Optical Zero Touch Networking - A Large Operator Perspective

Eric Breverman, Nancy El-Sakkary, Tad Hofmeister, Sean Ngai, Anees Shaikh, Vijay Vusirikala
Google Inc. 1600 Amphitheatre Parkway, Mountain View, CA 94043 USA
vijayvusiri@google.com

Abstract: A key area of innovation in optical networking has been enabling modern, vendor-agnostic APIs on devices. We provide specifics of how these new capabilities enable deployment and operational efficiencies.

1. Introduction

Over the last few years, operators have increasingly been asking for an entirely new model of device management. Instead of CLI, SNMP, TL1, etc., operators want modern, vendor agnostic APIs[1, 2]. Several operational goals, such as scalability, efficiency, and reliability in building and operating a large multi-vendor network, are enabled with these new features. We refer to this program as Optical Zero Touch Networking (ZTN) to stress that humans should not have to interact with devices. They need only indicate high level intent or context (e.g., software version should be R1.23, turnup device X now, or build a circuit that goes from A to B to C), and everything else happens automatically.

2. Challenge

The two core areas currently being addressed by Optical ZTN are:

- *how to cost effectively sustain rapid network growth*
- *how to minimize human errors and associated outages*

Traditionally, in order to maintain and build a rapidly growing network that might add 50% additional capacity per year, a company might need to hire 50% more people every year. Not only is this cost prohibitive, but more importantly, it is not possible to deploy enough skilled engineers fast enough to keep up with network growth. Automation is required to support existing engineers so that the network can continue to grow to meet capacity demands by eliminating repetitive, transactional work and/or increasing process efficiency.

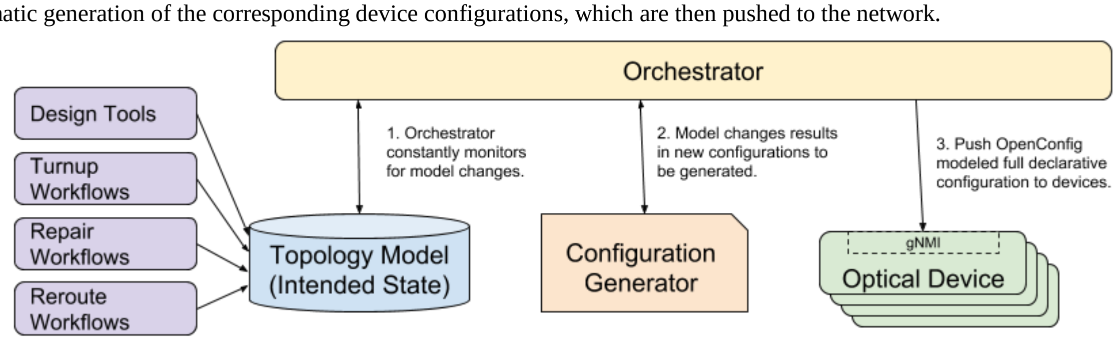
A significant number of network outages can be attributed to human accidents[3]. It is not uncommon for a human to type an incorrect command or enter an incorrect interface. Manual changes are also particularly risky when they are low level changes and the resulting higher level impact is not obvious. Optical ZTN enforces that all work happens through automated operations on an abstracted representation of the network, with adequate validation and safety checks. Automation doesn't guarantee unsafe changes or outages never occur, but provides an opportunity to incorporate fixes into reusable common workflows to ensure problems don't recur.

Use Cases

There are a growing number of use cases. Here we explore three of them which have been realized at Google.

Intent Driven Configuration

Intent driven configuration is a core concept and capability that eliminates humans from configuring equipment. The concept is that changes are made via manipulation of an abstract model of the intended network state. A change to the model results in automatic generation of the corresponding device configurations, which are then pushed to the network.



A vendor-agnostic topology model allows the intended state to be documented in a structured format. The topology model must be the authoritative source of truth for all network topology in the network. Design and planning tools update the intent based on new capacity demands.

To have devices configured per the intended state, an orchestration layer is needed to monitor all changes to configuration based on intent. When changes are made the orchestrator initiates the push of the new configuration onto the device. gNMI[4] has standardized the device API across platforms and makes device communication simple and consistent across all platforms. As a result, integrating new platforms into the orchestrator requires relatively minor changes.

Declarative configuration[5] support on the device plays a key role to support configuration pushes. With declarative configuration, the full device configuration (everything from basic configuration information such as hostname, usernames, etc. to circuit specific configuration such as admin states, cross-connects, etc.) is always sent to the device. Partial configurations are never allowed, which ensures a complete, valid configuration is always sent to the device. The device is responsible for checking for differences between the running configuration and the intended configuration and then updating the configuration to the intended state, with the lowest impact. The rationale of this approach is that the device has the best knowledge of how configuration changes should be sequenced, and results in significantly lower complexity in the operator's management software.

Another key element of the Optical ZTN architecture is configuration generators. Their role is to transform changes to the intended state topology model into full declarative vendor-agnostic device configurations. OpenConfig[6] is a vendor-agnostic device configuration model that is implemented on a growing number of platforms. It allows for common code to be used across different platforms and vendor combinations. The configuration generator's core logic interprets topology and combines it with business criteria (e.g., transponder output-power should be 1dBm for a given line system). Design and planning tools can simply populate the intent, which is pushed directly onto devices.

