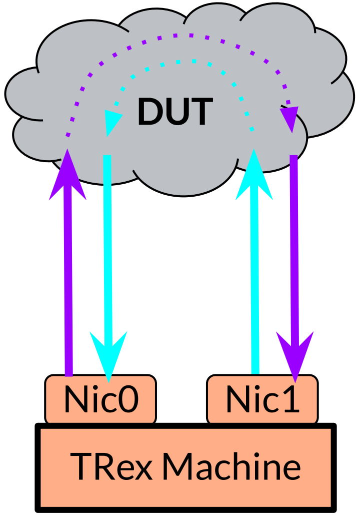
Config and Startup

Simple configuration:

```
- version: 2
  interfaces: ['5:00.0', '5:00.1']
  port_info:
      - ip        : 100.65.1.2        # T-Rex Nic0
        default_gw: 100.65.1.1        # DUT IPv4 A
      - ip        : 100.65.2.2        # T-Rex Nic1
        default_gw: 100.65.2.1        # DUT IPv4 B
```

Startup:

```
$ sudo ./t-rex-64 -i -c 6
$ ./trex-console
```



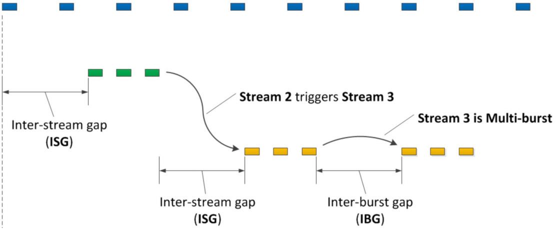
Realistic Traffic Generator

Stateless Traffic Profiles

Assemble packet streams with scapy:

- IPv4/IPv6 src/dst; proto; port src/dst; size; ratios; timings

```
self.ip_range = {'src': {'start': "16.0.0.1", 'end': "16.0.0.254"},
                 'dst': {'start': "48.0.0.1",  'end': "48.0.0.254"}}
```



```
# default IMIX properties
self.imix_table = [ {'size': 60,   'pps': 28,  'isg':0 },
                    {'size': 590,  'pps': 16,  'isg':0.1 },
                    {'size': 1514, 'pps': 4,   'isg':0.2 } ]
```

- Streams are *applied* on one or more ports
- Ports are configured to send a *rate* of traffic (bps, pps or % of line)

TRex
Realistic Traffic Generator

Method - Saturation Loadtest

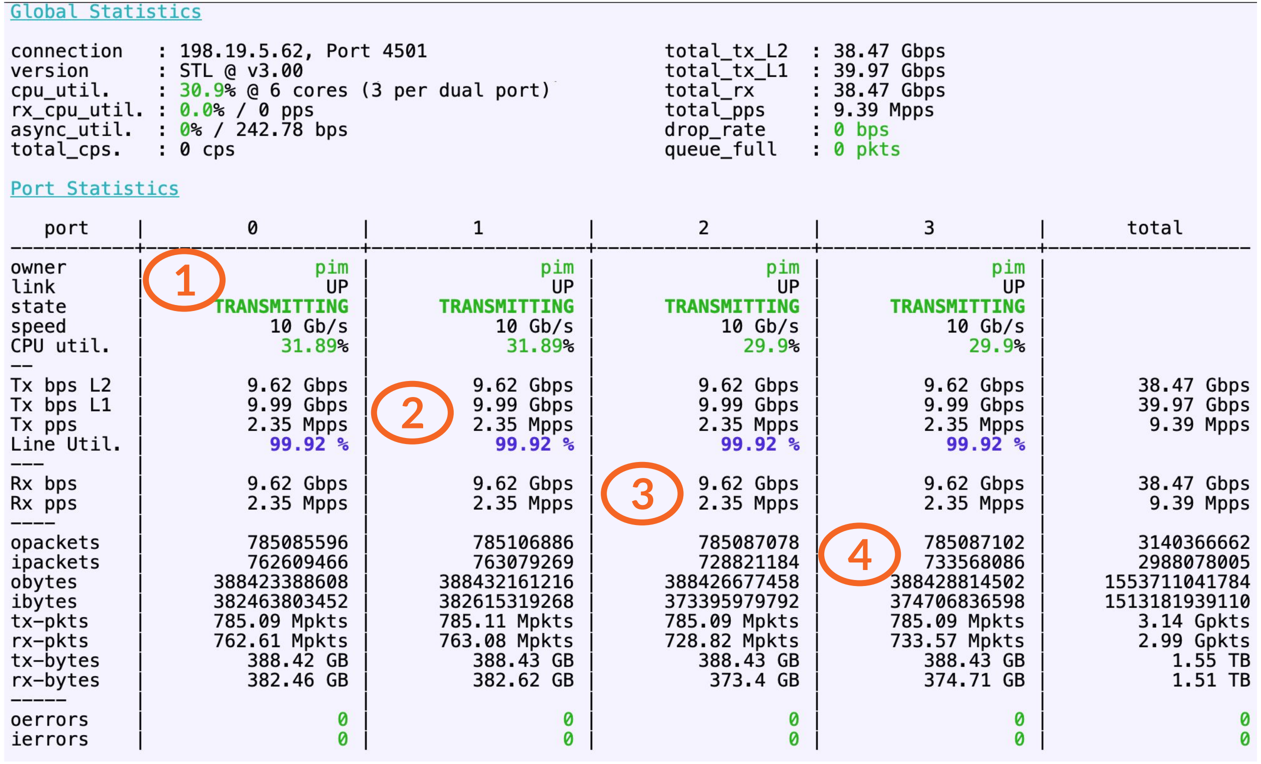
Legend:

