Implementation Note- Support for Multiple TCONTs

Introduction

This implementation note documents some low-level details on the impacts across various components in the ONOS/VOLTHA/OpenOLTAgent sub-system for multi-TCONT support.

Note: This implementation note is very technical and assumes strong understanding of existing design and code of tech-profile implementation in ONOS Apps, VOLTHA core, Openolt/onu Adapters and Openolt Agent.

Approach for multi-TCONT

Since each Technology Profile (TP) supports one scheduler (one TCONT), there are two ways to support multiple scheduler (or TCONTs)

1. Enhance the TP to support multiple scheduler (or TCONT)
2. One TP per scheduler

Approach (1) will reduce the number of TPs but has issues with mapping a given TP ID uniquely for a scheduler on a given Flow.

Thus Approach (2) is chosen due to constraint of Approach (1).

Implications on implementation

The Approach (2) has the following implication on multi-TCONT implementation

1. Each TP ID for a given `(pon_id, onu_id, uni_id)` tuple corresponds to a unique TCONT [Note 1].
2. Statement (1) conversely means that we cannot have two different TCONTs referring to same TP ID on a given `(pon_id, onu_id, uni_id)` tuple [Note 1].

The NBI systems must ensure that if a different TCONT is required for a given subscriber, they must use different TP [Note 1]. The database to store the TP related information (by openolt adapter) for each subscriber must appropriately adapt to this now constraint. These low-level details will be documented further.

Note 1

When the TP model indicates the instance is at the ONU Level (through 'Instance Control Attribute'), we would create an instance in the same way with the same index (including UNI) for the per ONU TCONT model also. Since each UNI shares the same TCONT but has its own GEM Ports we would create the first instance on the ONU with a unique TCONT(Alloc-ID) and unique GEM Ports. The second UNI to use the

Tech Profile (ONU instance = 'Single Instance') on the same ONU would reuse the TCONT from the first TP Instance but allocate its own GEM Ports. The code would need to ensure that only one TCONT is allocate for this case and it is returned to the resource manager only when all Flows which reference it are deleted.

The bandwidth profiles applied to these individual flows must be aggregated together for the upstream TCONT (DBA) scheduler. The ONU needs to rate limit the upstream GEMs to their individual Bandwidth profiles. Downstream the config is same as if the flows are individual flows – only upstream has special shared bandwidth at the Aggregate (TCONT) level.

Code examples

There are some code examples (function names, constants, variable names, code snippets) in this implementation note and are picked up from Python openolt adapter. Please look for Golang OpenOLT adapter counterparts when referring these code examples and adapt them accordingly.

Impacts on ONOS/Apps

The below API is currently used to provision service with specific tags for a subscriber

http://ip:port/onos/olt/oltapp/services/<subscriber-id>/<s-tag>/<c-tag>

The TP ID for this subscriber was chosen from the SADIS configuration based on the subscriber-id key.

This must change as below so that we can prompt the specific TP ID to be used for a given subscriber-id, s-tag and c-tag.

http://ip:port/onos/olt/oltapp/services/<subscriber-id>/<s-tag>/<c-tag>/<tp-id>

Note1: TP-ID is optional. If TP-ID is not specified, it is chosen from SADIS config.

Impacts on OFAgent

No impacts

Impacts on VOLTHA NBI and Core

No impacts

Impacts on vCLI

No impacts

Impacts in OpenOLT adapter

The way the data is stored and retrieved will change to accommodate multi-TCONT.

1. TP Path: NO UPDATE
   `service/voltha/technology_profiles/<technology>/<tp_id>`

2. TP Instance Path: NO UPDATE
   `service/voltha/technology_profiles/<technology>/<tp_id>/<uni_port_name>`
   Since TP ID is unique per TCONT, there is no impact.
3. "TP ID path" for (pon_id, onu_id, uni_id): UPDATE REQUIRED
   The 'tp_id' is stored at below path
   `service/voltha/openolt/<olt-device_id>/tp_id/<(pon_id, onu_id, uni_id)>`
   It must change as below,
   `service/voltha/openolt/<olt-device_id>/<(pon_id, onu_id, uni_id)>/tp_id`

