Closed Loop Engine for Network Automation

Pigilam Swetha*, Monicka Vijayakumar*, Jahfer Abdulazeez*, Prem Kumar B*

* Smart World & Communication, Larsen & Toubro Constructions

Abstract

In this paper, we discuss the Closed Loop Engine for automation of Network Deployment, Operation, and Maintenance. A closed-loop engine is a framework that consists of a combination of multiple components that can make the network functionalities automated with minimal manual interruption. It is a process of performing a blend of operations such as orchestration, management, telemetry, analytics, customer-facing function, and business support system with programmed actions. The telecommunication industry faces multiple challenges as the technology evolves, hence the network is expected to be vendor agnostic, support integrations, enable business use cases, service completion, withstand the complexities and ensure scalability. All these challenges will be modeled and resolved using well-designed workflows, which is the basis of a closed-loop engine. In this paper, we explain the proposed workflow that enables the closed-loop engine for network automation.

Keywords — Automation, closed-loop, NEs, Workflows, Orchestration, YANG, NetConf, SNMP, EMS, OSS, NMS.

1. Introduction

With the billion number of devices in today's world, it is impossible to drive these networks by humans. Hence in the Telco industry, there is a transition towards virtualization and automation of networks which reduces the need for human intervention.

The buzz word- Automation is a methodology by which an operation is performed in a controlled environment with a programmed approach. It provides simplification in the overall process by making it unmanned and overcoming the challenges owing to complexities caused by an increased number of devices in the network and emerging technologies such as IoT, 5G. To improve the operational efficiency, the processes which are repetitive, time-consuming and prone to errors need to be automated.

A closed-loop engine is an automatic control system in which operation, process, and mechanism are regulated by feedback. It provides fewer errors with high stability. A closed-loop engine does a job that is dependent on the control actions provided by the system. It is designed in such a way that all the network processes such as Orchestration, management, telemetry, analytics, customer-facing function, and business support system will be performed automatically. A feedback loop is used for communication which includes monitoring of all network devices, identifying the errors and optimizing the performance of the network. It monitors the real-time network traffic generated by the network elements, analyses the demands of the traffic, checks the availability of resources required and accurately places the traffic in order to optimize the performance of the network and provides greater service agility. It also monitors and provides access to the failure scenarios, congestion situations and accordingly corrects those issues by following certain workflows. The workflows design actions such as triggering the alarms, root cause analysis of the alarms and probing actions to clear the alarms. In this paper, a closed-loop engine is proposed for network infrastructure automation and network service automation. A Closed-Loop Automation also stitches the functionalities of the vital components like- the network elements and the software elements such as Network Management System, Business Support System, Automation and Orchestration, Element Management System and Analysing engine.

2. YANG Model

The programmability of actions starts with data modeling of the configurations that are to be pushed to the network elements. A data model is a blueprint that ensures visibility to structure, syntax, and semantics of the data, where the data is derived from various protocols. It models both the configuration and state of network elements. YANG- Yet Another Next Generation is the Data Model for the NetConf configuration management protocol. This combination of YANG (DataModel) and NetConf (Protocol) enables a standardized way to programmatically update and change the configuration for domain agnostic devices as shown in Fig 1.

