A Design of Dynamic Fault Management System for High Availability in NFV

Quang-Hiep Mai, YoungHan Kim*
Soongsil Univ

ABSTRACT

Network Function Virtualization enables flexibility and automation for network service by running those services on virtualized hardware. When a fault occurs in a VNF, an entire SFC will also go down. To enable High availability for the whole network, MANO shall include a fault management component that provides insights into the performance and I/O demands of VNFs, the underlying infrastructure resources metrics. Besides, understanding how those metrics affect the NFV environment results in appropriate reaction to the situation. In this paper, we propose a monitor architecture to ensure high availability in NFV and dynamic change the fault management policy.

I. INTRODUCTION

Network Function Virtualization (NFV) [1] is a crucial technology for telecom operators to deploy efficiency network services and improve customer's quality of experience by optimizing usage resources. NFV decouples the software implementation of network function from its underlying infrastructure by leveraging the virtualization technology and running them on commercial off-the-shelf hardware.

Along with its benefits, NFV brings several challenges to network operators [2], such as the guarantee of network performance, recovery when faults occur. When underlying infrastructure is lightly utilized, the performance of VNFs is significant instability [3]. Besides the network performance, monitoring and detecting failures of the VNFs is another major problem to provide efficiency insights of the system to support high availability and failure management.

In this paper, we present a novel approach to identify performance degradation in VNF service function chain (SFC). The key feature is that it can detect anomalies in multi-layer and classify them with similarity (e.q., Network services in the same service chain). Then, metrics data will be analyzed over time to infer potential abnormal behaviors, which cannot be detected by threshold-based classifiers and provide planners to adjust the system's observation policies for troubleshooting on demand.

II. PROPOSED ARCHITECTURE

A. Architecture

The dynamic fault detection system proposed in this paper includes Monitoring Driver, Alert Driver, and Analytic Driver. Primary functions of drivers are:

- Collect data from multiple sources, transform and transfer data to other Driver for further processing
- Handles metrics data and send alert to MANO when faults occur
- Analyze monitor data to adjust monitoring and alerting policy. Detect complex abnormal behaviors.

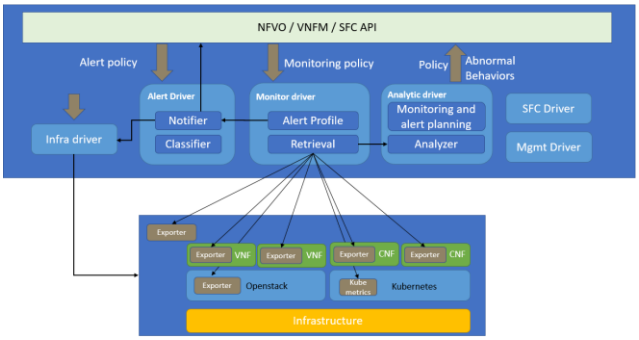

Figure 1. System Architecture

1. Monitor driver

Unlike existing network equipment that hardware and software are tightly coupled, fault detections in the NFV environment require multiple monitoring from underlying infrastructure to virtualization layer and the VNFs. In order to collect resource information, an exporter agent is installed. The agent fetches the host's resource information, including CPU, memory, and network utilization. In the case of the virtualization layer, it collects the APIs status and virtual resource statistic.

Pull-based monitoring is applied to provide dynamic monitoring. NFV orchestrator (NFVO) determines how long the monitor driver should scrape the metrics from agents. This policy can be enforced globally or specific for each service. When there is a likelihood of failure, NFVO can adjust the scrape interval to fetch data from agents more frequently for analytic.

An alert profile determines when to notify the Alert Driver when specific metrics reach the threshold that is described in the alert policy. The alert is generated with information on the metrics that exceed the threshold, resource type, and additional fields for alert Driver to classify them.

2. Alert Driver

Alert Driver receives multiple alerts from monitor driver and classifies them of similar nature. For example, when a network service fails in a service function chain, various alarms can be raised due to mutually dependent resources and NFVO would be overloaded with duplicate notifications.