1. NIC Info, T-Rex CPU utilization
2. Sent traffic (L1, L2, packets/sec)
3. Received traffic (L2, packets/sec)
4. Detailed packet/byte counters

Shown:

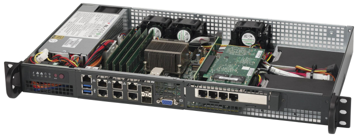
- 40Gbps, 512b frames: 9.4Mpps
- Using ~32% CPU on the T-Rex
- 4x10Gbps all making it through

Spoiler: MPLS works fine :)



```
Global Statistics

connection   : 198.19.5.62, Port 4501                      total_tx_L2  : 38.47 Gbps
version      : STL @ v3.00                                 total_tx_L1  : 39.97 Gbps
cpu_util.    : 30.9% @ 6 cores (3 per dual port)           total_rx     : 38.47 Gbps
rx_cpu_util. : 0.0% / 0 pps                                total_pps    : 9.39 Mpps
async_util.  : 0% / 242.78 bps                             drop_rate    : 0 bps
total_cps.   : 0 cps                                       queue_full   : 0 pkts

Port Statistics

   port    |         0          |         1          |         2          |         3          |       total
-----------+--------------------+--------------------+--------------------+--------------------+------------------
owner      |                pim |                pim |                pim |                pim |
link       |                 UP |                 UP |                 UP |                 UP |
state      |       TRANSMITTING |       TRANSMITTING |       TRANSMITTING |       TRANSMITTING |
speed      |            10 Gb/s |            10 Gb/s |            10 Gb/s |            10 Gb/s |
CPU util.  |             31.89% |             31.89% |              29.9% |              29.9% |
--         |                    |                    |                    |                    |
Tx bps L2  |          9.62 Gbps |          9.62 Gbps |          9.62 Gbps |          9.62 Gbps |        38.47 Gbps
Tx bps L1  |          9.99 Gbps |          9.99 Gbps |          9.99 Gbps |          9.99 Gbps |        39.97 Gbps
Tx pps     |          2.35 Mpps |          2.35 Mpps |          2.35 Mpps |          2.35 Mpps |         9.39 Mpps
Line Util. |            99.92 % |            99.92 % |            99.92 % |            99.92 % |
---        |                    |                    |                    |                    |
Rx bps     |          9.62 Gbps |          9.62 Gbps |          9.62 Gbps |          9.62 Gbps |        38.47 Gbps
Rx pps     |          2.35 Mpps |          2.35 Mpps |          2.35 Mpps |          2.35 Mpps |         9.39 Mpps
----       |                    |                    |                    |                    |
opackets   |          785085596 |          785106886 |          785087078 |          785087102 |        3140366662
ipackets   |          762609466 |          763079269 |          728821184 |          733568086 |        2988078005
obytes     |       388423388608 |       388432161216 |       388426677458 |       388428814502 |     1553711041784
ibytes     |       382463803452 |       382615319268 |       373395979792 |       374706836598 |     1513181939110
tx-pkts    |        785.09 Mpkts|        785.11 Mpkts|        785.09 Mpkts|        785.09 Mpkts|        3.14 Gpkts
rx-pkts    |        762.61 Mpkts|        763.08 Mpkts|        728.82 Mpkts|        733.57 Mpkts|        2.99 Gpkts
tx-bytes   |          388.42 GB |          388.43 GB |          388.43 GB |          388.43 GB |           1.55 TB
rx-bytes   |          382.46 GB |          382.62 GB |           373.4 GB |          374.71 GB |           1.51 TB
-----      |                    |                    |                    |                    |
oerrors    |                  0 |                  0 |                  0 |                  0 |                 0
ierrors    |                  0 |                  0 |                  0 |                  0 |                 0
```

