

A YANG Data model for Event Management

draft-wwx-netmod-event-yang-01

Michael Wang (wangzitao@huawei.com)

Qin Wu (bill.wu@huawei.com)

Chongfeng Xie (xiechf@ctbri.com)

Recap

- This draft defines a YANG data model for the ECA based network policy management
 - Provides the ability for the network management function (i.e. a controller, an orchestrator, or a network element) to control the configuration and monitor state change on the network element;
 - Take simple and instant action when a trigger condition on the system state is met.
- This draft was firstly discussed in last NETMOD meeting.
- The WG suggest to coordinate with Smart-filter draft.

Updates since last IETF 103

- Polish the texts
 - Clarify the difference between boolean trigger condition and threshold trigger condition.
 - Synchronize the terms with RFC8328 (a policy-based management framework)
- Split the Event Management model into “ietf-event-trigger.yang” module and “ietf-event.yang” module.
 - make the event-trigger reusable in other YANG models.
- Change “evt-smp-min” and “evt-smp-max” into “min-data-object” and “max-data-object” in the “ietf-event.yang” module.

Model Overview

```
module: ietf-event
+--rw events
+--rw evt-smp-min?          uint32
+--rw evt-smp-instance-max? uint32
+--rw event* [event-name type]
+--rw event-name           string
+--rw type                  identityref
...
| +--rw type                identityref
+--rw trigger* [name]
| +--rw name                 string
...
+--rw action* [action-name]
+--rw action-name           string
```

Before

Reusable Groupings:



```
module: ietf-event-trigger
```

```
grouping trigger-grouping
```

```
+-- (test)?
+--:(existences)
| +-- existences
|   +-- target* -> /events/event/target
+--:(boolean)
| +-- boolean
|   +-- comparison? enumeration
|   +-- value?       match-value
|   +-- target*      target
+--:(threshold)
+-- threshold
```

```
module: ietf-event
```

```
+--rw events
+--rw min-data-object?          uint32
+--rw max-data-object?         uint32
+--rw event* [event-name type]
+--rw event-name                string
+--rw type                       identityref
...
+--rw trigger* [name]
| +--rw name                     string
...
+--rw (test)?
+--:(existences)
...
+--:(boolean)
...
+--:(threshold)
...
+--rw action* [action-name]
+--rw action-name               string
```

Reuse the groupings defined in event-trigger module

After

Next Steps

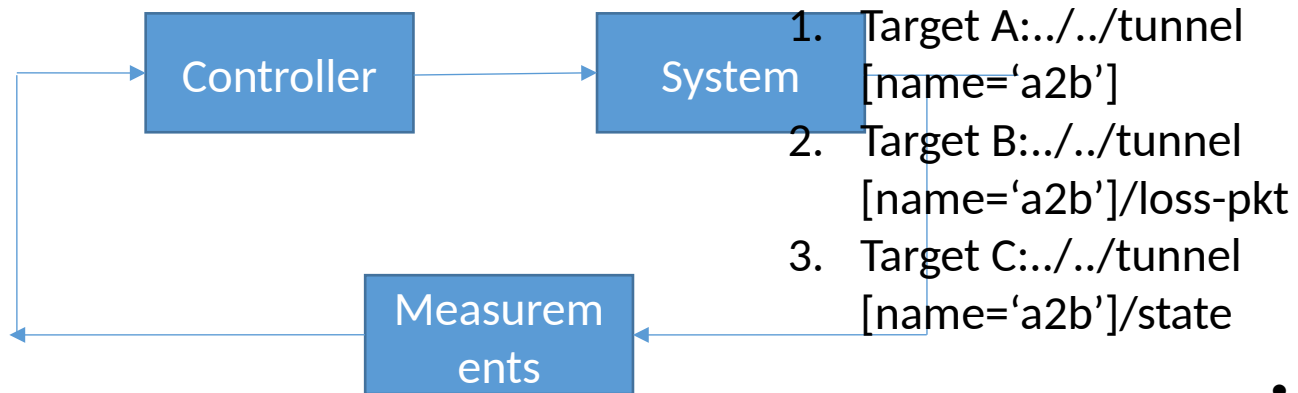
- Solicit more comments
 - Your comments and suggestions are welcome!
- Improve our solution and document
- Seeking WG adoption

What does the event yang do?

Action A : Monitor a2b's loss pkt

Action B: Notif EventB Action B': Switch the path

Action C: Notif EventC Action C': Active/standby failover



Trigger A: "../../tunnel [name='a2b']" exist

Trigger B: ../../tunnel [name='a2b']/loss-pkt > 100

Trigger C: ../../tunnel [name='a2b']/state = down

Usage Example: Lifecycle management for a VPN service

- **Event A:** A TE tunnel (tunnel a2b) is set up between VPN's site A and site B;
 - **Target:** ../../tunnel [name='a2b']
 - **Trigger :** this tunnel be set up successfully
 - **Action:** trigger a function: enable the performance monitoring of tunnel a2b
- Above action may involve a set of events, for example:
 - **Event B:** Monitor the tunnel a2b's loss packets number
 - **Target :** ../../tunnel [name='a2b'] /loss-pkts
 - **Trigger:** loss packets number cross the threshold
 - **Action:**
 1. sent a notification- "Loss packets number cross the threshold",
 2. and automatic set corresponding value, for example switch the path.
 - **Event C:** Monitoring the tunnel a2b's state
 - **Target :** ../../tunnel [name='a2b'] /state
 - **Trigger:** tunnel a2b down
 - **Action:**
 1. sent a notification - "a2b down",
 2. and trigger an another event—Event D: active/standby failover.