The O1 Simulator offers support for demonstrating the O-RAN-SC E-Release Network Slicing use case proposed by Radisys. This is available in the nts-ng-o-ran-du docker image, starting with version 1.4.3.

### Features

1. Support for o-ran-du-hello-world.yang model developed by the OAM project, containing slicing information
2. Support for VES PM stndDefined messages
3. Support for VES Subscription and PM Job creation

### Usage

Details about how to start a simulated O-DU and how to use the Network Slicing use case features are below. A video demonstrating these capabilities was presented in this meeting. The recording is available there.

For starting the OAM Controller and VES Collector, a minimal SMO deployment is needed. This can be started using the instructions from here and ignoring the NonRT-RIC and ONAP-Policy steps. Please replace the version of the simulators in the ".env" file with the current 1.4.3 version.

### Starting

A simulated O-DU can be started via docker-compose. The following .env and docker-compose.yaml files can be used to start such a simulator.

```
.env

NEXUS3_DOCKER_REPO=nexus3.o-ran-sc.org:10004/o-ran-sc/

NTS_BUILD_VERSION=1.4.3
CU_PREFIX=111
DU_PREFIX=${CU_PREFIX}1
IPv6_ENABLED=true
SSH_CONNECTIONS=1
TLS_CONNECTIONS=0
#NTS_HOST_IP=10.20.11.121
NTS_HOST_IP=2001:db8:1::1
NTS_HOST_BASE_PORT=50000
NTS_HOST_NETCONF_SSH_BASE_PORT=50000
NTS_HOST_NETCONF_TLS_BASE_PORT=52000
NTS_HOST_TRANSFER_FTP_BASE_PORT=54000
NTS_HOST_TRANSFER_SFTP_BASE_PORT=56000
NTS_NF_MOUNT_POINT_ADDRESSING_METHOD=docker-mapping
NTS_NF_STANDALONE_START_FEATURES="datastore-populate ves-heartbeat ves-file-ready ves-pnf-registration netconf-call-home web-cut-through"

SDN_CONTROLLER_PROTOCOL=http
SDN_CONTROLLER_IP=2001:db8:1:50::23
SDN_CONTROLLER_PORT=8181
SDN_CONTROLLER_CALLHOME_PORT=4335
SDN_CONTROLLER_CALLHOME_IP=2001:db8:1:50::23
SDN_CONTROLLER_USERNAME=admin
SDN_CONTROLLER_PASSWORD=Kp8bJ4SXszM0WXlhak3eHlcse2gAw84vaoGGMJvUy2U

VES_COMMON_HEADER_VERSION=7.2.1
VES_ENDPOINT_PROTOCOL=https
VES_ENDPOINT_IP=2001:db8:1:50::27
VES_ENDPOINT_PORT=8443
VES_ENDPOINT_AUTH_METHOD=basic-auth
VES_ENDPOINT_USERNAME=sample1
VES_ENDPOINT_PASSWORD=sample1
```

docker-compose.yaml
version: '3.8'
x-common_env: &common_env
  IPv6_ENABLED: ${IPv6_ENABLED}
  SSH_CONNECTIONS: ${SSH_CONNECTIONS}
  TLS_CONNECTIONS: ${TLS_CONNECTIONS}
  NTS_NF_MOUNT_POINT_ADDRESSING_METHOD: ${NTS_NF_MOUNT_POINT_ADDRESSING_METHOD}

  NTS_HOST_IP: ${NTS_HOST_IP}
  NTS_HOST_BASE_PORT: ${NTS_HOST_BASE_PORT}
  NTS_HOST_NETCONF_SSH_BASE_PORT: ${NTS_HOST_NETCONF_SSH_BASE_PORT}
  NTS_HOST_NETCONF_TLS_BASE_PORT: ${NTS_HOST_NETCONF_TLS_BASE_PORT}
  NTS_HOST_TRANSFER_FTP_BASE_PORT: ${NTS_HOST_TRANSFER_FTP_BASE_PORT}
  NTS_HOST_TRANSFER_SFTP_BASE_PORT: ${NTS_HOST_TRANSFER_SFTP_BASE_PORT}

