OpenSwitch

Overview

OpenSwitch (OPX) – an open source network operating system (NOS) and ecosystem - is an early adopter of emerging concepts and technologies (hardware and software disaggregation, use of open source, SDN, NFV and DevOps) which disrupt how networks and networking equipment are built and operated. Designed using a standard Debian Linux distribution with an unmodified Linux kernel, OpenSwitch provides a programmable high-level abstraction of network components, such as switching ASICs (Network Processors) and optical transceivers. Architected as a scalable, cloud-ready, agile solution, the open source OpenSwitch software implements a flexible infrastructure to enable both network operators and vendors to rapidly on-board open source Networking OS applications. OpenSwitch provides a YANG based programmatic interface, that can be accessed using Python, thus providing an environment well-suited for DevOps.

OpenSwitch Features

OpenSwitch (or OPX) provides an abstraction of hardware network devices in a Linux OS environment. It has been designed from its inception in order to support the newest technologies and concepts in the networking industry:

- In OPX, software is disaggregated from hardware, and software components are disaggregated as well.
  - OpenSwitch can be deployed on diverse networking hardware – only the low-level software layers SAI (Switch Abstraction Interface) and System Device Interface (SDI) are hardware specific and may need to be adapted. A minimum requirement is for hardware to be build around a standard ASIC, with Layer 2 switching, Layer 3 routing, ACL and QoS functionality.
- Makes extensive use of standard open source software, for instance:
  - ONIE installer
  - Linux Debian distribution with an unmodified Linux kernel
  - Switch Abstraction Interface (SAI) defined in Open Compute Project for interfacing with the networking ASIC.
- Integrates Linux native APIs with networking ASIC functionality. In OpenSwitch, networking features are also accessible using the Linux standard API's ("netlink"). Thus standard open source network packages (such as FRR) can be installed and supported in binary format.
- OPX supports containers and NFV. The Docker container environment (Docker CE), or any other Linux container environment, can be installed on any OpenSwitch device - in this environment, users can deploy their own containerized virtualized network functions (VNF).
- Supports programmability, automation and DevOps:
  - A robust and flexible programmatic interface – namely the Control Plane Services (CPS). The API is defined using YANG models and is accessible through Python (and C/C++).
  - The availability of a programmatic interface (CPS API/YANG models) allows integration with external orchestrators and SDN controllers
- Provides a rich set of networking features including full access to the networking ASIC ACL and QoS functionality using CPS API/YANG models.

OpenSwitch provides support for:

- L2 protocols: LLDP, LACP (link aggregation interfaces), 802.1q (VLAN interfaces), STP and bridge interfaces
- L3 protocols (e.g. BGP)
- ACL and QoS network functions (through CPS / YANG API's)
- Instrumentation: sFlow, telemetry
- Orchestration and management

Programmability and Automation

OpenSwitch supports a rich ecosystem for automated deployment, for instance:

- Ansible – various modules are already defined for OpenSwitch
- Zero-touch provisioning (ZTP) allows provisioning of OpenSwitch ONIE-enabled devices automatically, without manual intervention
- Puppet

