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

draft-ietf-opsawg-service-assurance-architecture-02
B. Claise (Huawei), J. Quilbeuf (Huawei), D. Lopez (Telefonica), D. Voyer (Bell Canada), T. Arumugam (Cisco)

draft-ietf-opsawg-service-assurance-yang-01
B. Claise (Huawei), J. Quilbeuf (Huawei), P. Lucente(NTT), P. Fasano (Telecom Italia Mobile), T. Arumugan (Cisco)

IETF 112, OPSAWG
One Slide Summary

• 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
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
Architecture Draft: Update in v 02

- Clearer DAG: adding arrows
Architecture Draft: Update in v 02

Circular Dependencies Covered

• Why it could happen?
  – Engineer A “Interface depends on Link”
  – Engineer B “Link depends on Interface”
  – Merge: circuit: “Link -> Interface -> Link”

• Circular dependencies should be avoided
• Orchestrator should detect them
• Graph transformation proposal

• New section 3.1.1 added
Removing circular dependencies:

- Strongly Connected Component

Before Transformation:
- a
- b
- c -> d
- ^
- e
- v
- g
- h

After Transformation:
- a
- b
- c
d
e
top
v
f
g
h

Same set of dependencies as “a” on the left

Same nodes without inner dependencies
YANG Module Draft

• This is now stable: no more open issues
• Unchanged since last time
Open Issues

• Refer to the Intent-based Networking NMRG documents (Intent Assurance, Service Intent: synonym for custom service model see [I-D.irtf-nmrg-ibn-concepts-definitions] and [I-D.irtf-nmrg-ibn-intent-classification])
  – Those drafts in IRSG review right now
Situation and Next Steps

• No hackathon, but we were thinking of:
  – assurance graph of a service to compute the status of that service based on the status of the involved subservices

• Next step: Actionable symptoms

• Not sure whether those would be part of the same drafts, a different one, or not standardized

• By the next IETF meeting, if no other issues are discovered, it should be time for last-call.
Service Assurance for Intent-based Networking Architecture & YANG Modules for Service Assurance

draft-ietf-opsawg-service-assurance-architecture-02
B. Claise (Huawei), J. Quilbeuf (Huawei), D. Lopez (Telefonica), D. Voyer (Bell Canada), T. Arumugam (Cisco)

draft-ietf-opsawg-service-assurance-yang-01
B. Claise (Huawei), J. Quilbeuf (Huawei), P. Lucente (NTT), P. Fasano (Telecom Italia Mobile), T. Arumugan (Cisco)

IETF 112, OPSAWG

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
      +--rw 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
        +--ro symptoms-history-start? yang:date-and-time
    +--rw symptoms
      | +--ro symptom* [start-date-time id]
      |    +--ro id string
      |    +--ro health-score-weight? uint8
      |    +--ro description? 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

Subservice Parameters

Health score and Symptoms per subservice

Dependency relationship
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

      +--rw (parameter)?
      |  +--:(service-instance-parameter)
      |   |  +--rw service-instance-parameter
      |   |   |  +--rw service string
      |   |   |  +--rw instance-name string
      |   +--:(service-assurance-interface:parameters)
      |    |  +--rw service-assurance-interface:parameters
      |    |    |  +--rw service-assurance-interface:device string
      |    |    |  +--rw service-assurance-interface:interface string
      |    +--:(service-assurance-device:parameters)
      |     |  +--rw service-assurance-device:parameters
      |     |    |  +--rw service-assurance-device:device string
      ..
    +--rw dependencies
      +--rw dependency* [type id]
        +--rw type -> /subservices/subservice/type
        +--rw id -> /subservices/subservice[type=current()/./.type]/id
        +--rw dependency-type? identityref

Two different subservices (device, and interface)
• “mandatory true” now added for the parameters

```yang
choice parameter {
  description "Specify the required parameters per subservice type.";
  container service-instance-parameter {
    when "derived-from-or-self(../type, 'service-assurance:service-instance-idty')";
    description "Specify the parameters of a service instance.";
    leaf service {
      type string;
      mandatory true;
      description "Name of the service.";
    }
    leaf instance-name {
      type string;
      mandatory true;
      description "Name of the instance for that service.";
    }
  }
}
```
YANG Module Draft: Update in v 01

- Added a “parameters” container for interface, to align the structure with the other subservice
  - “parameters” instead of the identity-name

```yang
augment "/service-assurance:subservices/service-assurance:subservice/service-assurance:parameter" {

description
  "Specify the required parameters for the interface-idty subservice type";
container parameters {
  when "derived-from-or-self(.//service-assurance:type, 'interface-idty')";
  description
    "Required parameters for the interface-idty subservice type";
leaf device {
  type string;
  mandatory true;
  description
    "Device supporting the interface."
};
leaf interface {
  type string;
  mandatory true;
  description
    "Name of the interface."
};
}
```
YANG Module Draft: Update in v 01

- Complete set of YANG modules for the architecture draft example
- Added the “IP connectivity” and “IS-IS” subservice
YANG Module Draft: Update in v 01

• New “Appendix A. Example of YANG instances”
  – Validated with yangson

• New “Appendix B. YANG Library for Service Assurance”
YANG Module Draft: Update in v 01

• New section on “guidelines for subservice extension”
  – Module name
  – Module namespace
  – Module prefix
  – Specific identity
  – Parameters
Open Issue, Feedback, and Questions

• Refer to the Intent-based Networking NMRG documents (Intent Assurance, Service Intent: synonym for custom service model see [I-D.irtf-nmrg-ibn-concepts-definitions] and [I-D.irtf-nmrg-ibn-intent-classification]).

• Thanks to those who provided/will provide feedback.
Flexible Architecture

Could be a single box

Agent could be in or off routers
Global Architecture

- Service
  - Configuration
  - Orchestrator

- Network
  - Service
  - Instance
  - Configuration

- Feedback Loop

- SAIN
  - Orchestrator
  - Collector

- Configuration
  - (assurance graph)

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

- Metric Collection

- Monitored Entities

- DxAgent

- gRPC

- VPP
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