Service Assurance for Intent-based Networking Architecture & YANG Modules for Service Assurance

draft-claise-opsawg-service-assurance-architecture-03
B. Claise (Cisco), J. Quilbeuf (Cisco), Y. El Fathi (Orange), D. Lopez (Telefonica), D. Voyer (Bell Canada)

draft-claise-opsawg-service-assurance-yang-05
B. Claise (Cisco), J. Quilbeuf (Cisco), P. Lucente (NTT), P. Fasano (Telecom Italia Mobile)

IETF 108, Virtual
Issue & Proposal

• Issues:
  – When a service degrades, where is the fault? what are the symptoms? what is the root cause?
  – When a network component fails, which services are impacted?

• Service Assurance for Intent-based Networking Architecture proposal:
  – Decompose the problem into smaller components (=subservices)
  – The assurance graph links those subservices to map the service “intent”
  – The subservices are assured independently
  – Infer a service health score

• This complements the end-to-end synthetic testing
Architecture
Flexible Architecture

Could be a single box

Agent could be in or off routers
Open Architecture with YANG Models

[Diagram showing a hierarchical structure with YANG models.]
Open Architecture with YANG Models
Changes since Last IETF

• Stressed the generic architecture:
  applicable to wireline, wireless, 5G, VIM, etc.

• Timing: NTP is required all over the place

• Introduced the symptoms history start

"Date and time at which the symptoms history starts for this subservice instance, either because the subservice instance started at that date and time or because the symptoms before that were removed due to a garbage collection process."
Assurance Graph PoC

ECMP fair load on sain-pe-3
GigabitEthernet0/0/0/0
GigabitEthernet0/0/0/1
Value: 0.5
Expression tree

List of impacted services:
- sain-tunnel CocaCola
- l2vpn-p2p HSBC
- sain-tunnel-ipv6 RedBull

Symptoms/Root causes:
- Output traffic on interface GigabitEthernet0/0/0/0 in ECMP bundle with(GigabitEthernet0/0/0/1) is not fairly balanced
- Output traffic on interface GigabitEthernet0/0/0/1 in ECMP bundle with(GigabitEthernet0/0/0/0) is not fairly balanced
Changes since Last IETF Hackathon: Lessons Learned

• Assurance graph version & last change now compulsory
  – while per sub-service last change remains optional
• Explain what a change means
• Rename symptoms “label” to “description”
  – Not to be confused with the subservice label
Service Assurance for Intent-based Networking Architecture & YANG Modules for Service Assurance

HACKATHON
Korian Edeline (Liège University)

IETF 108, Virtual
An open-source SAIN agent

dxtop

Console App (GUI)

dxweb

Web App

Shared Memory

gNMI Exporter

rules

Apply rules on normalized data & compute health scores

metrics

Normalize input data. Discover subservices and dependencies

inputs

Discover data sources and gather data

e.g., "Receive Errors Peak", lmin(delta(rx_error))>100

e.g., rx_bytes, Net/Rate/Rx,/if/rx-bytes => rx_bytes

XPCOM /proc netlink gNMI shmem ...

https://github.com/ekorian/dxagent
An open-source SAIN agent

https://github.com/ekorian/dxagent
Rule Engine: Highlighting symptoms

- Subservice expertise for anomaly-highlighting rules
- Variables (metrics), basic operators and more (temporality, selection, has_changed, ...)

<table>
<thead>
<tr>
<th>Event Description</th>
<th>Path</th>
<th>Color</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Flapping</td>
<td>/node/bm/net/if</td>
<td>Red</td>
<td>1min(dynamicity(changes_count)) &gt;= 6</td>
</tr>
<tr>
<td>Low Buffer Availability</td>
<td>/node/kb/mem</td>
<td>Orange</td>
<td>(buffer_free/buffer_total) &lt; 0.1</td>
</tr>
<tr>
<td>DPDK Buffer Alloc Errors</td>
<td>/node/kb/net/if</td>
<td>Orange</td>
<td>dynamicity(dpdk_alloc_errors) &gt; 0</td>
</tr>
<tr>
<td>Sensor reached critical temperature</td>
<td>/node/bm/sensors/sensor,</td>
<td>Red</td>
<td>input_temp &gt;= critical_temp</td>
</tr>
<tr>
<td>Non-standard Ethernet MTU</td>
<td>/node/bm/net/if</td>
<td>Red</td>
<td>(mtu!=1500) and (type==&quot;ether&quot;)</td>
</tr>
</tbody>
</table>
An open-source SAIN agent

- Client service monitoring by concatenating assurance trees
- From failing component, find impacted subservices
- Monitor multiple subservices or components
SAIN Hackaton: Open Architecture with YANG Models
SAIN Hackaton : Open Architecture with YANG Models

- **Service**
  - **Configuration**
  - **Orchestrator**

- **Network**
  - **Service**
  - **Instance**
  - **Configuration**

- **SAIN**
  - **Orchestrator**

- **SAIN**
  - **Collector**
    - **telegraf**
    - **pmacct**

- **dxagent**

- **Metric Collection**
  - **Metric Collection**

- **Influxdb**
- **Grafana**

- **Monitored Entities**
  - **VPP/virtual machine/container**
SAIN Hackaton: Open Architecture with YANG Model (Example)

- Monitor a VPP-in-VM instance
SAIN Hackaton: Open Architecture with YANG Model (Example)

- Health score decreases
$ ./dxweb -t 1.2.3.4:50051

$ ssh "dxtop"
Interoperability

• Dxagent exporting to the Cisco SAIN collector
Next Steps

- More complete rule engine
- Add end-to-end probing as input
- Multi-node architecture (path assurance)
- gNMI support in pmacct (Kannan Jayaraman)
- White paper on specific use case
- More input, more rules

https://github.com/ekorian/dxagent
Questions & Feedback on both Presentations

• Going in the right direction?
• Time to provide more feedback?

• If yes, please consider as WG adoption.