VPP: IPv4 Performance



Supermicro 5018D-FN8T

CPU: Xeon D1518 • 2.2GHz • 4C/8T

RAM:256kB L1, 1MB L2, 6MB L3, 32GB DDR4

Disk: 128GB mSATA

Price: CHF 1’350,-

NICs: 2x 25GbE SFP28
2x 10GbE SFP+
2x 1GbE i210
4x 1GbE i350

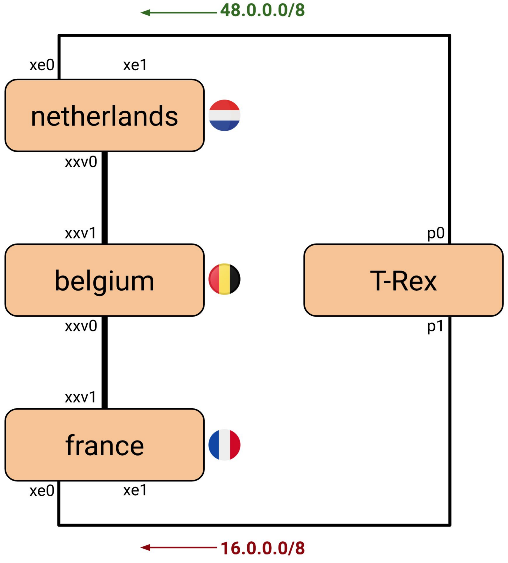
VPP Configuration:

- **3x DPDK threads, each NIC has 3x RX/TX queues with RSS**
- **IPv4 Routing: 9.31Mpps per core**

CPU is 35W TDP and hyperthreaded (4C/8T)
Hyperthreading reduces from 9.31Mpps/core to 6.30Mpps/core (but 8 threads!)
At 6 threads: 37.8Mpps forwarding at 48W → 1.56μJ per packet

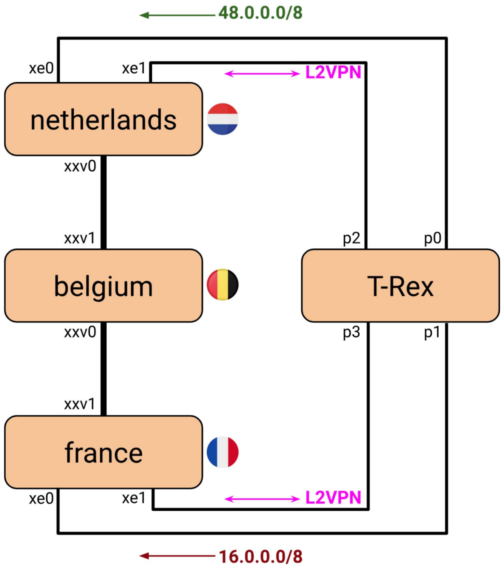
TRex
Realistic Traffic Generator

VPP: MPLS Configuration (L3)



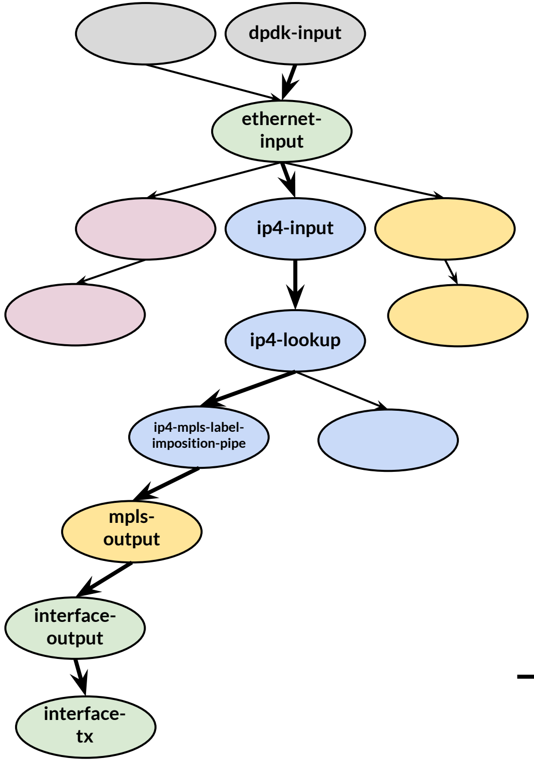
```
netherlands# set interface ip address xe0 100.64.1.2/30
netherlands# set interface state xe0 up
netherlands# ip route add 16.0.0.0/8 via 100.64.1.1
netherlands# ip route add 48.0.0.0/8 via 192.168.13.6 xxv0 out-labels 33
netherlands# mpls local-label add 31 eos via ip4-lookup-in-table 0

belgium# mpls local-label add 33 eos via 192.168.13.4 xxv0 out-labels 33
belgium# mpls local-label add 31 eos via 192.168.13.7 xxv1 out-labels 31

france# set interface ip address xe0 100.64.2.2/30
france# set interface state xe0 up
france# ip route add 48.0.0.0/8 via 100.64.2.1
france# ip route add 16.0.0.0/8 via 192.168.13.5 xxv1 out-labels 31
france# mpls local-label add 33 eos via ip4-lookup-in-table 0
```

