YANG model for NETCONF Event Notifications

draft-ahuang-netconf-notif-yang-05

A. Huang Feng, INSA-Lyon
P. Francois, INSA-Lyon
T. Graf, Swisscom
B. Claise, Huawei

September 19th 2024

YANG model for NETCONF Event Notifications

Problem statement - Netconf Notification

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
 <eventTime>2022-09-02T10:59:55.32Z</eventTime>
 <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
   <id>101</id>
   <datastore-contents>
     <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
       <interface>
         <name>eth0</name>
         <oper-status>up</oper-status>
       </interface>
     </interfaces>
   </datastore-contents>
 </push-update>
</notification>
```

```
{
   "ietf-notification:notification": {
      "eventTime": "2017-10-25T08:00:11.22Z",
      "ietf-yang-push:push-update": {
         "id": 1011,
         "datastore-contents": {
            "ietf-interfaces:interfaces": [
               "interface": {
                  "name": "eth0",
                  "oper-status": "up"
               }
            ]
         }
      }
   }
}
```

RFC 5277 - Netconf Event Notifications

RFC 8641 - YANG Push

YANG encodings:

- RFC 7950 - YANG XML
- RFC 7951 - YANG JSON
- RFC 9254 - YANG CBOR

Implementation Issues:

(1) YANG module not defined

(2) Non-existing Normative text defining this header

YANG-Push Specifications

Dependencies

- RFC8641 YANG-Push
  - Defines the “push-update” and “push-change-update” Notifications statements
  - Relies in Subscribed Notifications [RFC8639] for
    - RPCs for configuring the subscription
    - How the notification is encoded
- RFC8639 Subscribed Notifications
  - Defines the RPCs for configuring subscriptions to Datastores
  - Relies in Netconf Event Notifications [RFC5277] for
    - How the notification is encoded [XML]
- RFC5277 Netconf Event Notifications
  - Defines XML-based mechanism for sending notifications
  - Model of the notification structure implemented in XML

YANG-Push Specifications

Dependencies

- What happens if we look for the Notification definitions from a YANG perspective?
  - RFC7950 YANG 1.1 XML
  - RFC7951 JSON encoding for YANG 1.1
  - RFC9254 CBOR encoding for YANG 1.1

Notifications encoded in XML/YANG-JSON/YANG-CBOR

- RFC5277 (NETCONF Event Notifications) Defines the structure of the Notification and XML examples:

```
<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
   <eventTime>2007-07-08T00:02:00Z</eventTime>
   <event xmlns="http://example.com/event/1.0">
       <eventClass>fault</eventClass>
       <reportingEntity>
           <card>Ethernet2</card>
       </reportingEntity>
       <severity>critical</severity>
    </event>
</notification>
```

- RFC7950 (YANG 1.1) Section 4.2.10, defines how Notifications should be encoded when modeled in YANG:

YANG Example:

```
notification link-failure {
  description
    "A link failure has been detected.";
  leaf if-name {
    type leafref {
      path "/interface/name";
    }
  }
  leaf if-admin-status {
    type admin-status;
  }
  leaf if-oper-status {
    type oper-status;
  }
}
```

NETCONF XML Example:

```
<notification
    xmlns="urn:ietf:params:netconf:capability:notification:1.0">
  <eventTime>2007-09-01T10:00:00Z</eventTime>
  <link-failure xmlns="urn:example:system">
    <if-name>so-1/2/3.0</if-name>
    <if-admin-status>up</if-admin-status>
    <if-oper-status>down</if-oper-status>
  </link-failure>
</notification>
```

Notifications encoded in XML/YANG-JSON/YANG-CBOR

- RFC7951 (YANG-JSON):
  - Notifications are **not explicitly** covered
  - Example in Section 5.5 covering how “anydata” statements should be encoded:

```
Example: For the anydata definition

anydata data;

the following is a valid JSON-encoded instance:

"data": {
  "ietf-notification:notification": {
    "eventTime": "2014-07-29T13:43:01Z",
    "example-event:event": {
      "event-class": "fault",
      "reporting-entity": {
        "card": "Ethernet0"
      },
      "severity": "major"
    }
  }
}
```

eventTime as defined in RFC5277 present

Notifications encoded in XML/YANG-JSON/YANG-CBOR

- RFC9254 (YANG-CBOR):
  - Notifications are **not explicitly** covered
  - Defines a Notification as a “container-like” structure:

> **container-like instance:**
> An instance of a container, a YANG data structure, notification contents, RPC input, RPC output, action input, or action output (Section 4.2); a list entry in a list (Section 4.4); or an anydata node (Section 4.5).

  - An example of a “notification” statement is in Section 4.5 covering “anydata” statements