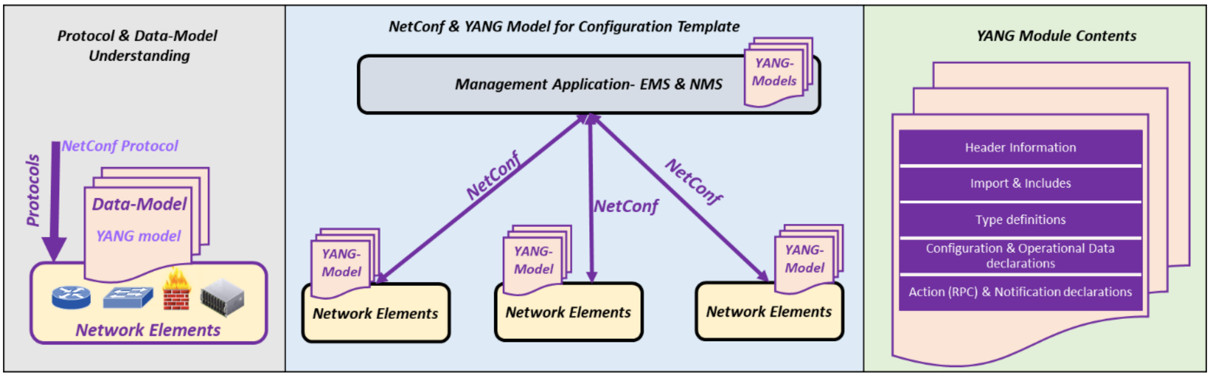
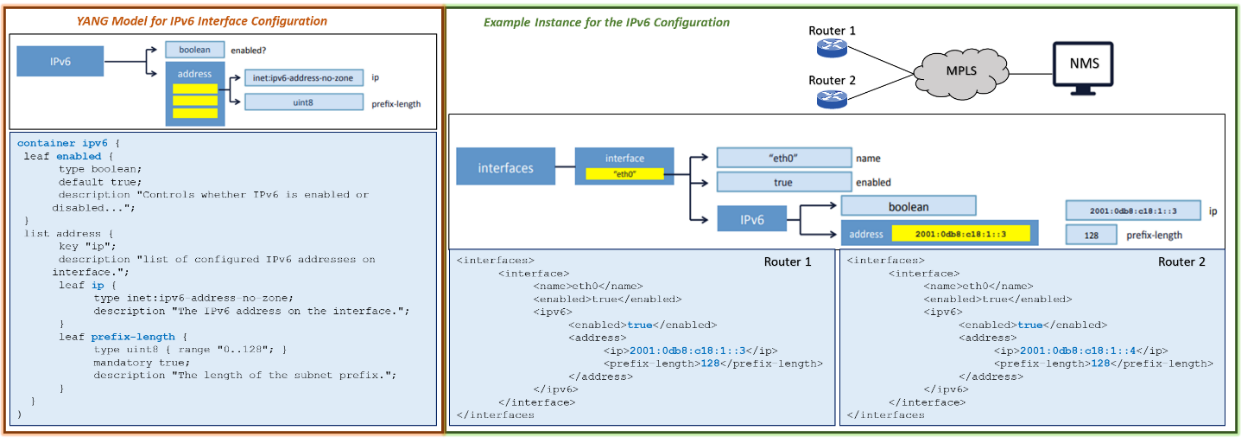


Fig 1. YANG Model

The YANG data model is structured into modules and sub-modules. YANG module has a tree structure as shown in Fig 1. It contains the header information, Imports, Includes, Type definitions, configurations and operational data declarations with action and notification declarations. It has built-in types which are binary, bits, Boolean, decimal64, empty, enumeration, and identityref. String is used to name any interface, identityref is used to represent the type. Boolean is used to show whether the interface is enabled or not and enumeration is used to show the link status. New types also can be defined using typedef command.

Consider a network with two routers connected over MPLS as shown in Fig 2. As shown, Router 1 has an Ethernet interface with IPV6 address 2001:db8:c18:1::3/128 and Router 2 has an interface with IPV6 address 2001:db8:c18:1::4/128. To configure IPV6 addresses to these interfaces, the commands are shown in Fig 2.



```
container ipv6 {
 leaf enabled {
      type boolean;
      default true;
      description "Controls whether IPv6 is enabled or
     disabled...";
 }
 list address {
      key "ip";
      description "list of configured IPv6 addresses on
     interface.";
      leaf ip {
           type inet:ipv6-address-no-zone;
           description "The IPv6 address on the interface.";
      }
      leaf prefix-length {
           type uint8 { range "0..128"; }
           mandatory true;
           description "The length of the subnet prefix.";
      }
  }
)
```

Router 1

```
<interfaces>
      <interface>
            <name>eth0</name>
            <enabled>true</enabled>
            <ipv6>
                  <enabled>true</enabled>
                  <address>
                        <ip>2001:0db8:c18:1::3</ip>
                        <prefix-length>128</prefix-length>
                  </address>
            </ipv6>
      </interface>
</interfaces
```