VPP: MPLS Configuration (L2)



```
netherlands# mpls tunnel l2-only via 192.168.13.6 xxv0 out-labels 331
netherlands# mpls local-label 311 eos via l2-input-on mpls-tunnel0
netherlands# set interface state mpls-tunnel0 up
netherlands# set interface l2 xconnect xe1 mpls-tunnel0
netherlands# set interface l2 xconnect mpls-tunnel0 xe1

belgium# mpls local-label add 331 eos via 192.168.13.4 xxv0 out-labels 331
belgium# mpls local-label add 311 eos via 192.168.13.7 xxv1 out-labels 311

france# mpls tunnel l2-only via 192.168.13.5 xxv1 out-labels 311
france# mpls local-label 331 eos via l2-input-on mpls-tunnel0
france# set interface state mpls-tunnel0 up
france# set interface l2 xconnect xe1 mpls-tunnel0
france# set interface l2 xconnect mpls-tunnel0 xe1
```

VPP: Feature Performance

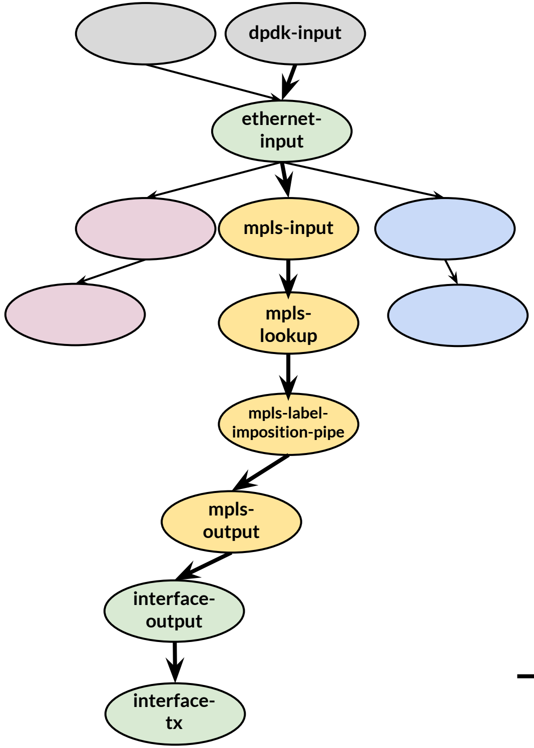


What is *Netherlands (PE-Ingress)* actually doing?

`dpdk-input`	Receives packets from DPDK
`ethernet-input`	Handles ingress Ethernet packets
`ip4-input-no-checksum`	Handles IPv4 packets (w/ hardware cksum offload)
`ip4-lookup`	Performs IPv4 FIB lookups
`ip4-mpls-label-impo…`	Encapsulates packets as MPLS
`mpls-output`	Handles egress MPLS packets
`interface-output`	Handles L2 lookups for (ethernet) nexthops
`interface-tx`	Sends packets to DPDK for marshalling

VPP: Feature Performance

What about *Belgium (P-Router)*?



`dpdk-input`	Receives packets from DPDK
`ethernet-input`	Handles ingress Ethernet packets
`mpls-input`	Handles MPLS packets
`mpls-lookup`	Performs MPLS FIB lookups
`mpls-label-imposit...`	Encapsulates packets as MPLS
`mpls-output`	Handles egress MPLS packets
`interface-output`	Handles L2 lookups for (ethernet) nexthops
`interface-tx`	Sends packets to DPDK for marshalling

VPP: Show runtime (CLI)

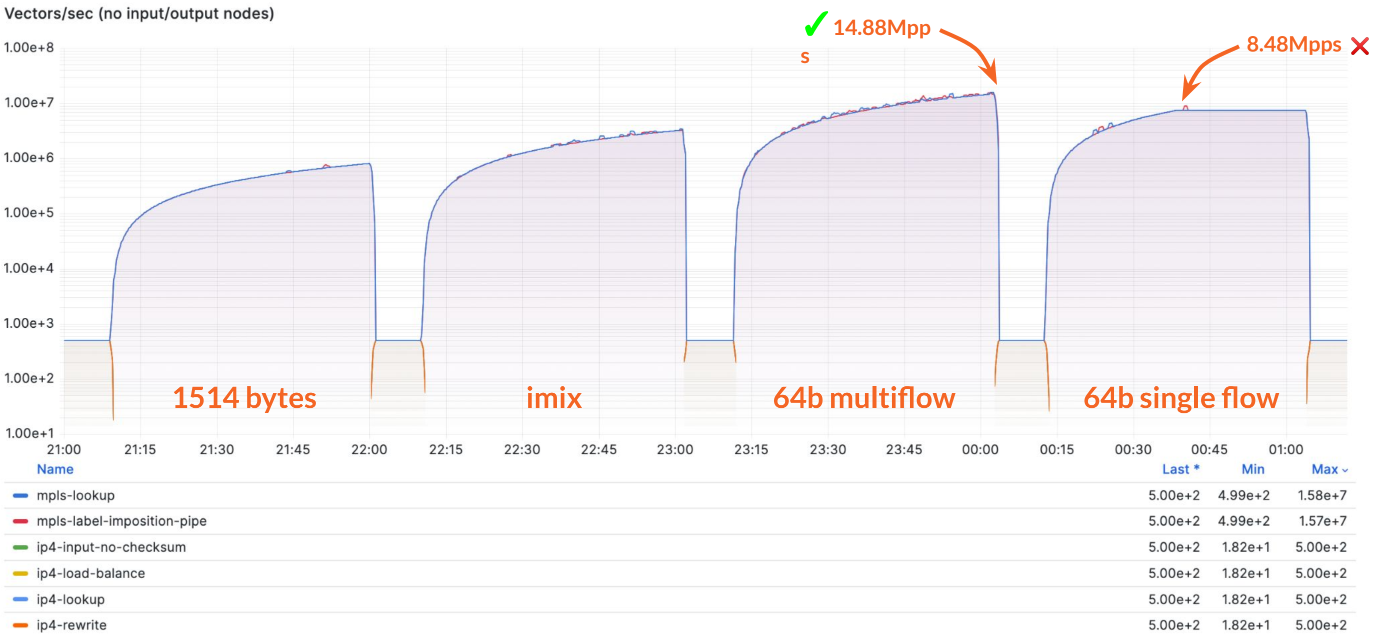
Belgium - P Router:

```
Time 16.8, 10 sec internal node vector rate 256.00 loops/sec 10711.51
  vector rates in 8.4848e6, out 8.4848e6, drop 0.0000e0, punt 1.1916e-1
             Name                 State         Calls          Vectors        Clocks  Vectors/Call
dpdk-input                       polling         185435      142414081        4.58e1        768.00
ethernet-input                   active          556306      142414081        1.84e1        255.99
mpls-input                       active          556305      142414079        1.86e1        255.99
mpls-label-imposition-pipe       active          556305      142414079        2.17e1        255.99
mpls-lookup                      active          556305      142414079        2.81e1        255.99
mpls-output                      active          556305      142414079        2.88e1        255.99
xxv0-output                      active          556305      142414079        6.96e0        255.99
xxv0-tx                          active          556305      142414079        8.90e1        255.99
```

