Independent market research and competitive analysis of next-generation business and technology solutions for service providers and vendors

HEAVY READING

WHITE PAPER

Enabling Network Programmability & Automation With NETCONF/YANG

A Heavy Reading white paper produced for Tail-f Systems

tail-f a Cisco company

AUTHOR: JAMES CRAWSHAW, SENIOR ANALYST, HEAVY READING

INTRODUCTION

The networking industry is going through a period of unprecedented change with new paradigms such as SDN and NFV. The ancient Greek philosopher Heraclitus observed that change is the only constant. Yet as Canadian prime minister Justin Trudeau pointed out earlier this year, "The pace of change has never been this fast, yet it will never be this slow again."

So how do network operators cope with exponential change? Through automation. According to Network Programmability and Automation by Edelman, Lowe & Oswalt, network automation is being "about simplifying the tasks involved in configuring, managing, and operating network equipment, network topologies, network services, and network connectivity."

There are many different components involved in network automation, including application programming interfaces (APIs), operating systems, scripting languages, data modeling languages, templating languages, system admin tools and new DevOps methodologies. Network automation use cases include:

- Device provisioning – using templates with configuration parameter files to generate consistent config files
- Data collection – using open source tools to pull the data you want from network devices, not just what your existing monitoring tools allow
- Configuration validation – checking security compliance and automatically triggering remediation if there is a violation
- Troubleshooting – automating the analysis of fault and performance data to find root causes more quickly and reliably, and to predict future issues

Networking is already highly automated at the data plane (packet forwarding) and control plane (e.g., protocols such as OSPF and BGP). The aim of SDN is to enable even greater automation by separating the data plane, distributed across multiple networking devices, from the control plane, which is centralized to enable holistic control of the network, rather than having to implement control changes on a node-by-node basis.

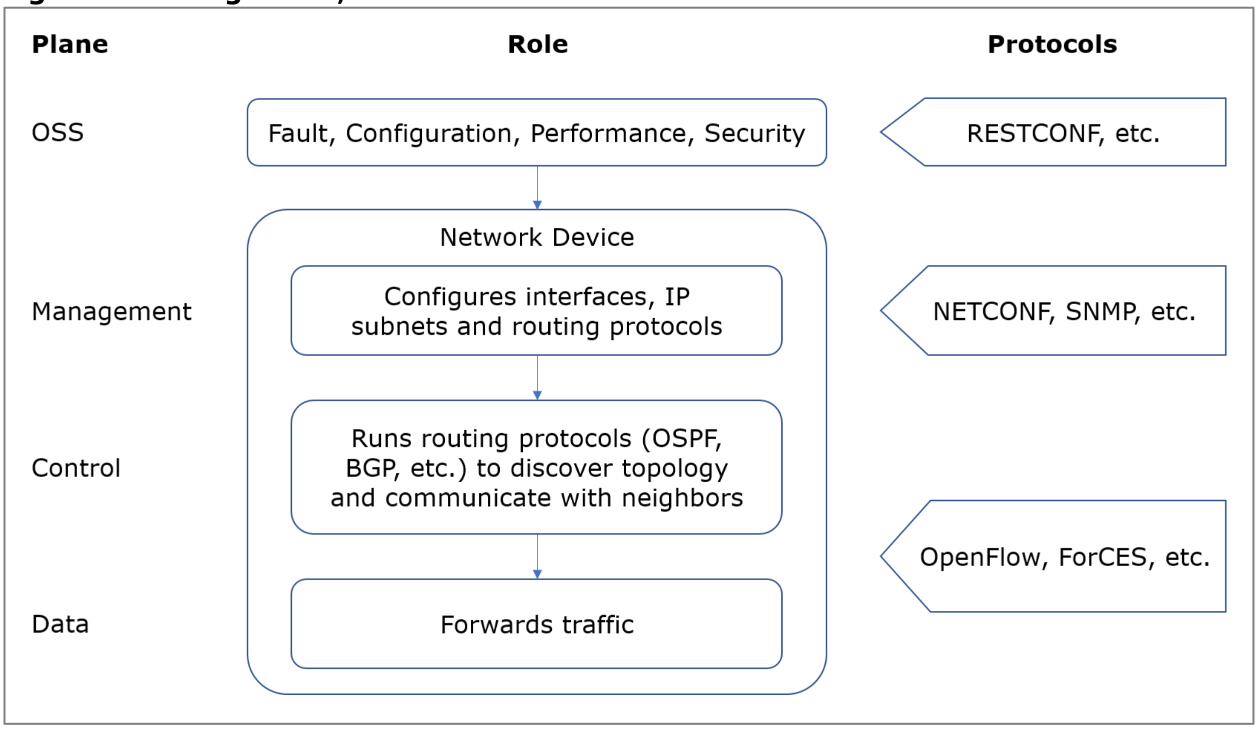
The least automated part of networking is the management plane (see **Figure 1**), where human interfaces such as command-line interface (CLI) are often still used to configure the network, and legacy protocols such as Simple Network Management Protocol (SNMP) are used for monitoring. This is fine for static networks that are configured once and then left to run; however, the pace of change of networks (driven by server virtualization, new applications, mobile devices, etc.) now requires greater programmability of the management plane. The manual processes involved and the lack of common interfaces across vendors means that CLI will not scale cost-effectively to meet rising network demands.

