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

draft-claise-opsawg-service-assurance-architecture-04
B. Claise (Huawei), J. Quilbeuf (Independent), D. Lopez (Telefonica), D. Voyer (Bell Canada), T. Arumugam (Cisco)

draft-claise-opsawg-service-assurance-yang-06
B. Claise (Huawei), J. Quilbeuf (Independent), P. Lucente (NTT), P. Fasano (Telecom Italia Mobile), T. Arumugan (Cisco)

IETF 110, OPSAWG
Status

• Presented a few times already
• Last presentation at IETF 108
• Drafts were simply « refreshed » this time, almost no updates
• Presentation goals:
  – Quick refresh
  – Evaluate the interest to work on it as WG document
  – Introduce the opensource effort (Liège University)
Issues & 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) monitoring
Flexible Architecture

Could be a single box

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

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 GigabitEthernet0/0/1 in ECMP bundle with(GigabitEthernet0/0/0/0) is not fairly balanced
Open and Flexible Architecture

• Open architecture for multi-vendor support
  – How? With a YANG module:
    • Can augment the YANG module
    • Even for vendor-specific subservices

• Open architecture for multi-domains (wireline, wireless, 5G, VIM, etc.)
  – How? By linking domain-specific assurance graph
Feedback

• Valid problem to solve industry-wide?
• At the IETF?
• Going in the right direction?

• If yes, you might consider it as WG adoption.
Diagnostic Agent

Korian Edeline, Justin Iurman, Thomas Carlisi, Benoit Donnet
Objectives

• Main goal?
  – evaluate the SAIN architecture

• How?
  – opensource tools
  – test our implementation w.r.t. use cases
    • basic VPP health
    • VPP in VM monitoring
Global Architecture

- Service
  - Configuration
  - Orchestrator

- Network
  - Service
  - Instance
  - Configuration

- SAIN
  - Orchestrator

- SAIN
  - Collector

- Health Status
  - (Score + Symptoms)
  - Streamed
  - via Telemetry

- Metric Collection

- Monitored Entities

- VPP

- DxAgent

- gRPC
DxAgent

• An Open-Source SAIN Agent

```
metrics.csv
```

```
input.csv
```

```
rules.csv
```

**Input**
- XPCOM
- /proc
- IOAM
- netlink

**Metrics**
- e.g., rx_bytes, Net/Rate/Rx, /if/rx_bytes ⇒ rx_bytes

**Rules**
- e.g., "Receive errors peak", 1min(Δ(rx_error)) > 100

**Shared Memory**
- dxtop

**gNMI exporter**
- dxweb
DxTop

- Console app displaying data collected by DxAgent
DxWeb
• Web interface displaying dependency graph, health metrics, and symptoms from DxAgent
Rules

- Highlighting symptoms
  - subservice expertise for anomaly-highlighting rules
  - variables (metrics), basic operators and more
    (temporality, selection, has_changed, ...)

```plaintext
"Interface Flapping", /node/bm/net/if, Red, 1min(dynamicity(changes_count))>=6
"Low Buffer Availability", /node/kb/mem, Orange, (buffer_free/buffer_total)<0.1
"DPDK Buffer Alloc Errors", /node/kb/net/if, Orange, dynamicity(dpdk_alloc_errors)>0
"Sensor reached critical temperature", /node/bm/sensors/sensor, Red, input_temp>=critical_temp
"Non-standard Ethernet MTU", /node/bm/net/if, Red, (mtu!=1500) and (type="ether")
```
Use Case

• Monitor a VPP-in-VM instance
On-going Work

• IOAM agent for traffic telemetry
  – queue depth
  – buffer occupancy
• Cross-Layer Telemetry (CLT)
  – IOAM + OpenTelemetry
• Python wrapper for OWAMP C implementation
• Advanced application observation and service assurance
  1. a customer expresses a specific SLO between herself and an app running at the edge
  2. DxAgent+IOAM+CLT in charge of discovering the most appropriate app instance and the most appropriate path
  3. DxAgent+IOAM+CLT connects the customer to the selected app instance, through the selected path
  4. DxAgent+IOAM+CLT observes whether the app instance/path is still the most appropriate
Conclusion

• Working prototype
  – still lots of work to do (see previous slides)

