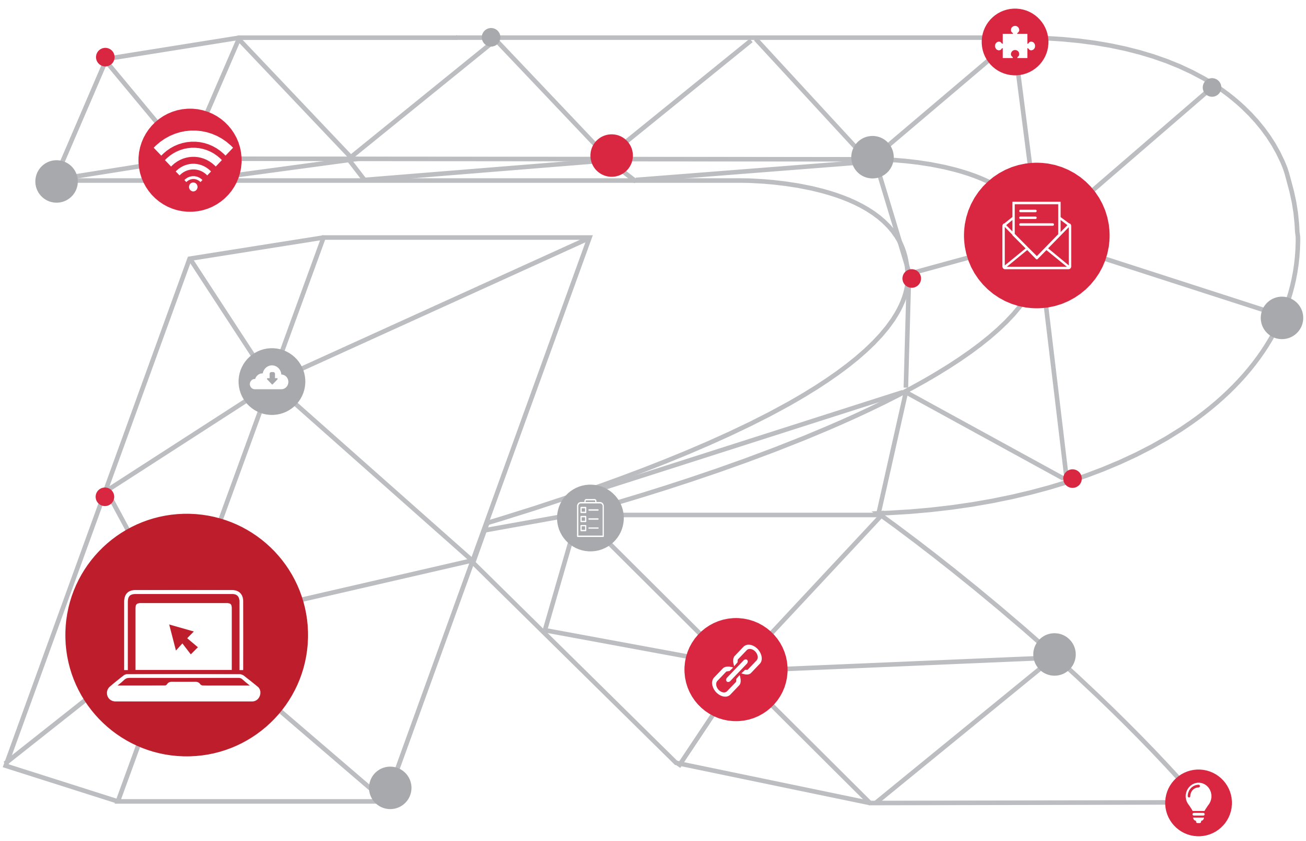
White Paper

A Feast of Technologies The "Sword of Giant" for Maintenance in IDCs



Contents

Do Technology Innovations for Service-Driven Networks Bring an "Angel" or a "Devil"?

With the vigorous development of Internet businesses, technologies such as big data, artificial intelligence (AI), and Remote Direct Memory Access (RDMA) have been widely used. They cause continuous traffic growth in Internet data centers (IDCs) and require end-to-end, low-latency, and lossless forwarding on infrastructure networks. To address the challenges, the industry promotes upgrade of Ethernet switch chips, which brings the following benefits:

* Upgraded chip performance

The conventional 10G Ethernet is upgraded to the current popular 25G Ethernet. Some providers have even begun to deploy high performance computing (HPC) clusters based on 100G Ethernet.