  SDN_CONTROLLER_PROTOCOL: ${SDN_CONTROLLER_PROTOCOL}
  SDN_CONTROLLER_IP: ${SDN_CONTROLLER_IP}
  SDN_CONTROLLER_PORT: ${SDN_CONTROLLER_PORT}
  SDN_CONTROLLER_CALLHOME_IP: ${SDN_CONTROLLER_CALLHOME_IP}
  SDN_CONTROLLER_CALLHOME_PORT: ${SDN_CONTROLLER_CALLHOME_PORT}
  SDN_CONTROLLER_USERNAME: ${SDN_CONTROLLER_USERNAME}
  SDN_CONTROLLER_PASSWORD: ${SDN_CONTROLLER_PASSWORD}

  VES_COMMON_HEADER_VERSION: ${VES_COMMON_HEADER_VERSION}
  VES_ENDPOINT_PROTOCOL: ${VES_ENDPOINT_PROTOCOL}
  VES_ENDPOINT_IP: ${VES_ENDPOINT_IP}
  VES_ENDPOINT_PORT: ${VES_ENDPOINT_PORT}
  VES_ENDPOINT_AUTH_METHOD: ${VES_ENDPOINT_AUTH_METHOD}
  VES_ENDPOINT_USERNAME: ${VES_ENDPOINT_USERNAME}
  VES_ENDPOINT_PASSWORD: ${VES_ENDPOINT_PASSWORD}

x-du_env: &du_env
  NTS_NF_STANDALONE_START_FEATURES: "datastore-populate ves-heartbeat ves-file-ready ves-pnf-registration web-cut-through"

x-nf: &common_nf
  stop_grace_period: 5m
  cap_add:
    - SYS_ADMIN
    - SYS_PTRACE

services:
  ntsim-ng-o-ran-du-1:
    <<: *common_nf
    image: "$NEXUS3_DOCKER_REPO/nts-ng-o-ran-du:$NTS_BUILD_VERSION"
    container_name: ntsim-ng-o-ran-du-${DU_PREFIX}
    hostname: O-DU-${DU_PREFIX}
    environment:
      <<: *common_env
      <<: du_env

networks:
  default:
    external:
      name: oam

Please adjust the values in the docker-compose.yaml according to your needs: e.g. specify the correct VES Endpoint IP address, SDN Controller IP address, rename or remove the oam network, as applicable to your specific host.

Starting can be done via:

docker-compose up -d
This will start a docker container containing the o-ran-sc-du-hello-world YANG model. The user can then start a NETCONF session to this device (default is port 830 and the IP address can be the one of the docker container).

Configure VES details

NETCONF

After the simulated O-DU is up and running, the VES subscription details can be configured. These are part of the o-ran-sc-du-hello-world model. An example edit-config NETCONF operation that configures VES endpoint details:

```xml
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <duhw:network-function xmlns:duhw="urn:o-ran-sc:yang:o-ran-sc-du-hello-world">
        <duhw:subscription-streams>
          <duhw:id>stream-1</duhw:id>
          <duhw:administrative-state>unlocked</duhw:administrative-state>
          <duhw:user-label>stream1</duhw:user-label>
          <duhw:ves-endpoint-protocol>https</duhw:ves-endpoint-protocol>
          <duhw:ves-endpoint-auth-method>basic-auth</duhw:ves-endpoint-auth-method>
          <duhw:ves-endpoint-ip>10.20.11.121</duhw:ves-endpoint-ip>
          <duhw:ves-endpoint-port>8443</duhw:ves-endpoint-port>
          <duhw:ves-endpoint-username>sample1</duhw:ves-endpoint-username>
          <duhw:ves-endpoint-password>sample1</duhw:ves-endpoint-password>
        </duhw:subscription-streams>
      </config>
    </edit-config>
  </rpc>
```

