

O-DU Slice Assurance usecase

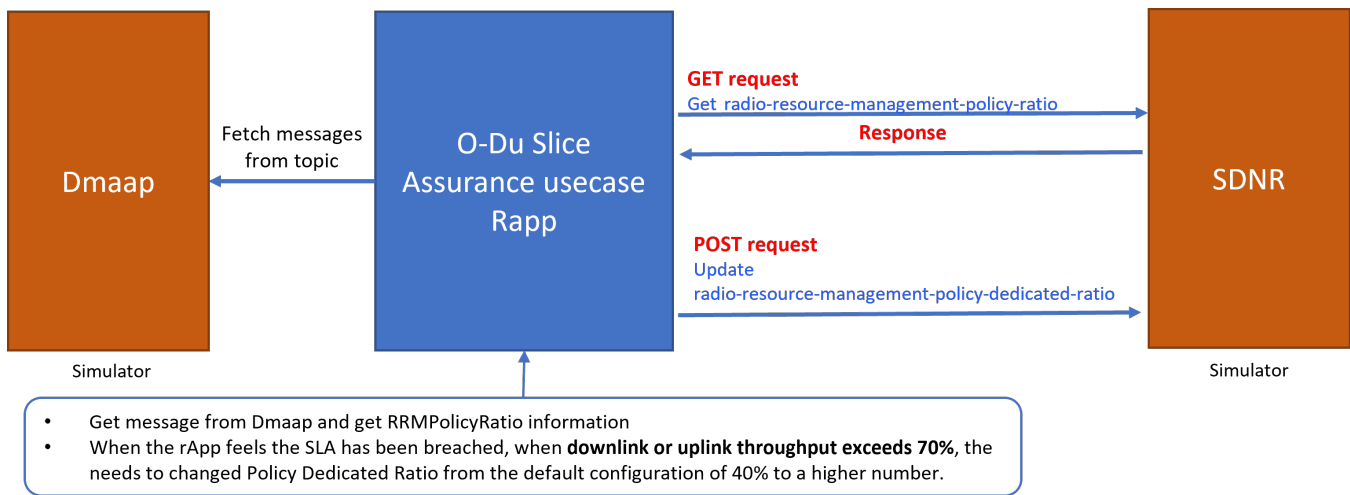
This page describes how to run current implementation for the O-DU Slice Assurance usecase.

- Standalone script version
 - Prerequisites
 - Run
 - Run Dmaap MR / SDNR stub
 - Run O-DU Slice Assurance Rapp
- Control loop version

Standalone script version

The standalone script version of the usecase is implemented in Golang.

It provides a simulator that stub both Dmaap MR and SDNR (orange boxes in the picture below), so both processes can share its data.



Prerequisites

The following need to be installed to run the script according to these instructions:

1. Go must be installed, see <https://go.dev/doc/install>.
2. Pull the nontrric repo, <https://gerrit.o-ran-sc.org/r/admin/repos/nontrric>.

Run

Run Dmaap MR / SDNR stub

This stub has been coded on Go, similar to the simulator used in O-RU O-DU Closed loop recovery use case previously described. However, this stub simulates both Dmaap VES messages and also SDNC. By default, the stub listens to port 3905, can be changed with the flag "--sdnr-port".

To run the stub, follow the steps below:

1. Goto "test/usecases/odusliceassurance/goversion/stub" in the repo.
2. Build the stub, "go build".
3. Start the stub, "./stub"

Run simulator

```
cd nonrtric/test/usecases/odusliceassurance/goversion/stub
go build
./stub [--sdnr-port <portNo>] [--dmaap-port <portNo>]
```

Example:

```
$ ./sdnr --sdnr-port 3906
Starting DmaapMR stub on port: 3905
Starting SDNR stub on port: 3606
```

Run O-DU Slice Assurance Rapp

The application takes a number of environment variables for configuration, but only MR_HOST and MR_PORT are required, others are optional. More information can be found in README.md file

