

# Data Models for Optical Devices in Data Center Operator Networks

**Eric Breverman, Nancy El-Sakkary, Tad Hofmeister, Anees Shaikh, Vijay Vusirikala**

*Google Inc. 1600 Amphitheatre Parkway, Mountain View, CA 94043 USA*

*vijayvusiri@google.com*

**Abstract:** Standardized, vendor agnostic data models deliver major operational benefits. OpenConfig has been implemented on multiple platforms and is an ideal data model to take advantage of these benefits. This document provides an overview. © 2019 The Author(s)

## 1. Introduction

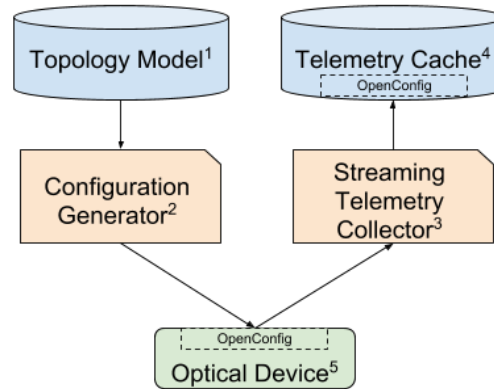
Network operator goals of software defined networking (SDN), zero-touch provisioning (ZTP), and automating operational tasks have become common across the industry. In a multi-vendor / multi-platform environment, having a standardized and vendor agnostic model of configuration and telemetry data enables tremendous simplification and reduces operational complexity.

These data models enable operators to evolve to a plug-and-play approach to integration of new platforms. New technologies can be quickly deployed in production and higher layer automation / SDN controllers can be greatly simplified by supporting a common model that does not require platform specific knowledge. OpenConfig is a maturing model-driven management ecosystem for achieving these goals, and this approach is validated by the growing number of implementations in the industry.

## 2. OpenConfig Based Software Defined Network

The value of a platform with a standardized, vendor agnostic model is realized when it is used as a core feature of a software defined network. Applications can then be built which only interact with the top two boxes (1 and 4) in Fig.1. A proven approach to building a software defined network is described below, at a high level, to help provide context for this document.

- Vendor neutral Topology Model (1) describes the intended end-to-end topology of the complete network (i.e., details of how each circuit traverses from source to destination).
- Configuration Generator (2) is a service which consumes the vendor neutral topology, applies business logic, and produces device-level configurations as OpenConfig-modeled data.
- Streaming Telemetry Collector (3) is a service which accepts and stores streamed monitoring data from devices in a vendor-neutral format based on OpenConfig. The data is made available via a Telemetry Cache (4) for other services to use.
- Optical Device(s) that have implemented OpenConfig (5).



**Fig. 1 High level representation of an OpenConfig based SDN optical network**

### 3. Discussion of Use Cases

#### *New Product Introduction (NPI)*

As automation becomes more prevalent in a network, the first challenge is that an NPI can grow tremendously in scope. In addition to typical NPI tasks like lab testing and defining install processes, integrating with management software is a large component of the NPI. A new platform might provide significant cost and performance improvements or enable a critical feature, so delays can be impactful for operators. However, if the platform supports a standardized, vendor agnostic management interface, the core software infrastructure can integrate simply in a plug-and-play approach.

#### *Configuration*

With a standardized, vendor agnostic model, the configuration of devices becomes largely identical among different platforms. Various platforms will have some unique characteristics, but configuration generators should be built in a modular fashion (i.e. AAA, NTP, IP, Client Port, etc.) to accommodate this. Then, modules can simply be pieced together to create the full device configuration. There might be some hardware specific information (i.e. device has X cards, each with Y client ports and Z line ports), but the specifics of how to configure a port type or setup device management become identical across platforms. This enables configuration generator code to be written quickly and allows operators to easily apply standard configuration across all platforms.