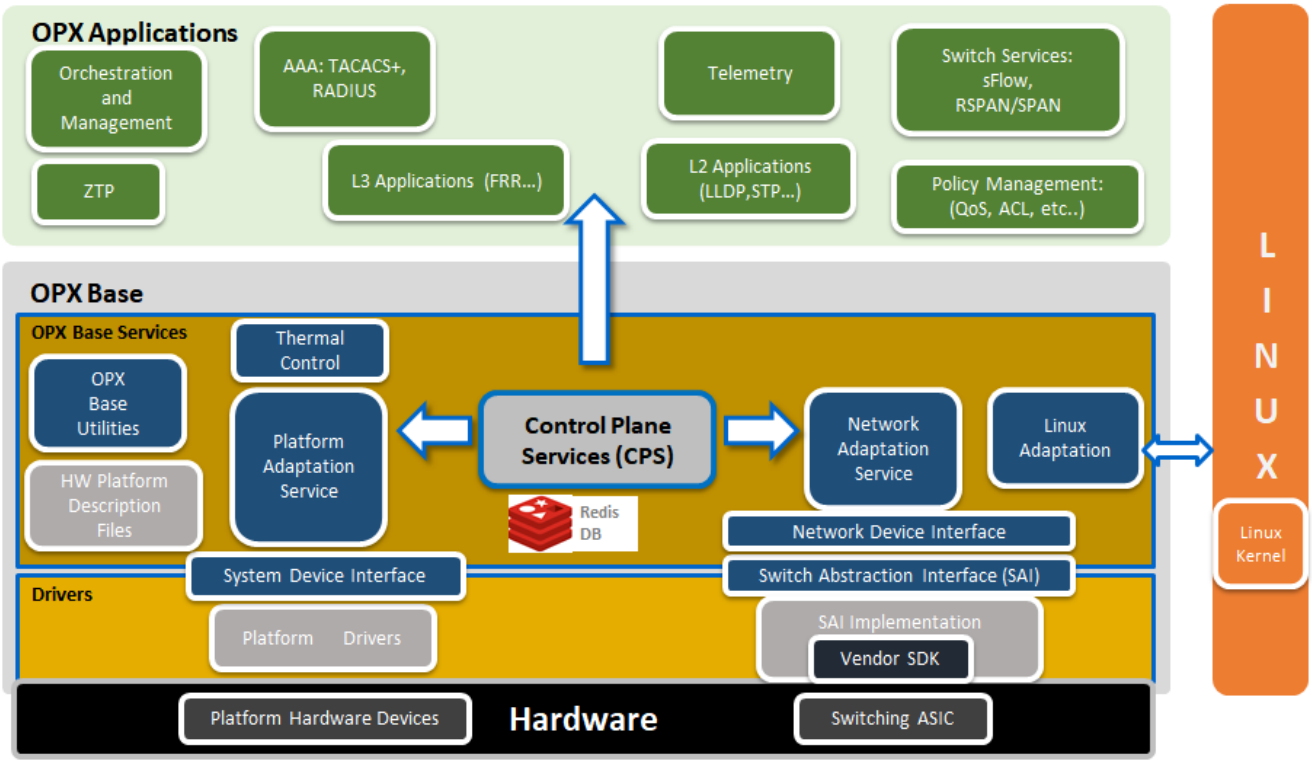
North-Bound Programmatic Interfaces

The OpenSwitch CPS programmatic interface is defined using YANG models, and in combination with Python, provides support for programming the network functionality, automation and DevOps. While the CPS API is the native OpenSwitch API, a REST API can be added as well, by mapping REST requests to CPS.

In addition, a set of OpenSwitch specific commands are available and can be invoked from a Linux shell (e.g. display the current software version, hardware inventory etc.).

OpenSwitch Architecture

The figure below illustrates the main areas of the OpenSwitch architecture:



OPX Base

The key components of OPX Base are:

NAS – Network Adaptation Service

- Manages the high level abstraction of the switching ASIC
- NAS manages the middle-ware that associates physical ports to Linux interfaces, and adapts Linux native API calls (e.g. netlink) to the switching ASIC

PAS- Platform Adaptation Service

- A higher-level abstraction and aggregation of the functionality provided by the SDI component

CPS – Control Plane Service

- Object centric framework
- Mediates between application software components and the platform
- Provides a pub/sub model and set/get/delete/create
- Provides the framework for defining YANG modeled APIs - with Python and C/C++ bindings. In OPX, YANG models are used with an efficient CPS binary encoding.

SAI – Switch Abstraction Interface

- SAI API is an open interface that abstracts vendor-specific switching ASIC behavior

SDI – System Device Interface

- An API that provides a low level abstraction of platform specific hardware devices (e.g. fans, power supplies, sensors…)

OPX Applications

A variety of open source or vendor specific applications have been tested and can be deployed with OpenSwitch:

- FRR - BGP
- AAA: TACACS+, RADIUS
- Telemetry: Broadview, Packet Trakker
- Inocybe OpenDaylight integration
- NetSNMP
- Puppet
- Chef

It should be noted that these applications are not pre-installed with OpenSwitch. In a "disaggregated" model, users select applications to install them based on the requirements of a given network deployment.

In general, since OpenSwitch is based on Linux Debian distribution with an unmodified kernel, any Debian binary application can be installed and executed on OpenSwitch devices.

Hardware Simulation

OPX software supports hardware virtualization (or simulation). Software simulation of basic hardware functionality is also provided (simulation specific SAI and SDI components), and the higher layer software functionality can be developed and tested on generic PC/server hardware. OPX hardware simulation can be executed under Virtual Box, GNS3 / QEmu etc.