As Adrian Iliesiu writes in his blog, Defining Network Programmability: "The times when we could configure and manage one device at a time are going away. In a dynamic DevOps world when entire testing and development environments are built and destroyed within minutes, we expect the network to keep up and be just as flexible."

We posit that the solution to this problem is NETCONF, a protocol that has been around for more than 10 years and has become a key requirement in the next-generation architecture of leading communications service providers such as Deutsche Telekom for the last five years. Since then, other operators such as AT&T have embraced it, and nowadays it is a requirement for selling network devices to many Tier 1 operators across the globe.

In this paper, we consider the importance of the NETCONF protocol, and its associated modeling language YANG, in management plane programmability and automation.

Figure 1: Management, Control & Data Planes in Network Devices



Source: Heavy Reading

PROGRAMMABILITY IS THE KEY TO AUTOMATION

The key to network automation is making networking devices more programmable. This is the ultimate objective of making the network "software-defined," enabling it to be managed by SDN controllers and service orchestrators. Programmability is about configuring and managing network devices using software, as opposed to manual tinkering via CLI. Programmability is key to managing large-scale networks with devices from multiple vendors, running multiple different operating systems and even variants of OS within each vendor domain. CLIs are cumbersome to use, error-prone, and vary widely across different vendors and even between products from the same vendor. With programmability, network management becomes an exercise in computer science, where network administrators manipulate data, not devices.

Consequently, most vendors are migrating away from CLI-based device management to API-driven devices. APIs provide an abstraction of the underlying technology, so that a (network) programmer does not need to know unnecessary details but has access to: what data is available (attributes); how the data is organized (schema); and how the data can be accessed (methods). That makes it a lot easier to write and maintain scripts or programs to configure and manage network equipment. APIs enable automated access to all the data in network devices such as flow-level data, routing tables, FIB tables, interface statistics, MAC tables, VLAN tables and serial numbers.

Network programmability enables:

- Time and cost savings
- Fewer errors
- Easier troubleshooting
- Easier innovation in network and service management

Instead of using proprietary (albeit REST-based) APIs, vendors should adhere to standardized APIs to make it easier for communications service providers to manage multivendor networks. There are two such standards that we discuss below: NETCONF and RESTCONF. In addition, there is a data modeling language, YANG, which is key to defining devices and services in a consistent, parsable manner.

A STANDARDIZED APPROACH TO PROGRAMMABILITY

To address the network device configuration challenge, the Internet Engineering Task Force (IETF) has defined three key standards: NETCONF, YANG and RESTCONF.

Network Configuration Protocol (NETCONF) – RFC 6241

NETCONF is an XML-based, RPC-style protocol that supports operations such as installing, querying, editing and deleting configuration data on network devices. It can be conceptualized as four layers:

1. Content – configuration data for the network device
2. Operations – methods for communicating between agent and manager, e.g., <get-config> or <kill-session>
3. RPC – used for encoding a Remote Procedure Call
4. Transport – communications path between server and client using secure protocols such as SSH and TLS/SSL

One of the key advantages of NETCONF over SNMP or CLI is that it is transactional. Configuring a device can involve multiple steps. Usually these actions cannot be done partially, as this would leave the device in an undefined state. If any step of the end-to-end provisioning fails (or is undertaken in the wrong order), there is a need to roll back, i.e., undo all previous actions, to revert to the original configuration. This requires extensive programming when transaction management is not supported, as is the case with CLI. NETCONF, on the other hand, does support transactionality, so service providers can be confident that either all the configurations in a sequence are applied or the entire update is rolled back. Additionally, transactionality at the device level makes it easier to implement network-wide transactions that involve multiple devices when provisioning services across a network.

As Edelman, et al., state in Network Programmability and Automation, "All commands succeed or they are not applied. This is in contrast to the more common scenario of entering a series of commands and having a command somewhere in the middle fail, yielding a partial configuration."