VPP exports very precise time bookkeeping:

sum(clocks) = 259; CPU clockspeed = 2.2GHz ⇒ 8.49Mpps

VPP: Show runtime (API)



Vectors/sec (no input/output nodes)

✓ 14.88Mpp
s

8.48Mpps ✗

1514 bytes

imix

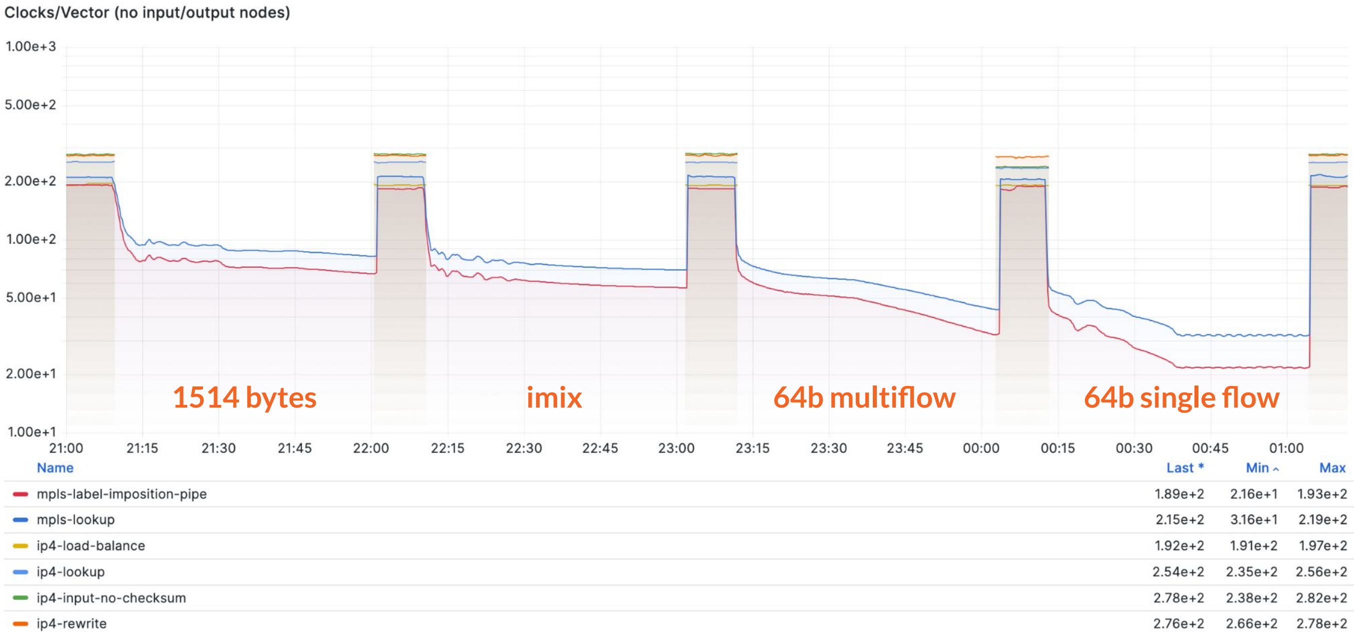
64b multiflow

64b single flow

Name	Last *	Min	Max ⌄
mpls-lookup	5.00e+2	4.99e+2	1.58e+7
mpls-label-imposition-pipe	5.00e+2	4.99e+2	1.57e+7
ip4-input-no-checksum	5.00e+2	1.82e+1	5.00e+2
ip4-load-balance	5.00e+2	1.82e+1	5.00e+2
ip4-lookup	5.00e+2	1.82e+1	5.00e+2
ip4-rewrite	5.00e+2	1.82e+1	5.00e+2