Router 2

```
<interfaces>
      <interface>
            <name>eth0</name>
            <enabled>true</enabled>
            <ipv6>
                  <enabled>true</enabled>
                  <address>
                        <ip>2001:0db8:c18:1::4</ip>
                        <prefix-length>128</prefix-length>
                  </address>
            </ipv6>
      </interface>
</interfaces
```

Fig 2. IPv6 Interface configuration using YANG

In the above configuration, the usage of container, list and leaf statements is shown. The container is one argument identifier that defines an interior data node that has no value but contains a list of child nodes. The list represents a collection of entries that consists of one or more nodes. In the above configuration, the list is used to define the set of IPv6 addresses and prefix lengths using a set of leaf nodes. Leaf Name can serve as a list key.

3. Network Automation

Network Automation can be understood as deploying, provisioning and activating network and services in an unmanned method. On a day to day basis, a large amount of network traffic is generated by multiple devices and multiple applications which involves video, audio, text, etc. To support this exponential growth of network traffic and to mitigate troublesome network risks, driving towards automation will be a one-stop solution. Network Automation can be implemented for cloud environments, local area networks, data centers and virtualized environments. It provides enhanced network agility, reliability, and visibility into the network. Network Automation is broadly classified into two. 1. Network Infra Automation 2. Network Service Automation.

3.1 Network Infra Automation – Zero Touch Provisioning

Generally, setting up the network infrastructure requires a lot of manual work in configuring every network device. This is where Zero Touch Provisioning plays a vital role. A typical ZTP helps in reducing the several manual steps involved in provisioning a network element to two simple steps of Racking and Stacking and Powering the device on.

Consider a Scenario where Network Elements such as routers, switches, etc. has to be connected to an existing network with DHCP enabled. As shown in the Fig 3, the network device will send a DHCP Discovery message to get assigned an IP address. In return, DHCP server will send DHCP offer message to the network device. Then network device will request for IP address and other relevant information which it requires, by tagging an ID to the DHCP request packet such as option 32 to the DHCP server. Finally, DHCP server will send an IP address and a URL along with the appropriate details to the network device. Now, the NE downloads the appropriate configuration or script based on the NE's unique identifier from the configuration server. Finally, NE updates the EMS components.

