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



Open Architecture with YANG Models





Assurance Graph PoC



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, <u>Benoit Donnet</u>

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



DxAgent

An Open-Source SAIN Agent



DxTop

• Console app displaying data collected by DxAgent

CPU Memory	Proces	ses N	etworki	ng Vi	rtual M	achines	1	/PP	Health
vm-count: 0 kb-count:0 symptoms-count: 3									
No ARP Entry For Default Gateway: /node[name=sam]/bm/net/if[name=ens6f6]									
Metrics									
/node[name=sam]/bm/cpus/cpu health:100									
сри0-сри7									
idle time	100 %	99 %	94 %	99 %	100 %	100 %	100	э %	100 %
system_time	0 %	0%	1 %	1 %	0%	0%	0 9	6	0 %
user_time	0 %	1 %	5 %	0 %	0 %	0 %	0 9	6	0 %
guest_time	0 %	0%	0%	0%	0%	0%	0 9	6	0%
idle time	81 %	100 %	100 %	100 %	100 %	100 %	100	9 %	100 %
system time	0 %	0 %	0 %	0 %	0 %	0 %	0 9	6	0 %
user time	19 %	0 %	0 %	0 %	0 %	0 %	0 9	6	0 %
guest_time	st_time 0% 0% 0% 0% 0% 0%					0 9	6	0 %	
/node[name=sam]/bm/net/if health:79									
			eno	1 healt	h:100				
rx packets	x packets 119158855 908						i -		
rx bytes	rx bytes 11908174732 MB 700					700	36		
rx error	rx error			0				0	
rx drop			0	0				0	
tx packets			440	44076236				996	
tx bytes			192	19246201603 MB				263184	
tx error			0	0				0	
tx_drop 0			0	0				0	
up_count	p_count 1				Θ				
down_count		1			0				
changes_count			2	2			0		
state			up	up 0			0		
wireless			0					0	
mtu			150	0 B				0	
numa			0	0 0					
cpulist			0-1	0-15 0					
tx_queue			100	1000 0					
gw_in_arp			1	0					
type			ether 0						
dns_server	ns_server 127.0.0.53 0								
dhcp_server	dhcp_server 0								
driver				e1000e				0	

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, ...)

"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

	Main		😂 😗 UTC Local 📑 🎯 Variables 🕈 Annotations 🎜 10s 🔻 🕒 Past 15m 💌 🗐
0	Interfaces Health	Node health	Disks Health
	100%		j 100%
⊞	labort.nean		ever tikker district tikker
	0% 15:35 15:40 15:45	0 0	under being
3		Tesi2 05	15:35 15:40 15:45
\sim			
ß			15:40 15:45 mean_node/bm/disks/disk[/health/node/bm/disks/disk/name=systemd-1]



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
 - <u>https://people.montefiore.uliege.be/bdonnet/telemetry/</u>
 - DxAgent implementation
 - <u>https://github.com/ekorian/dxagent</u>
 - IOAM Agent implementation
 - <u>https://github.com/lurmanJ/ioam-agent</u>
 - CLT
 - <u>https://github.com/lurmanJ/cross-layer-telemetry</u>

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 string +--rw id +--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 +--rw symptoms +--ro symptom* [start-date-time id] +--ro id string +--ro health-score-weight? uint8 +--ro label? string yang:date-and-time +--ro start-date-time +--ro stop-date-time? yang:date-and-time +--rw dependencies +--rw dependency* [type id] -> /subservices/subservice/type +--rw type -> /subservices/subservice[type=current()/../type]/id +--rw id +--rw dependency-type? identityref

Assurance Tree API

```
+--rw subservices
```

```
+--rw subservice* [type id]
```

+--rw type

+--rw id

- identityref

```
•••
```

```
+--rw dependencies
```

```
+--rw dependency* [type id]
```

+--rw type -> /subservices/subservice/type

string

- +--rw id -> /subservices/subservice[type=current()/../type]/id
- +--rw dependency-type? identityref

```
Dependency
relationship
```

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
                                   yang:date-and-time
        +--ro stop-date-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] identityref +--rw type +--rw id string +--ro last-change? yang:date-and-time +--ro label? string boolean +--rw under-maintenance? **NEW** +--rw maintenance-contact string +--rw (parameter)? +--:(service-instance-parameter) Subservice +--rw service-instance-parameter **Parameters** +--rw service? string +--rw instance-name? string

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? string
- +--:(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 vendor-specific subservice type