CPU saturation in 64b single flow: flatlines at 8.48Mpps

VPP: Show runtime (API)



1514 bytes

imix

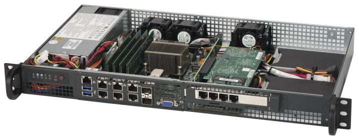
64b multiflow

64b single flow

Name	Last *	Min ^	Max
🟥 mpls-label-imposition-pipe	1.89e+2	2.16e+1	1.93e+2
🟦 mpls-lookup	2.15e+2	3.16e+1	2.19e+2
🟨 ip4-load-balance	1.92e+2	1.91e+2	1.97e+2
🟦 ip4-lookup	2.54e+2	2.35e+2	2.56e+2
🟩 ip4-input-no-checksum	2.78e+2	2.38e+2	2.82e+2
🟧 ip4-rewrite	2.76e+2	2.66e+2	2.78e+2

CPU time ~10x improvement under load (193 → 21.6 clocks/packet)

VPP: MPLS Performance



Create bottleneck by forcing VPP to use only one CPU, example:

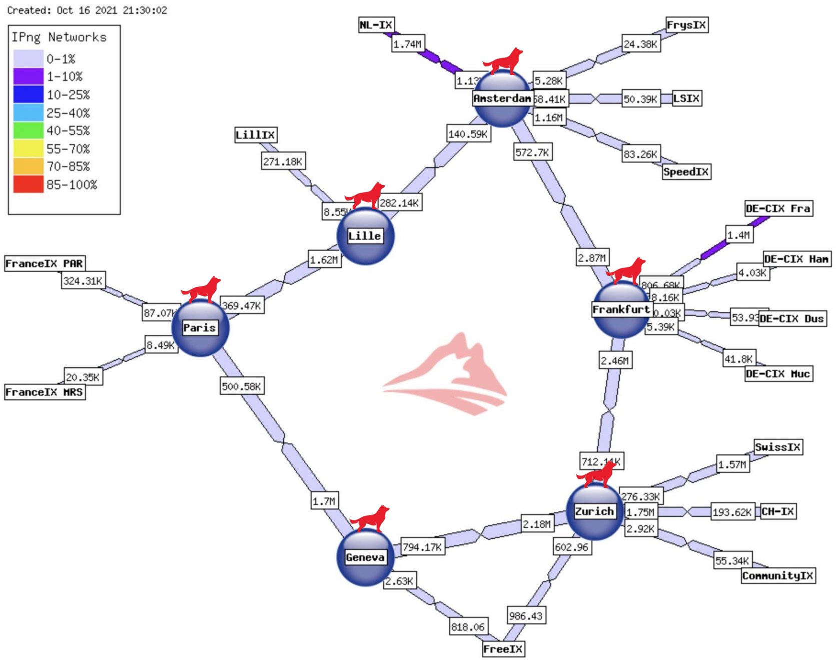
```
> set interface rx-placement xxv0 queue [0-2] thread 0
```

Results (*per core* of Xeon D1518):

PE IPv4 Ingress:	7.43Mpps
P Router:	8.48Mpps
PE IPv4 Egress:	7.37Mpps
P Router w/ PHP:	8.94Mpps
PHP IPv4 Egress:	9.31Mpps (= IPv4 Router)
PE L2VPN Ingress:	5.40Mpps
PE L2VPN Egress:	8.65Mpps

```
vpp v23.10-rc0~12-g94a0f57a6 built by pim on hippo at 2023-05-31T10:40:22
```

Questions, Discussion



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

Useful Resources

- VPP: fd.io
- VPP Linux CP: Github
- Articles: ipng.ch
- Mastodon: @IPngNetworks