In addition, once a platform is in production, updating configurations can happen easily across multiple platforms. For example, if a new feature needs to be supported on all platforms, the configuration generators are updated once with the new modeling and the effort to support the new feature is complete (assuming all vendors support the feature). This is a vast improvement over the previous process which required repeating this work for multiple platforms, doing this N x number of platforms.

#### *Telemetry*

With a legacy managed platform, if a new automation initiative requires a set of telemetry data, the developers need to investigate how to get the telemetry data for each of their platforms and translate vendor specific data into a common format. With a standardized, vendor agnostic model, common telemetry data is clearly defined and identical across platforms.

For example, retrieving the frequencies of all ports requires one identical request for the data to all platforms, followed by a lookup that the model defines it as a uint64 in units of MHz. Today, as trivial as that task seems, there are platforms that use GHz, THz, nanometers, custom channel number mappings, etc. Data retrieval may be through SNMP, TL1, or CLI, which requires writing platform specific parsers to interpret strings written primarily for human consumption. These strings are also likely to change from one software revision to the next so maintenance is

an ongoing challenge. Finally, while this is just one variable, there are hundreds of others that are represented differently across platforms.

### ***Lab Testing and Qualification***

Operators aspire to automate as much of their lab testing and qualification process as possible. However, in a multi-vendor environment this is usually impractical because each platform is unique, so integrating a new platform requires writing unique code for that platform.

For OpenConfig platforms, this paradigm completely changes as all platforms conform to a common set of requirements. Any effort to automate a lab qualification benefits not only the new platform, but future platforms as well. This opens the possibility of taking this further and having open source testing suites for both operators and vendors. Google is taking the first step by developing an open source testing suite<sup>1</sup> for streaming telemetry.

There is also a cyclical effect that will occur as network automation gains traction. As an operator develops a Software Defined Network (SDN), they will start making more requests with the vendor for software features that are unrelated to hardware. As a result, vendors will deliver more regular software releases and operators will want to perform an increasing number of software qualifications and rollouts. Today, operators might only want to roll out one or two new software upgrades per year. Automated lab testing would be an absolute requirement if this increased to a monthly cadence as that is the only way new releases could be properly qualified in time.

### **4. Conclusion**

Multiple platforms from multiple vendors<sup>2,3,4</sup> now support OpenConfig. Within the Google optical network, these implementations are being used on thousands of devices carrying production traffic. Over the last few years, the implementations continue to grow more comprehensive as gaps in modeling are being closed. As a result, Google has been able to take advantage of OpenConfig's vendor agnostic and standardized models. This has enabled automation gains in building and operating the optical network. Examples of these gains are seen in automating diagnostics, repair, software rollouts, traffic rerouting, capacity additions, and device turnup, to name a few.

### **5. References**

1. gnmitest - A Test Framework for gNMI, <https://github.com/openconfig/gnmitest>
2. Anees Shaikh, OpenConfig - progress toward vendor-neutral network management, Oct 2017, [https://pc.nanog.org/static/published/meetings/NANOG71/1535/20171004\\_Shaikh\\_Lightning\\_Talk\\_Openconfig\\_v1.pdf](https://pc.nanog.org/static/published/meetings/NANOG71/1535/20171004_Shaikh_Lightning_Talk_Openconfig_v1.pdf).
3. Ciena - WaveServer AI Datasheet, 2018, <https://media.ciena.com/documents/Waveserver-Ai-DS.pdf>
4. Cisco - Data Models Configuration Guide for Cisco NCS 1001, March 2018, [https://www.cisco.com/c/en/us/td/docs/optical/ncs1001/b\\_Datamodels\\_cg\\_ncs1001/b\\_Datamodels\\_cg\\_ncs1001\\_chapter\\_011.html](https://www.cisco.com/c/en/us/td/docs/optical/ncs1001/b_Datamodels_cg_ncs1001/b_Datamodels_cg_ncs1001_chapter_011.html)