* Rich operation and maintenance (O&M) features

The upgraded chips provide more enhanced capabilities, such as shared buffer, in-band network telemetry (INT), priority-based flow control (PFC), explicit congestion notification (ECN), mirror-on-drop (MOD), and transient capture buffer (TCB).

For example, when RDMA is used, a switch needs to use a complex combination of features to ensure its stable running. "Light coupling" with services increases maintenance difficulty.

"Sword of Giant" for Maintenance

As network device technologies become increasingly complex, to ensure reliable running of services, you must have a deep understanding of the internal statuses of network devices by visualizing the status changes. Nowadays, development and operations (DevOps) is dominant, and the selection of a northbound interface for a switch becomes very important.

Conventional tools such as the command-line interface (CLI) and Simple Network Management Protocol (SNMP) cannot meet the requirements of automated maintenance in terms of performance, efficiency, and automation. Based on practices of some Internet giants in the industry and a deeper understanding of Google remote procedure Call (gRPC), we can foresee that a gRPC-based maintenance interface is expected to be the most important automated maintenance tool in the future. Before sharing gRPC with you, let's first analyze the bottlenecks of switch maintenance in IDCs.

Bottlenecks of Switch Maintenance

To implement automated maintenance, switches must support the following operations:

* Get: Obtain status and configuration information.

The maintenance platform obtains key configuration or software and hardware status from the switch as needed. The configuration information includes the Border Gateway Protocol (BGP) and security configuration. The status information includes information about the interface traffic, interface status, buffer queue length, and packet loss.

The operation must meet requirements for preventive maintenance inspection (PMI) and troubleshooting in IDCs.

* Set: Deliver configurations.

The maintenance platform delivers configuration changes to the switch, such as shutting down a port, configuring an IP address, and configuring a threshold.

The operation must meet requirements for routine service changes.

* Alarm: Report abnormal statuses.

The switch reports notifications to the maintenance platform when certain trigger conditions are met. For example, the switch reports notifications when the CPU utilization reaches the security threshold, the queue reaches the threshold, and the port goes up or down.

The operation must meet requirements for reporting alarms upon abnormal statuses.

* Push: Periodically report key status information.

Devices periodically report some status information, such as information about the interface traffic, queue threshold, and error packets received on interfaces.

The operation must meet requirements for continuous monitoring on key metrics.

Neither the conventional CLI + Syslog and SNMP nor the popular Network Configuration (NETCONF) and OpenConfig can meet all the requirements on the preceding routine operations. In addition, bottlenecks related to performance, compatibility, scalability, and standardization occur. A combination of maintenance interfaces must be used to meet the requirements for rapid and continuous integration on the automated maintenance platform. Table 1 analyzes these maintenance interfaces.

Table 1: Capability Analysis on the Four Maintenance Interfaces

Maintenance Interface	Item	Get	Set	Alarm	Report
CLI+Syslog	Capability	It provides complete **show** commands.	It provides complete **config** commands.	It generates alarms when conditions are triggered.	N/A
	Disadvantage	>The **show** command varies with the manufacturer. >Different **show** command outputs may have different formats. Secondary adaptation on the platform is complex.	>CLI commands provided by different manufacturers are incompatible with each other. >Development is inflexible. Continuous unification of CLI commands may fail. >Remote operations have low reliability. The platform must adapt to the command output.	>Flexibility is low. Alarms are generated based on key events of the OS. >Only a few features can be adjusted.	>The CLI cannot periodically report results. >Scripts need to be compiled on the console to periodically collect monitoring results by using the CLI.
SNMP	Capability	It provides a complete readable management information base (MIB).	It allows modifying a few features by using the MIB.	It provides complete trap alarms.	N/A