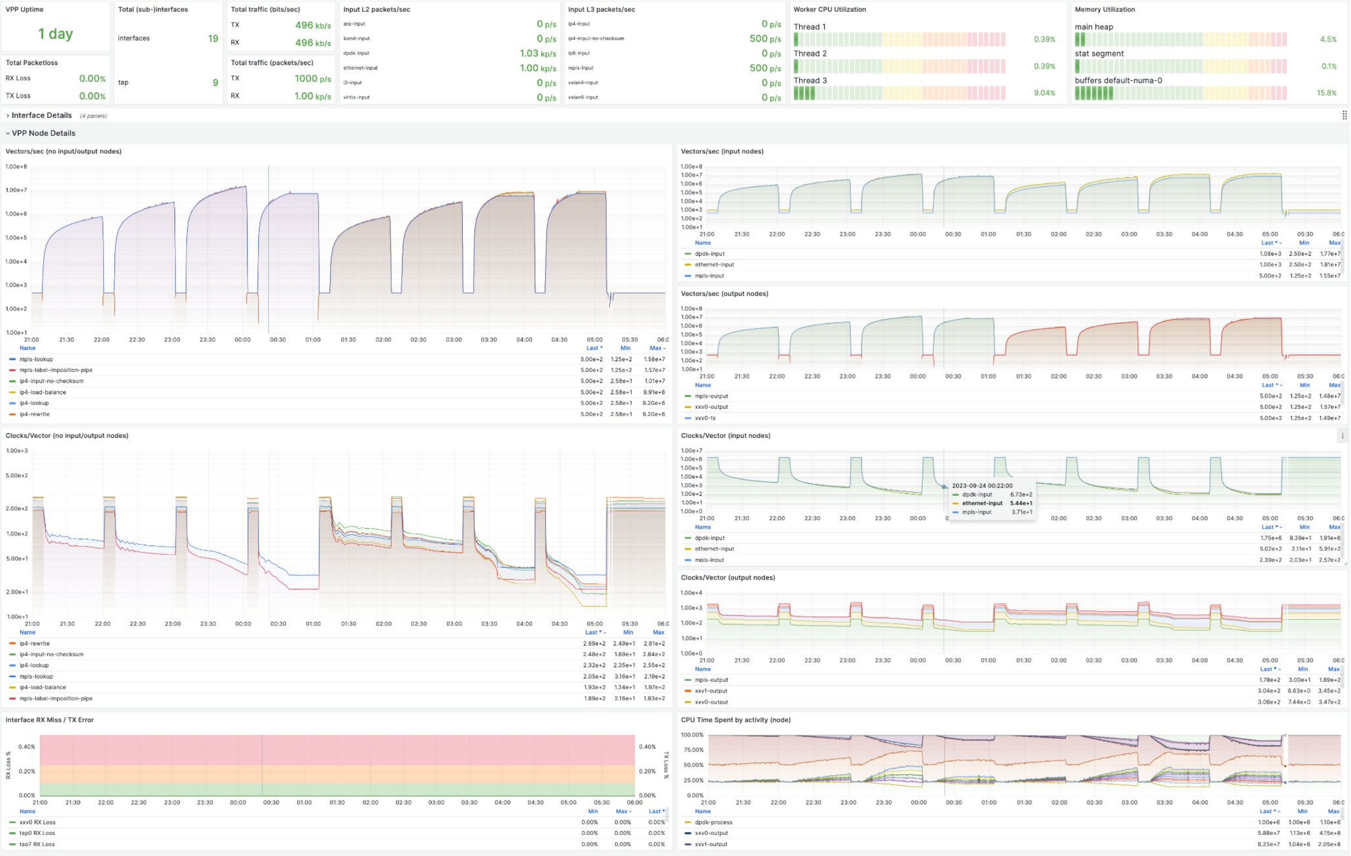
Also: thanks for listening!

BONUS material

Bonus Slide:

Grafana VPP Dashboard

Coming soon!



VPP Uptime

1 day

Total Packetloss

RX Loss	0.00%
TX Loss	0.00%

Total (sub-)interfaces

interfaces	19
tap	9

Total traffic (bits/sec)

TX	496 kb/s
RX	496 kb/s

Total traffic (packets/sec)

TX	1000 p/s
RX	1.00 kp/s

Input L2 packets/sec

arp-input	0 p/s
bond-input	0 p/s
dpdk-input	1.03 kp/s
ethernet-input	1.00 kp/s
l2-input	0 p/s
virtio-input	0 p/s

Input L3 packets/sec

ip4-input	0 p/s
ip4-input-no-checksum	500 p/s
ip6-input	0 p/s
mpls-input	500 p/s
vxlan4-input	0 p/s
vxlan6-input	0 p/s

Worker CPU Utilization

Thread 1	0.39%
Thread 2	0.39%
Thread 3	9.04%

Memory Utilization

main heap	4.5%
stat segment	0.1%
buffers default-numa-0	15.8%

› Interface Details *(4 panels)*

⌄ VPP Node Details

Vectors/sec (no input/output nodes)

Name	Last *	Min	Max ⌄
mpls-lookup	5.00e+2	1.25e+2	1.58e+7
mpls-label-imposition-pipe	5.00e+2	1.25e+2	1.57e+7
ip4-input-no-checksum	5.00e+2	2.58e+1	1.01e+7
ip4-load-balance	5.00e+2	2.58e+1	9.91e+6
ip4-lookup	5.00e+2	2.58e+1	9.20e+6
ip4-rewrite	5.00e+2	2.58e+1	9.20e+6

Vectors/sec (input nodes)

Name	Last * ⌄	Min	Max
dpdk-input	1.08e+3	2.50e+2	1.77e+7
ethernet-input	1.00e+3	2.50e+2	1.81e+7
mpls-input	5.00e+2	1.25e+2	1.55e+7

