Robert Peschi, Nokia,

Joey Boyd, Adtran

July 3rd, 2024

Achieving Scalable YANG Modules for Large Systems

Executive summary

- The telecom industry needs to deploy YANG models for very large systems (30K+ interfaces), building on IETF existing YANG models (example: Passive Optical Network (PON) managing an OLT combined with many ONUs)
- Current standard models do not scale at that size level which make them increasingly difficult to deploy for such Use Cases. This is a serious issue for which a deployable solution is urgently needed.
- In the Broadband Forum (BBF), taking the example of ONU management, studies have concluded that YANG scalability in the OLT can be achieved by optimizing the structure of the model and reducing the size of the data store:
  - Defining a **list of ONUs** with each entry regrouping all data nodes of the ONU
  - Configuring at OLT root level (outside the list of ONUs) a centralized list of profiles that each ONU can refer to (e.g., list of profiles)
  - Configuring once for all a single ONU "model" - call it "ONU template" and configure an ONU instance with just a reference to an ONU template and only few data nodes with values specific to the ONU. The data to be configured to the ONU is then created by making a copy of the ONU template, further tuned with values specific to this ONU.
- These techniques lead to the following 3 requirements:
  1) Data nodes (including IETF's) originally defined at the root of standard YANG models are "moved" deeper in the YANG tree. For instance: `interfaces/interface -> onus/onu/interfaces/interface` (same for ONU templates)
  2) ONU instance data nodes can have their mandatory and default statements removed to avoid interference with the template.
  3) ONU instance and ONU template data nodes can refer to data nodes located at the root of the tree (e.g., shared profiles)
- Although IETF Schema mount (RFC8528) supports requirement (1), it does not support (2) - its very objective is to mount *unmodified* existing modules - , and also it puts restrictions on (3).
- "`draft-jouqui-netmod-yang-full-include`" proposes an extension of RFC8528 but satisfies neither (2) nor (3) in the current version of the draft.
- This presentation gives more insight about what is missing in Schema-mount to make it a solution for YANG scalability.

YANG: strong point and weak point

- One of the greatest assets of YANG is that it allows embedding semantic constraints in a model definition, e.g., ‘when’, ‘must’, etc.
  - any off-the-shelf YANG tool can automatically impose or verify the consistency of a configuration: no dependency on vendors software that are potentially prone to interpretation errors, jeopardizing interoperability

- A weak point is that running YANG validation rules can however become very computationally intensive
  - For very large data stores, performance *collapses* far below practical levels !
  - *YANG models must be specially designed when they are to be used for large-scale devices*

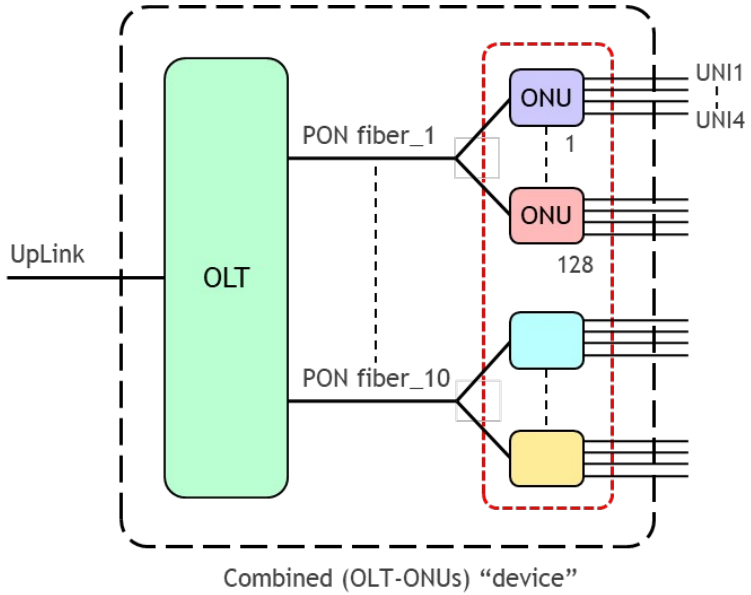
- PON Access Networks targeted by BBF lead to huge YANG datastores (where greater than 10K interfaces need to be managed).
  - Many current standard models run into performance issues when scaled to that extreme.
  - To address this, model structure must be adapted, and other modeling techniques used, to
    - Reduce the size of the datastore (to reduce YANG validation time and more generally OAM processing burden)
    - More efficiently organize data nodes (rather than define new ones).

