



Roadmap to Operating SDN-based Networks

Berkeley Lab from July 14th - 16th, 2015

David Bainbridge
Ciena Agility

Disclaimer

I don't speak for OpenDaylight

The following is a mixture of opinion, rough consensus, and reality. I will try to differentiate which is which, but no promises.

If in doubt, ask.

OpenDaylight

OpenDaylight is an open platform for network programmability to enable SDN and create a solid foundation for NFV for networks at any size and scale.

- Model driven (YANG)
- Modularized (dynamic feature loads/updates)
- Basics of scale and HA (but not mature)
- Large community of contributors (466 for Li)
- Supports legacy interfaces
- Application Container v. Controller
- Internal tension: research v. operational
- Projects run relatively independently

SDN “Opportunities” v. OpenDaylight

- Quality / Testing
 - Defined best practice
 - Unit testing
 - Integration test
- Abstractions
 - Multiple (intent, GBP, VTN, etc.)
 - Not always compatible

SDN “Opportunities” v. OpenDaylight

- Discovery
 - DIDM project to identify and discover network elements
 - DIDM can support information feed for service discovery
 - Neighbor discovery

SDN “Opportunities” v. OpenDaylight

- Crisp, clear, interfaces
- Problem isolation / Troubleshooting
- No SDN standard
- Defined by YANG models
- Logging
- No real tooling

Operational Concerns (1/3)

- Logging
 - Defined best practice
 - Centralized logging being proposed for next release
- Documentation
 - Documentation project
 - Required from each project to be part of release

Operational Concern (2/3)

- Release management
 - Published release schedule
 - Supports service release
 - *Inter-release changes*
- Packaging
 - Core release features
 - Packaged differently by vendors

Operational Concern (3/3)

- Support
 - Current and one release back officially supported
 - Provided by vendor for their release
- Security
 - Introduced in latest release (Lithium)
 - Minimal
 - No data isolation

Operational Concern (3/3)

- Scaling / HA
 - Supports single site HA via clustering
 - Immature
- E2e and Multi-domain
 - No
 - Suppose it could be done, but not core and would be significant work

Engagement with R&E

- Workshops such as this
 - Feedback will be given to the ODL TSC / Community
- Needs to be put through the ringer
 - Testing is by developer, not user perspective
 - Enter bugs into ODL's bug tracking system
- Contribute to specific projects
 - Use cases
 - Code
 - Documentation updates
- Key is participation and not just use of the project