Maintenance Interface	Item	Get	Set	Alarm	Report
SNMP	Disadvantage	>MIBs are proprietary to manufacturers. >MIB coverage varies with the manufacturer. >The polling interval is long and the real-timeliness is low. >Traversal performance is poor. >Time precision is low.	>MIB nodes vary with the manufacturer. >Only a few features are covered. >MIBs are proprietary to manufacturers.	>MIBs are proprietary to manufacturers. >MIB capabilities vary with the manufacturer.	>SNMP does not support this operation. Scripts must be compiled on the console to periodically collect key information. >The polling interval is long and the real-timeliness is low.
NETCONF	Capability	It obtains the configurations and status information based on the YANG model.	It delivers configurations based on the YANG model.	It sends alarm notifications based on the YANG model.	N/A
	Disadvantage	YANG models are proprietary to manufacturers.	YANG models are proprietary to manufacturers.	YANG models are proprietary to manufacturers.	NETCONF does not support this operation. Scripts must be compiled on the console to periodically collect key information.
OpenConfig	Capability	It obtains configurations and status information based on the standard YANG model.	It delivers configurations based on the standard YANG model.	It sends alarm notifications based on the standard YANG model.	It allows setting a report period based on the subscription information.
	Disadvantage	Many features are not defined.	Many features are not defined.	Many features are not defined.	Only a few features are available for subscription.

Table 2 summarizes the advantages and disadvantages of the four maintenance interfaces based on the preceding analysis.

Table 2: Advantages and Disadvantage of the Four Maintenance Interfaces

Maintenance Interface	Advantage	Disadvantage
CLI+Syslog	It supports complete **config** and **show** commands and can perform all configuration and status queries.	>The CLI commands and command outputs of different manufacturers are incompatible with each other. The maintenance platform needs to adapt to commands and command output, or perform secondary encapsulation by using software, which requires heavy workload. >The upgrade of the switch OS may cause changes to the CLI commands and command output and the maintenance platform needs to apply the changes accordingly. >This interface has weak capability and low flexibility for alarm generation and periodical reporting.
SNMP	>It is mature and supports many third-party network management tools. >It supports basically complete MIBs.	>The MIBs are proprietary to manufacturers. The platform needs to adapt to the MIBs multiple times. >The polling mechanism has low efficiency. Only a few nodes can be monitored. The network scale is limited. >The time granularity is coarse. Data cannot be obtained in real time. >Frequent Get operations may increase the CPU load of the switch. >This interface is weak in performing the Set operation and lacks the capability for periodic Push operations.
NETCONF	>It uses a unified and mature transmission framework. >It is extensively used. >It supports complete Get and Set mechanisms.	>YANG models are proprietary to manufacturers. The platform needs to adapt to the YANG models multiple times. >XML is used as the data description language, which is poor in writing and readability and has low transmission efficiency. >This interface lacks the capability for periodic Push operations.
OpenConfig	>The YANG model is defined based on the OpenConfig work group and is adapted in a unified manner. >The underlying transmission framework is flexible and standard. >The YANG model supports data description in a tree structure, and therefore is easy to manage and scale.	>The YANG model supports incomplete features. It is not standardized enough to meet the requirements of architecture evolution. >Notifications and periodic reporting cannot be flexibly defined and are inefficient. >This interface is not widely supported or used and is immature.

According to the preceding summary, the current common northbound interfaces are not mature enough. They cannot meet the requirements for unified maintenance and continuous integration in the multi-manufacturer network in the future. From another perspective, the preceding northbound interfaces are generally not easy to modify and are uncontrollable. This leaves little room for maintenance personnel to redefine them. What is the ideal northbound maintenance interface for maintenance personnel?

Ideal Northbound Maintenance Interface in the Future

Based on the preceding analysis, we believe that we need to redefine northbound maintenance interfaces to support continuous, simple, and unified integration on the automated maintenance platform. An ideal maintenance northbound interface in the future needs to support the following features:

* Independent of manufacturers

A standardized model is defined based on the maintenance platform. Continuous adaptation and modification on devices from various manufacturers are not required.

* Standardized YANG model

A unified standard YANG model is defined based on the maintenance system of the customer. The unified standard YANG model can continuously iterate and evolve, and is not limited to the OpenConfig organization or proprietary YANG models of manufacturers.

* Comprehensive maintenance capabilities

Get, Set, Alarm, and Push capabilities are supported. The four capabilities are delivered and subscribed to on a unified interface.

* Unique maintenance interface

The unique maintenance interface is redefined. The automated maintenance platform can manage devices of all manufacturers by using the unique standard interface. From the perspective of technical details, the future northbound maintenance interface must support the following capabilities:

* Structured northbound interface

Learning from the layered protocol architectures of NETCONF and OpenConfig, the interface separates the modules for data encoding, capability model, remote calls, data transmission, and security. With the layered protocol architecture, the interface can achieve decoupling, ensuring rapid iteration of the standard interface.