Yet Another Next Generation (YANG) Data Modeling – RFC 6020

For NETCONF to be useful, it requires a consistent data modeling language to model both configuration data as well as state data of network elements. Although many data modeling languages exist, the IETF decided that the idiosyncrasies of the telecom industry required its own domain-specific language. YANG – Yet Another Next-Generation (data modeling language) – was first published in 2010.

YANG models define configuration data, operational state data and actions. YANG data models that describe devices and services (configuration and state information) are transported via the NETCONF protocol. The configuration is plain text and human readable/writable. Importantly, YANG models are easily parsed by a computer, unlike CLI. This is the key to enabling programmability.

Standard YANG models are being developed by a number of organizations, including the IETF, IEEE, MEF, OpenDaylight, BBF and OpenConfig (a consortium of webscale and telecom operators). As Rich Bennett and Steven Nurenberg of AT&T write in Building the Network of the Future: "YANG templates, which are vendor-neutral, create a portable and more easily maintained definitive representation of desired network functionality."

Their colleague, Hank Kafka, writes in the same book: "Consistent use of NETCONF and YANG for the management of network devices and the specification of data models plays a critical role in driving interoperability between access technologies and network elements from multiple suppliers… NETCONF and YANG, in and of themselves, provide a strong technology base for driving simpler, more cost effective, and more robust configuration management. Additional flexibility, adaptability, and cost savings will be achieved by the definition of open and standard YANG configuration data models." Kafka goes on to cite a number of BBF YANG models for G.fast, PON and other access technologies, adding that "Use of these models by network access equipment… will be essential for achieving the benefits of the new access architecture."

In their chapter on NFV in the same book, John Medamana and Tom Siracusa explain how AT&T has created a YANG data model design studio with models pushed to the VF (Virtual Function) via NETCONF. They use YANG to define the data model for a service at the device layer, network layer (e.g., VLANs), service layer (e.g., Ethernet WAN) and finally the API layer (which connects to BSS).

In Raj Savoor and Kathleen Meier-Hellstern's chapter on Network Measurements, they note that "Utilizing YANG models that cover not only the configuration but also the operational state of various feature components supported by each network element, one can transform a fairly rigid component measurement solution, utilizing legacy technologies such as CLI and SNMP, to a flexible measurement solution, centered on data models."

Representational State Transfer (REST) Configuration Protocol – RFC 8040

The IETF standardized RESTCONF as a REST-like API to access YANG data models. RESTCONF is still a relatively new protocol, and the ecosystem is comparatively small. Some orchestrator vendors are using RESTCONF in their northbound API to give access to their abstracted

YANG data models. For southbound communication, between the orchestrator and network functions, the NETCONF/YANG approach is used (see **Figure 1**).

The rationale for SDN controllers and NMS/EMS (or any network connectivity management-related software) to use REST and/or RESTCONF northbound (to connect to IT/OSS systems) is to reach a wide audience of developers that already know REST APIs well and have toolsets and frameworks to handle this type of integration.

REST is an API style that is the current *de facto* standard for IT system integration, with a huge existing ecosystem. However, REST is a concept, not a standard – hence, while REST APIs offer programmability, they are proprietary. In order to manage multi-vendor networks and ensure interoperability, the industry needs a standards-based approach such as RESTCONF or NETCONF.

NETCONF/YANG SUPPORTS NFV

NETCONF/YANG is a key enabler of the NFV paradigm, where services can be designed at a high level, independent of the complexities and device dependencies of the underlying infrastructure. This "infrastructure" will be assembled dynamically from a range of virtualized functions from multiple vendors. For NFV to succeed, the provisioning of services must be highly automated, which requires straightforward management and configuration.

The dynamic nature of heterogeneous virtual networks, with functions being spun up and changed continuously, requires highly automated, programmable management. The combination of NETCONF and YANG provides operators with greater flexibility in the programmable management of network devices, both physical and virtual.

A wide range of SDN projects and industry organizations, such as the Open Networking Foundation, the OpenDaylight Project and the Open Platform for NFV promote NETCONF/YANG as a universal southbound protocol for the configuration and management of both virtual network functions (VNFs) and physical network devices in SDN environments.

SDN/NFV promise to revolutionize network operations through automation and open APIs, driving down operating costs and increasing the speed with which new services can be delivered. However, many of these benefits can be achieved today, without the complications of virtualization, simply by adopting a programmatic and standards-based way of writing configurations to any network device from any vendor, replacing the manual configuration of tens or hundreds of devices that service providers must currently undertake to deliver a service to an individual customer.