YANG scalability is a general problem for large systems

- Although the YANG scalability issue has been pointed out for the management of OLTs with many ONUs, it is a general YANG problem encountered by any large system.
- This is a major issue faced by the telecom industry !
- For the sake of concreteness, the next slides detail the methods used to improve the scalability of OLTs with many ONUs, however these methods can be generalized to any large system that contains repetitive amounts of similar data nodes. For instance, systems with many interfaces carrying the same service.

PON Access Network, a matter of OLT and ONUs

- A PON Access Network is comprised of an OLT (Optical Line Termination) and ONUs (Optical Network Unit) entities each with their own data nodes (interfaces, hardware, etc…).
- The most common management mode is to consider the OLT and ONUs as *a single **large** device*.
  [BBF TR-385 "*OLT-ONUs Combined NE mode"*]
- Defining a scalable YANG model for this Combined "OLT-ONUs device" is crucial for achieving a usable and interoperable model that can be used by Northbound systems.
- An example of improvement is to develop an *ONU management model* that scales efficiently for OLT-ONU Combined NE mode



OLT-ONUs as a single large device

Dimension example:

- 3 VLAN interfaces per UNI
- 4 UNIs per ONU
- 128 ONUs per PON
- 10⁺ PONs per OLT

This results in 15K⁺ interfaces, just on the ONU side.

Moreover, each of these interfaces has a counterpart on the OLT side resulting in a grand total in the combined model of 30K⁺ interfaces related to ONUs.

How to improve YANG scalability of OLTs with many ONUs ?

Three silver bullets:

1. *Clustering* data nodes in the datastore per ONU
2. Reducing datastore size by using *shared profiles* shared across ONUs
3. Reducing datastore size by using ONU *templates*

1. *Clustering* data nodes in the datastore, per ONU (illustrating figure on next slide)

- The typical YANG model for a combined (OLT-ONUs) does not separate OLT data nodes from ONU data nodes (e.g., in the case of interfaces, there is only one flat list randomly populated with OLT and ONUs interfaces).
- Moving to 30K^{+} interfaces, this lack of structure is very inefficient, for instance when it comes to running validation rules between interfaces or deleting an interface without leaving orphan interfaces behind, higher in the stack.
- The strong decoupling between OLT and ONUs in term of validation rules or deletion constraints is not exploited nor the potential configuration similarity between ONUs that may have identical service models applied to them.

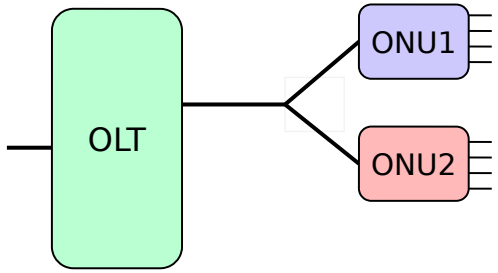
Analysis suggests that the best way to proceed would be a model that **clusters** data nodes of individual ONUs, alongside the OLT's, as if individual autonomous devices. Doing this scopes ONU validation rules to a much smaller number of data nodes (think for instance about interfaces).

This can be achieved by defining a **list of ONUs** with each entry regrouping all data nodes of the ONU.

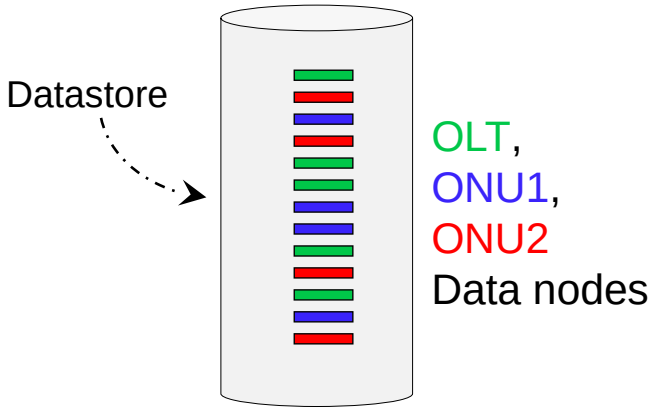
The next slide illustrates the principle.

