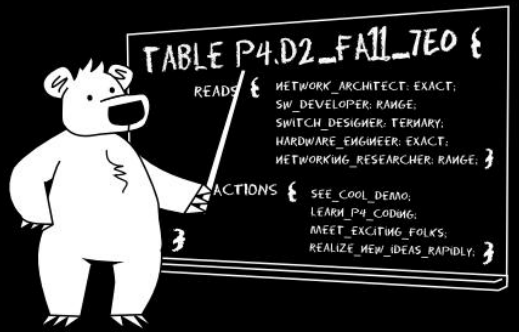
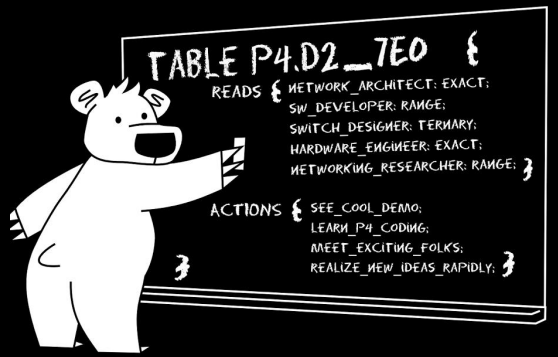




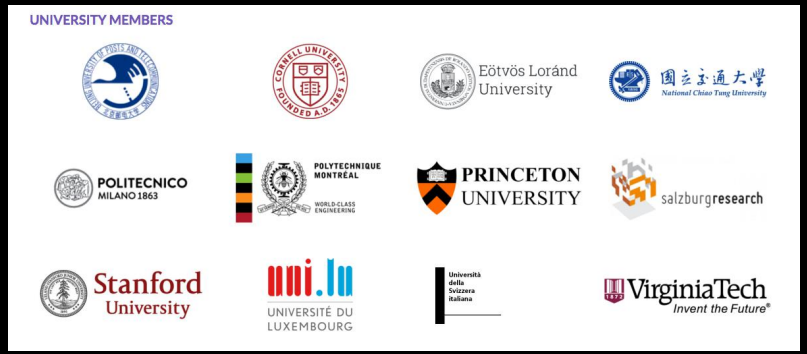
P4.org Update

Membership Growth

28 → 41 → 56

Nov 2015 May 2016 Oct 2016

Membership Growth



L2/L3 Data Plane in P4

switch.p4

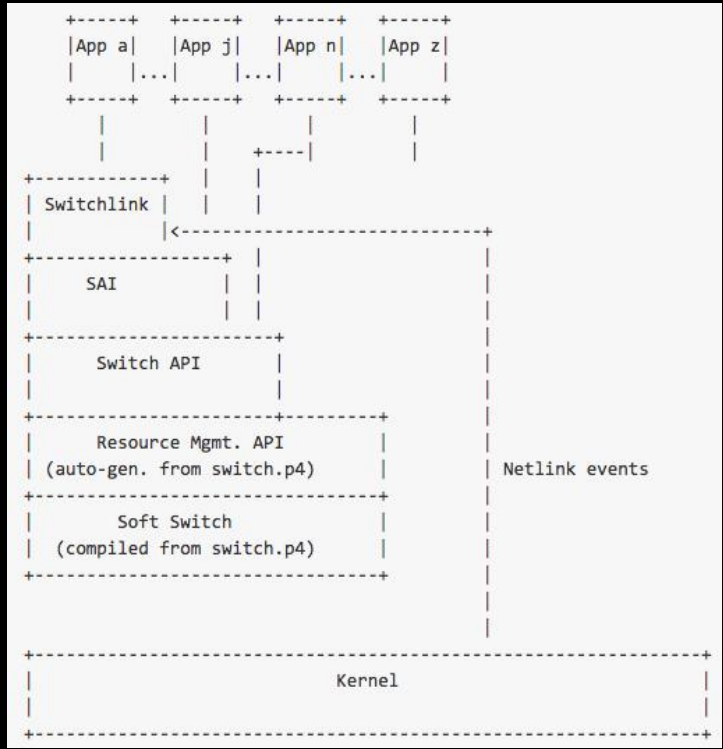
The switch.p4 program describes a data plane of an L2/L3 switch.

