NMS Application for SDN Networks

Hema Gopalakrishnan, Manohar SL, Dimple Jain, Deepthi V V and Gaurav Bhagwani.

- Ericsson
Agenda

• Introduction
• Existing ODL notification
• Standard SNMP MIBs
• Tools : Tsdr and Esper
• Graphical View of Fault and Topology
• Fault Correlation
• Future Enhancements
INTRODUCTION
Will SDN replace NMS ???

• No, NMS and SDN can work Coherently.

• Provisioning, Inventory Management, Fault Management, Traffic Engineering can be enhanced with SDN and NMS working together.

• Integrate SDN and NMS to manage the network more effectively.
FCAPS

• FCAPS is crucial for any networks, particularly early fault detection and analysis of the faults in the network.

• SDN Controller provides a centralized access of the underlying network infrastructure.

• Realizing NMS as an application based on SDN Controller enables use cases like:
  • Fault Detection and Correction.
  • Proactive fault identification.
  • Bulk Operations on the Nodes.
Design Proposal

• The proposal is to develop NMS as an NSF application.

• Fault Diagnosis NSF does the following:
  • Consume these notifications and store them in a database. This can be utilized for historical analysis.
  • Correlate these notifications, analyze them to identify the primary cause of failure.
  • The NSF also facilitates relaying these Notifications to the existing SNMP based NMS solutions.
NMS in ODL Architecture
Architecture: NMS

Existing NMS Apps

SNMP Traps

Fault Diagnosis NSF

Opendaylight YANG Notifications/DataChangeListeners

MD SAL

Openflow Asynchronous Message (ofp_port_mod)

OpenFlow Switches
ODL Notifications
ODL Notifications

• The NMS NSF will be using following ODL notifications for handling Fault Management:
  
  • Node UP/DOWN
  • Node Connector UP/DOWN
Standard SNMP MIBs

iso (1) . org (3) . dod (6) . internet (1) . snmpV2 (6) . snmpModules (3) . snmpMIB (1)

- -- snmpMIBObjects (1)
  + -- snmpStats (1)
  + -- snmpV1 (2)
  + -- snmpOR (3)
  + -- snmpTrap (4)
- -- snmpTraps (5)
  -- coldStart (1)
  -- warmStart (2)
  -- linkDown (3) object Details
  -- linkUp (4)
  -- authenticationFailure (5)
+ -- snmpSet (6)
# Standard SNMP Trap - Link Down

<table>
<thead>
<tr>
<th>Object</th>
<th>linkDown</th>
</tr>
</thead>
<tbody>
<tr>
<td>OID</td>
<td>1.3.4.1.6.3.1.1.5.3</td>
</tr>
<tr>
<td>Status</td>
<td>current</td>
</tr>
<tr>
<td>MIB</td>
<td>IF-MIB</td>
</tr>
<tr>
<td>Trap Components</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ifIndex</td>
</tr>
<tr>
<td></td>
<td>ifAdminStatus</td>
</tr>
<tr>
<td></td>
<td>fOperStatus</td>
</tr>
<tr>
<td>Description</td>
<td>A LinkDown trap signifies that the SNMP entity, acting in an agent role, has detected that the ifOperStatus object for one of its communication links is about to enter the down state from some other state. This present state is indicated by ifOperStatus value.</td>
</tr>
</tbody>
</table>

*Ericsson*

*OpenDaylight For India 2015*
Tools: Tsdr and Esper
Notification Handling

• NMS Application will handle ODL notifications using:
  • tsdr (Time Series Data Repository)
  • Esper (Complex Event Processing (CEP) Engine)

• Useful for use-cases like highlighting historical events.
Time Series Data Repository (TSDR)

- Already proposed in OpenDaylight (ODL).
- Contains time series data repository + time series data services.
- Services collects, stores, queries and maintains time series data.
Time Series Data Repository (TSDR)

- Statistics data are made available on MD-SAL messaging bus.

- TSDR subscribes to this data store them as time series data into TSDR.

- Northbound APIs retrieve this time series data from the TSDR.
TSDR Architecture

ODL Controller

Northbound APIs (RESTCONF)

TS NB APIs

TS Data Services

MD-SAL

Data Collection Service

Poll/Query

Poll/Query

Data Storage Service

Persist

Data Query Service

Query

Data Purging & Aggregation Service

Aggregate/Purge

ODL Data Store

HBase YSDR Plugin

ODL Time Series Data Repository

Network

ODL-Controlled Devices

Southbound Protocol Plugin

Poll/Query

Southbound “Push” Plugin

Push

Poll/Query

HBase + Hadoop clustering system

Standby Name Node (Standby Master Server)

Data Node (Region Server)

Data Node (Region Server)

Data Node (Region Server)

ODL Forum India 2015
• Complex Event Processing (CEP) includes event data analysis.

• It emphasis more on patterns of events.

• Esper is one of the CEP Engine having both Open Source GPL and EsperTech commercial.

