NOKIA

T-PCE Nokia's contributions

Code review

Javier Errea & Dominique Verchere

26-11-2020

© 2019 Nokia

Contribution status

Delivered:

- Honeycomb (v1.2.1) code extension → Circuit pack handling
- T-PCE NETCONF change notification handling (v1.2.1)
- openroadm-topology & service updates:
  - Interface D: Topology Manager ←→ Service Handler

To be delivered:

- TAPI module:
  - OpenROADM service ←→ TAPI connectivity service
  - TAPI topology service
  - TAPI notification service → Kafka broker
- Honeycomb (v2.2.1) code extension → Circuit pack handling ??
- T-PCE NETCONF event handling (v2.2.1) ??
- otn-topology updates ??
- Functional tests for code...

© 2019 Nokia

Contribution step I

- DeviceListener.java (v1.2.1): processing NETCONF change notification (i.e. Port State change)
- JIRA TRNSPRTPCE-249 workaround → creating new Thread per notification

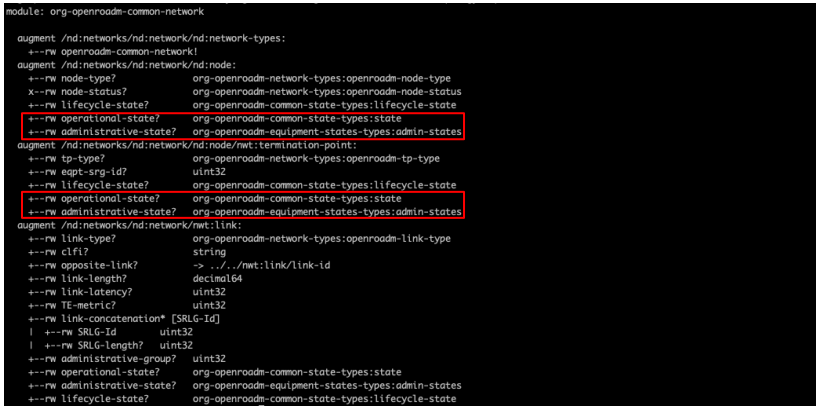
To improve:

- ChangeNotification may include a list of configuration modifications. At the moment, just handling the modification at position 0
- Same implementation for DeviceListener221.java ??
- Orange feedback ??

© 2019 Nokia

Contribution step II

- *org-openroadm-common-network.yang*:



- openroadm-topology populated with operational and administrative states

To improve:

- State should be retrieved from device configuration (at portmapping creation??). At the moment, it is assumed to be ***InService***
- Orange feedback ??

© 2019 Nokia

Contribution step III

- NetworkModelServiceImpl.java: *updateOpenRoadmTopology()*:
  - Termination Points: updating only **DEGx-TTP**, **SRGx-PPyy**, **XPDRx-NETWORKy** and **XPDRx-CLIENTy**, as they have a 1-to-1 mapping with physical device ports (external)
  - Links: new state is based on the 2 TPs at the edge of the link
- OrdLink.java: *getTpofNode()* (same as Rdm2XpdrLink.java):
  - Add state to ROADM-TO-ROADM links

To improve:

- *updateOtnTopology()* ?? As in *createOpenRoadmNode()* → just check for device version, and if it is v2.2.1 perform the otn-topology update
- Node state (i.e. ROADM/XPONDER or DEGREE/SRG): can it be OutOfService ?? How could that be related to device configuration ??
- Orange feedback ??

© 2019 Nokia

Contribution step IV

- Interface D models: *transportpce-common-ord-topology-types.yang* & *transportpce-networkmodel.yang*:

```
module: transportpce-networkmodel

  notifications:
    +---n topology-update-result
       +--ro notification-type?      transportpce-common-ord-topology-types:topology-notification-types
       +--ro ord-topology-changes* [id]
          +--ro id       string
          +--ro state    org-openroadm-common-state-types:state
```

```
typedef topology-notification-types {
  type enumeration {
    enum openroadm-topology-update {
      value 1;
    }
    enum openroadm-network-update {
      value 2;
    }
    enum clli-network-update {
      value 3;
    }
    enum otn-topology-update {
      value 4;
    }
  }
}
```

- NetworkModelListener empty implementation in ServiceHandler listeners directory

To improve:

- Extend yang models to complete interface D implementation
- Orange feedback ??

© 2019 Nokia

Contribution step V

- Implementation of NetworkModelListener.java: receiving openroadm-topology-update notifications
- NetworkModelServiceImpl.java:
  - Creation of topology-changes list → this list includes the topology elements modified and their new state
  - Added NotificationPublisherService to send openroadm-topology-update notifications. Notifications are sent only if the topology-changes list is not empty (the *updateOpenRoadmTopology()* function is triggered by any NETCONF change notification which includes a Circuit Pack update)
- Rdm2XpdrLink.java: added states on link creation (forgotten before)

To improve:

- If *updateOtnTopology()* function is implemented, the topology-changes could include a triplet **(elementId, newState, openroadm-networkLayer)**
- Orange feedback ??

© 2019 Nokia

Contribution step VI

- PCE nodes and PCE links are built with operational and administrative states obtained from the openroadm-topology
- PCE module checks links and nodes states when performing **PathComputation** & **GraphCalculation**:
  - The objective is to create a **PathDescription** with only InService openroadm-topology elements
- otn-topology populated with operational and administrative states (forgotten before → causing errors on tests)

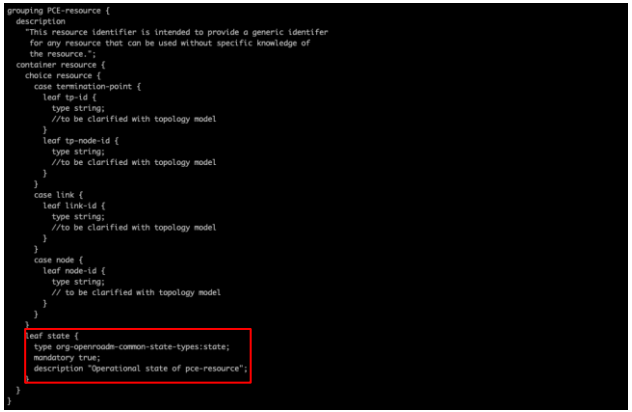
To improve:

- Orange feedback ??

© 2019 Nokia

Contribution step VII

- *transportpce-pathDescription.yang*:
  - **AToZ** & **ZToA** → use grouping **hop**
  - **hop** → uses **PCE-resource**
- NetworkModelListener.java: processing of topology-update notification (**openroadm-topology type**):
  - Added functions to ServiceDataStoreOperations.java:
    - *getServicePaths()*
    - *modifyServicePath()*
  - Retrieving **ServicePathList** which includes (among others):
    - ServiceName
    - PathDescription
  - Looking for established services affected by the topology-update notification:
    - Compare topology-changes list with path-description (AToZ & ZToA)
    - Update path-description & services accordingly



To improve:

- **Bug** → NullPointerException if no services are created
- If otn-topology update is implemented, handle **otn-topology type** notification
- Orange feedback ??

© 2019 Nokia

Contribution step VIII

- Honeynode simulator (v1.2.1 branch):
  - Implementation of Circtuit Pack notification creation in DeviceNotificationProducer.java
  - Implementation of Circuit Pack update datastore in DeviceChangeListener.java
- Tested & worked with the *sample_configs* **xxxx-full.xml**

To improve:

- Implementation for v2.2.1 branch ??
- Orange feedback ??

© 2019 Nokia

NOKIA