   Also, this key will not map to a single tp_id, but a list of tp_id since we start supporting multiple tconts.
4. "Meter ID path": UPDATE REQUIRED
   The meter_id path should change from
   `service/voltha/openolt/<olt-device-id>/meter_id/<(pon_id, onu_id, uni_id)>/<direction>/<meter_id>`

   to

   `service/voltha/openolt/<olt-device-id>/ <(pon_id, onu_id, uni_id)>/<tp-id>/ meter_id/<direction>/<meter_id>`

   The 'update_meter_id_for_onu', 'get_meter_id_for_onu' and 'remove_meter_id_for_onu' functions should now carry the 'tp_id' as this is necessary to store, retrieve and delete the meter_id information.

5. Do not get alloc_id and gem_ports from KV store. Use TP Instance JSON blob to get the required alloc_ids and gemport IDs during flow processing. The change is shown below. The get_alloc_id_from_tp_instance function should be implemented to fetch alloc_id and gemport_ids from tp_instance json blob.

```
    def divide_and_add_flow(self, intf_id, onu_id, uni_id, port_no, classifier,
                            action, flow, tp_id, us_meter_id, ds_meter_id):

        self.log.debug('sorting flow', intf_id=intf_id, onu_id=onu_id,
                       uni_id=uni_id, port_no=port_no,
                       classifier=classifier, action=action,
                       tp_id=tp_id, us_meter=us_meter_id,
                       ds_meter=ds_meter_id)

        tp_instance = self.get_tech_profile_instance(intf_id, onu_id, uni_id, tp_id)
        if tp_instance is None:
            self.log.error("flow-not-added--tp-instance-unavailable")
            return

        pon_intf_onu_id = (intf_id, onu_id, uni_id)
-        alloc_id = \
-            self.resource_mgr.get_current_alloc_ids_for_onu(pon_intf_onu_id)
-        gem_ports = \
-            self.resource_mgr.get_current_gemport_ids_for_onu(pon_intf_onu_id)
+
```

```
+         alloc_id = self.get_alloc_id_from_tp_instance(tp_instance)
+         gemports = self.get_gemports_from_tp_instance(tp_instance)

         if alloc_id is None or gem_ports is None:
             self.log.error("alloc-id-or-gem-ports-unavailable",s
                            alloc_id=alloc_id, gem_ports=gem_ports)
             return

         self.create_us_scheduler_queues(intf_id, onu_id, uni_id, tp_instance, us_meter_id)
         self.create_ds_scheduler_queues(intf_id, onu_id, uni_id, tp_instance, ds_meter_id)
```

6. The 'get_current_alloc_ids_for_onu' in openolt_resource_manager.py should return the list of alloc_ids for the ONU and not just the first alloc_id.
7. Clean-up:

   In the below two functions in openolt_flow_mgr.py, loop through all the tp_id for the ONU and clean-up the TP Instance associated with those tp_id. Currently, it is done only for one tp_id, but we will have multiple tp_id when multi-Tcont is supported.

   a. _clear_flow_id_from_rm
   b. clear_flows_and_scheduler_for_logical_port

Impacts in Tech-Profile module

When the TP model indicates the instance is at the ONU Level (through 'Instance Control Attribute'), we would create an instance in the same way with the same index (including UNI) for the per ONU TCONT model also. Due to the face that each UNI shares the same TCONT but has its own GEM Ports we would create the first instance on the ONU with a unique TCONT(Alloc-ID) and unique GEM Ports. The second UNI to use the Tech Profile (ONU instance = 'Single Instance') on the same ONU would reuse the TCONT from the first TP Instance but allocate its own GEM Ports. The code would need to ensure that only one TCONT is allocate for this case and it is returned to the resource manager only when all Flows which reference it are deleted.

Impacts in Resource Manager module

1. The 'update_alloc_ids_for_onu' over-writes old alloc_ids. This needs to change to append new alloc_ids with the old contents.
2. The 'update_gemport_ids_for_onu' over-writes old gemport_ids. This needs to change to append new gemport_ids with the old contents.