• It works with any Java Process and is OSGI enabled.
ESPER IN ODL

OPEN DAYLIGHT
“HELIUM”

LEGEND
AAA: Authentication, Authorization & Accounting
AuthN: Authentication
BGPP: Border Gateway Protocol
COA: Common Open Policy Service
DLUX: OpenDaylight User Experience
DDB: Distributed Data Base
DOCSIS: Data Over Cable Service Interface Specification
FPM: Forwarding Rules Manager
GBP: Group Based Policy
LISP: Locator/Identifier Separation Protocol
OVsDB: Open vSwitch DataBase Protocol
PCEP: Path Computation Element Communication Protocol
PCAM: Packet Cable MultiMedia
PlugInOOC: Plugin To OpenContrail
SDN: SDN Interface (Cross-Controller Federation)
SFC: Service Function Chaining
SNBI: Secure Network Bootstrapping Infrastructure
SNMP: Simple Network Management Protocol
TTP: Type Type Patterns
VTN: Virtual Tenant Network

Base Network Service Functions
- Topology Manager
- Stats Manager
- Switch Manager
- FRM
- Host Tracker

Service Abstraction Layer (SAL)
- Plugin Manager, Capability Abstractions, Flow Programming, Inventory, etc.

Esper ENGINE

OpenDaylight APIs (REST)
- AAA - AuthN Filter

Controller Platform

Network Applications Orchestrations & Services
- Network Applications
- Orchestration Services

SDN Wrapper
DDS Protection

OpenStack Neutron

AAA - AuthN Filter

OpenFlow Enabled Devices
- Open vSwitches

Additional Virtual & Physical Devices
- Additional Virtual & Physical Devices

Southbound Interfaces & Protocol Plugins
- Southbound Interfaces
- Protocol Plugins

OpenFlow
- OpenFlow 1.0, 1.3
- TTC

OVsDB
- OVSDB

NETCONF
- NETCONF

PCMM/COPS
- PCMM/COPS

SNBI
- SNBI

LISP
- LISP

GBP
- GBP

PCEP
- PCEP

SNMP
- SNMP

PluginOOC
- PluginOOC

For more information, visit OpenDaylight.org
API Overview

• EPService Provider
  • Acts as Engine

• EPStatement
  • Queries written to EQL

• UpdateListener
  • Receives updated data as soon as it’s processed for the statement.

• Subscriber
  • Provides method by name update to receive insert stream events row by row.
NMS Application: Esper/ Tsdr

• The proposed NMS application can make use of either Esper or Tsdr.

• With Tsdr, the NMS application satisfies various fault based use cases.

• With Esper, the NMS application satisfies various analytical use cases.
Graphical View

• The graphical view will enable the User to understand the Network Topology and also debug/correct problems on real time basis.

• Faster corrective actions can be taken due to integration with SDN Controller.

• The various Views which can be realized are
  • Topology View
  • Device View
  • Alarms View
  • Health Report View
Topology View (1/2)
Topology View (2/2)
### Alarms View

<table>
<thead>
<tr>
<th>Message</th>
<th>Source</th>
<th>Severity</th>
<th>Last Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Down: No response from device</td>
<td>sw6</td>
<td>Critical</td>
<td>25 Mar 2015 04:41:53 PM IST</td>
</tr>
<tr>
<td>Link down: eth05</td>
<td>Ixiacom-009</td>
<td>Attention</td>
<td>27 Mar 2015 06:23:12 PM IST</td>
</tr>
<tr>
<td>Threshold violated for 10.60.100.175</td>
<td>Linux80865</td>
<td>Trouble</td>
<td>29 Mar 2015 08:45:42 PM IST</td>
</tr>
<tr>
<td>Threshold violated for 10.50.128.172</td>
<td>RedhatLinux109337</td>
<td>Trouble</td>
<td>1 Apr 2015 10:23:12 PM IST</td>
</tr>
<tr>
<td>Service down: Port eth01 down</td>
<td>Ixiacom-001</td>
<td>Critical</td>
<td>3 Apr 2015 12:09:15 PM IST</td>
</tr>
<tr>
<td>Device Down: No response from device</td>
<td>Ixiacom-004</td>
<td>Critical</td>
<td>3 Apr 2015 12:49:19 PM IST</td>
</tr>
</tbody>
</table>
Fault Correlation
Fault Correlation

• Faults are often the result of underlying problems:
  • hardware or software failures
  • performance bottlenecks
  • configuration inconsistencies
  • intrusion attempts

• Often a single problem event in one resource can cause many symptom events in related resources.

• It is very important to be able to correlate the events to identify and localize underlying problems.
Fault Correlation Design

- Design of intuitive and smart Fault Correlation engine includes:

  - The diagnostic knowledge.

  - Algorithms that analyze this knowledge in the context of the current network state to detect root cause of failures.

  - Both the knowledge model and diagnostic processing must scale to large and complex networks.
Fault Correlation in SDN Context

• Correlating the multiple ODL Notifications to specific fault in the Network.

• Based on the pattern of problems, corrective actions can also be suggested/performed through the SDN Controller.

• Fault Correlation can be realized to the next level of self-correcting process in the context of SDN Networks.
Fault Correlation Evolution

Management Plane

Control Plane

Data Plane

Management Plane

Control Plane

Data Plane
Future Enhancements
Future Enhancements

• To realize the other FCAPS features, for greater degree of NMS realization.

• NMS Analytics.

• Auto Corrective capability.
To Summarize

• Realizing NMS NSF on SDN Networks.

• Integration with ODL based Notifications.

• Tsdr/Esper for Notification management and storage.

• Graphical representation.

• Fault Correlation and Correction.
Thank You