CONCLUSIONS

NETCONF provides a standardized network API that, unlike CLI, enables programmability and provides a common approach to configuration across multi-vendor environments. Critically, NETCONF implies transactionality, which significantly reduces the scope for network configuration errors due to partial implementation. Riding on top of NETCONF is YANG, the language that specifies the schema for defining device and service models, enabling operators to leverage common definitions from standards bodies such as BBF and consortia such as OpenConfig.

Service providers are driving the adoption of NETCONF/YANG, demanding that network equipment vendors provide equipment that fits within their automation environment. NETCONF/YANG support is a "must have" requirement to virtual and physical equipment providers in service provider RFQs.

Arguably, Deutsche Telekom has led the way, adopting NETCONF/YANG as part of the next-generation OSS it is adopting for its TeraStream infrastructure. (See this presentation from 2013 about DT's TeraStream project, which outlines the importance of using data models in the new architecture.)

AT&T has also been an early adopter of NETCONF/YANG. As Michael Satterlee and John Gibbons of AT&T discuss in their chapter of Building the Network of the Future: "Today, the industry has more or less standardized on using YANG as a data modeling language for networking environments, [which] has allowed us to improve both cost and time associated with service development while reducing the typical errors seen in prior processes. Just like an architect creates a blueprint reference before constructing an actual building, the network designers can now create YANG-based blueprints before worrying about the choice of vendor device, capability of each device, etc. Once the service models and device models are ready, software tools are used to generate device-specific CLIs, thus restricting human intervention only to the service design stage."

Verizon is similarly a supporter of NETCONF/YANG, noting in its SDN-NFV Reference Architecture white paper that "The choice of the southbound protocol depends on the use case and will also require additional adaptations/extensions. NETCONF (RFC 6241) is emerging as the primary candidate for provisioning network functions in general." The paper goes on to state: "YANG has emerged as the data modeling language for the networking domain."

Given this strong demand for NETCONF/YANG, many vendors now support these protocols on their latest-generation network devices. However, as this blog from a Deutsche Telekom network architect shows, implementing NETCONF/YANG is not simply a tick-box exercise for an RFQ. Network device vendors should not make device management an afterthought. Updating multiple management interfaces adds cost for network device vendors. As such, they should aim for reuse in management interfaces, rather than writing separate code for each. It may make sense to use third-party tools to develop complete and reliable interfaces in order to accelerate time to market.

The benefits of network automation are clear: It is about speeding up operations and reducing errors. And it is about reducing the time needed for deployments and configuration changes, so that service providers can be more responsive to their customers, particularly on the enterprise side.

Automation is also about reducing errors that impact customer experience and the bottom line, if SLAs are breached or churn rises. Different engineers working on the same network will often implement changes via CLI in slightly different ways, and if they are copying and pasting changes from Notepad, it is easy to introduce an error.

By automating, we can make configuration changes more consistent and reduce the scope for human errors. Fewer errors translate to lower operating cost. Automation can also reduce capital cost, as it increases the reliability of the network, so network planners don't have to massively overprovision to compensate for uncertainty. By specifying and providing the network API, NETCONF and YANG are key to enabling this automation.

ABOUT TAIL-F

Tail-f, a Cisco company, is a pioneer and leader in network programmability and data model-driven device management. Tail-f has contributed heavily to the development of the NETCONF, RESTCONF and YANG standards in IETF. As the worlds of SDN and NFV have evolved, the benefits of using programmable devices with standard APIs and protocol interfaces have come a long way.

The problem is, the network has evolved. Traditional methods of managing devices are costly and slow, especially with the sprawl of new devices needing to speak to other devices and objects. They don't speak the same languages, and enhancements to the devices are often not supported. Tail-f realized that the key to fixing this situation is understanding the benefits of network orchestration in helping organizations reduce opex, make deploying and removing services easier, and improve reliability and network flexibility. Ultimately, network equipment providers need a way to simplify operations (which ultimately reduces network costs) and replace the traditional, slow, high-touch programmability model with a real-time, automated approach.

Seeing this need, Tail-f developed a unified data model-driven software framework solution for all management interfaces (NETCONF, RESTCONF, CLI, SNMP, WebAPI), supporting any user preference and described by the standard YANG data modeling language. Tail-f's ConfD software framework leverages the NETCONF and RESTCONF protocols and YANG data models and a centralized configuration database to deliver unified, programmable management capabilities.

Today, creating new physical and virtual devices has never been easier or faster. Network equipment providers can now expedite the development and deployment of new hardware in less time while meeting their business objectives. Tail-f is the first to deliver this flexible data model-driven framework that includes a variety of open standards-based management interfaces, including NETCONF and RESTCONF. Tail-f's ConfD solution is ideal for producing programmable network elements that meet the next-generation networking requirements for NFV and SDN technologies.

For more information, please visit www.tail-f.com.