Supported Features

1. Basic L2 Switching: Flooding, learning and STP
2. L2 Multicast
3. Basic L3 Routing (unicast): IPv4 and IPv6 and VRF
4. L3 Multicast
5. LAG
6. ECMP
7. Tunneling: VXLAN and NVGRE (including L2/L3 Gateway), Geneve, GRE and IPinIP
8. Basic ACL: MAC and IP ACLs
9. Unicast RPF check
10. MPLS: LER, LSR, IPVPN, VPLS, L2VPN
11. Host interface
12. Mirroring: Ingress and egress mirroring with ERSPAN
13. Counters/Statistics
14. Ingress Policers
15. Inband Network Telemetry (INT)

Upcoming Features

1. NAT
2. QoS



APIs

SwitchAPI

The switchapi library exposes a higher level API on top of lower level resource management API auto-generated from the switch.p4 program in p4factory repository. Refer to p4factory/targets/switch/README.md for more details on switch.p4 program.

Supported Features

1. Basic L2 switching: Flooding, learning and STP
2. L2 Multicast (IGMP/MLD snooping)
3. Basic L3 Routing: IPv4, IPv6 and VRF
4. L3 Multicast (Sparse mode, SSM and Bidir)
5. LAG
6. ECMP
7. Tunneling: VXLAN and NVGRE (including L2/L3 Gateway), Geneve, GRE and IPinIP
8. Basic ACL: MAC and IP ACLs
9. Unicast RPF check
10. MPLS: LER, LSR, IPVPN
11. Host interface
12. Mirroring: Ingress and egress mirroring with ERSPAN
13. Counters/Statistics
14. Ingress Policers
15. Lookup bypass in Cpu Tx path
16. Netfilter Rx/Tx support
17. QoS (Quality of Service) - Buffers, Queues
18. Nat
19. CoPP (Control Plance Policing)

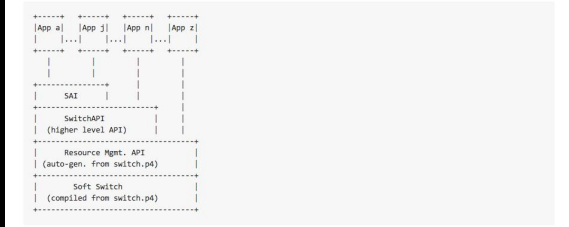
Upcoming Features

1. VPLS

SwitchSAI

The switchsai library exposes the standard Switch Abstraction Interface (SAI) API on top of the switchapi API library written to program the data plane described in the switch.p4 program in p4factory repository.

Refer to https://github.com/p4lang/p4factory/tree/master/targets/switch for more details on switch.p4 program and https://github.com/p4lang/switchapi for details on switchapi library.



Supported Features

1. Basic L2 switching: VLAN flooding and STP, learning, aging
2. L2 Multicast
3. Basic L3 Routing: IPv4, IPv6 and VRF
4. L3 Multicast
5. LAG
6. ECMP
7. Basic ACL: MAC and IP ACLs
8. Host interface
9. Ingress Policers
10. Statistics: VLAN, ACL
11. Qos (Quality of Service) - Buffers, Queues
12. CoPP (Control Plane Policing)