RESTCONF

After the simulated O-DU is mounted in the SDN Controller (this should be done automatically via VES pnRegistry), the VES details could also be configured via RESTCONF.

```bash
### @name setSubscriptionStream  
# baseUrl needs to be replaced by the OAM IP address and port of the SDN Controller RESTCONF interface  
# node needs to be replaced by the name of the O-DU (e.g. O-DU-1211)  
```

PM Job creation

NETCONF

Generating VES PM stdDefined events can be done by adding PM Job entries. An example edit-config NETCONF operation that adds a new PM job:
<xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <duhw:network-function xmlns:duhw="urn:o-ran-sc:yang:o-ran-sc-du-hello-world">
          <duhw:id>pm1</duhw:id>
          <duhw:administrative-state>unlocked</duhw:administrative-state>
          <duhw:user-label>pm-1</duhw:user-label>
          <duhw:granularity-period>30</duhw:granularity-period>
          <duhw:stream-target>stream-1</duhw:stream-target>
        </duhw:performance-measurement-jobs>
      </duhw:network-function>
    </config>
  </edit-config>
</rpc>

- The `granularity-period` specifies the amount of seconds between two "gathered" PM values, and it will be also the number of seconds between two generated VES events.
- `stream-target` points to a previously created stream (describing to a VES endpoint)

**RESTCONF**

```bash
### @name setPerformanceMeasurementJobs
# baseUrl needs to be replaced by the OAM IP address and port of the SDN Controller RESTCONF interface
# node needs to be replaced by the name of the O-DU (e.g. O-DU-1211)

cat payload.json

{"performance-measurement-jobs":{"id":"pm-1","administrative-state":"unlocked","user-label":"pm","job-tag":"my-job-tag","performance-metrics":[]}}

```

**PM Job deletion**

**NETCONF**

A PM Job can be deleted, thus the generation of new VES PM events associated with that job will be stopped. An example edit-config NETCONF operation that deletes a PM job:
<?xml version="1.0" encoding="utf-8"?><rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <duhw:network-function xmlns:duhw="urn:o-ran-sc:yang:o-ran-sc-du-hello-world">
          <duhw:id>pm1</duhw:id>
        </duhw:performance-measurement-jobs>
      </duhw:network-function>
    </config>
  </edit-config>
</rpc>

### @name deletePerformanceMeasurementJobs
# baseUrl needs to be replaced by the OAM IP address and port of the SDN Controller RESTCONF interface
# node needs to be replaced by the name of the O-DU (e.g. O-DU-1211)

Example VES PM stndDefined

This is an example JSON VES PM stndDefined message generated by the O-DU Simulator:
"event": {
  "commonEventHeader": {
    "domain": "stndDefined",
    "eventId": "pm1_1638984365",
    "eventName": "stndDefined_performanceMeasurementStreaming",
    "eventType": "performanceMeasurementStreaming",
    "sequence": 24,
    "priority": "Low",
    "reportingEntityId": "",
    "reportingEntityName": "O-RAN-O-DU-1",
    "sourceId": "",
    "sourceName": "O-RAN-O-DU-1",
    "startEpochMicrosec": 1638984365000000,
    "lastEpochMicrosec": 1638984370000000,
    "nfNamingCode": "SIM-O-DU",
    "nfVendorName": "O-RAN-SC SIM Project",
    "stndDefinedNamespace": "o-ran-sc-du-hello-world-pm-streaming-oas3",
    "timeZoneOffset": "+00:00",
    "version": "4.1",
    "vesEventListenerVersion": "7.2.1"
  },
  "stndDefinedFields": {
    "stndDefinedFieldsVersion": "1.0",
    "data": {
      "id": "pm1_1638984365",
      "start-time": "2021-12-08T17:26:05.0Z",
      "administrative-state": "unlocked",
      "operational-state": "enabled",
      "user-label": "pm-1",
      "job-tag": "",
      "granularity-period": 5,
      "measurements": [{"measurement-type-instance-reference": "/o-ran-sc-du-hello-world:network-function/distributed-unit-functions[id='O-DU-1211']/cell[id='cell-1']/supported-measurements[performance-measurement-type='user-equipment-average-throughput-downlink']","value": 12355,"unit": "kbit/s"}]
    }
  }
}