Clustering data nodes per ONU

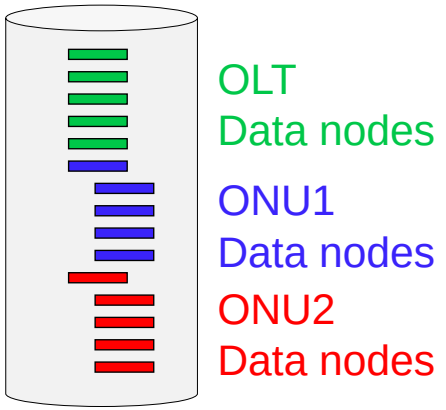
Example of system to be modeled



Traditional model:
all data nodes interleaved



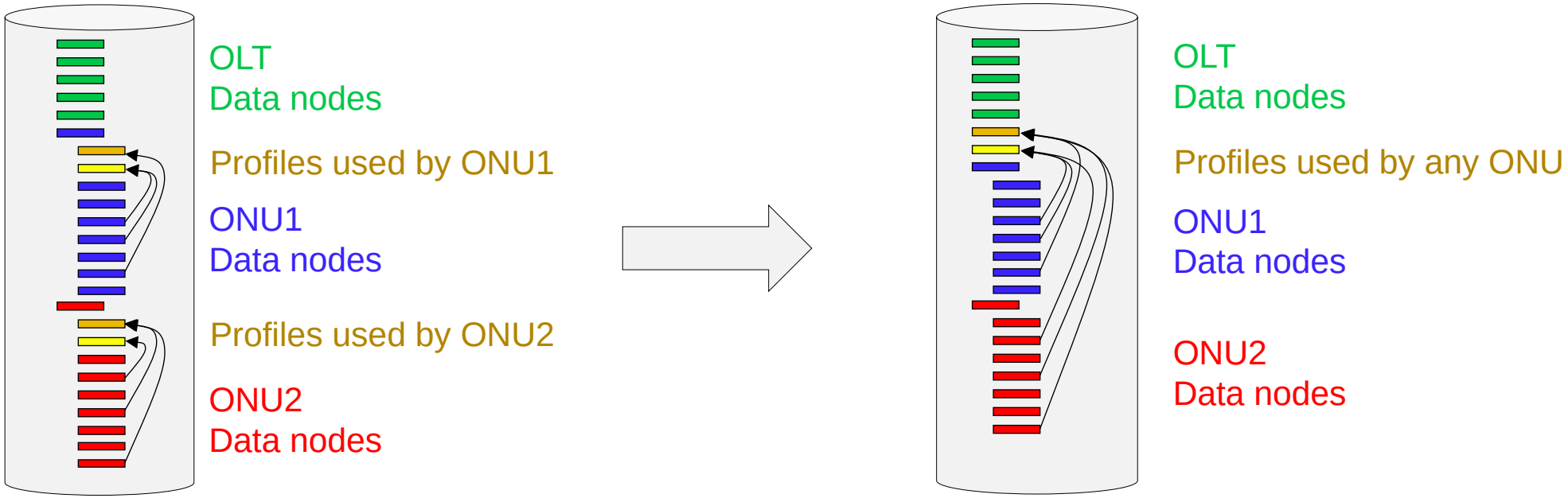
Improved model:
ONU data nodes “clustered” per ONU



2. Reducing datastore size by using shared profiles shared across ONUs

- Obviously if large systems suffer from scalability, reducing the size of the datastore will improve scalability !
- It is common practice in YANG standards to define profiles. A profile is a set of data nodes (parameters) that give identical properties to several elements in the system, for instance to interfaces of a certain type. Augmenting each interface with a reference to a profile is more efficient than explicitly augmenting each interface with each properties separately.
- The YANG configuration of an ONU extensively makes use of profiles. That on its own is already an efficient way to reduce data store size.
- This can be exploited to a maximum extend by configuring at OLT root level a *centralized list of profiles that each ONU can refer to*. Indeed, although “clustered” ONUs are individual autonomous entities, it would be wasteful to configure each ONU with the profiles it needs because many ONUs are expected to use the same profiles.
- This is illustrated in the next slide

