



Netconf Notifications Over Web Sockets

ODL Magnesium DDF, Antwerp - September 2019

Balaji Varadaraju
Lumina Networks

Agenda

- Current State
- Proposes solution to emit Netconf Notifications over web sockets

Current State

- When a NETCONF device is mounted onto the controller all the YANG models are available for CRUD operations through RESTCONF
- MDSAL Datastore change events and YANG notifications are available over websockets
- However NETCONF notifications terminate at the controller and are not available over websockets

Application Overview

- This is a simple application module to overcome this limitation.
- The application module can intercept the NETCONF notifications from the intended devices , wrap them in a YANG notification and make them available over websocket.

Application Overview

- Application uses data driven approach. The intended capability to identify a device, it's stream id to subscribe to, and YANG notification identifier are defined in an external JSON file.
- The application will listen on NETCONF device mount events and automatically subscribe to the devices for defined streams.
- Application will capture the events and wrap them in the out of the box defined YANG notification and emit it
- This YANG notification can be subscribed over web socket.

Application YANG snippet

```
grouping notification {
  leaf node {
    type string;
  }
  leaf time-stamp {
    type string;
  }
  anyxml payload {
    mandatory true;
    description
      "Encapsulated notification. The format is the XML representation of
      a notification according to RFC6020 section 7.14.2.";
  }
}

notification event-notification {
  uses notification;
}
```

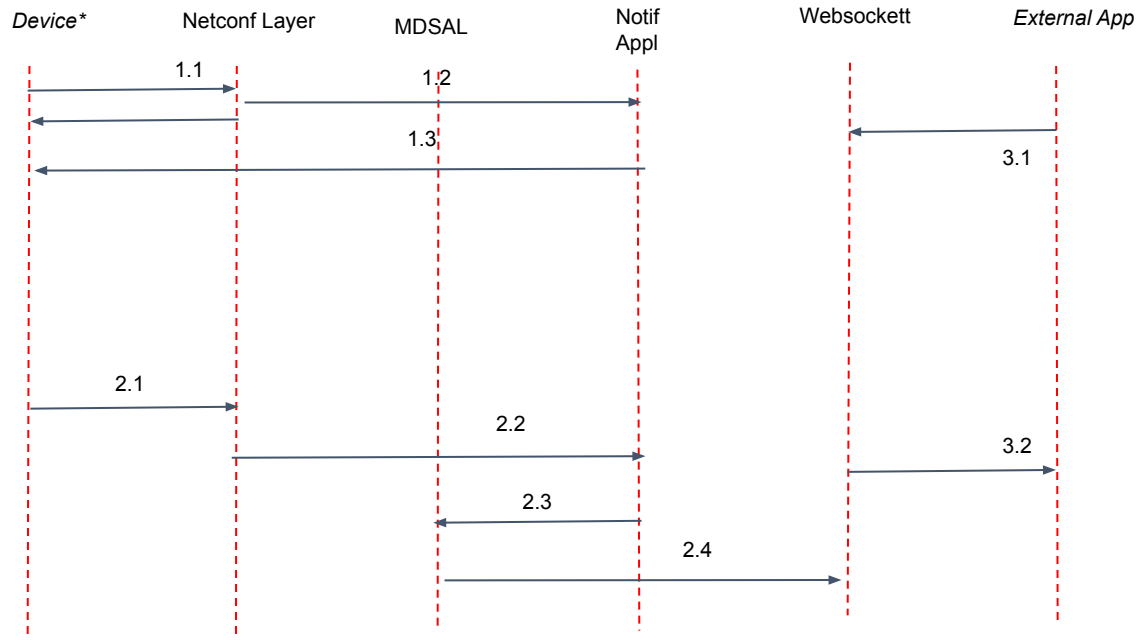
Declarative JSON Snippet

```
{  
  "notifications" : [  
    {  
      "yang-id" : "event-notification",  
      "vendor" : "vendor1",  
      "capabilities" : ["http://www.vendor1.com/ns/xyz"],  
      "streams" : ["stream1"]  
    }  
  ]  
}
```

Application Workflow

- A NETCONF Device gets mounted onto the controller and successfully gets connected
- Application will get the call back on connection status
- Application will examine the JSON file for list of intended devices to subscribe
- If the capability matches any of the items listed in the JSON file, application will subscribe to notifications for the listed streams.
- Application will register a callback for all notifications for that device by examining the schema context of mounted device
- Device will publish NETCONF events to controller which will be fed to the application.
- Application will wrap this device notification in a YANG notification and publish
- YANG notification is available over websocket

Application Workflow



1. Device connection and notification subscription
Device gets mounted either conventional or callhome. Notif application gets the mount status, examines JSON file to check if subscription is needed and if so subscribes to notifications from the device. It will also register for call backs for all notifications defined for that device by examining the schema context
2. Notification Processing
Device emits NETCONF notification to the controller. Since the application has registered callback for all notifications they will be sent to callback routine. Application then wraps this notification in a MDSAL YANG notification as fined in the YANG file and publishes it. This notification is then available over websockett.
3. External clients obtaining the notifications over WS
External clients subscribes to the YANG notification as described in the application YANG model using the standard ODL subscription mechanism. Notifications are then available to them over websockett

Limitations

- The device needs to support notifications.yang file. If not we need to sideload this YANG file during device mount.
- If the user needs different wrapped YANG notification per vendor (Say notification-x for vendor-x and notification-y for vendor-y), it needs to be defined in application YANG file. This is explained in the next slide

YANG and JSON snippet to define new notification

- YANG file snippet

```
notification event-notification-xyz {  
    uses notification;  
}
```

- JSON File snippet

```
{  
    "yang-id" : "event-notification-infinera",  
    "vendor" : "infinera",  
    "capabilities" : ["com:infinera:base"],  
    "streams" : ["Alarm"]  
}
```



Thanks