```
module example-port {
  ...

  notification example-port-fault {   // SID 60200
    leaf port-name {                  // SID 60201
      type string;
    }
    leaf port-fault {                 // SID 60202
      type string;
    }
  }
}
```

```
{
  60123 : {                   / last-event (SID 60123) /
    47(60200) : {             / event-port-fault (SID 60200) /
      1 : "0/4/21",           / port-name (SID 60201) /
      2 : "Open pin 2"        / port-fault (SID 60202) /
    }
  }
}
```

Using YANG-SID

```
{
  "event-log:last-event" : {
    "example-port:example-port-fault" : {
      "port-name" : "0/4/21",
      "port-fault" : "Open pin 2"
    }
  }
}
```

ietf-notification:notification container missing

eventTime leaf missing

Using names in keys

YANG model for NETCONF Event Notifications

Proposal Summary

- (1) Use Normative text to explicit how message need to be encoded [mimick RFC8040 RESTCONF]
- (2) Definition of the notification structure in a YANG
- (3) RESTCONF out of the scope of the document

- Updates multiple RFCs:
  - RFC5277 (NETCONF Notifications) → The Notification is defined in this RFC, using XML
  - RFC8639 (Subscribed Notifications) → The Notification uses the definition of RFC5277
  - RFC7951 (YANG JSON) → Notifications are not **explicitly** defined
  - RFC9254 (YANG CBOR) → Notifications are “container-like” instances

YANG model for NETCONF Event Notifications

Proposal (1) – Use Normative Text

- (1) Use Normative text to explicit how message need to be encoded [mimick RFC8040]

- Normative text for each encoding and including an example: XML, JSON, CBOR

> A YANG notification encoded in JSON is structured as a root "notification" container. The namespace of this container is the name of the YANG module "ietf-notification" defined in Section 5.¶
>
> Two child nodes within the "ietf-notification:notification" container are expected, representing the event time and the notification payload. The "eventTime" node is defined within the same namespace as the "ietf-notification:notification" container and is compliant with [RFC3339].

Section 4.2. JSON encoding

YANG model for NETCONF Event Notifications

Proposal (2) – Define a YANG module

- (2) Definition of the notification structure in a YANG

- Definition of the notification structure in a YANG
- Uses the same XML URI as RFC5277
- “eventTime” in CamelCase following model defined in RFC5277

```
sx:structure notification {
  leaf eventTime {
    type yang:date-and-time;
    mandatory true;
    description
      "The date and time the event was generated by the event source.
      This parameter is of type dateTime and compliant to [RFC3339].
      Implementations must support time zones.
      The leaf name in camel case matches the name of the XSD element
      defined in Section 4 of RFC5277.";
  }
}
```