• See
  – our work on telemetry
    • https://people.montefiore.uliege.be/bdonnet/telemetry/
  – DxAgent implementation
    • https://github.com/ekorian/dxagent
  – IOAM Agent implementation
    • https://github.com/IurmanJ/ioam-agent
  – CLT
    • https://github.com/IurmanJ/cross-layer-telemetry
Backup Slides
module: ietf-service-assurance

   +--ro assurance-graph-version?       yang:counter32
   +--ro assurance-graph-last-change?   yang:date-and-time
   +--rw subservices
       +--rw subservice* [type id]
           +--rw type                                identityref
           +--rw id                                  string
           +--ro last-change?                        yang:date-and-time
           +--ro label?                              string
           +--ro under-maintenance?                  boolean
           +--rw maintenance-contact                 string
           +--rw (parameter)?
               |   +--:(service-instance-parameter)
               |       +--rw service-instance-parameter
               |           +--rw service?         string
               |           +--rw instance-name?   string
               +--ro health-score?                       uint8
           +--rw symptoms
               |   +--ro symptom* [start-date-time id]
               |       +--ro id                     string
               |       +--ro health-score-weight?   uint8
               |       +--ro label?                 string
               |       +--ro start-date-time       yang:date-and-time
               |       +--ro stop-date-time?       yang:date-and-time
           +--rw dependencies
               +--rw dependency* [type id]
                   +--rw type               -> /subservices/subservice/type
                   +--rw id                 -> /subservices/subservice[type=current()]/../type/id
                   +--rw dependency-type?   identityref
Assurance Tree API

module: ietf-service-assurance

+-rw subservices
  +-rw subservice* [type id]
    +-rw type    identityref
    +-rw id      string

...

+-rw dependencies
  +-rw dependency* [type id]
    +-rw type    -> /subservices/subservice/type
    +-rw id      -> /subservices/subservice[type=current()]/../type]/id
    +-rw dependency-type? identityref
Health Score and Symptoms API

module: ietf-service-assurance

  +--ro assurance-graph-version?       yang:counter32
  +--ro assurance-graph-last-change?   yang:date-and-time
  +--rw subservices
    +--rw subservice* [type id]
      ....
    +--ro health-score?                 uint8
    +--rw symptoms
      |  +--ro symptom* [start-date-time id]
      |     +--ro id                     string
      |     +--ro health-score-weight?   uint8
      |     +--ro label?                 string
      |     +--ro start-date-time        yang:date-and-time
      |     +--ro stop-date-time?        yang:date-and-time

Health score and Symptoms per subservice
Subservice Parameters API

module: ietf-service-assurance

+-ro assurance-graph-version?   yang:counter32
+-ro assurance-graph-last-change? yang:date-and-time
+-rw subservices
  +--rw subservice* [type id]
    +--rw type        identityref
    +--rw id          string
    +--ro last-change? yang:date-and-time
    +--ro label?      string
    +--rw under-maintenance? boolean
    +--rw maintenance-contact string
  +--rw (parameter)?
    |   +--:(service-instance-parameter)
    |   +--rw service-instance-parameter
    |   +--rw service? string
    |   +--rw instance-name? string

NEW
New Subservices, with different Sets of Parameters

+--rw (parameter)?
  | +--:(service-instance-parameter)
  | | +--rw service-instance-parameter
  | |   +--rw service?   string
  | |   +--rw instance-name?   string
  | +--:(service-assurance-device:device-idty)
  | | +--rw service-assurance-device:device-idty
  | |   +--rw service-assurance-device:device?   string
  | +--:(service-assurance-interface:device)
  | | +--rw service-assurance-interface:device
  | |   +--rw service-assurance-interface:device?   string
  | +--:(service-assurance-interface:interface)
  | | +--rw service-assurance-interface:interface
  | |   +--rw service-assurance-interface:interface?   string
  | +--:(example-service-assurance-device-acme:acme-device-idty)
  | | +--rw example-service-assurance-device-acme:acme-device-idty
  | |   +--rw example-service-assurance-device-acme:device?   string
  | |   +--rw example-service-assurance-device-acme:acme-specific-parameter?   string

New subservice type
New vendor-specific subservice type