Clustered ONUs using shared profiles

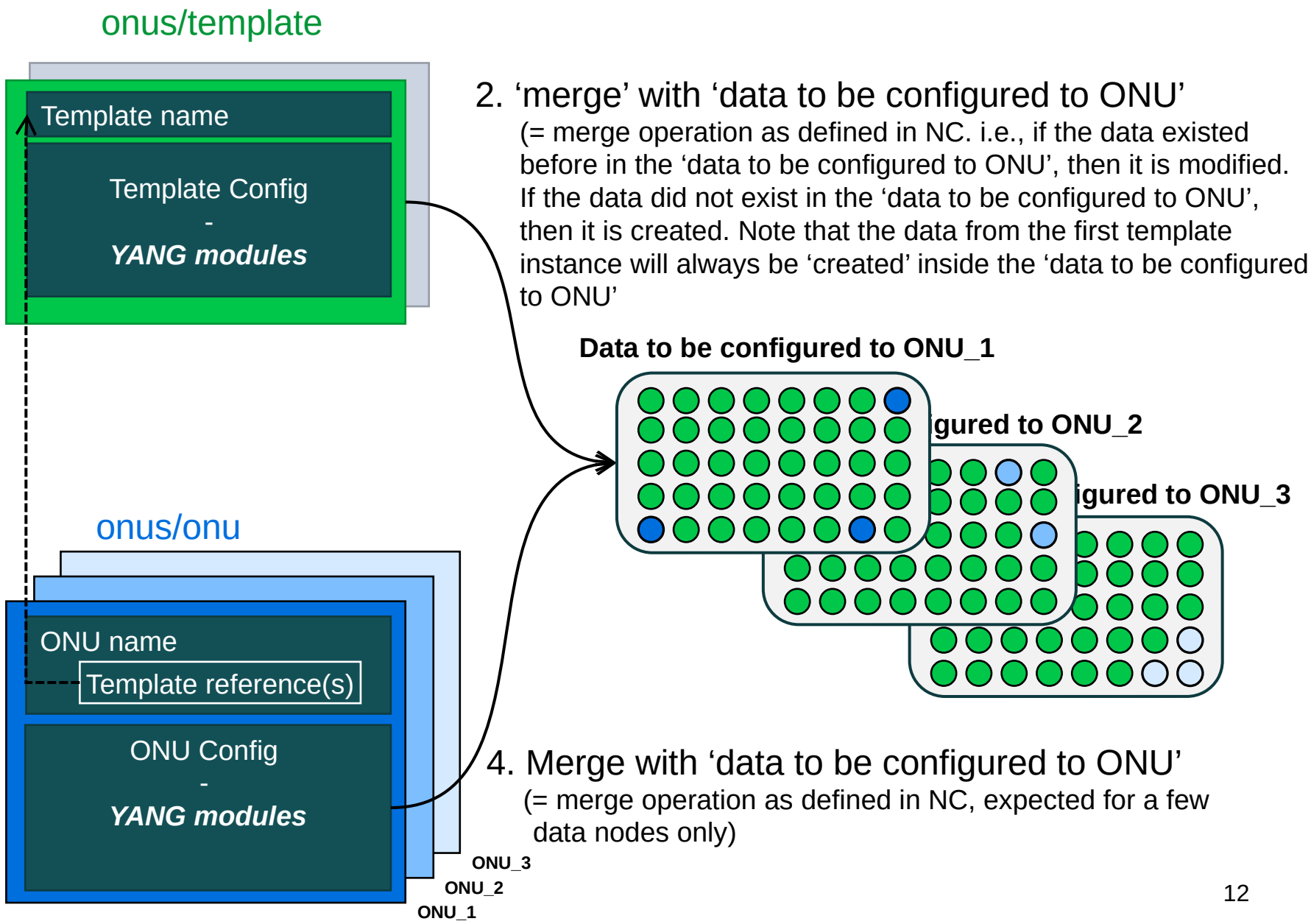


3. Reducing datastore size by using ONU templates

- ONUs generally have very similar - almost identical - configurations.
- Creating new ONU instances requires configuring the same data node values and run the same validation rules, again and again for each ONU instance. This causes large datastores and high CPU load.
- A better way would be to configure once for all a single ONU “model” - call it “ONU template”. Configuring an ONU instance with just a reference to an ONU template is all it takes for the OLT to generate the data nodes to be configured to the ONU with a configuration copied from the ONU template.
- In practice the configuration of actual ONUs is not fully identical from instance to instance. Some data nodes must have a unique value per ONU. Also, it may be decided that only part of the ONU configuration should come from a basic template and the remaining data nodes be explicitly configured in the instance. So, when creating on ONU instance it must be possible to configure any data nodes of the ONU: the resulting data nodes will be a “merge” of the template data nodes and the instance data nodes, the latter ones taking precedence over the template.
- For maximum flexibility, if should also be possible that an ONU instance refers to several ONU templates, each one providing the data nodes for a specific domain. For instance, a template could provide the data nodes related to hardware components and another one could provide the data nodes related to higher layer services. This can be seen as building an ONU instance sequentially, fragment-wise. When templates cover orthogonal domains, using multiple template per instance could allow a lot of ONU instance variations with only a limited number of templates.
- The principle of ONU templates is illustrated in the next slide