Impacts in OpenONU adapter

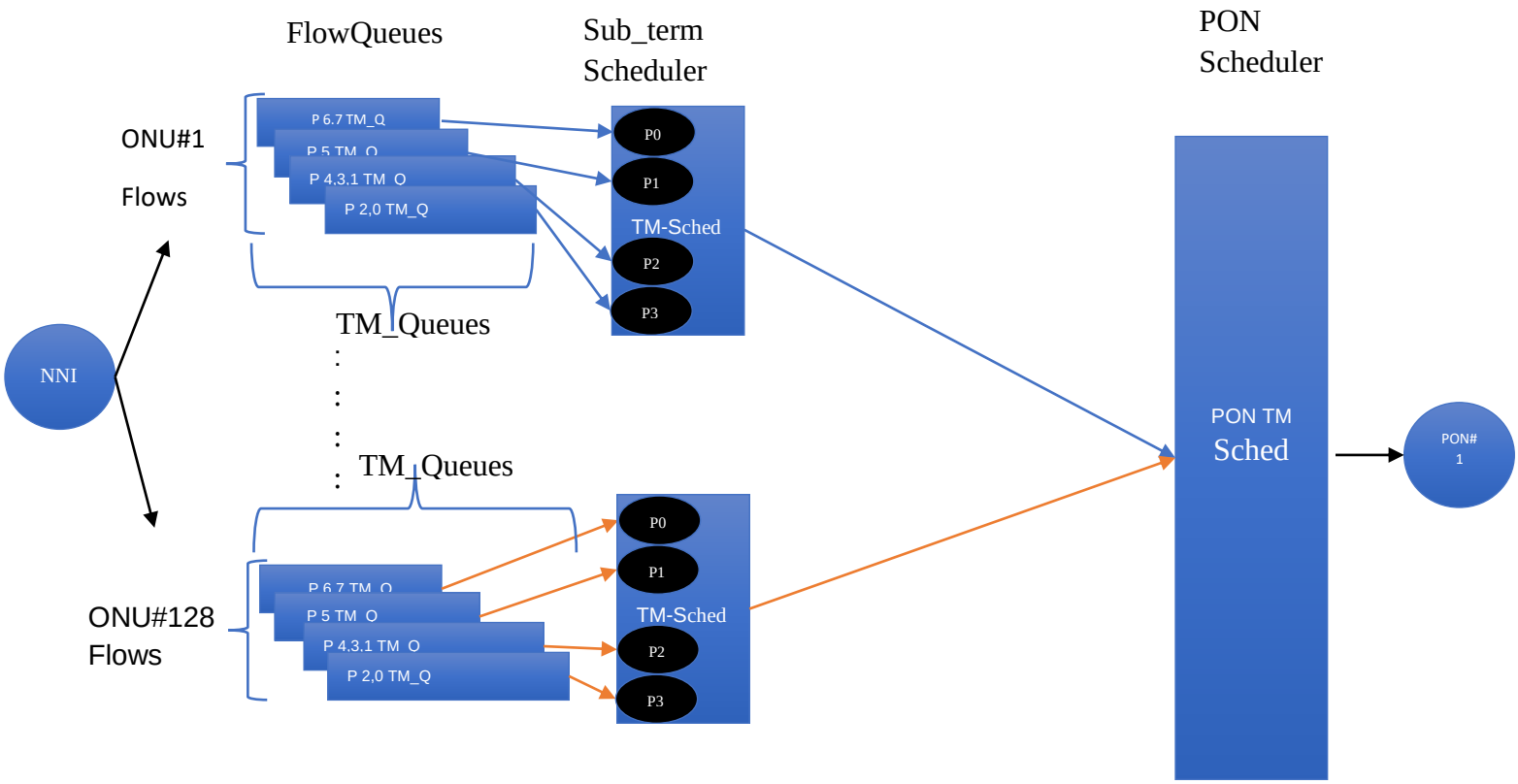
The ONU adapter already maintains states on a per TP Instance Path for a given UNI port. Since, each TCONT has a unique TP path, there are no issues and no changes required.

Impacts in OpenOLT agent

The Queue and Scheduler model at OpenOLT agent is below.