Alert configuration profile is pre-defined by operator to specify how long it should wait to action for a group of alerts and what action it should take based on the severity of the fault. For example, an Operator can
Inject a policy to wait up to 30 seconds for the alerts come from the SFC A; after that, it sends notifications to the infra driver to reschedule the fault service or notify the MANO for further investigation.
An inhibition rule can be applied to mute an alert if another alert with higher priority has already fired. This mechanism prevents the system from having to deal with lower priority alerts when an emergency problem occurs. Assume a scenario when the infrastructure is unreachable, any SFC abnormal symptom is muted until correlation operations are taken in the infrastructure.

3. Analytic Driver

The Analytic Driver enables dynamic fault management. Running as a central data storage, Analytic Driver can process data from multiple streams and sources.
The most characteristic feature of analytic Driver is to collect metrics and analyze to derive any abnormal behaviors of the system that can not be detected by mere metrics. This capacity can be achieved by having multi-layer metrics from the infrastructure, virtualization layer, and virtual network service layer.
Analytic Driver also provides a planner component to adjust the monitoring and alert policy on demand which allows the tool to troubleshoot specific conditions. For example, If Analytics Driver root cause of a fault, it can reduce the scrape interval and define additional metric of targeted components. Those policies are sent to NFVO and then applied to Monitoring Driver.

B. DYNAMIC FAULT MANAGEMENT PROCEDURE

Figure 2 describes in detail the procedure of Dynamic fault management procedure. First, Metrics are pulled from VNF to Monitor-driver. Whenever the metrics reach a certain threshold, an alert will be fired to the Alert-Driver. After classifying the alert, correlation maintenance actions will be taken by the NFVO.
Second, the data flow also be routed to the Analytic-Driver to analyze any abnormal behaviors, and if any fault is detected, it will notify NFVO for maintenance. Finally, if Analytic-Driver can request directly to the NFVO to dynamically change the monitor and alert policies.

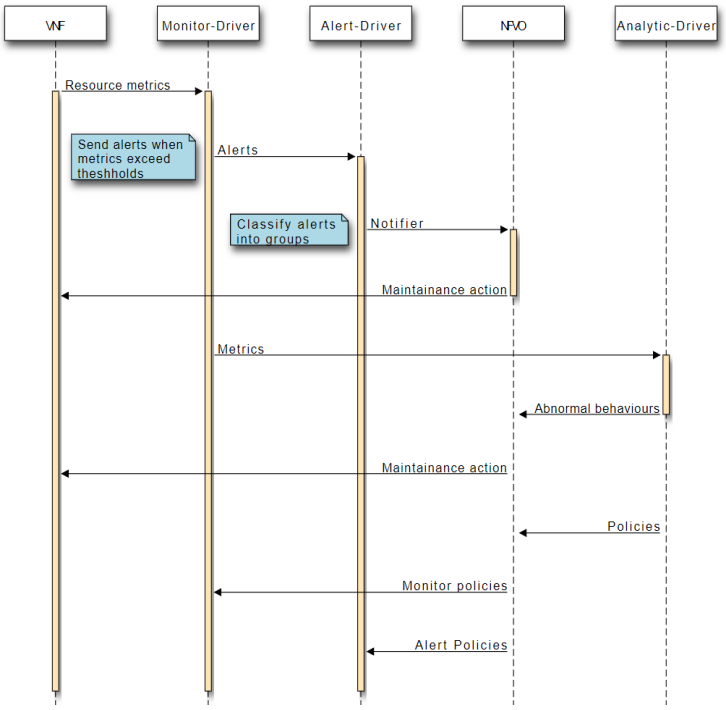

Figure 2. Dynamic fault management procedure

III. CONCLUSION

In this paper, we proposed a dynamic fault management system to support high availability of SFC. The Monitoring and Alert Driver was integrated to monitor multi-layer resources and provide correlation actions when fault occurs. Moreover, we achieved dynamic management by deriving abnormal behavior from analytic Driver and adjusting both monitor and alert policy.

ACKNOWLEDGMENT

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2020-2017-0-01633) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation).

REFERENCES

[1] ETSI GS NFV 002 v1.1.1, "Network Functions Virtualization: Architectural Framework", http://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf, accessed July 2017.
[2] Bo Han; Vijay Gopalakrishnan; Lusheng Ji; Seungjoon Lee, "Network function virtualization: Challenges and opportunities for innovations", IEEE Communications Magazine Volume: 53, Issue: 2, Feb. 2015
[3] G. Wang and T. S. E. Ng, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center", Proc. INFOCOM'10, pp. 1163-71, 2010-Mar.