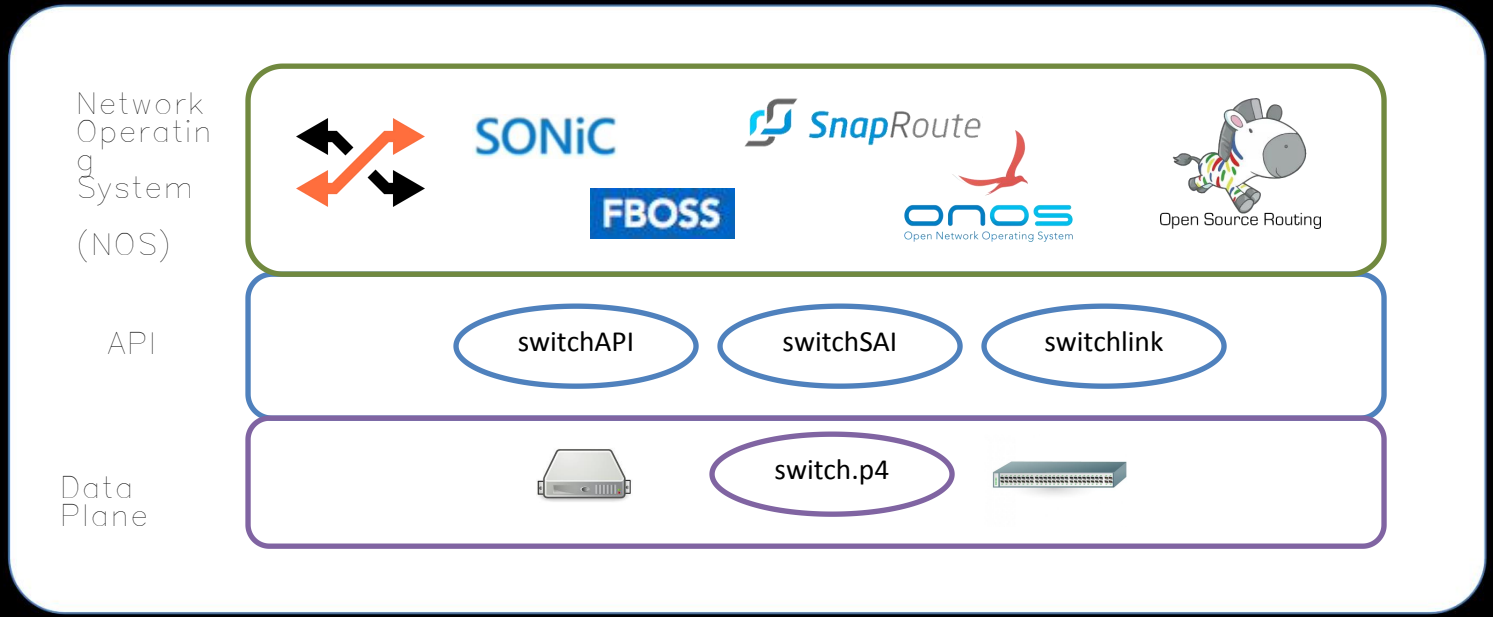
For the list of supported APIs and attributes, please refer to sai_support.pdf file in the doc directory.

Switchlink

The switchlink library provides a Netlink listener that listens to kernel notifications for network events (link, address, neighbor, route, etc.) and uses the switchsai library to program the data plane described in the switch.p4 program in the p4factory repository.

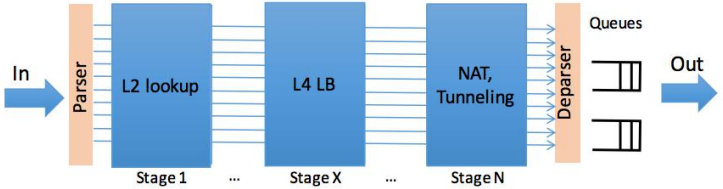
The switchlink library listens to NETLINK messages only on 'swp' interfaces. The 'swp' interfaces are TUNTAP interfaces that represent the softswitch's physical ports. Each 'swp' interface is mapped to a physical port (see src/switchlink_db.c). Applications (bridge-utils, mstpd, iputils, Quagga, etc.) interact with the softswitch via the 'swp' interfaces. Users can use native Linux tools to configure the 'swp' interfaces.

