OpenDaylight: Introduction, Lithium and Beyond

Colin Dixon

Technical Steering Committee Chair, OpenDaylight
Senior Principal Engineer, Brocade

Some content from: David Meyer, Neela Jaques, and Kevin Woods
Outline

• Introduction...
  • ...to SDN
  • ...to OpenDaylight...
    • ...with a brief aside into YANG

• New in Lithium

• Plans for Beryllium
Traditional SDN (OpenFlow)

The separation of the control and data planes

- Modern switches
  - Control/data plane both on switch
  - Data plane: fast, reads tables
  - Control plane: slow, writes tables

- SDN
  - Decouple control/data planes
  - Data plane on the switch
  - Control plane elsewhere, e.g., an x86 server, can do fancier things
Modern, Inclusive SDN

Vendor A

Vendor B

Vendor C

Logically Centralized SDN Controller

Northbound API

Industry Standard Control/Management Protocols

Standard Modeling Language
What is OpenDaylight

OpenDaylight is an Open Source Software project under the Linux Foundation with the goal of furthering the adoption and innovation of Software Defined Networking (SDN) through the creation of a common industry supported platform.

**Code**

To create a robust, extensible, open source code base that covers the major common components required to build an SDN solution.

**Acceptance**

To get broad industry acceptance amongst vendors and users:
- Using it directly or through vendor products
- Vendors using OpenDaylight in commercial products

**Community**

To have a thriving and growing technical community contributing to the code base, using the code in commercial products, and adding value above, below and around.
OpenDaylight Releases

• **Hydrogen** (first release)
  • February 2014
  • 13 projects, 1.3m lines of code

• **Helium** (second release)
  • October 2014
  • 25 projects, 2.1m lines of code

• **Lithium** (latest release)
  • June 2015
  • 40+ projects, 2.3m lines of code
Core Architecture

Model-Driven Service Abstraction Layer (MD-SAL)

App/Service

App/Service

Plugin

Plugin

Controllers in a Cluster

RPCs

Data

Notifications

YANG Models
What is YANG?

• Data modeling language for NETCONF
  • RFC 6020

• Great, what is NETCONF?
  • Think of it as an SNMP replacement with nice features
  • YANG models ~= SNMP MIBs

• OK, fine, but what is YANG?
What is YANG?

• Three core abstractions
  • Data
  • RPCs (just data in and data out)
  • Notifications (just data out)

• So, it’s really all about the data
What does YANG data look like

- container ~= struct
- list ~= map/dictionary
- leaf ~= primitive types
- grouping ~= interface
- Others: typedef, pointers, constraints, etc.
OpenDaylight Community

• Like any Open Source Project, OpenDaylight primarily consists of those who show up to do the work.

• Running around 250 commits per week over 12 months, trending up
  • 30 Days: ~625 commits, ~100 contributors (7/13/15–8/12/15; during a release)
    • Spikes to ~2x this near releases
  • 12 Months: ~13,250 commits, ~365 contributors (8/12/14–8/12/15)

• Strong integration and testing community
  • This stuff really matters

Source: https://www.openhub.net/p/opendaylight
OpenDaylight Community
New in Lithium
Major Shifts in Lithium

• Deprecation of the AD-SAL
  • Move to MD-SAL: OVSDB, LISP, etc.

• OpenStack/Neutron Integration
  • Significant closing of the feature gap
  • More implementations
    • OVSDB, GBP, VPN Svc, VTN, LISP

• Service chaining/NFV
  • SFC project, integrating with GBP

• Security
  • Formal process defined
  • Handled many issues/fixes

• More on policy
  • NIC as a vendor-neutral layer
  • Big push on SFC+GBP

• Release process refinement
  • Better documentation process
  • Better integration/test process
  • Offsets for coordination

• 20+ new projects
OVSBDB migration in Lithium

- Split network virt. and driver
- Moved both to use the MD-SAL
- Significantly better CI and testing with OpenStack
- Aspirations of moving to support policy/NIC
New Projects in Lithium

• Meta
  • Release Engineering - autorelease
  • Release Engineering - Builder
  • Controller Core Functionality Tutorials

• Drivers
  • CAPWAP-Support
  • Distributed LLDP with Auto Attach Capability
  • Link Aggregation Control Protocol
  • Internet of Things Data Management (IoTDM)
  • SNMP Plugin
  • Source Group Tag eXchange Protocol (SXP)
  • Unified Secure Channel

• Apps
  • Application Layer Traffic Optimization (ALTO)
  • Integration of Maple Programming Model
  • Network Intent Composition
  • Neutron Northbound

• Services
  • Persistence
  • Device Identification and Driver Management
  • Discovery
  • Time Series Data Repository
  • Topology Processing Framework
  • VPN Service
Lithium Dependency Diagram
S3P: Security, Stability, Scalability, Performance

• Goal to improve the quality of key features
  • OpenFlow, OVSDB, NETCONF, MD-SAL, etc.

• Significant progress on OpenFlow (thanks Luis/Integration team)
  • See following slides

• Broader progress expected in Beryllium
OpenFlow flows/sec performance

Throughput Mode

Latency Mode


OpenFlow REST flows/sec, scalability


https://jenkins.opendaylight.org/releng/view/openflowplugin/job/openflowplugin-csit-1node-periodic-scalability-daily-only-stable-lithium/plot/Inventory%20Scalability/
S3P To Dos

OpenFlow
- Stability testing
  - Longevity beyond an hour – Test written waiting for infra OK
  - Under different kinds of load – Summer intern will take care
- Performance in a cluster
  - OpenFlow doesn’t currently work in a cluster
- Scalability beyond switches
  - Flows, hosts, links, etc. – Host and Links tests are coming

More SB Protocols
- NETCONF – Pantheon and Carol (BRCD) will bring tests
- OVSDB – Intel working on this (asked Colin for BRCD thoughts)
- BGP
- PCEP

More NB interfaces (use cases)
- Neutron
- SFC
Beryllium Plans
Beryllium Releas

• Focus on S3P, Migration, and HA/clustering

• Tries to balance maturity (the above) with feature velocity
  • Some projects will be *mature*
  • Some of the Karaf features in *mature* projects will be *stable*
  • Stable features will have S3P, Migration, and HA/clustering requirements
  • Stable and “normal” distribution; stable only has stable features

• Trying to drive appropriate projects/features to mature/stable

Trying to move offset 0 projects to mature

• Controller
• MD-SAL
• NETCONF
• AAA
• YANG Tools
• odlparent