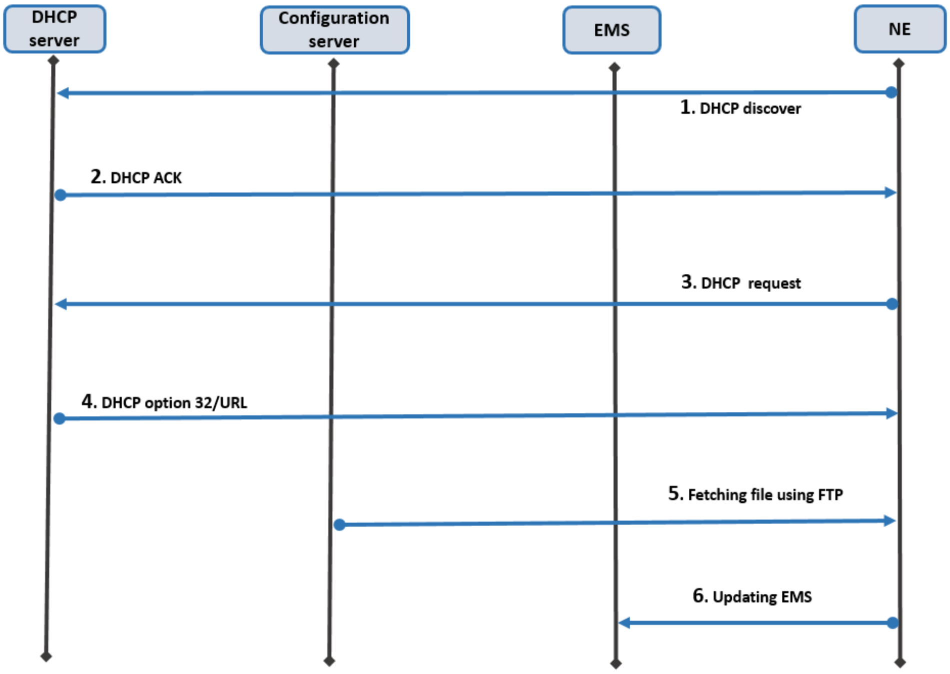


Fig 3. Zero Touch Provisioning in Network Infra Automation

This is a typical case of Zero Touch Provisioning for the network devices. ZTP avoids any unwanted protocol churn by permitting the switch to only use its ports to find and download the required software and configuration data. Here provisioning can be done just by a single click without touching the network devices physically. On a single click, all these steps will be done automatically. This is called Network Infra Automation.

3.2 Network Service Automation

Traditionally, element managers need to be updated with the creation and deployment of services end to end and then these changes have to be integrated with the Operational Support System (OSS). Network Services Automation refers to adding, changing and deleting the services without disrupting overall service and ensuring that services are delivered in real-time.

Consider the diagram shown in Fig 4. Let's assume that every router has 3 slots of 1G ports. For example, if there is a request for an L2 VPN connectivity between A and D, provisioning has to be done in the first step so that the particular service can be enabled between A and D. To start with, a few checks have to be done such as availability of L2 VPN, required resources and inventory check. Post this, resources will be reserved and corresponding templates will be pushed to A and D to enable the service. All these processes will be accomplished automatically in Network Service Automation.

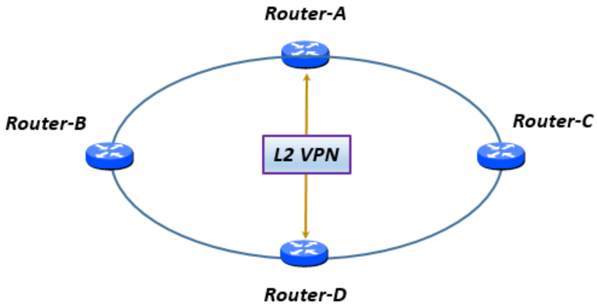


Fig 4. Basic Network Topology

OSS will maintain the Network Service Catalogue and Automation and Orchestration (A&O) platform will maintain the service inventory, system resource inventory required for service provisioning such as VLAN, IP address, etc. A&O is responsible for setting up all the processes involved to enable a service in network elements. The network flow for the service provisioning is as shown in Fig 5. As the request arrives, OSS fetches the inventory from EMS. Then it checks the feasibility of providing the requested service by the network devices by checking the resources such as ports, VLANs, IP addresses required for service and the capability of providing service. If they are available, resources will be reserved in the network elements to provide the requested service. Then A&O will push the corresponding service configuration to the network device through EMS. Once the service configuration is done in the network device, the service status will be updated in EMS. Then A&O will perform the service provisioning for the network devices by pushing the corresponding templates using YANG Model to the network devices. A&O updates NMS with current resources to update the inventory and NEs also updates EMS about the changes done.