Device Turnup

In a multi-vendor environment where new platforms are being introduced regularly, it is challenging for engineers to become experts on every platform. Automating device turnup has increased the speed of each turnup and subsequently increased the rate of new devices being deployed in the network.

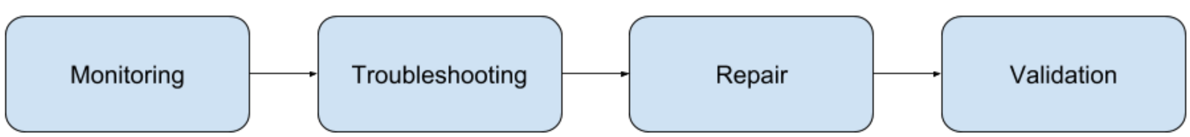
To perform a device turnup after physical installation, an engineer can start a workflow by simply providing a device name. The automation then performs a sequence of tasks:

1. Enable the management link.
2. Use DHCP or the console serial port via a terminal server to configure an IP address on the platform automatically.
3. Install the latest software on the device.
4. Enable the device to be intent driven and let the orchestrator, (described above) to push full configuration to the device.
5. Perform an acceptance audit on the device (e.g., alarms) and if successful, end the workflow.

Workflows can initiate multiple device turnups, so multiple devices can be turned up in parallel. This enables full device turnups in typically under one hour after the physical install. Also critical is that local technicians who perform the physical installation are empowered to get a device accepted into the network without having to wait for the support of optical engineers.

Repair

Repair work is typically a repeatable series of steps which lends itself to being automated. To automate network repair, the following high level systems are needed:



A monitoring system needs to alert based on defined rule sets. Rule sets may be alarm matching (e.g., "hardware failed") or based on link level indicators such as router interfaces that are down. The key requirement is monitoring the state of the device. Streaming telemetry solves this challenge by providing a mechanism for near-real time collection of all information desired from the device. In a multi-vendor environment, OpenConfig defines standard structure and format of the data across platforms and vendors so it can easily be interpreted.

Based on the alert, an appropriate series of troubleshooting steps can take place. Two common troubleshooting scenarios are:

- Router interfaces are down for an individual circuit

    - Loopback testing is started in an attempt to troubleshoot. Performance monitoring variables (optical power, PreFEC, etc.) are correlated to create a diagnosis, such as faulty fiber jumper, failed transceiver, etc. An escalation is then raised to a human with the recommended action. Automation may also help in coordinating maintenance windows or requesting the right team to perform the repair. At the conclusion of the repair, validation determines if the problem was resolved. If not, automation repeats troubleshooting and the process repeats.
- Hardware failure alarm
    - For obvious hardware failures, automation can move quickly to perform repairs. Initially, a proactive reboot can be performed. The OpenConfig gNOI[7] API standizes this action across all platforms. Using the topology model, the related router interfaces can be identified and drained before performing the reboot. At the conclusion of the reboot, validation checks to see if the problem was resolved. If not, an RMA can be initiated with the vendor.

7. Conclusion

The approach to building and operating an optical network is dramatically changing as a result of Optical ZTN. Best in class processes can be created to perform troubleshooting or device turnup, replacing the days when each engineer had their own unique procedures. While humans are still required for some work, such as physical installation or hardware replacement, this can be done by local technicians empowered to complete an end-to-end process without involving a limited pool of specially trained engineers.

In a large multi-vendor network, Optical ZTN can only be efficiently realized by modernizing the device management approach with technologies such as OpenConfig, modern API's, and declarative configuration. In addition, the same approach is relevant for Layer 3 equipment, so services can be reused across the entire network. This new management approach is enabling scalability and efficiency improvements to allow terminal and line system optical devices to be built and maintained without linearly scaling operational staff.

References

1. O. F. Yilmaz, S. St-Laurent, and M. Mitchell, "Automated Management and Control of a Multi-Vendor Disaggregated Network at the L0 Layer," in Optical Fiber Communication Conference, OSA Technical Digest (online) (Optical Society of America, 2018), paper Tu3D.9.
2. A. Shaikh, T. Hofmeister, V. Dangui, and V. Vusirikala, "Vendor-neutral Network Representations for Transport SDN," in Optical Fiber Communication Conference, OSA Technical Digest (online) (Optical Society of America, 2016), paper Th4G.3.
3. Ramesh Govindan , Ina Minei , Mahesh Kallahalla , Bikash Koley , Amin Vahdat, Evolve or Die: High-Availability Design Principles Drawn from Googles Network Infrastructure, Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference, August 22-26, 2016, Florianopolis, Brazil
4. gNMI - gRPC Network Management Interface - https://github.com/openconfig/gnmi
5. E. Breverman, N. El-Sakkary, T. Hofmeister, A. Shaikh, V. Vusirikala, "Optical Network Control & Management Plane Evolution", Optical Fiber Communication Conference, 2019.
6. OpenConfig - http://www.openconfig.net/
7. gNOI - gRPC Network Operations Interface - https://github.com/openconfig/gnoi