NOS Integrations



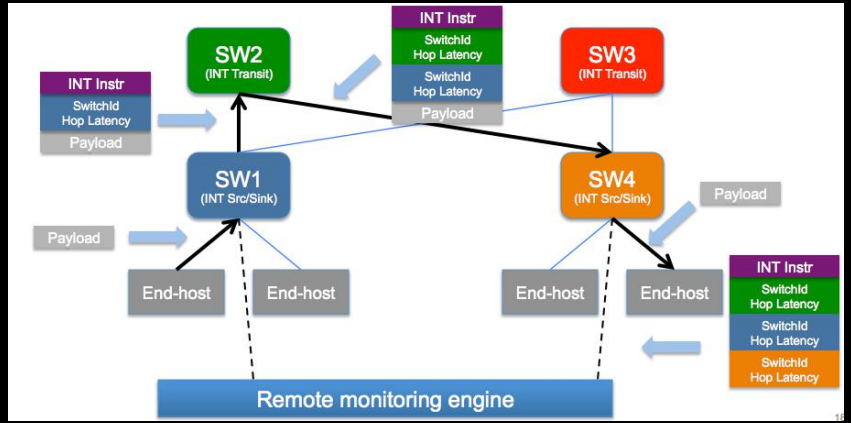
P4 Applications

L4 Load Balancer



- Benefits
  - High throughput (Tbps, Gpps), zero-latency, ubiquitous
  - Predictable performance even under availability attacks
- Challenges
  - Don't break existing connections during DIP pool update
  - Maintain millions of connection states in switch SRAM

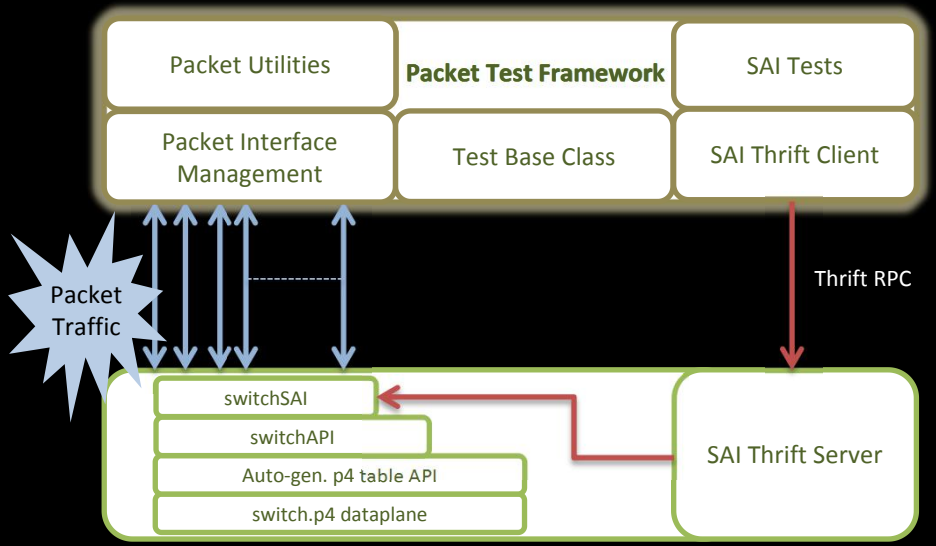
Path & Latency Tracking (PLT)



Many more ... ...

Test Tools

PTF – Packet Test Framework



NTF - Network Test Framework

This repository contains sample mininet and docker infrastructure required to perform network-wide tests of various applications on Behavioral Model version 2.

The directory structure of NTF repository is shown below:

```
── ntf                              Docker and Mininet Infrastructure framework
    ├── apps                        Reference applications
    ├── bmv2                        Folder to run BMv2 model
    │   ├── Makefile.bmv2           Makefile to build and compile BMv2
    │   ├── run_build_for_ptf.sh    Script to run PTF tests
    │   └── run_build_submodules.sh Script to build and Compile BMv2
    ├── docker
    │   ├── scripts                 Scripts to install dependencies inside docker
    │   ├── DockerFile              Specify dependencies for docker image
    │   └── startv2.sh              Script to setup CPU ports and front panel ports inside docker
    ├── makefiles
    │   └── docker.mk               makefile to specify docker target
    ├── mininet                     Mininet scripts
    │   ├── docker                  Scripts for bmv2 docker support
    │   ├── int_cfg.py              Helper script to simulate network for INT
    │   └── int_ref_topology.py     Mininet Script for INT
    ├── pull_modules.sh             script that clones required modules
    ├── update_modules.sh           script that updates requires modules
    ├── install_ntf_deps.sh         Dependency install script
    └── tools                       Various setup scripts
```

Thank You