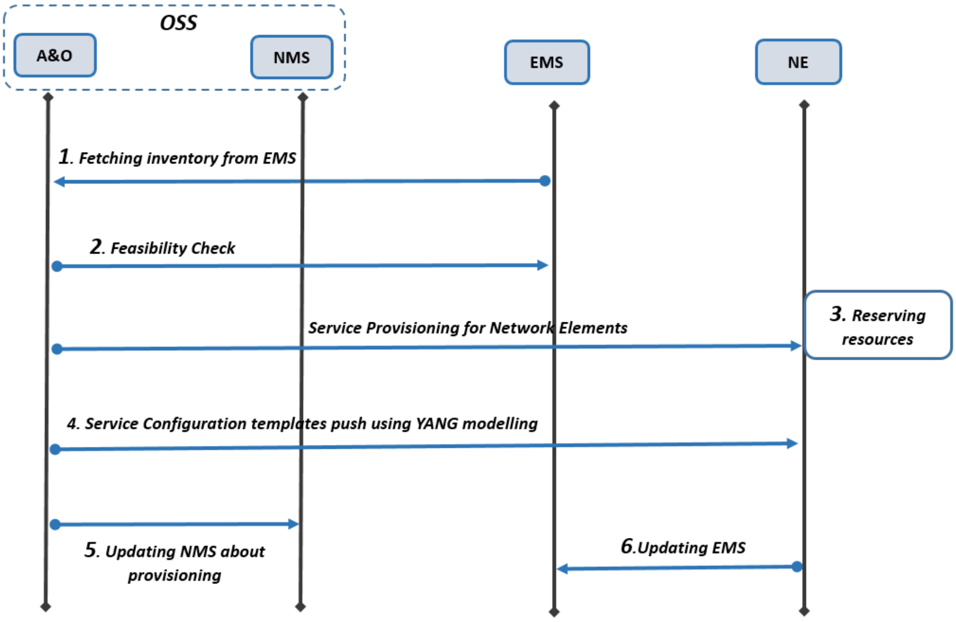

OSS
A&O
NMS
EMS
NE
1. Fetching inventory from EMS
2. Feasibility Check
Service Provisioning for Network Elements
3. Reserving resources
4. Service Configuration templates push using YANG modelling
5. Updating NMS about provisioning
6.Updating EMS

Fig 5. Network Service Automation Flow

Here YANG modeling is a data modeling language used for the configuration of network devices. All the processes involved in service provisioning will be automated to improve the operational efficiency of providing service to users without any manual process.

4. Closed loop Automation

Closed loop automation is a strategic installation in a service provider network or a wide enterprise network to maintain the end to end network health and manage the huge amount of data traffic. It is a continuous feedback loop which will monitor, verify, configure and maintain the network on a real time basis and offers self-healing to the network on which it is deployed. The feedback loop process involves continuously validating the network on a real time basis, in case any fault is found, necessary remediation is done and then again validates the network like a loop.

Closed loop automation is automating the communications happening between network elements and other software elements like Orchestration, BSS, EMS, NMS. This communication involves configuring devices by pushing templates, clearing alarms by pushing the correction template, updating the changes, monitoring devices etc. Network infra automation and network service automation along with closed loop engine leads to the performance optimization of the network.

It is explained using a failure scenario where an alarm will be generated and how this closed loop automation will resolve the alarm as shown in Fig 6. Whenever there is a change in some parameters of network devices such as temperature, CPU utilization, etc., alarms will be pushed to the EMS which will be updated in the NMS. Policy engine collects the alarms from NMS and send it to the analysing engine for analysing the alarm. A policy engine works with a combination of programmed rules and the network analytical engine. When a network alarm is triggered by a NE on an EMS or a NMS system, rather than notifying some network personnel about the alerts in forms of ticketing and other services, Policy engine will apply varying degrees of automation taking into consideration several factors such as type of alarms, analysing the history and correlating the events with a rule association technique and finally taking the corrective measures to fix the alarm or resolve the incident by providing the necessary algorithm.

