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

draft-claise-opsawg-service-assurance-architecture-01
draft-claise-opsawg-service-assurance-yang-02

Benoit Claise and Jean Quilbeuf, Cisco
IETF 106, Singapore
Issues

• A service being configured doesn’t mean it’s operating correctly
• Too much data in telemetry: needle in a haystack
• 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?
• How to solve closed loop automation, as a first step?
Proposal

• The end goal: self-healing/driving/whatever networks or intent-based networking
• Intent: the top down approach, declarative way is a nice concept
  – Mainly working for greenfield deployments
  – We have to solve this differently
• Service Assurance for Intent-based Networking Architecture proposal
  – Decompose the problem into smaller components
  – Those components are assured independently
  – Complement the end-to-end synthetic tests
Assurance Graph

Service Instance

Subservice Instances

- Tunnel
  - Service Instance
  - Tunnel
    - Service Instance
      - Peer1
        - Tunnel
          - Interface
        - Peer1
          - Physical
            - Interface
        - Peer1
          - Device
      - Peer2
        - Tunnel
          - Interface
        - Peer2
          - Physical
            - Interface
        - Peer2
          - Device
  - IP Connectivity
    - IS-IS Routing Protocol
Score & Symptoms

An inferred health score + a series of symptoms

- Tunnel
  - Service Instance
    - Peer1
      - Tunnel Interface
        - Peer1
          - Physical Interface
            - Peer1
              - Device
        - Peer2
          - Physical Interface
            - Peer2
              - Device
    - Peer2
      - Tunnel Interface
        - Peer2
          - Device
(Impacting or Informational) Dependencies

- Tunnel
  - Service Instance
    - Peer1
      - Tunnel Interface
        - Peer1
          - Physical Interface
            - Peer1
              - Device
    - Peer2
      - Tunnel Interface
        - Peer2
          - Physical Interface
            - Peer2
              - Device
    - IP Connectivity
      - IS-IS Routing Protocol
(Impacting or Informational) Dependencies

```
+---------------------+
| Tunnel              |
| Service Instance    |
+---------------------+

```

**Informational Dependency**

```
+-----+----------------------------+
| Peer1 | Tunnel | Interface                  |
+-----+----------------------------+
| Peer1 | Physical | Interface                |
+-----+----------------------------+
| Peer1 | Device           |
+-----+----------------------------+

```

```
+-----+----------------------------+
| Peer2 | Tunnel | Interface                  |
+-----+----------------------------+
| Peer2 | Physical | Interface                |
+-----+----------------------------+
| Peer2 | Device           |
+-----+----------------------------+

```

```
+-----+----------------------------+
| IP   | Connectivity               |
+-----+----------------------------+
| IS-IS | Routing | Protocol                 |
+-----+----------------------------+

```

ECMP
So far, we know...

• 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
Architecture

Service
 Configuration
 Orchestrate

Network
 Service
 Instance
 Configuration

SAIN
 Orchestrate

Configuration
 (assurance graph)

SAIN
 agent

^ ^ ^

Metric Collection

^ ^ ^

Feedback
 Loop

SAIN
 Collector

Health Status
 (Score + Symptoms)
 Streamed
 via Telemetry

Monitored Entities
Flexible Architecture

Could be a single box

Agent could be in or off routers
Open Architecture

• Why? multi-vendor
• How? With a YANG module
  – Can augment the YANG module
  – Even for vendor-specific subservices
Open Architecture with YANG Models
Open Architecture with YANG Models
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 (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-timeyang: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
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 (parameter)?
        +--:(service-instance-parameter)
        |    +--rw service-instance-parameter
        |    +--rw service?    string
        |    +--rw instance-name? string
New Subservices

++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
  | ++--:(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
Feedback/Flame/Tomatoes

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