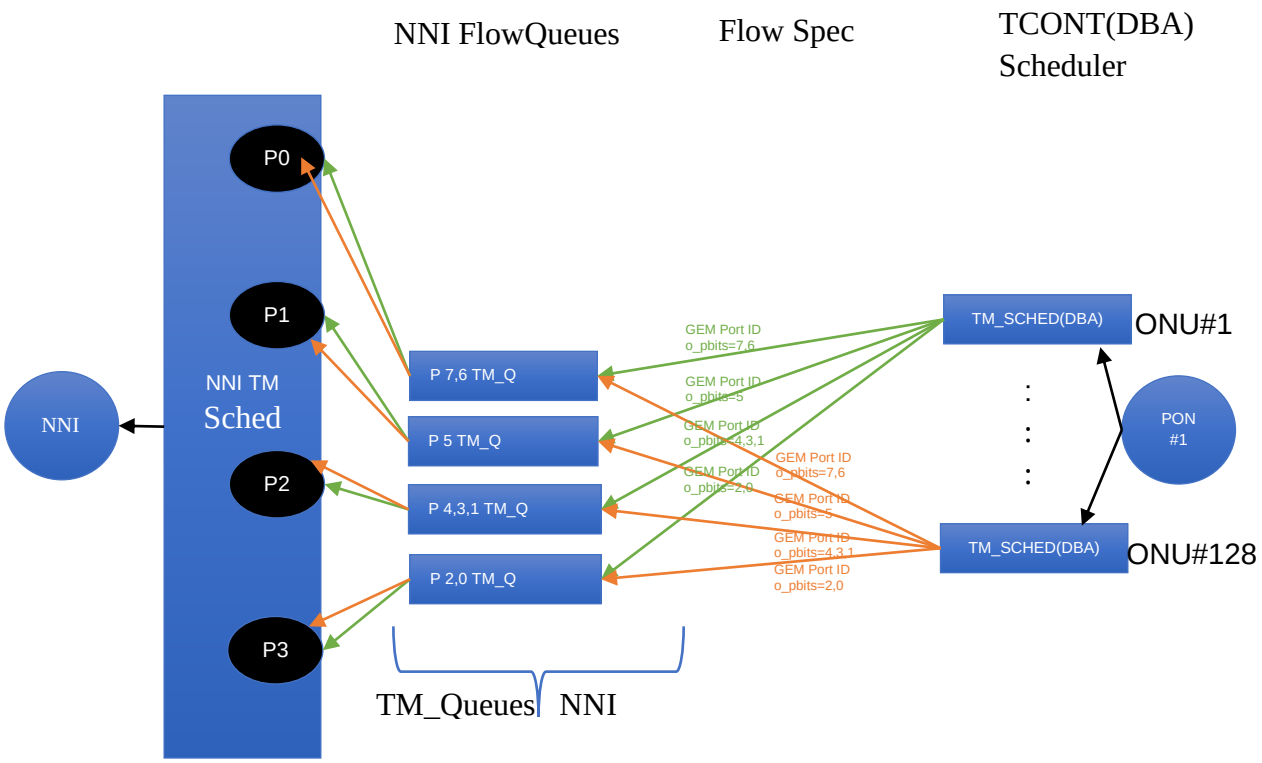
Downstream

In the Downstream Direction the OLT Traffic Management for AT&T could be as the Figure Below:



Upstream

In the Upstream Direction the OLT Traffic Management could be as the Figure below:



Impact on management of Scheduler IDs

As seen in the above section, in the current model, we have unique schedulers per subscriber in upstream and downstream direction. The sched_id for these schedulers is not managed from VOLTHA resource manager but are generated at OpenOLT agent. With single-tcont, the sched_id is unique per (pon_intf_id, onu_id, uni_id, direction) key. When we start supporting multiple-tcont we need slight adjustment to get unique sched-ids.

Uplink

Add alloc_id to (pon_intf_id, onu_id, uni_id, direction) to get unique IDs

Downlink

Although alloc_id is not relevant in Downlink, adding alloc_id should solve the problem for downlink too.

Impact on management of Queue IDs

No impact.