Principles of using ONU templates

1. Select template
3. (repeat step 1 and step 2 for any referenced template instance in the order of the ordered list of references)



A caveat with default and mandatory statements

- From previous slides, the YANG model of an ONU template and an ONU instance contain the same data nodes (one just expects that only a few of them are configured in the ONU instance).
- There is however an issue when data nodes imported from published modules (IETF, BBF or IEEE) have a mandatory or default statement.
- The ONU instance should not contain such statements:
  - A default statement will (silently !) overrule a different value coming from the template if not explicitly configured to repeat the template value.
  - A mandatory statement forces an ONU instance to repeat a data node already configured in the template.
- Clearly, such default and mandatories are defeating the purpose to minimize the number of data nodes to be explicitly configured in the ONU instance
- We cannot modify the IEEE / IETF / ITU-T published modules that contain the data nodes (moreover removing defaults is not even backward compatible)
- Hence solving these issues requires *new modules without* mandatories and defaults.

The problem illustrated with mandatories / defaults and templates

Illustrated with ietf-interfaces: ‘type’ is mandatory, ‘enabled’ has default ‘true’

Data node characteristic	Running DS > ONU template	Running DS > ONU instance	Data to be configured to the ONU (with source of the data)	Issue
Mandatory	Configure	*Configure (same as template)*	ONU instance	Unintended redundant configuration (memory)
	Configure	Configure (≠ from template)	ONU instance	Intended overrule
default	Configure	Configure	ONU instance	Intended overrule
	Configure	*Default*	ONU instance	Unintended overrule (silent side effects only known by YANG specialists)
	Default	Configure	ONU instance	Intended overrule
	Default	*Default*	ONU instance	Unintended redundant configuration (memory)
optional	Configure	Configure	ONU instance	Intended overrule
	Configure	Not configured	Template	Intended
	Not configured	Configure	ONU instance	Intended overrule
	Not configured	Not configured	Not existing	Intended not existing

Solving YANG scalability issue: recap

To solve YANG scalability issue when managing ONUs, we have the following requirements:

1. We need a list of ONU templates and list of ONU instances in the OLT YANG model.
2. YANG models of ONU templates and ONU instances use the same data nodes as defined in external standards (IETF, IEEE, BBF) *except,*
   - for their location in the YANG tree: that are located relative to their respective list entry instead of relative to the root of the original YANG model
   - for their mandatories and defaults that are not to be used in ONU instance model nor template when several templates are combined. This means that YANG models for ONU templates and instances need to differ from the original standard modules for that reason.
3. We need YANG model to allow that data nodes in ONU templates and ONU instances refer to shared profiles defined once for all in the OLT root.
4. Market needs a solution that can be deployed without delay

Schema mount RFC 8528 cannot currently be used for scaling YANG

- The primary selling point of schema mount is that it allows to reuse unmodified YANG modules so that their data nodes originally defined at YANG tree root at different places in a YANG tree (under a mountpoint) without editing the original module.
- If schema mount was to be used for our purpose, in mountpoints in the ONU template list and ONU instance lists, one should anyway mount ***altered versions of the original standard modules*** (i.e., without mandatories and defaults). Schema mount does not currently allow that.
- Schema mount has restrictions when a mounted data node need to refer a data node outside the mount point (cf RFC 8528 “parent-reference”) which limits its applicability.
- Internet draft “draft-jouqui-netmod-yang-full-include” proposes an extension of RFC8528. However, our understanding of the current draft is that it suffers the same drawback as RFC 8528 when it comes to our scalability goals.

Hence, to be useable for YANG scalability, an improved version of schema mount should at least:

1. Allow alteration of mounted standard data nodes to control their mandatory & default statement and
2. Allow unrestricted capability to refer to data nodes outside the mount point.

Note that there is a need from the market for a quickly deployable solution, including available tool chain.

Summary

- To support the industry to manage access network equipment, there is an urgent need for YANG modules that efficiently scale to support large-scale systems.
- Current standard YANG modules (including IETF) are unpractical to deploy in such large-scale systems.
- Neither existing RFC 8528 Schema mount, nor current Internet draft “draft-jouqui-netmod-yang-full-include are suitable for helping YANG scalability.
- To support YANG scalability, an enhanced version of the Internet draft should at least:
  - Allow alteration of mounted data nodes to control their mandatory & default statement,
  - Allow unrestricted capability to refer to data nodes outside the mount point and
  - Require as little as possible specific support from tool chain

Thank You.