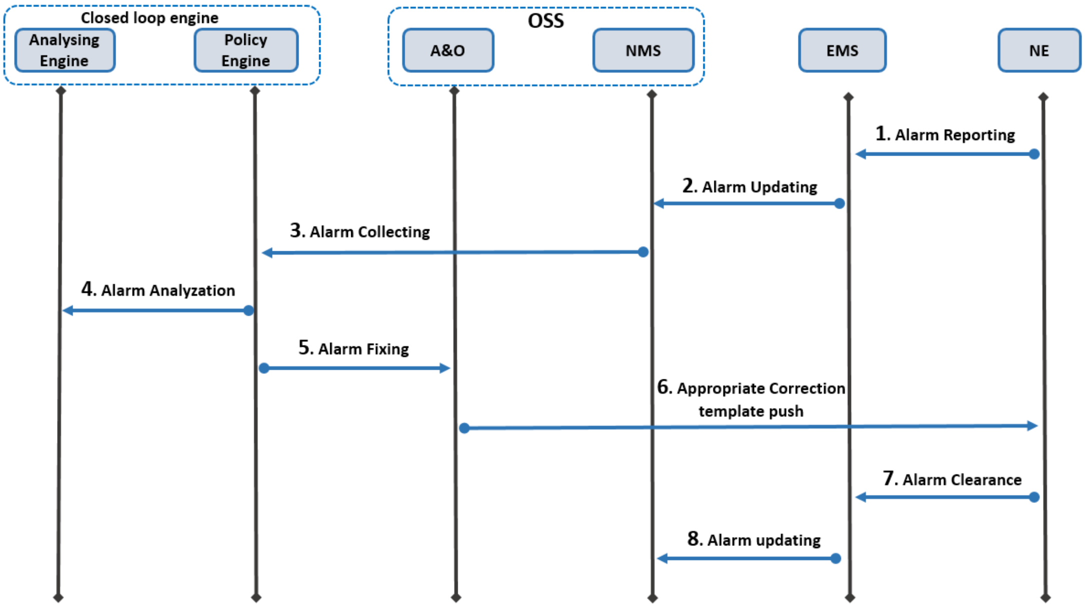


Fig 6. Closed loop Automation flow

Automation and Orchestration will push the corresponding correction template to NEs. Finally, the Alarms will be cleared and network elements update about alarm clearance status to the EMS. This is how an alarm generated at the network elements will be analyzed and cleared with a closed loop automation.

5. Use cases

Closed loop automation helps to perform automated troubleshooting. It involves the heavy amount of data collection from different elements which can be automated easily. This helps in changing the device configurations very easily whenever there is a requirement by automating the process. This also helps in developing the self-driving networks where the system responds to changes and external events and adjusts the configuration accordingly in the real-time in the network devices automatically. This can be used to perform predictive analytics by monitoring the network for some time and taking a decision accordingly regarding corrective actions. This can be used mainly for the processes which are error prone by reducing human intervention. This helps in analysing the traffic for security issues which may inject malfunction into the network and to change the policies accordingly to avoid the malfunctions. As this can help to perform the calculation of bandwidth demands and resources availability, this provides optimal service quality by determining the placement of best amount of traffic according to the defined policies. On the whole, Closed Loop Automation can help in analysing different types of network traffic, collecting the performance metrics and optimising the overall network performance.

6. Conclusions

By implementing closed loop engine for network automation, there will be enhanced noise reduction ability and more accuracy by eliminating the manual and tedious processes. It also provides more stability, resiliency, and reliability by simplifying the troubleshooting process which involves the collection of a heavy amount of data. This will help to reduce the risk by decreasing downtime and increasing agility. This, in turn, will also minimize the operational expenses with automation and optimizes the performance of the network that would help to increase the business values.

7. References

[1] Jürgen Schönwälder, Martin Björklund ,Phil Shafer "Network configuration management using NETCONF and YANG",October 2010, IEEE Communications Magazine 48(9):166 - 173.

[2] Tarik Taleb , Ibrahim Afolabi, et al,"On Multi-Domain Network Slicing Orchestration Architecture and Federated Resource Control", IEEE Network (Volume: 33, Issue: 5, Sept.-Oct. 2019).

[3] Yuri Demchenko, Sonja Filiposka, Raimundas, et al, "Enabling Automated Network Services Provisioning for Cloud-Based Applications Using Zero Touch Provisioning", 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC).

[4] Miguel Lopes, Braga, Portugal, Antonio Costa, et al"Automated network services configuration management",2009 IFIP/IEEE International Symposium on Integrated Network Management-Workshops.

[5] Thomas Szyrkowiec, Michele Santuari “Automatic intent-based secure service creation through a multilayer SDN network orchestration” ,The International Journal of Computer and Telecommunications Networking, October 2016.

[6] Dr. Prithwish Kangsabanik, “OSS/BSS Impact with 5G Applications and Services”, IEEE 5G.

[7] Adnan Hamad, DIngli Yu, J B Gomm, Mahavir S Sangha, “Fault Detection and isolation for engine under closed-loop control”, Sept 2012, IEEE.