```
module: ietf-notification

  structure notification:
    +-- eventTime    yang:date-and-time
```

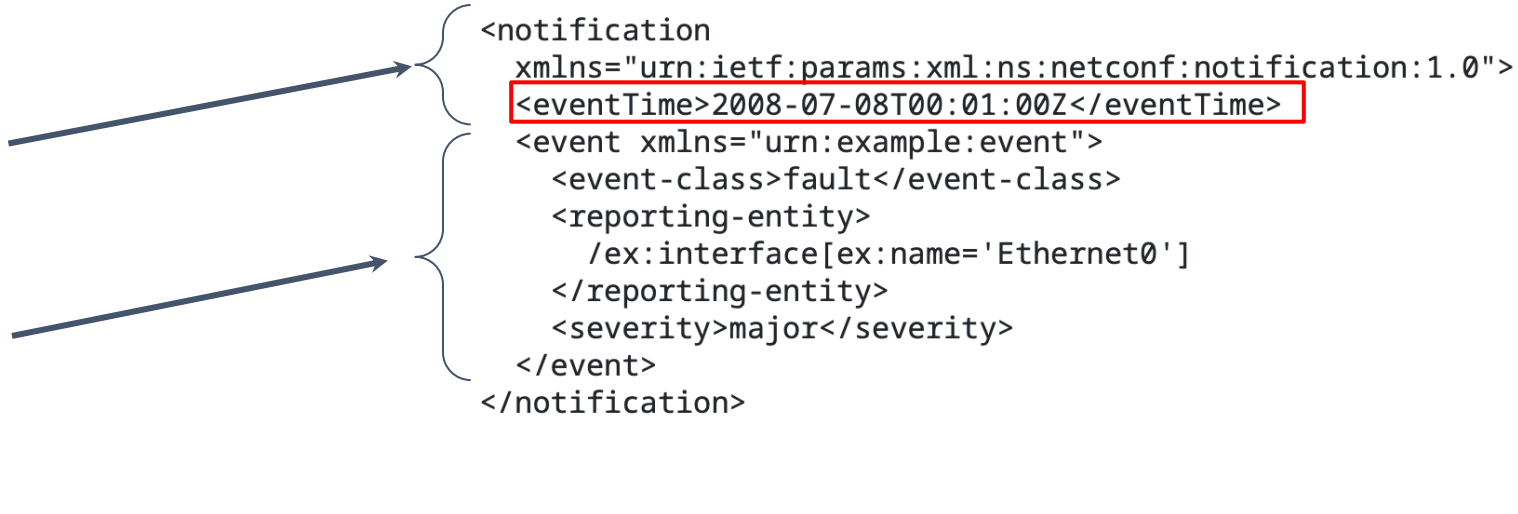
*Structure definition based on the feedback on the list [Oct 17, 2022]: https://mailarchive.ietf.org/arch/msg/netconf/q2nf9Da05R3YIusPhbt6TwMo4Q8/

Why the YANG module defines a “structure” rather than a “container”?

Section 7.16.3 of RFC7950 defines:

```
module example-event {
  yang-version 1.1;
  namespace "urn:example:event";
  prefix "ev";

  notification event {
    leaf event-class {
      type string;
    }
    leaf reporting-entity {
      type instance-identifier;
    }
    leaf severity {
      type string;
    }
  }
}
```



```
<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2008-07-08T00:01:00Z</eventTime>
  <event xmlns="urn:example:event">
    <event-class>fault</event-class>
    <reporting-entity>
      /ex:interface[ex:name='Ethernet0']
    </reporting-entity>
    <severity>major</severity>
  </event>
</notification>
```

Why the YANG module defines a “structure” rather than a “container”?

If we define the notification as a container:

```
container notification {
  leaf eventTime {
    type yang:date-and-time;
    mandatory true;
    description
      "The date and time the event was generated by the event
      source. This parameter is of type dateTime and compliant
      to [RFC3339]. Implementations must support time zones.
      The leaf name in camel case matches the name of the XSD
      element defined in Section 4 of RFC5277.";
  }
}
```

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-09-01T10:00:00Z</eventTime>
</notification>
```

The notification defined in Section 7.16.3 of RFC7950 needs to be defined as augmentation rather than using the “notification” statement. Otherwise, the relationship is between “example-event” and “ietf-notification” is non-existant.

```
module example-event {
  yang-version 1.1;
  namespace "urn:example:event";
  prefix "ev";

  import ietf-notification {
    prefix inotif;
  }

  augment "/inotif:notification" {
    container event {
      leaf event-class {
        type string;
      }
      leaf reporting-entity {
        type instance-identifier;
      }
      leaf severity {
        type string;
      }
    }
  }
}
```

YANG model for NETCONF Event Notifications

Proposal (3) – RESTCONF out of scope

- (3) RESTCONF out of the scope of the document

- Namespace of a notification for RESTCONF remains “ietf-restconf” as defined in RFC8040 (Sec. 6)

```
<notification
   xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
   <eventTime>2013-12-21T00:01:00Z</eventTime>
   <event xmlns="http://example.com/event/1.0">
      <event-class>fault</event-class>
      <reporting-entity>
          <card>Ethernet0</card>
      </reporting-entity>
      <severity>major</severity>
    </event>
</notification>
```

XML notification from RFC8040

```
{
  "ietf-restconf:notification" : {
    "eventTime" : "2013-12-21T00:01:00Z",
    "example-mod:event" : {
      "event-class" : "fault",
      "reporting-entity" : { "card" : "Ethernet0" },
      "severity" : "major"
    }
  }
}
```

JSON notification from RFC8040

YANG model for NETCONF Event Notifications

Status (last IETF meeting)

- Clear interest from the WG
  - Plenty of support during the WG adoption call
  - Addressed Mohamed Boucadair’s comments on -05 iteration
- Triggered other discussions related to YANG-Push

YANG model for NETCONF Event Notifications

Questions to the WG

- (1) Is updating the RFCs the way to fix this issue?
- (2a) Is normative text the way to solve this gap?
- (2b) Is providing a YANG module definition the way to solve this gap?
  - Do we have a rough consensus on using a struct in the YANG definition?
- (3) Do we have a rough consensus that this header should be extensible?