Vectors/sec (output nodes)

Name	Last * ⌄	Min	Max
mpls-output	5.00e+2	1.25e+2	1.48e+7
xxv0-output	5.00e+2	1.25e+2	1.57e+7
xxv0-tx	5.00e+2	1.25e+2	1.49e+7

Clocks/Vector (no input/output nodes)

Name	Last * ⌄	Min	Max
ip4-rewrite	2.69e+2	2.49e+1	2.81e+2
ip4-input-no-checksum	2.48e+2	1.89e+1	2.84e+2
ip4-lookup	2.32e+2	2.35e+1	2.55e+2
mpls-lookup	2.05e+2	3.16e+1	2.19e+2
ip4-load-balance	1.93e+2	1.34e+1	1.97e+2
mpls-label-imposition-pipe	1.89e+2	2.16e+1	1.93e+2

Clocks/Vector (input nodes)

2023-09-24 00:22:00

dpdk-input	6.73e+2
ethernet-input	5.44e+1
mpls-input	3.71e+1

Name	Last * ⌄	Min	Max
dpdk-input	1.75e+6	8.39e+1	1.91e+6
ethernet-input	5.02e+2	2.11e+1	5.91e+2
mpls-input	2.39e+2	2.03e+1	2.57e+2

Clocks/Vector (output nodes)

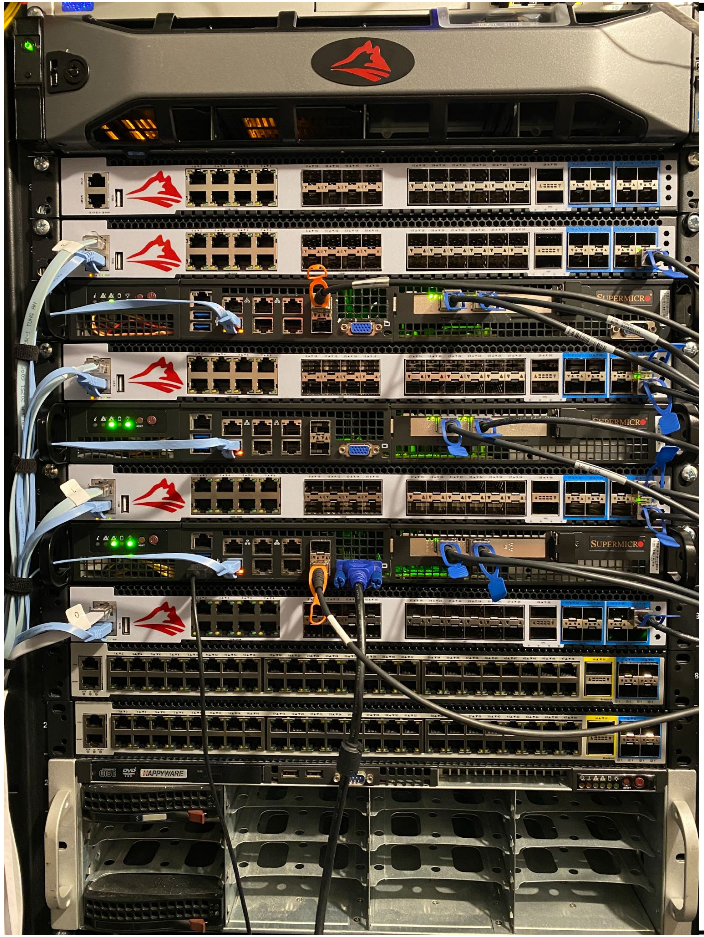
Name	Last * ^	Min	Max
mpls-output	1.78e+2	3.00e+1	1.89e+2
xxv1-output	3.04e+2	6.63e+0	3.45e+2
xxv0-output	3.06e+2	7.44e+0	3.47e+2

Interface RX Miss / TX Error

Name	Min	Max ⌄	Last *
xxv0 RX Loss	0.00%	0.00%	0.00%
tap0 RX Loss	0.00%	0.00%	0.00%
tap7 RX Loss	0.00%	0.00%	0.00%

CPU Time Spent by activity (node)

Name	Last * ^	Min	Max
dpdk-process	1.00e+6	1.00e+6	1.10e+6
xxv0-output	5.88e+7	1.13e+6	4.15e+8
xxv1-output	6.23e+7	1.04e+6	2.05e+8



The rack in my basement upon which this code was written and these loadtests performed.

Bottom:	hippo.net.ipng.ch (Ryzen 5950X)
Switches:	Centec S5612X
Sandwich:	Supermicro Xeon D1518 (5018D-FN8T)
Switches:	Centec S5548-4X
Top:	lab.ipng.ch (+ VMs)

Call to Action:

I'd like to break the Terabit/sec and Billion packets/sec barrier with VPP. If you want to help out with this, reach out to <pim@ipng.ch>.

September 26, 2003: Happy 20th Outbreak Day anniversary

This useless trivia brought to you by HBO and YouTube.