Run Rapp

```
cd nonrtric/test/usecases/odusliceassurance/goversion/
go build
./oduclosedloop
```

Environment variables can be defined as part of the command line as follow:

Run rapp with environment variables

Example:

```
$ MR_HOST=http://localhost MR_PORT=3905 ./oduclosedloop
```

Control loop version

The usecase can also be instantiated as a control loop that will create a micro-service for the O-DU slice assurance app. The control loop can be created by following the detailed instructions provided in [b\) Control loop for script version](#). However, for this usecase, only one helm chart (for odu-app) needs to be copied into the kubernetes-participant and it is located at this path in the nonrtric repo:

```
nonrtric/test/usecases/odusliceassurance/goversion/helm/odu-app/
```

Moreover, the tosca template to be commissioned for this control loop is given below:

commission.yaml

```
tosca_definitions_version: tosca_simple_yaml_1_1_0
data_types:
  onap.datatypes.ToscaConceptIdentifier:
    derived_from: tosca.datatypes.Root
    properties:
      name:
        type: string
        required: true
      version:
        type: string
```

```

        required: true
node_types:
  org.onap.policy.clamp.controlloop.Participant:
    version: 1.0.1
    derived_from: tosca.nodetypes.Root
    properties:
      provider:
        type: string
        required: false
  org.onap.policy.clamp.controlloop.ControlLoop:
    version: 1.0.1
    derived_from: tosca.nodetypes.Root
    properties:
      provider:
        type: string
        required: false
      elements:
        type: list
        required: true
      entry_schema:
        type: onap.datatypes.ToscaConceptIdentifier
  org.onap.policy.clamp.controlloop.ControlLoopElement:
    version: 1.0.1
    derived_from: tosca.nodetypes.Root
    properties:
      provider:
        type: string
        required: false
      participant_id:
        type: onap.datatypes.ToscaConceptIdentifier
        required: true
  org.onap.policy.clamp.controlloop.K8SMicroserviceControlLoopElement:
    version: 1.0.1
    derived_from: org.onap.policy.clamp.controlloop.ControlLoopElement
    properties:
      chart:
        type: string
        required: true
      configs:
        type: list
        required: false
      requirements:
        type: string
        required: false
      templates:
        type: list
        required: false
      entry_schema:
        type: string
        required: true
topology_template:
  node_templates:
    org.onap.domain.sliceassurance.SliceAssuranceControlLoopDefinition1:
      version: 1.2.3
      type: org.onap.policy.clamp.controlloop.ControlLoop
      type_version: 1.0.1
      description: Control loop for Slice Assurance
      properties:
        provider: Ericsson
        elements:
          - name: org.onap.domain.sliceassurance.OduAppK8SMicroserviceControlLoopElement
            version: 1.2.3
  org.onap.k8s.controlloop.K8SControlLoopParticipant:
    version: 2.3.4
    type: org.onap.policy.clamp.controlloop.Participant
    type_version: 1.0.1
    description: Participant for k8s
    properties:
      provider: ONAP
  org.onap.domain.sliceassurance.OduAppK8SMicroserviceControlLoopElement:

```

```
version: 1.2.3
type: org.onap.policy.clamp.controlloop.K8SMicroserviceControlLoopElement
type_version: 1.0.1
description: Control loop element for odu-app
properties:
  provider: ONAP
  participant_id:
    name: K8sParticipant0
    version: 1.0.0
  participantType:
    name: org.onap.k8s.controlloop.K8SControlLoopParticipant
    version: 2.3.4
  chart:
    chartId:
      name: odu-app
      version: 0.1.0
    releaseName: odu-app
    repository:
      repoName: chartmuseum
  namespace: nonrtric
  overrideParams:
    image.repository: nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-o-du-slice-assurance
    image.tag: 1.0.0
    messagerouter.host: http://message-router.onap
    messagerouter.port: 3904
    sdnr.address: http://sdnr-simulator:9990
```

NOTE: The default hostname/port for sdnr and message-router are specified in **overrideParams** of the above file. They should be replaced with actual values if using different hostname/port.