Next steps:

➢ Find consensus

➢ WG Adoption

Notification Header Extensions

Extend Netconf Notifications with Hostname and Sequence Number

Extends ietf-notification and ietf-system-capabilities

```
module: ietf-notification-sequencing

  augment /sysc:system-capabilities/notc:subscription-capabilities:
    +--ro sysname-sequence-supported?   notification-support {sysname-sequence}?

  augment-structure /inotif:notification:
    +-- sysName            inet:host
    +-- sequenceNumber     yang:counter32
```

```
{
  "ietf-notification:notification": {
    "eventTime": "2023-03-25T08:30:11.22Z",
    "ietf-notification-sequencing:sysName": "example-router",
    "ietf-notification-sequencing:sequenceNumber": 1,
    "ietf-yang-push:push-update": {
      "id": 6666,
      "ietf-yp-observation-time:observation-time": "2023-02-04T16:30:09.44Z",
      "ietf-yp-observation-time:point-in-time": "current-accounting",
      "datastore-contents": {
        "ietf-interfaces:interfaces": [
          {
            "interface": {
              "name": "eth0",
              "type": "iana-if-type:ethernetCsmacd",
              "oper-status": "up",
              "mtu": 1500
            }
          }
        ]
      }
    }
  }
}
```

- YANG-Push [RFC8641] uses the same structure defined in [RFC5277]. [RFC5277] defines the structure using XML encoding. For other YANG encodings, [I-D.ahuang-netconf-notif-yang] describes how notifications should be encoded in JSON and CBOR.
- An implementation supporting this document extends the notification defined in Section 4 of [I-D.ahuang-netconf-notif-yang] adding a sysName and a sequenceNumber. Two more child nodes within the "notification" root container are expected, representing the sysName and the sequenceNumber.
- YANG-related system capabilities can be discovered by subscribing to the datastore defined in [RFC9196]. This document augments the "ietf-notification-capabilities" defined in Section 5 of [RFC9196] so that the YANG-Push receiver can learn the capabilities defined in this documents through the datastore.

Extend Netconf Notifications with Hostname and Sequence Number

And adds a new Netconf Capability

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
        urn:ietf:params:xml:ns:netconf:base:1.0
      </capability>
      <capability>
        urn:ietf:params:netconf:capability:startup:1.0
      </capability>
      <capability>
        urn:ietf:params:netconf:capability:notification:1.0
      </capability>
      <capability>
        urn:ietf:params:netconf:capability:notification-sysname-sequence:1.0
      </capability>
    </capabilities>
  <session-id>4</session-id>
</hello>
```

- Adds a new capability to NETCONF so that the NETCONF server can notify the support of the sysName and sequenceNumber to the NETCONF client.
- In the capability exchange between the NETCONF client and server, the server will announce this capability along with the supported capabilities by the server as shown in the example.

BACKUP

BACK UP: Difference with ***draft-ietf-netconf-notification-messages***

draft-ahuang-netconf-notif-yang

```
module: ietf-notification

  structure notification:
    +-- eventTime    yang:date-and-time
```

draft-ietf-netconf-notification-messages

```
structure message
   +--ro message!
      +--ro message-header
      |  +--ro message-time            yang:date-and-time
      |  +--ro message-id?             uint32
      |  +--ro message-generator-id?   string
      |  +--ro notification-count?     uint16
      +--ro notifications*
      |  +--ro notification-header
      |  |  +--ro notification-time         yang:date-and-time
      |  |  +--ro yang-module?              yang:yang-identifier
      |  |  +--ro subscription-id*          uint32
      |  |  +--ro notification-id?          uint32
      |  |  +--ro observation-domain-id?    string
      |  +--ro notification-contents?
      |  +--ro notification-footer!
      |     +--ro signature-algorithm    string
      |     +--ro signature-value        string
      |     +--ro integrity-evidence?    string
      +--ro message-footer!
         +--ro signature-algorithm    string
         +--ro signature-value        string
         +--ro integrity-evidence?    string
```