* Intuitive and efficient data description

The data model can be described in JSON, instead of XML or Protocol Buffer (Protobuf). This simplifies data compilation and improves readability. In addition, changes to the data model do not affect serialized transmission of underlying data, such as Protobuf data.

* Unified tree-structured YANG model

Based on the capability model of switches, a tree-structured YANG model is built based on different functional modules, such as BGP, Open Shortest Path First (OSPF), security, and interface modules. In addition, Get, Set, Alarm, and Push capabilities are integrated in different functional modules.

* Efficient data transmission

Data is serialized into binary data for transmission and then deserialized. This ensures efficient data transmission in the conventional text mode.

A single TCP connection can be multiplexed to transmit multiple streams, thereby improving efficiency.

* RPC-based remote call decoupling

Remote calls are made by using the APIs implemented based on the RPC framework. In this way, switches are decoupled from the maintenance platform, ensuring that they are transparent and independent of each other.

* Secure and reliable data transmission

RPC calls require a complete authentication mechanism.

Data transmission requires secure encryption.

Although the preceding description is only a vision for the future northbound maintenance interface, comprehensive and unified management of switches is an urgent demand. The interface must enable all switches to support the Get, Set, Alarm, and Push operations based on the maintenance platform. Does such an interface exist in reality? An interface based on gRPC and Protobuf may be a possible choice.

gRPC-based Unified Maintenance Interface

The following figure shows the maintenance model based on gRPC and Protobuf.

1. The controller subscribes to or unsubscribes from real-time or periodic events.

2. The switch saves or deletes the address and port number of the subscription server, and subscribed events.

3. Based on the subscribed events, the switch constructs data in JSON, encapsulates the data into packets by using Protobuf, and sends a Proto Request message to the server over gRPC.

4. After receiving the Proto Request message, the server decapsulates the message by using Protobuf to restore the data in JSON for service processing.

5. After processing the data, the server encapsulates response data by using Protobuf, and sends a Proto Reply message to the switch over gRPC.

6. After the switch receives the Proto Reply message, the gRPC-based interaction ends.

gRPC is the key transmission framework in the design of the unified maintenance interface, but it is not the only concerning factor. You also need to focus on the following factors:

* Data: the final data to be transmitted, used to support the Get, Set, Alarm, and Push operations.

* Unified YANG model: unified description of the data model in JSON. The YANG model integrates the network architecture and maintenance requirements in a tree structure.

* gRPC: unified northbound interface. Data is sent or obtained by using RPCs, so that remote objects are called as local ones.

* Protobuf: defines the RPC interface services in .proto files, serializes data, and deserializes the data, to improve data transmission efficiency and lower required bandwidth.

* Netty + HTTP 2.0: provides bidirectional stream multiplexing on a reliable network connection. A combination of HTTP 2.0 and Netty simplifies network programming.

gRPC is an HTTP 2.0-based high-performance, open-source and general-purpose RPC framework. The most important and difficult point in gRPC lies in establishment of a unified YANG model. Although OpenConfig defines a large number of standard YANG models to resolve the problem of uniformity and compatibility, such YANG model definition mode cannot meet the requirements for rapid iterations of infrastructure network maintenance. Therefore, leading Internet companies are called on to take the lead in sorting out a unified YANG model that can be continuously supplemented and improved by everyone. This can reduce the cost of multi-party interaction on the maintenance platform and allows the maintenance platform to focus on maintenance capabilities.

Summary

The gRPC and Protobuf-based northbound maintenance interface has been used in Ruijie switches to meet the maintenance requirements of some features. For example, this interface can comprehensively manage buffers of switches, to be specific, monitor the buffers for ingress and egress ports and queues in real time, collect metric data such as the number of buffer threshold exceeding times periodically or at an interval of several seconds for ports and queues, and automatically trigger alarms when the ingress or egress port buffers encounter packet loss due to insufficient space, or port buffers exceed the thresholds. In a word, this interface can meet the requirements for visualized and real-time maintenance. However, there is still a long way to go before SNMP and other protocols are replaced. It is believed that unified management on more maintenance capabilities can be implemented based on gRPC in the future.

Regarding the principle of gRPC, we will provide further description in subsequent articles. Stay tuned.